Machine learning for DQM and DC in CMS

Data Quality Monitoring and Data Certification

Draft

Mantas Stankevičius (Vilnius University, LT)
on behalf of the CMS collaboration

Outline

- Current DQM
    - Tools
    - Online: Detector monitoring
    - Offline: Data certification
    - Limitations
- ML-based DQM
    - Fit
    - Challenges
    - Applicability studies
        - Online
        - Offline

DQM

Data Quality Monitoring

Data Quality Monitoring

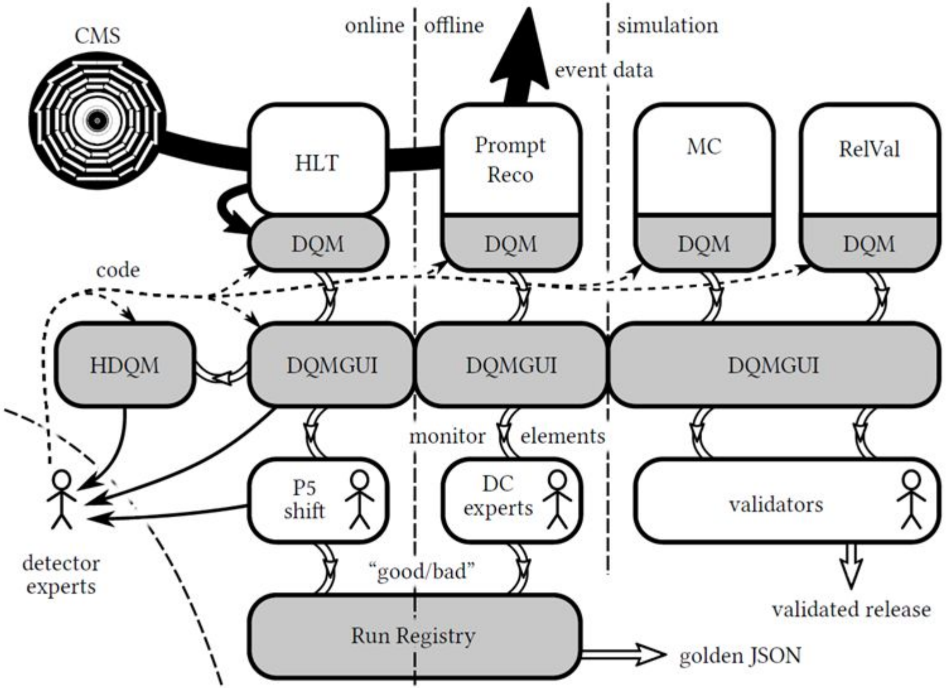
Collection of tools and processes to provide:

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

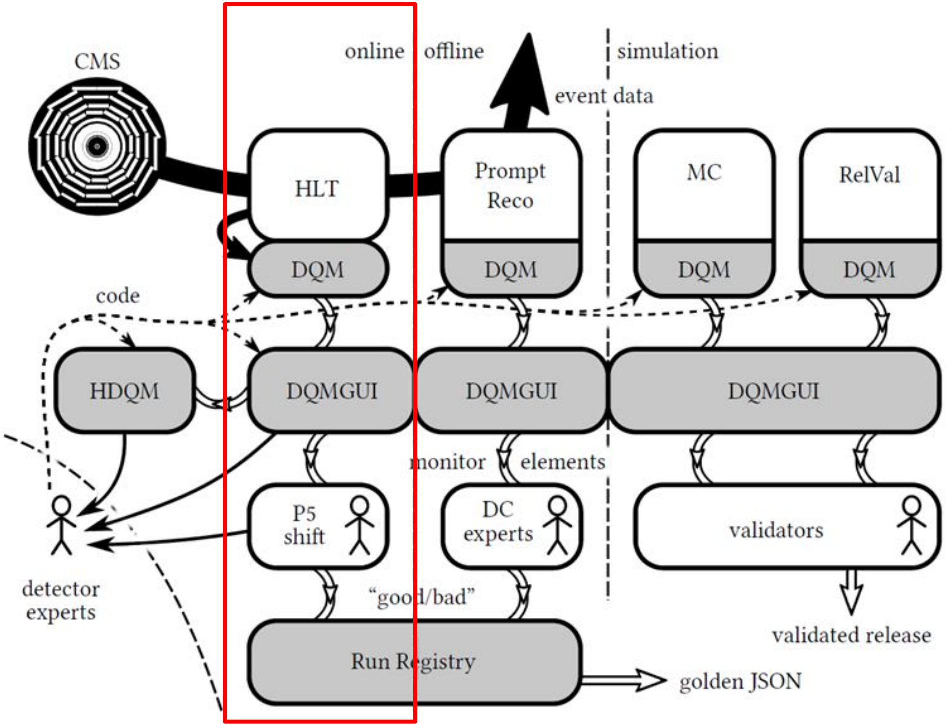
Collision data and detector status constantly flow from detector

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

Predefined rule-based 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



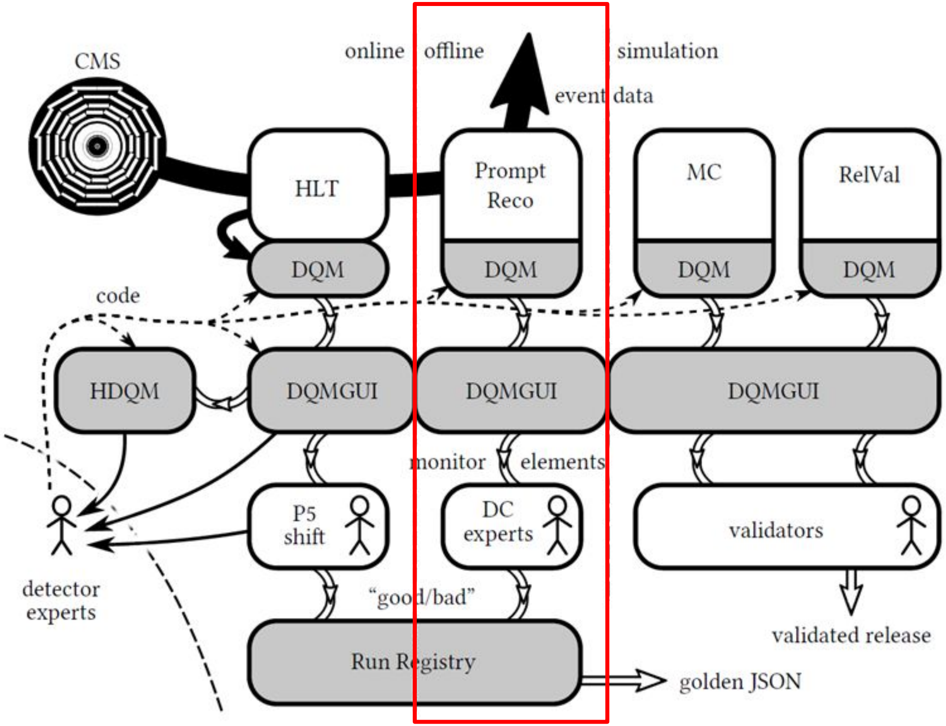
Data Quality Monitoring: Offline

Data is fully reconstructed and calibrated after approx 48 hours

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

Certification is made on Run and Lumisection* levels

GoldenJSON is produced. List of only GOOD Runs and Lumisections



* Lumisection is a ~23sec data-taking interval

DQM GUI

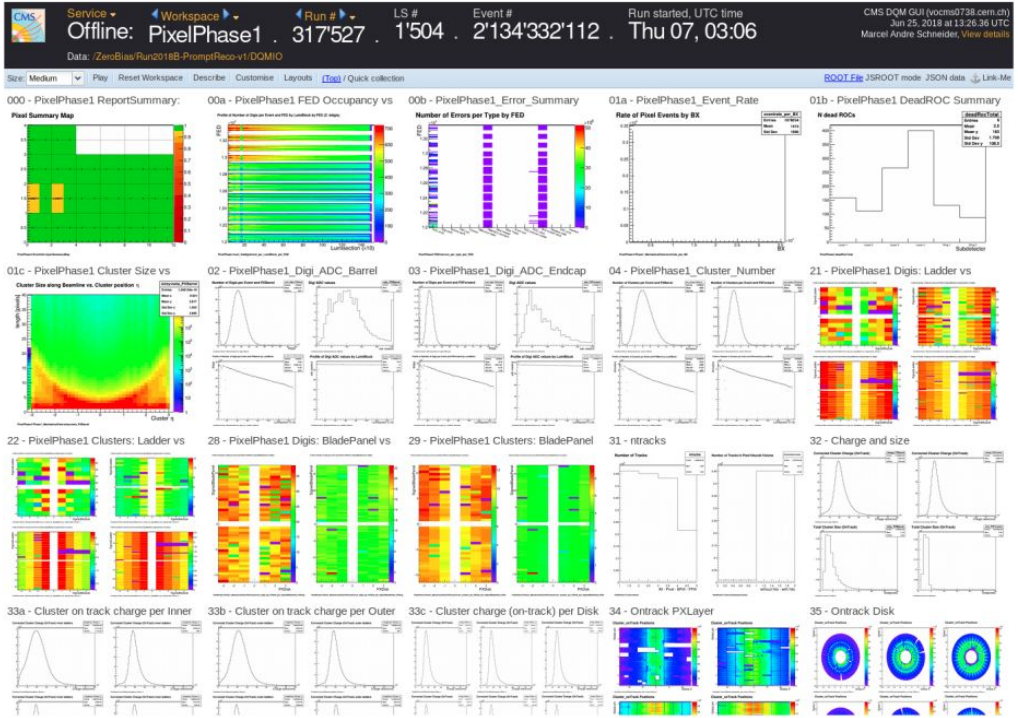
Web service to collect and archive monitoring elements (ME)

Provides APIs for scripts

Web based interface to browse realtime and historical data

DQMGUI provides access to:

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



Run Registry

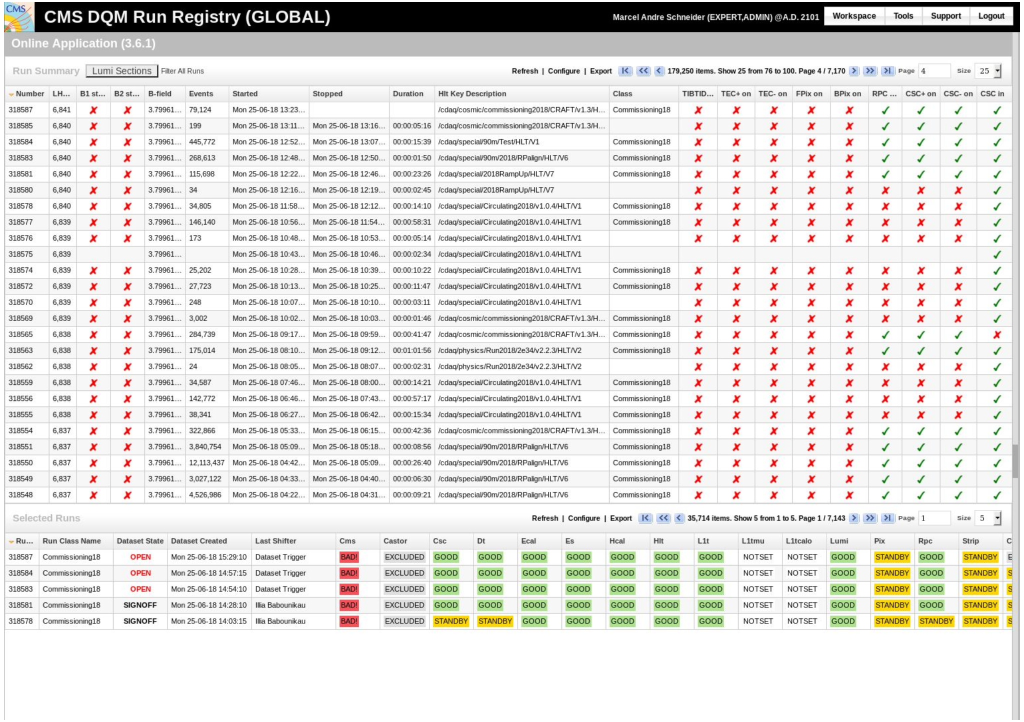
Automatically collects Run and Lumisection data

Web interface for experts to manually certify data

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

Currently under redesign 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
  - 132k hours in 2018 of DC
- 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

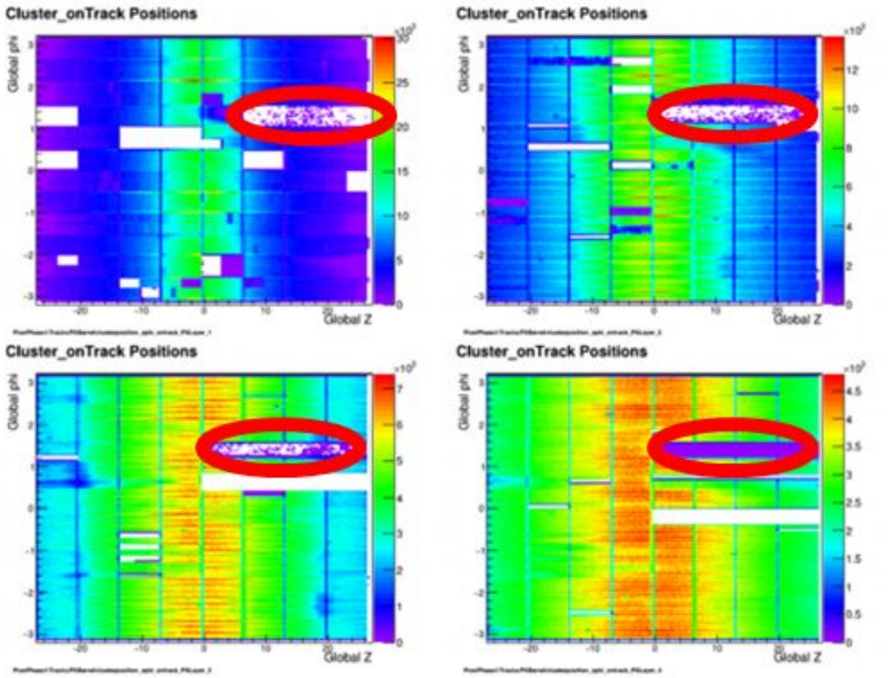
Power supply issue on the Pixel detector

- 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 Chip (ROC)s are not optimal

- OK - randomly distributed dead ROCs
- NOT OK - dead cluster

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



Towards ML-based DQM

From rules to (un)supervised models

ML fit in DQM operations

Reduce manual labor by doing tedious work faster (compare histograms)

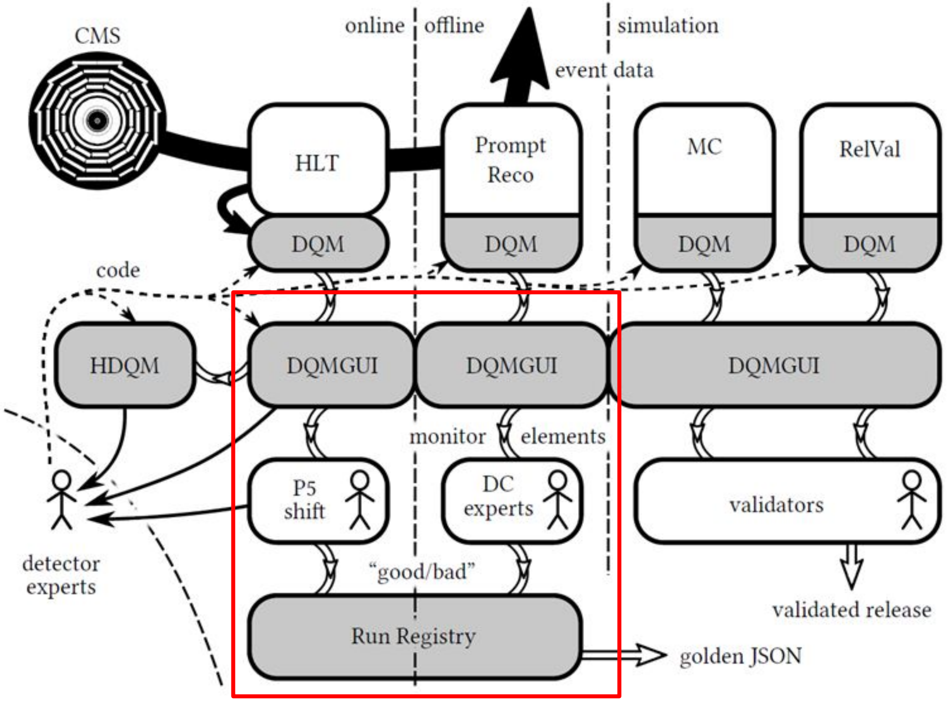
Minimize human errors and optimize human input

Detect anomalies with lower latency

Improve certification quality on lumisection level

Dynamically adapt to conditions change

Provide report of the classification results



Challenges

- Tons of data (in form of histograms)
- Sparse anomalies
- Changing running conditions

Offline

- Label contamination
- Class imbalance

Online

- Almost no labeled data
- Normalization is very difficult

Brief introduction in 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:

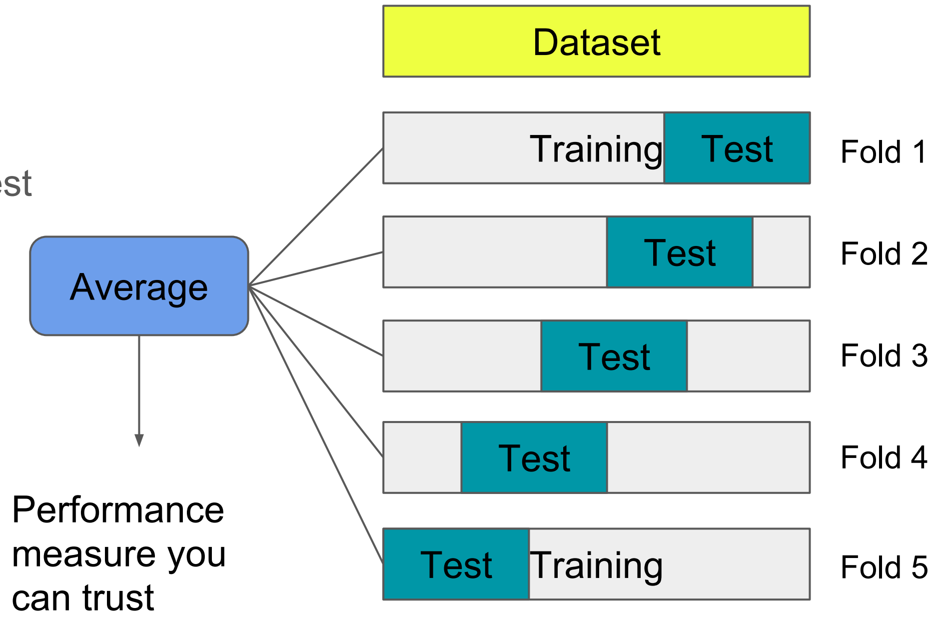
- Clustering
- Association

Cross validation

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

Performance measure is independent from train:test distribution

Solution to overfitting

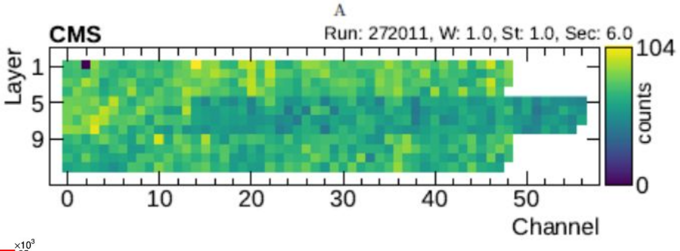


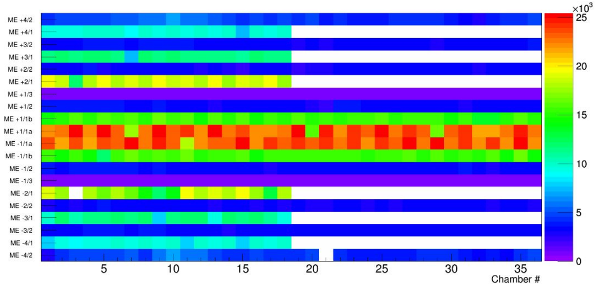
Online: detector monitoring

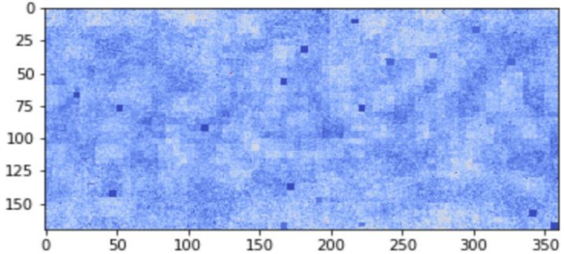
Occupancy plots

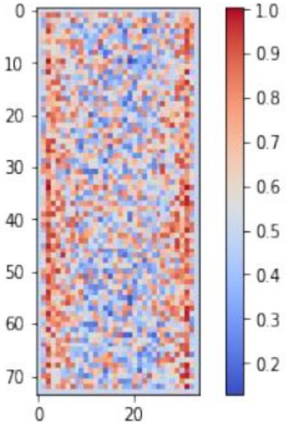
Overall occupancy plots are among the most important DQM plots

They show the frequency of hits in given detector channels and are used to identify and diagnose problems









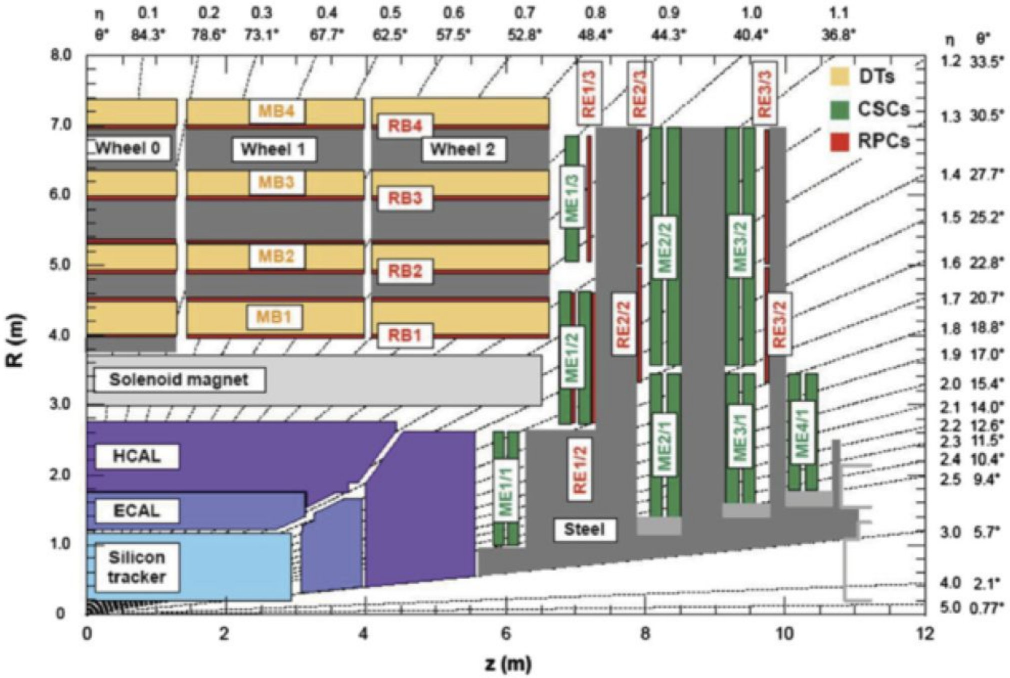
Drift Tubes (DT)

Barrel Muon sub-detector (|η|≲1.1):
o(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

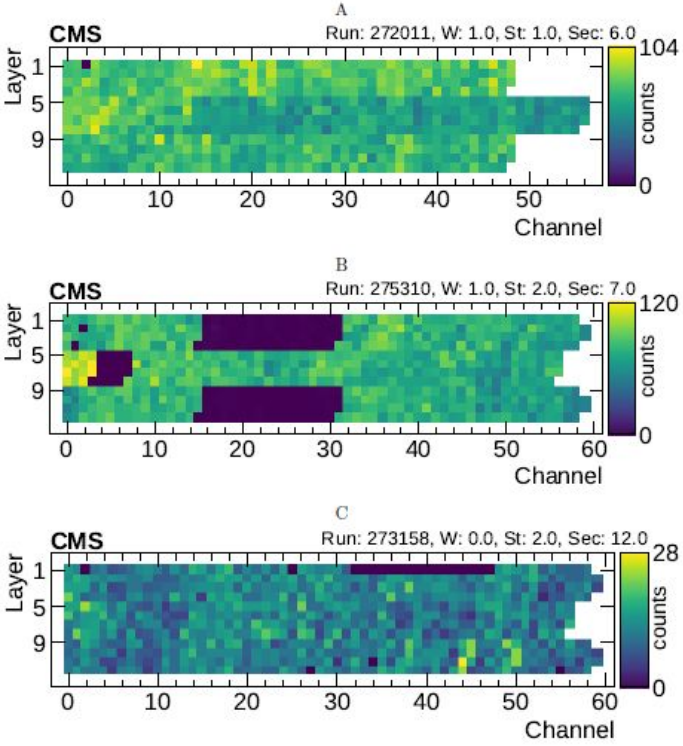
Labels:

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

A - Dead one channel

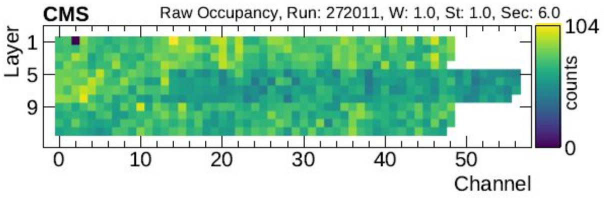
B - Dead regions in multiple layers

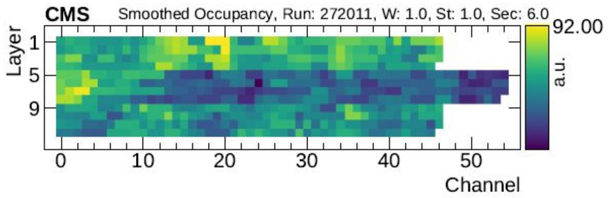
C - Dead region in one layer



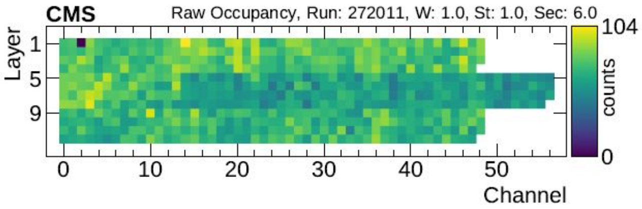
Data preprocessing

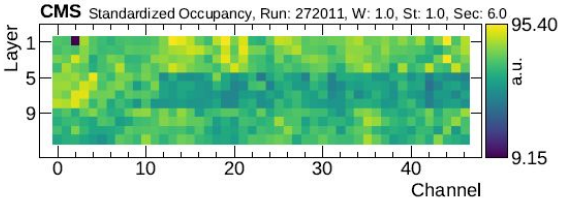
Smoothing. According to CMS DT experts isolated misbehaving channels are not considered a problem





Standardization into fixed dimensionality. 1D Linear interpolation



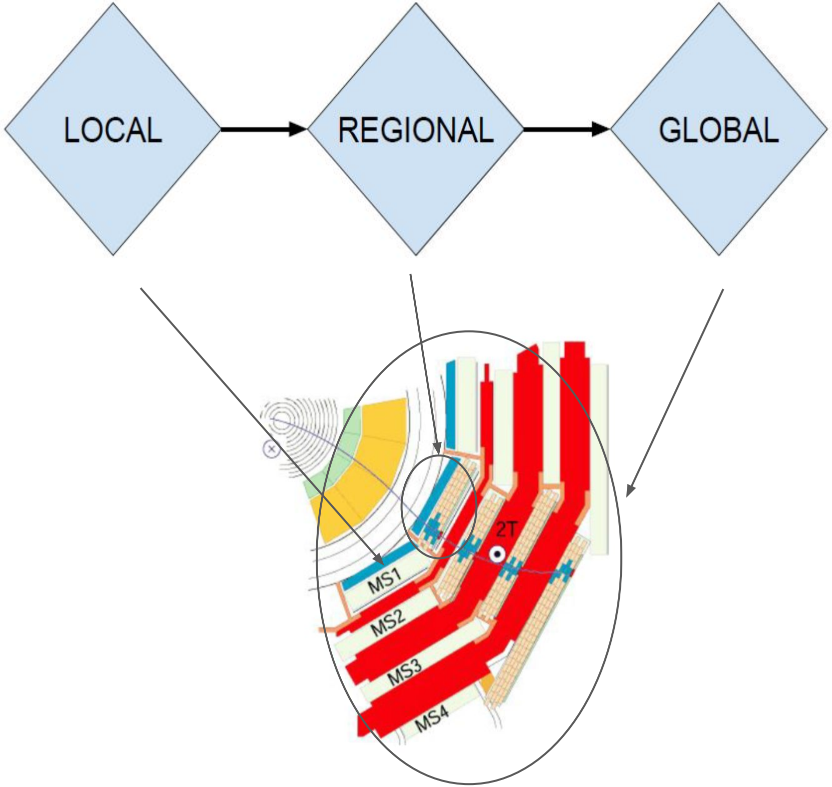


Approaches to the anomaly detection in DT

Local: each chamber layer is treated independently from the other layers

Regional: extend the local approach to account for intra-chamber problems; simultaneously consider all layers in a chamber, but each chamber independently from the others

Global: simultaneously use the information of all the chambers for a given acquisition run; the position of the chamber in the CMS detector impacts expected occupancy distribution of the channel hits



Local strategy: scope, methods & results

Convolutional neural network (CNN)

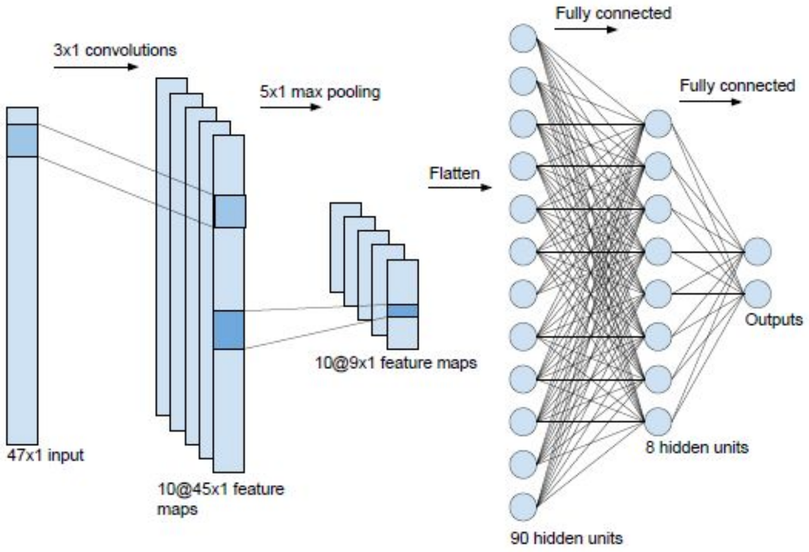
Activations: ReLU and softmax

Optimizer: Adam

Loss function: cross entropy

Class weights: more attention to minority class

ROC AUC = 0.995



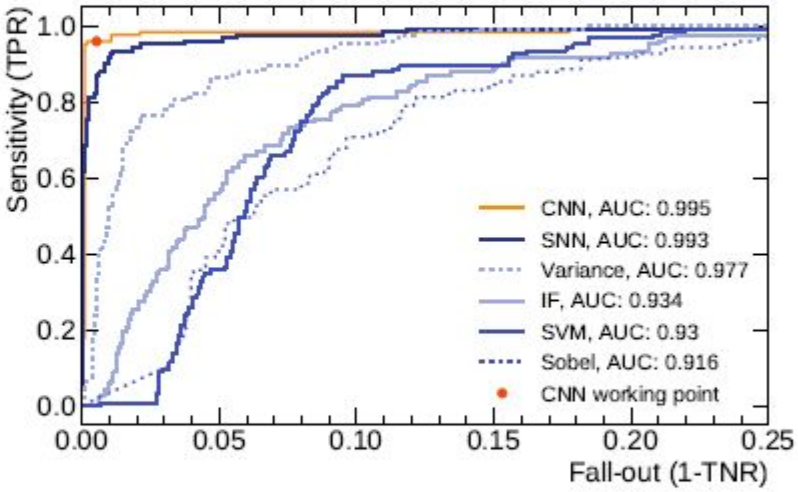
Local strategy: scope, methods & results

Filters out most of the anomalies

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

Each chamber layer is treated independently from the other layers

Convolutional neural network (CNN) outperforms other methods



Supervised

- Shallow neural network (SNN)
- Convolutional neural network (CNN)

Semi-supervised

- SVM
- Isolation Forest

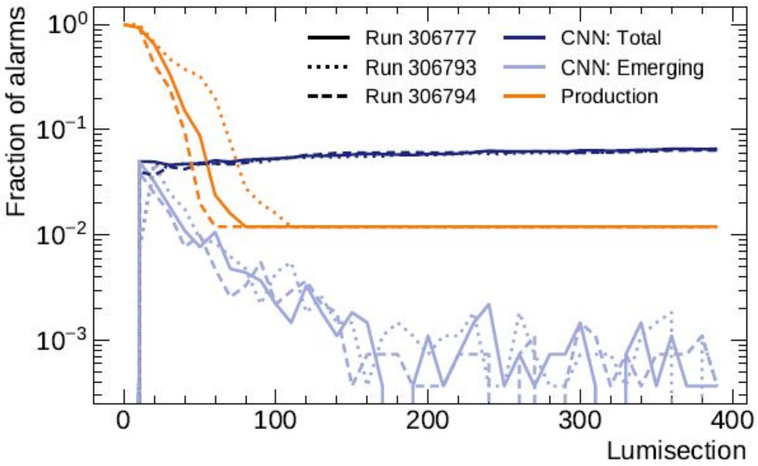
Unsupervised

- Sobel filter

Local strategy: scope, methods & results

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

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

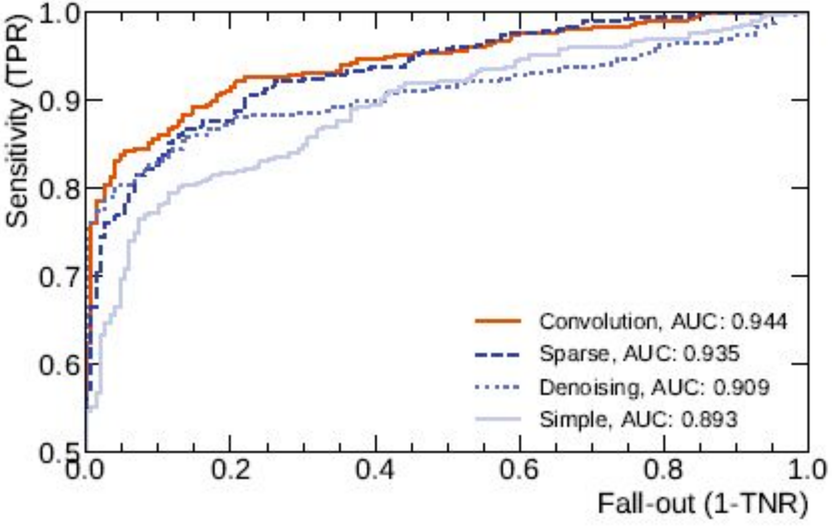


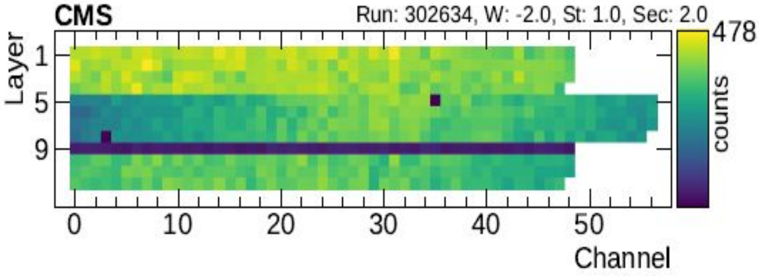
Regional strategy: scope, methods & results

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

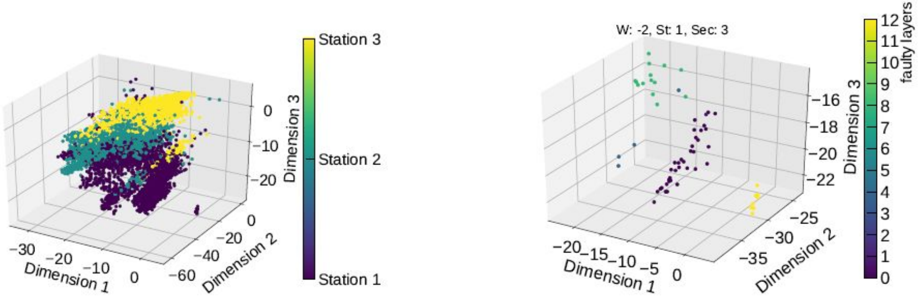
Global strategy: scope, method

Simultaneous use of all the chambers data. The position impacts expected occupancy pattern

Autoencoders learn a compressed representation of chamber data

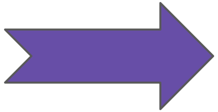
When the bottleneck of the autoencoder is 3-dimensional one can visually inspect those representation

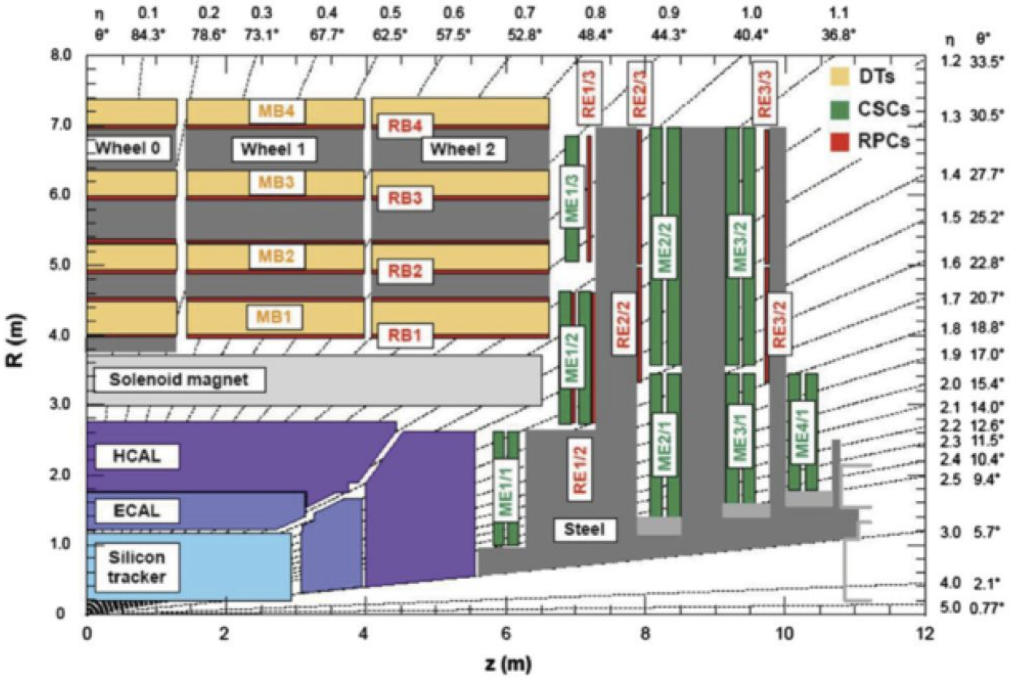
The global approach is then potentially capable to spot an unusual behavior of DT chambers taking into account the geographical constraints



Compressed representations of the chamber-level data

HCAL





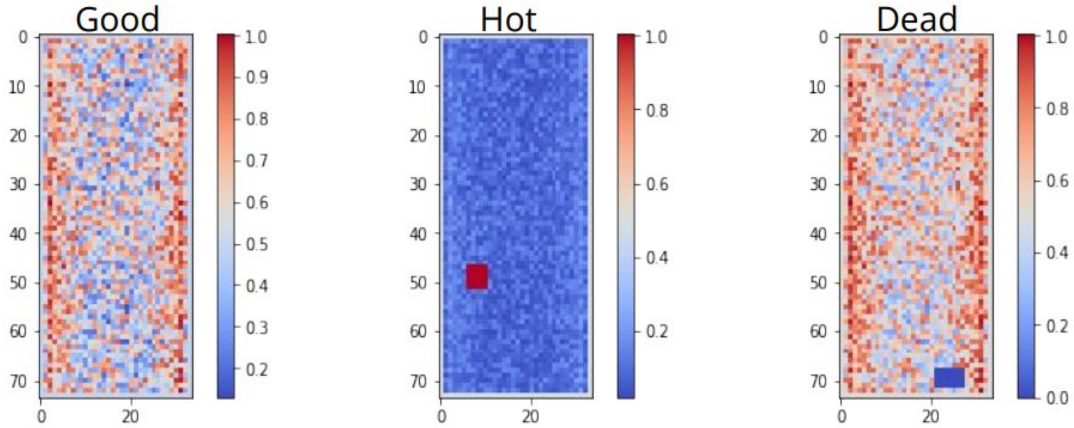
DTs
CSCs
RPCs
Wheel 0
Wheel 1
Wheel 2
MB4
MB3
MB2
MB1
RB4
RB3
RB2
RB1
Solenoid magnet
HCAL
ECAL
Silicon tracker
Steel
ME1/1
ME1/2
ME1/3
ME2/1
ME2/2
ME3/1
ME3/2
ME4/1
RE1/2
RE1/3
RE2/2
RE2/3
RE3/2
RE3/3
R (m)
z (m)
η
θ°

HCAL: dataset

Have mostly good data

Manually simulate bad data by setting region

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



HCAL: supervised

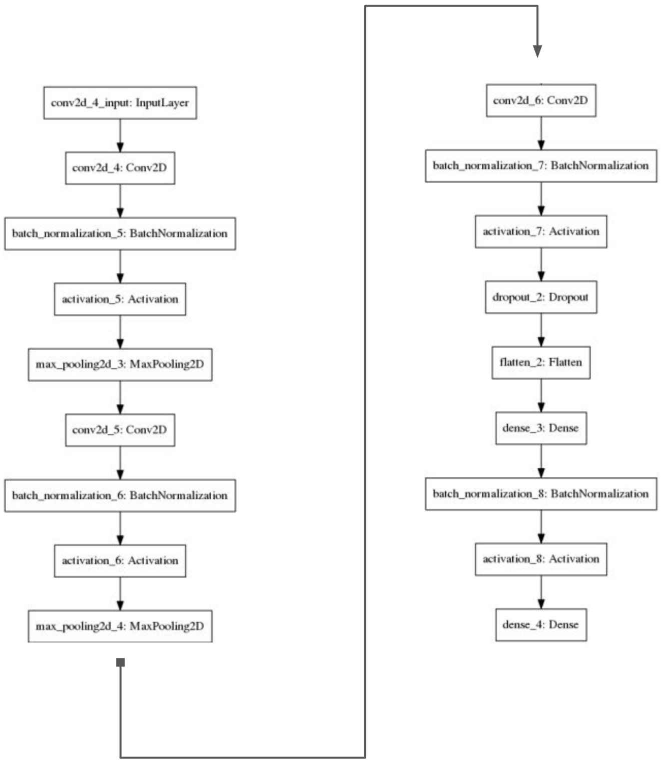
Convolutional neural network

3 convolutional layers

Activation: ReLU

Optimizer: Adam

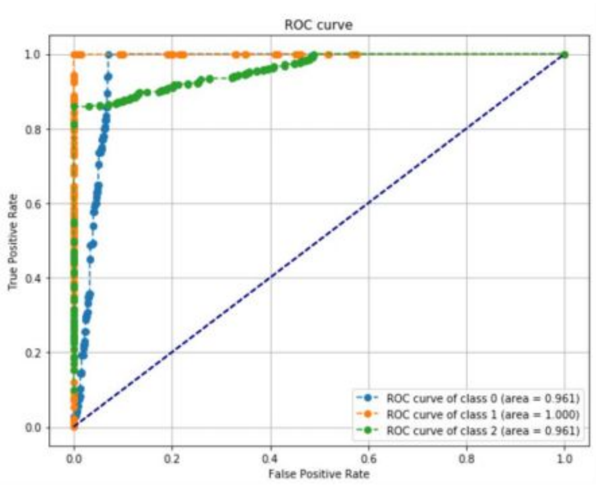
Loss function: categorical cross entropy

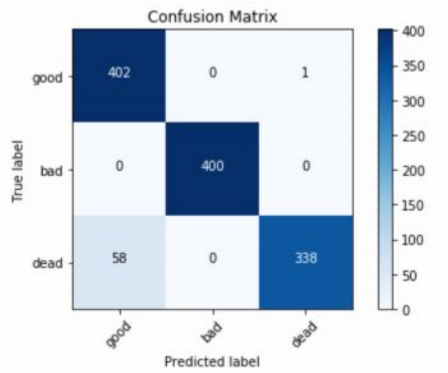


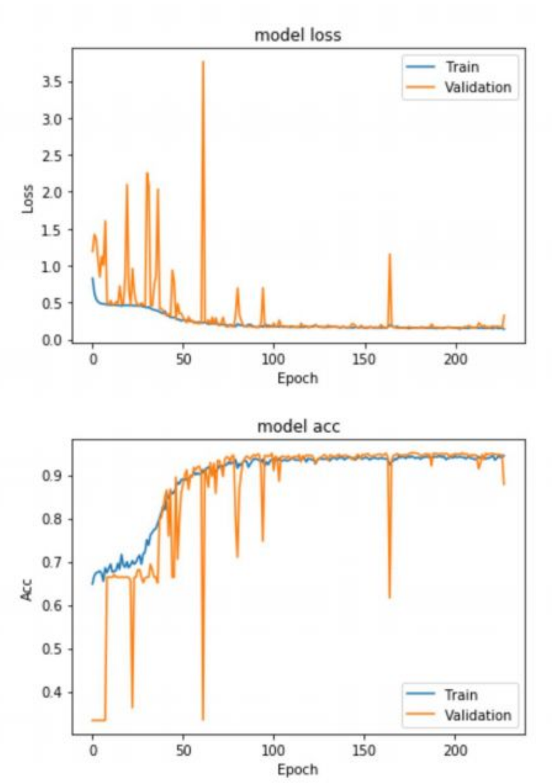
HCAL: supervised results

Accuracy: 0.95

ROC AUC: 1, 0.961, 0.961







HCAL: semi-supervised

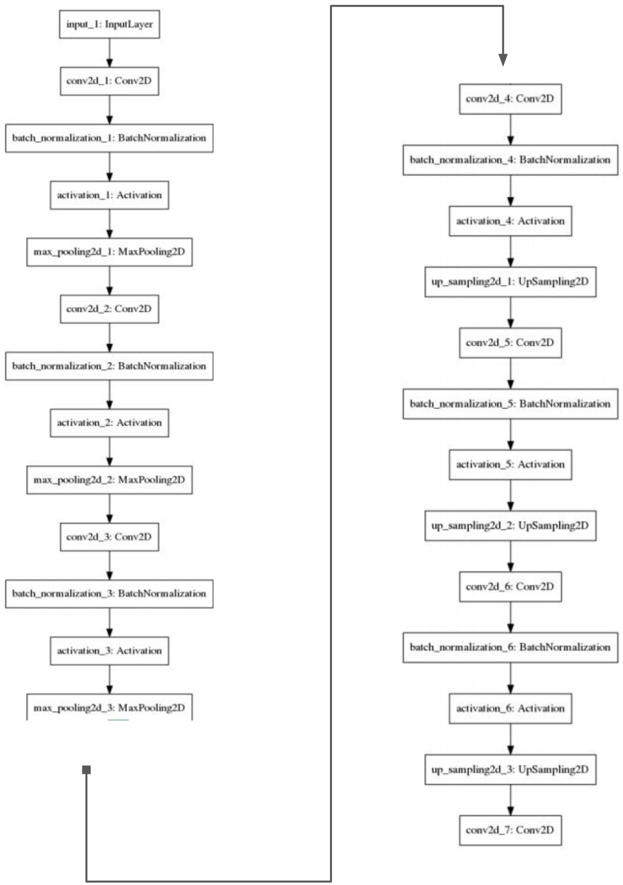
Bottleneck autoencoder

Encode & decoder: 3 convolutional layers each

Activation: ReLU

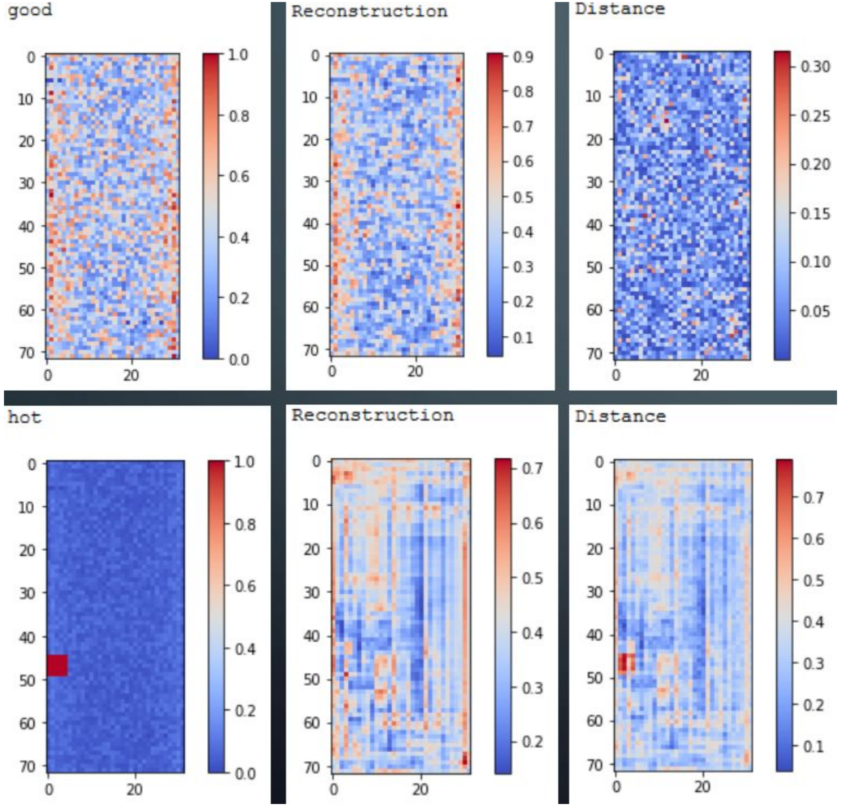
Optimizer: Adadelta

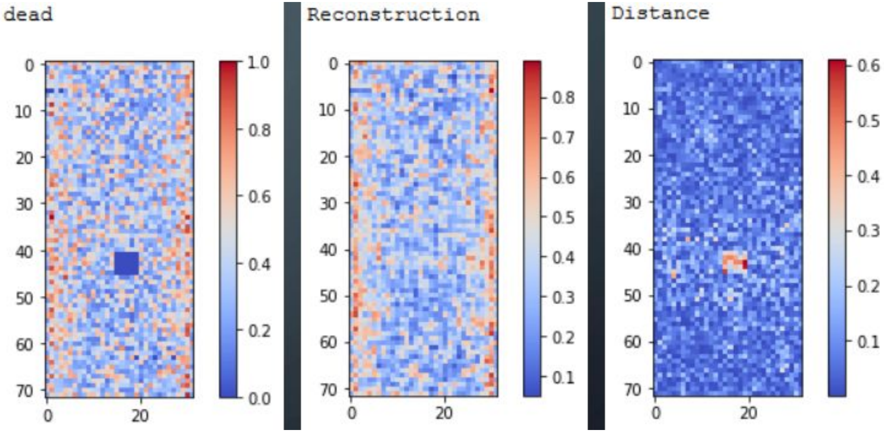
Loss function: mean square error



HCAL: semi-supervised results

Reconstruction of good, dead and hot

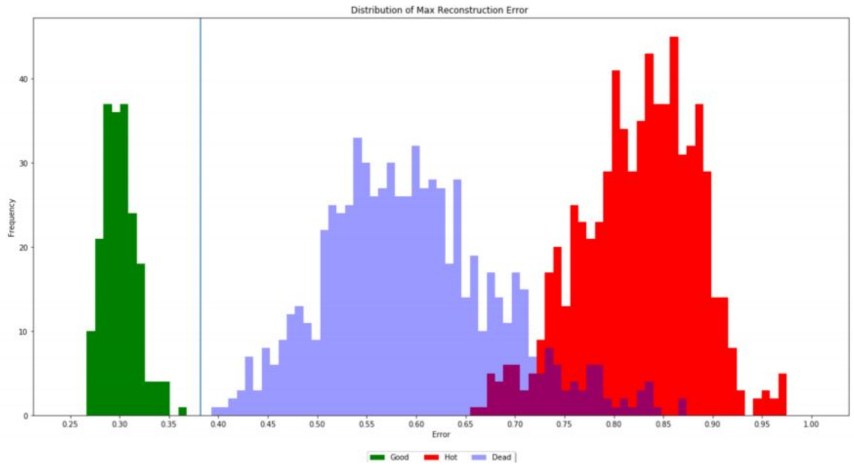


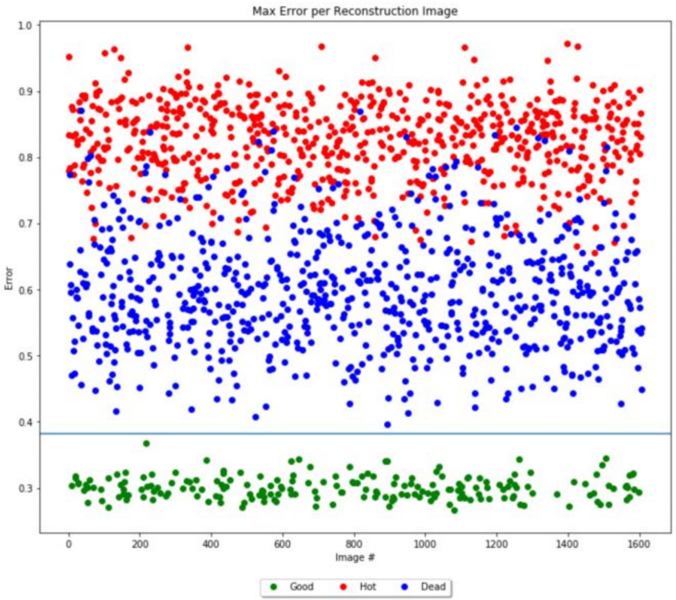


HCAL: semi-supervised: reconstruction

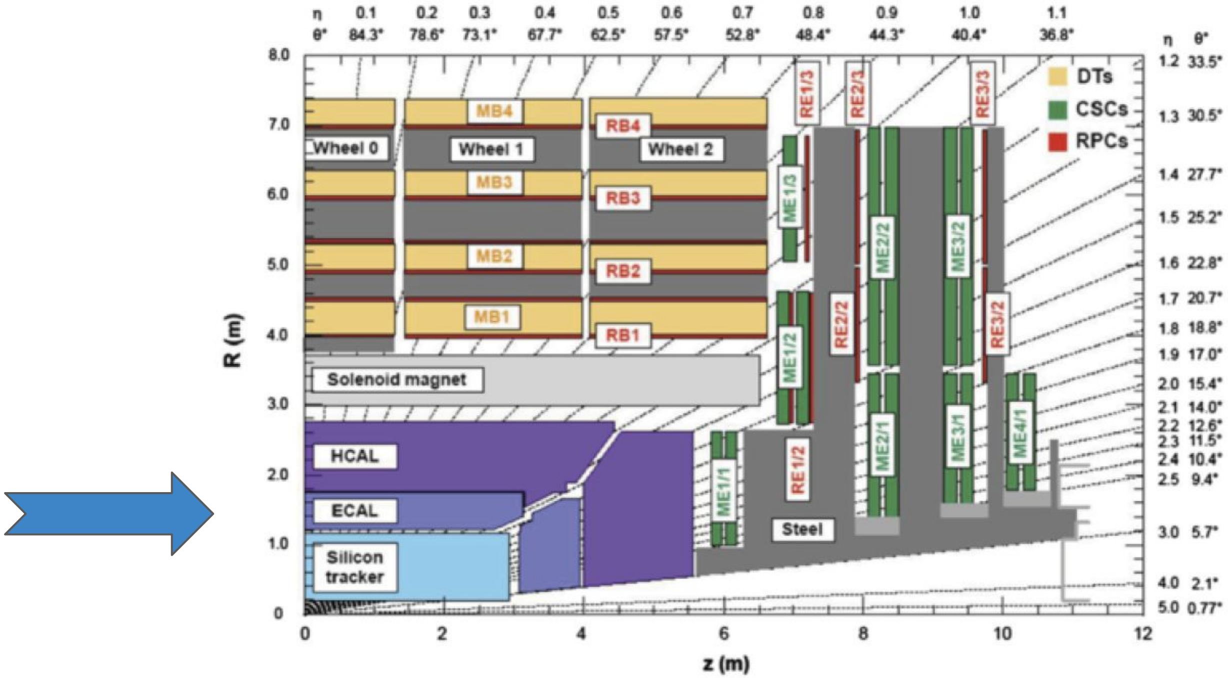
Distribution of reconstruction error

Good vs BAD are well differentiable even with simple parameters





ECAL



DTs
CSCs
RPCs
MB1
MB2
MB3
MB4
RB1
RB2
RB3
RB4
Wheel 0
Wheel 1
Wheel 2
Solenoid magnet
HCAL
ECAL
Silicon tracker
Steel
ME1/1
ME1/2
ME1/3
ME2/1
ME2/2
ME3/1
ME3/2
ME4/1
RE1/2
RE1/3
RE2/2
RE2/3
RE3/2
RE3/3
R (m)
z (m)
η
θ°

ECAL slide

No activity from ECAL for quite some time

https://indico.cern.ch/event/705916/contributions/2896872/attachments/1602624/2541404/19022018_ADforECAL_DQMML_Nab.pdf

Offline: data certification

Dataset 2010

Collected by CMS in 2010

Available through CERN OpenData

891 features

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

16.000 lumisections

75:25 class distribution ratio (GOOD:BAD)

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

Classification into 3 categories

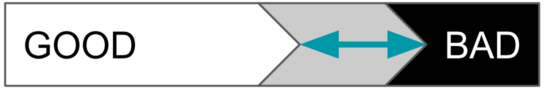
- Definitely GOOD (white zone)
- Definitely BAD (black zone)
- Ambiguous (gray zone)
    - Decision can’t be made automatically
    - Human intervention is required

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

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

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

Aim to minimize gray zone (Rejection Rate)



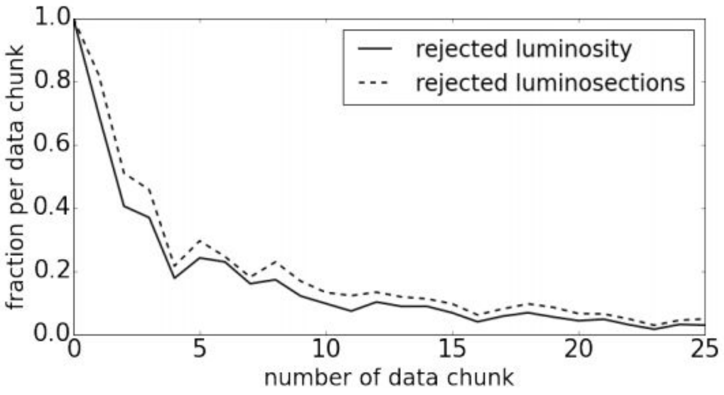
Gradient Tree Boosting classifier

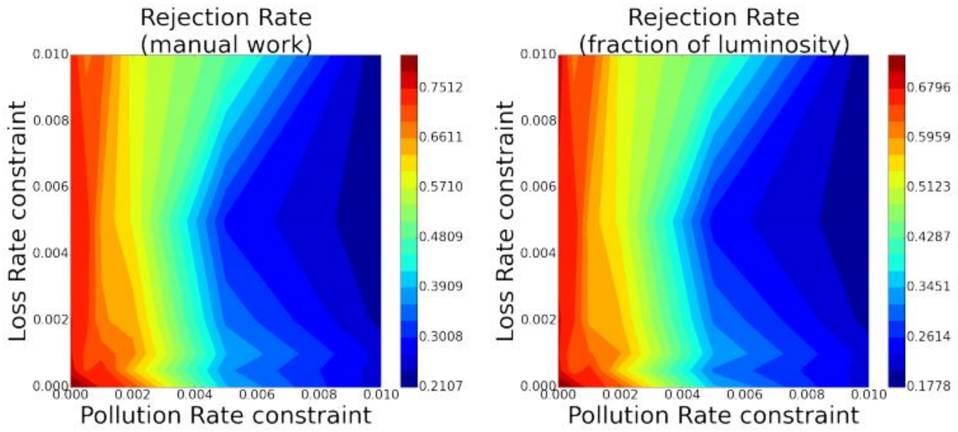
10 fold cross validation

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

System is able to automatically process at least 20% of samples and 30% of total luminosity keeping pollution and loss rates on negligible level

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





(a) Fraction of rejected samples.

(b) Fraction of rejected luminosity.

Deep learning for inferring cause of data anomalies [2]

Determine which sub-detector is responsible for anomaly

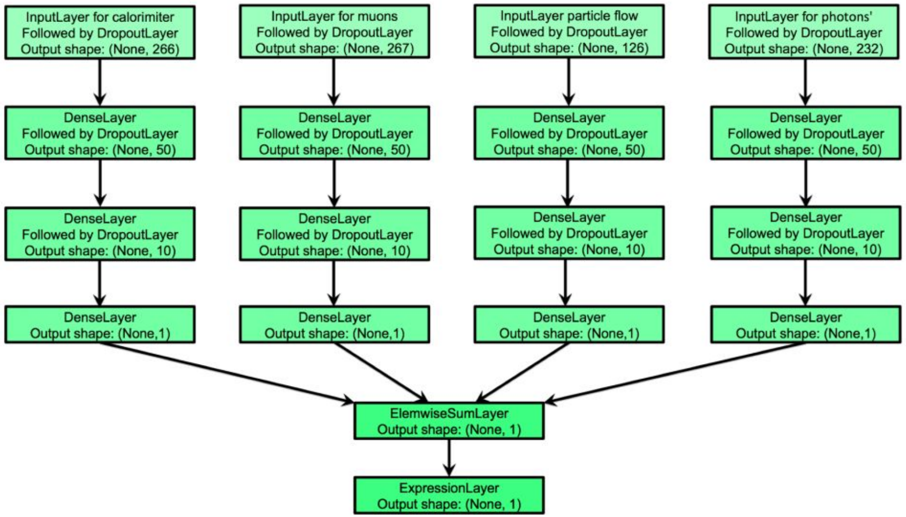
4 Neural Networks depending on particle type

- Photons
- Muons
- Particle Flow Jets
- Calorimeter Jets

Output is determined by `Fuzzy AND`

Loss function: dynamic cross-entropy

$$L' = (1 - C) \cdot L + C \cdot (L_1 + L_2 + L_3 + L_4)/4,$$



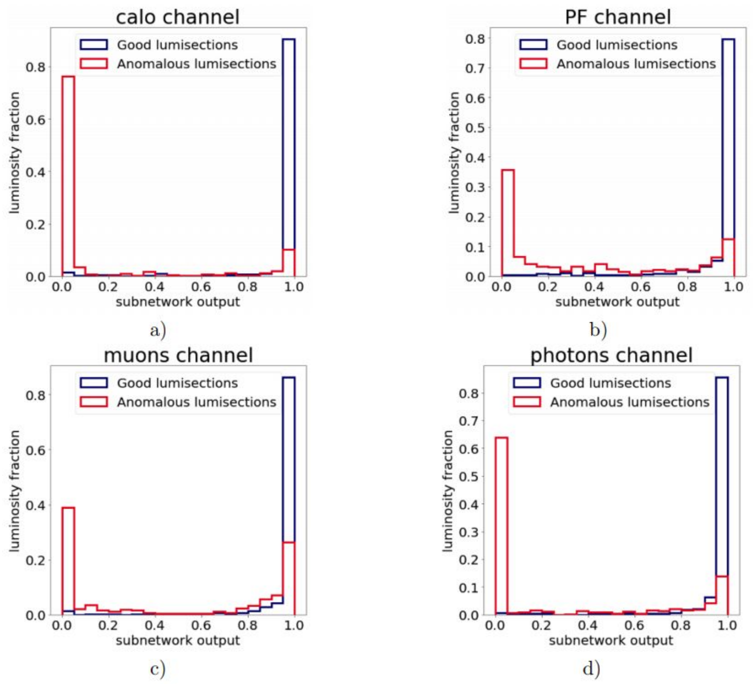
Deep learning for inferring cause of data anomalies [2]

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

Dataset for Jet analysis

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

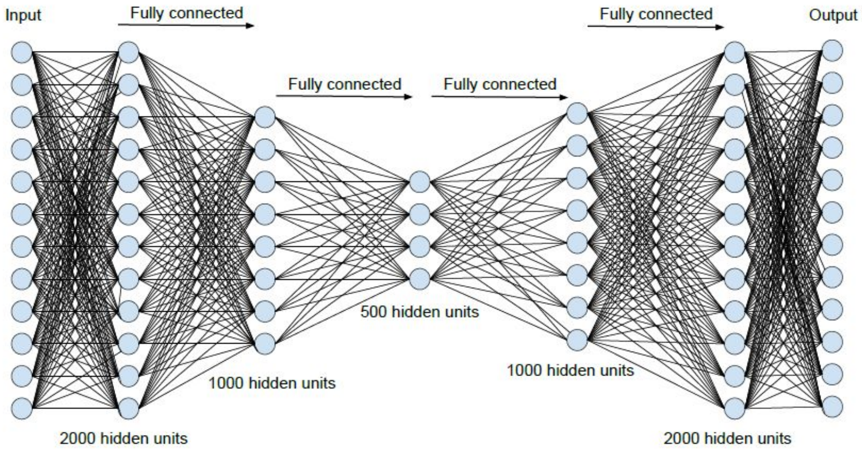
Data is sorted time-wise

Activations: PReLU

Optimiser: Adam (LR=0.0001)

Loss function: mean square error

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

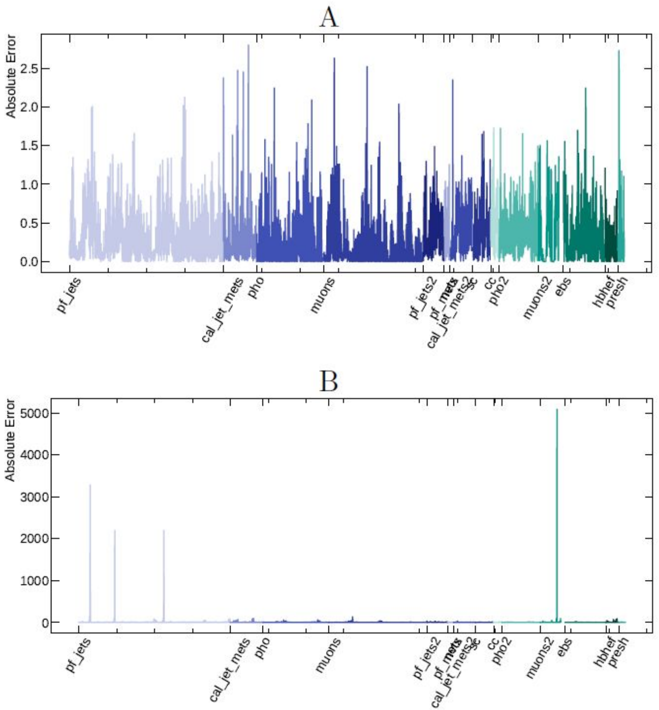


Anomaly detection using Autoencoders [3]

Features are grouped by physics object

A: reconstruction error for a GOOD lumisection.
Similar y-axis amplitude across GOOD lumisections

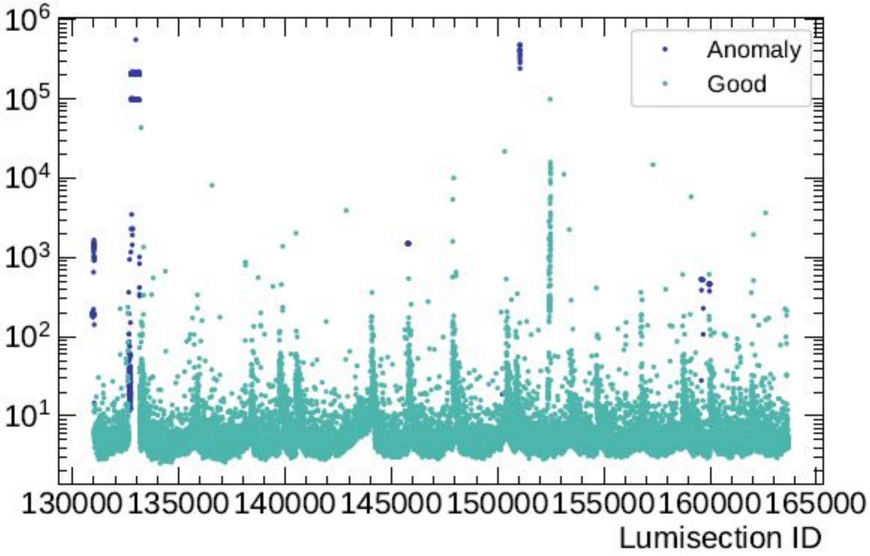
B: reconstruction error for a BAD lumisection.
Observable peaks for anomalous features.
Problematic muons and jets



Anomaly detection using Autoencoders [3]

Error waves depend on instantaneous luminosity

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
- XGBoost
  - Good predictive power
  - Average training speed
  - High memory usage during training

Comparison of supervised ML models [6]

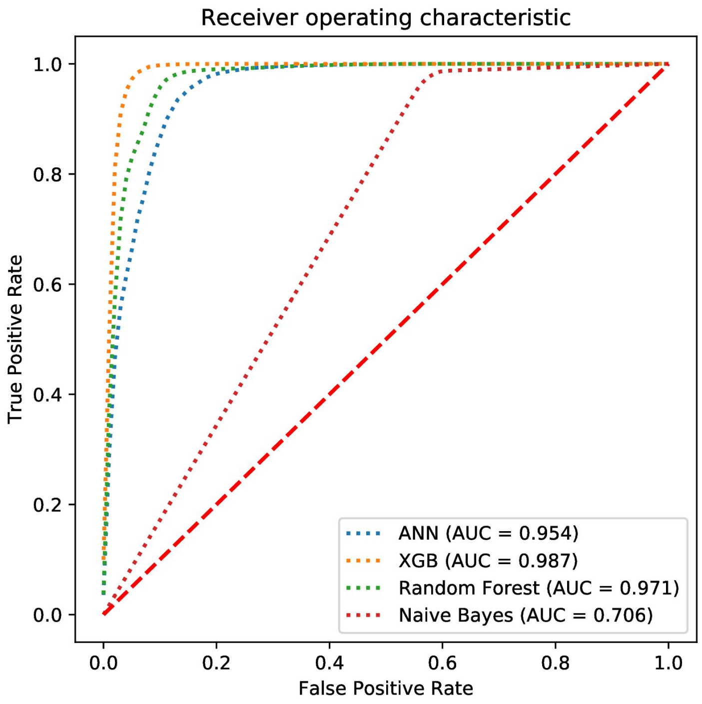
Class weights - more attention to minority class

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



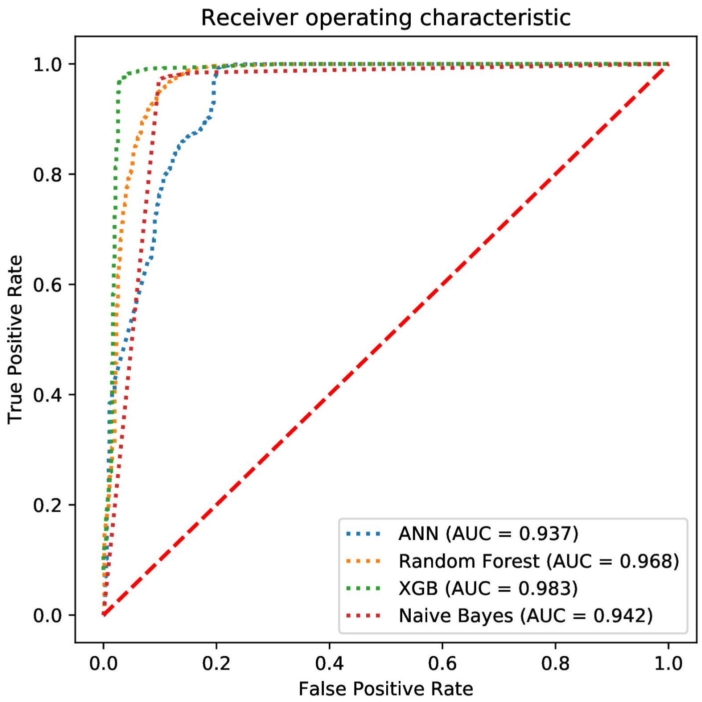
Comparison of supervised ML models [6]

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 engaged in partnership, successfully, with IBM and Yandex, agreed within the 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]

Towards Run 3

Experience we learned from studying ML4DQM and ML4DC has been extremely valuable, indicating positive promises

But so far, most of the ML studies which have been done are independent (outside) of the real online DQM, offline Data Certification process work-flow

As we continue to optimize ML techniques, we plan to start working on the implementation of ML into the real work-flow of the CMS online DQM and Data Certification process

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

Conclusion

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, http://cms.cern.ch/iCMS/jsp/db_notes/showNoteDetails.jsp?noteID=CMS%20CR-2018/207

[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, http://cms.cern.ch/iCMS/jsp/db_notes/showNoteDetails.jsp?noteID=CMS%20CR-2018/202

[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, http://cms.cern.ch/iCMS/jsp/db_notes/showNoteDetails.jsp?noteID=CMS%20CR-2018/276

[5] Marcel Andre Schneider et al, “The Data Quality Monitoring Software for the CMS experiment at the LHC: past, present and future”, CHEP 2018, http://cms.cern.ch/iCMS/jsp/db_notes/showNoteDetails.jsp?noteID=CMS%20CR-2018/228

[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://cms-mgt-conferences.web.cern.ch/cms-mgt-conferences/conferences/pres_display.aspx?cid=2237&pid=16934

[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