IBM Machine Learning project

via CERN openlab

with CMS and LHCb

V. Azzolini
M. Pierini
G. Cerminara
A. Pol
J-R. Vlimant
M. Andrews
T. Mudholkar
N. Dev
N. Marinelli
C. Jessop

Daniel Hugo Cámpora Pérez
Niko Neufeld
Xavier Vilasís Cardona
Michele Piero Blago

Outline

CERN long standing tradition of collaboration with industry and research institutes

CMS experiment and LHCb experiment engaged a partnership with IBM with the objective to improve operations, detector reconstruction and to generate benchmarking technological results, respectively.

In this joint talk we are going to present the research goals, agreed within the CERN Openlab framework, how we hope they will mature and be achieved through a collaborative contribution of technologies and/or resources.

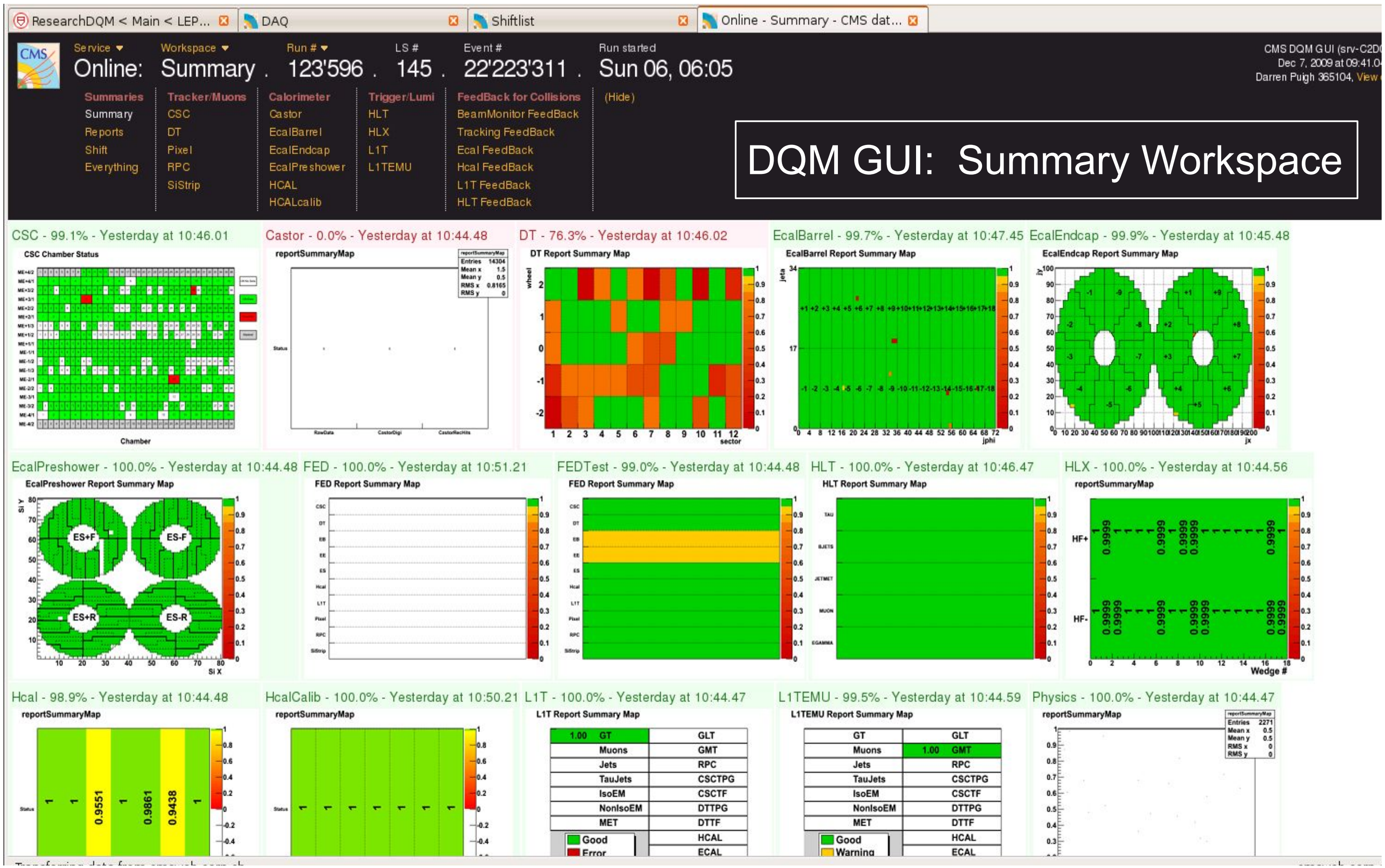
Virginia Azzolini, on behalf of CMS collaboration
Daniel Hugo Cámpora Pérez, on behalf of LHCb collaboration

CMS - IBM partnership

Improve Data Quality Monitoring (DQM) operations:

Monitors and ensures data quality of each data

Anomaly detection



One plot per sub system

Virginia Azzolini

Limits of a Human-based DQM

The current system works but

expensive, in terms of human resources

Online : 8h shift, 24/7 + the effort to train her, maintain instructions, etc

volume budget problem

There is a limit to the amount of quantities that a human can process in a finite time interval. Summary dashboard plus 15 subsystem dedicated ones. This can cause delay in spotting a problem or cause a transient problem to be overlooked

It makes assumptions on our level of understanding

the quantities are compared against a pre-defined reference visually or via automatic threshold checking. Static threshold, led by actual conditions understanding, do not scale

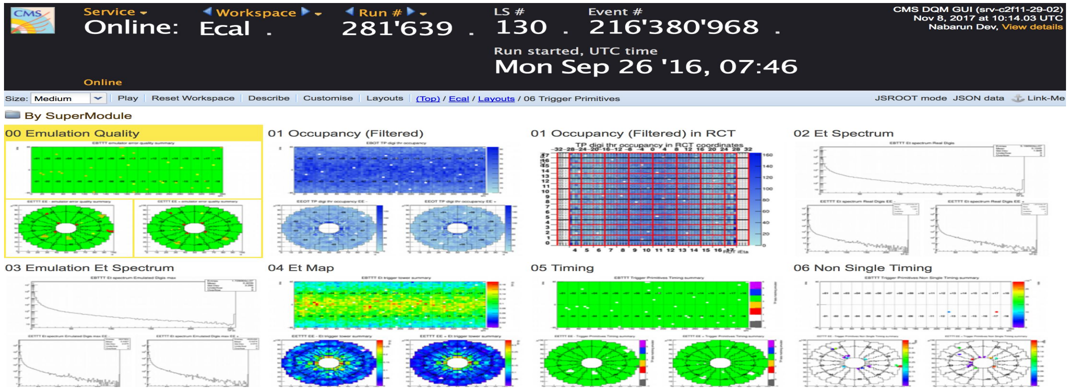
Strategy tailored to certain failure modes,

the certain set of quantities monitored might not have enough discriminatory power against all the possible problems

Project goals

intelligent
Integrate Machine Learning technique in the current Data Quality monitoring tools and operations, to make them less expensive in term of human intervention and more efficient

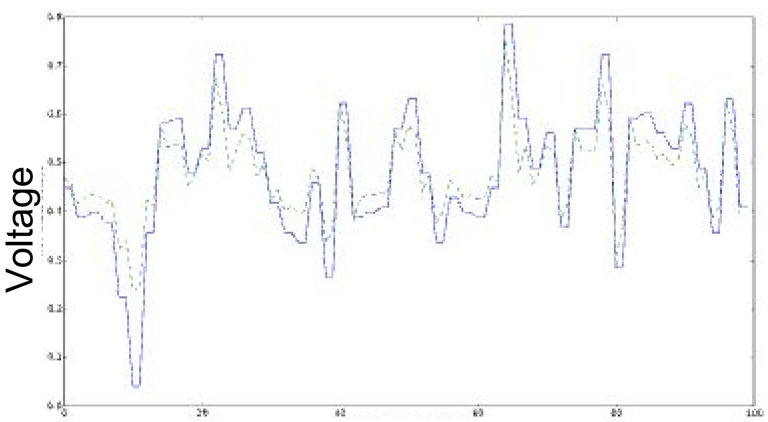
predictive
develop a demo application
for the Electromagnetic Calorimeter
for anomaly detection purposes
using Deep learning technique
(phase 1: see following slides)



Auto-Encoders

proactive
integrate detector metadata (a.k.a. subsystems readiness to take data (Detector Control System LOW /HV voltage)) into the ML-application
for a more omni-comprehensive monitoring of the detector.
Predictivity of hardware failures.
Recursive NN
(phase 2: second half of 2018, not yet started)

E.g.LHC*: predict one step ahead



Recursive NN
blue: real
green: predicted

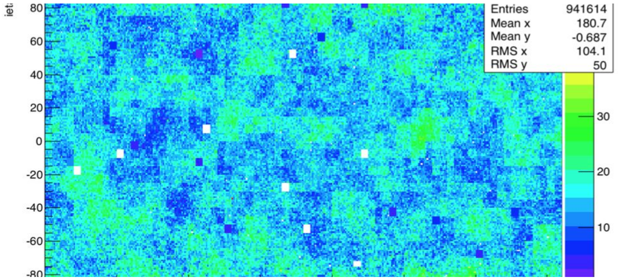
* monitoring the LHC magnets - DS@HEP
https://indico.fnal.gov/event/13497/

ECAL(phase 1) : from rules to unsupervised interpretation

ECAL DQM plots divided into:

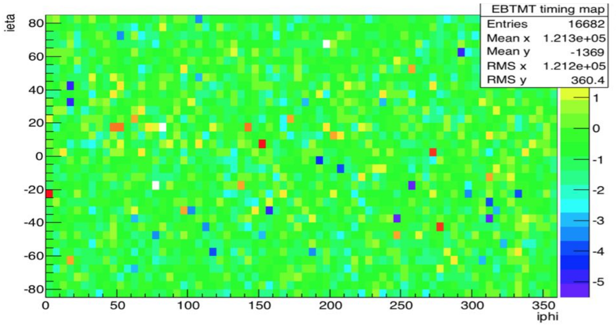
Task histograms: purely statistical description

ECAL BARREL rechit occupancy



Client histograms: provide quality interpretation

ECAL BARREL timing map



Rule-based DQM

. Quality thresholds based on basic operating envelopes of detector

. Difficult anticipate every way the detector can go wrong, especially with different levels of granularity

. More rules ⇒ more code complexity

ML-based DQM

. Unsupervised learning of quality interpretation

. Could potentially flag any “unusual” looking features in Task-histograms and interpret it accordingly in Client-histograms

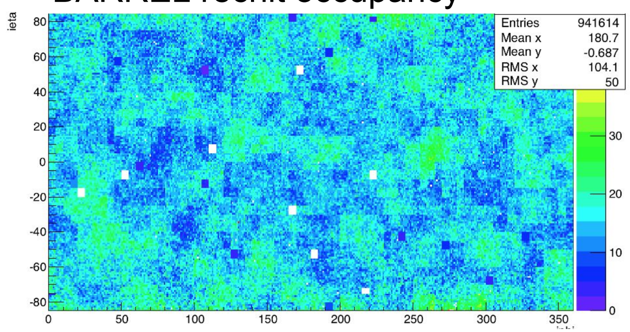
. Could eliminate need for hand-coded rules

Method, feature, dataset and metric

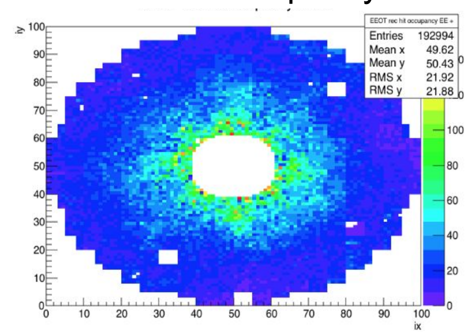
Semi-supervised learning:
Auto-Encoder with convolutional layers, in framework Keras library (tensorflow backend), trained on normal instances* only assuming imbalance of normal/anomalous instances

Feature:
Monitoring temperature maps of the ECAL detector: rechit occupancy and timing plots

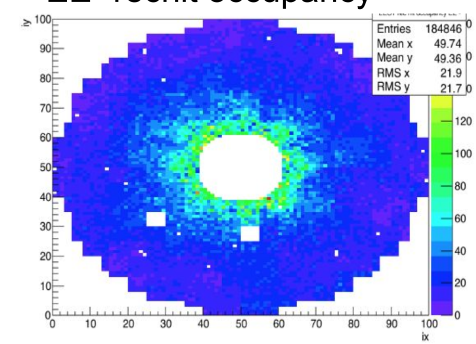
BARREL rechit occupancy



EE+ rechit occupancy



EE- rechit occupancy



Dataset: emulation of the online DQM running conditions, producing one sample (set of images) per LumiSection**

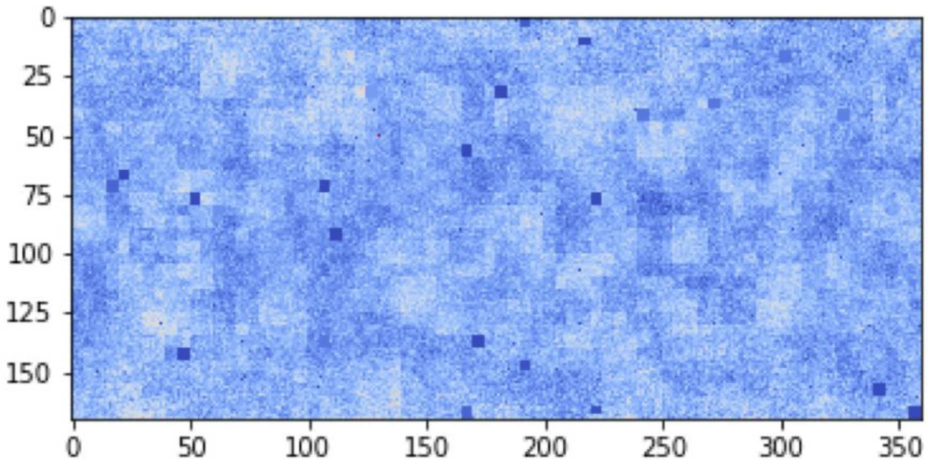
Loss is the metric :
Good instances should be reconstructed with low loss
Bad instances should be reconstructed with higher loss

*Current dataset : ~40000 samples from 2016 data from lumisections marked as good
** Lumisection: minimum quantum in data taking time

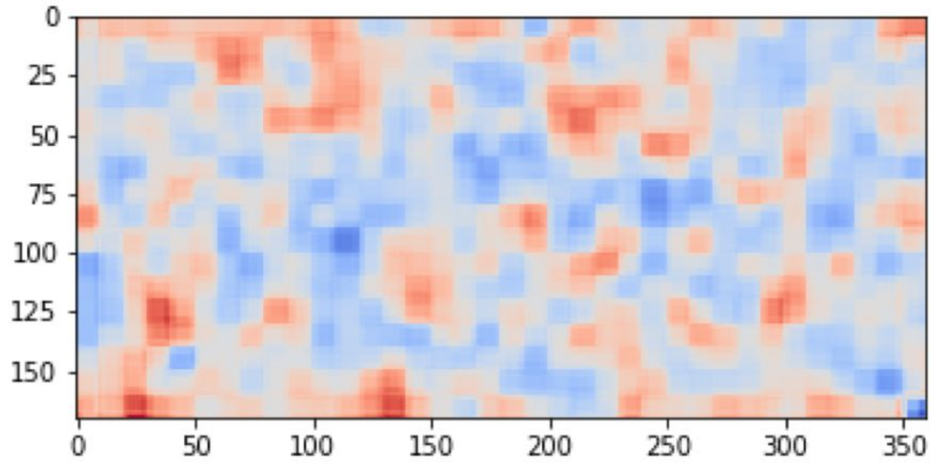
AE Model: GOOD input and output

ECAL barrel rechit occupancy map:

GOOD input image



AE model output, image reconstruction



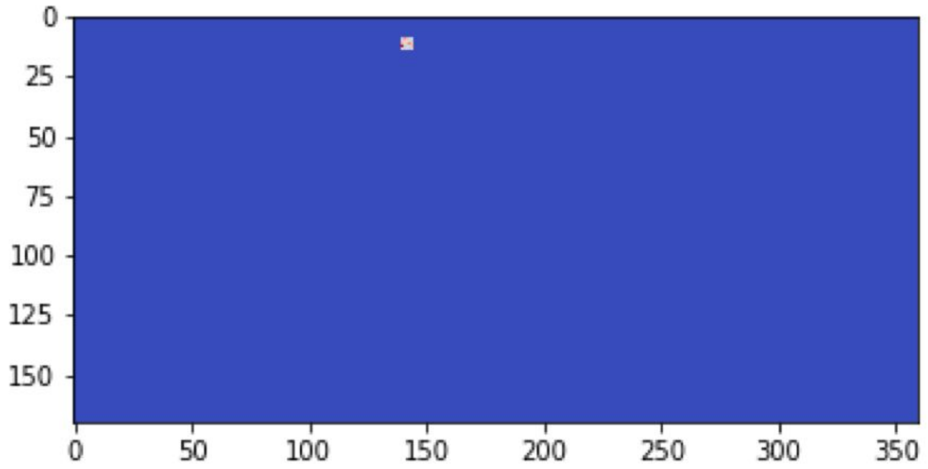
Model seems to generalize well as far as reconstructing images are concerned.
similar testing and training loss spectrum

We want in reality detect anomalies → (next slide) we look at BAD input images

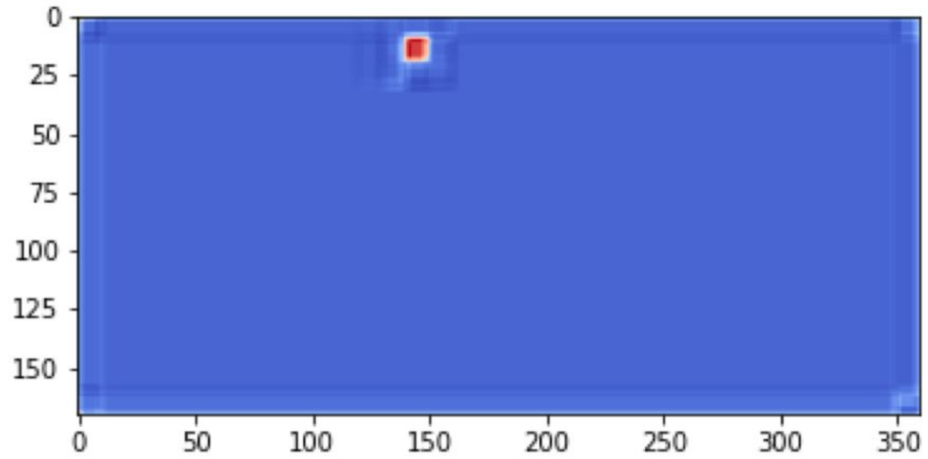
AE Model 0: BAD input and output

ECAL barrel rechit occupancy map:

BAD input image (Ecal hot tower*)

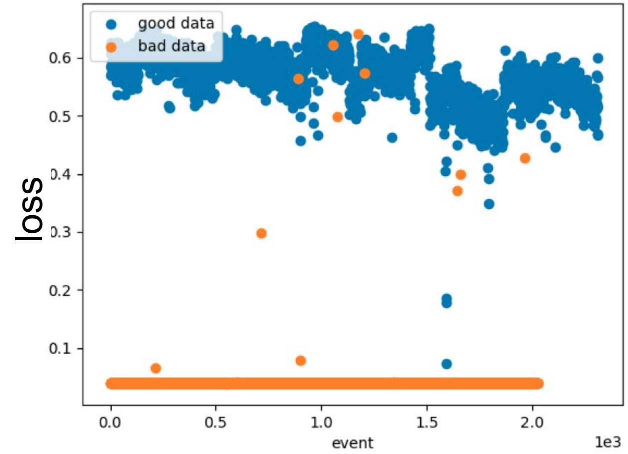


AE model output, image reconstruction



Model is able to 'detect' hot towers
and
reconstructs images containing hot towers
with a loss spectrum different from good images

Roc(TP_vs_FP)_AUC is >0.99



Missing module(test in backup):
results not so good. Investigating different pre-processing techniques and more sophisticated models

*simulated images

new Models and future tests

. ongoing tests:
model 0: Auto-encoder with convolutional layers, in framework Keras library (tensorflow backend)

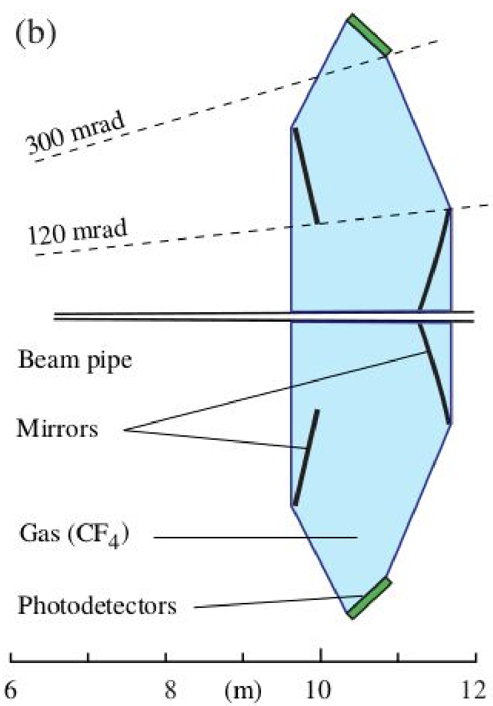
. trained on gpu, batch size of 30
. 60:20:20 split of train:test:validation
. Patience =5
. Optimizer: Gradient Descent (learning rate 0.01),
. Loss Function: Binary-Cross Entropy loss

model 1: more layers, small batch size (to avoid memory gpu bottlenecks)
No decrease in loss, trains faster probably due to smaller batch size
model 2: more more layers, less pooling in the NN
obligation to increase patience to reach the same performance of less layers

. future steps:
increase training size, possibility to include 2017 data
include anomalous examples and evaluate model on them
test more sophisticated networks (e.g. bigger autoencoders with sparsity constraints), use other images besides occupancy(e.g. timing), compare performances with other supervised technique (e.g. SVM)

LHCb-IBM partnership

The Ring-imaging Cherenkov (RICH) detectors determine the velocity of particles coming from proton-proton collisions at the LHCb detector at CERN. When particles pass by a C4F10 radiator gas, they emit cones of photons whose angle is linked to the particle speed and particle type. These photons are reflected in two mirrors prior to being detected in Hybrid Photo-Detectors (HPDs), translating into an array of pixels.



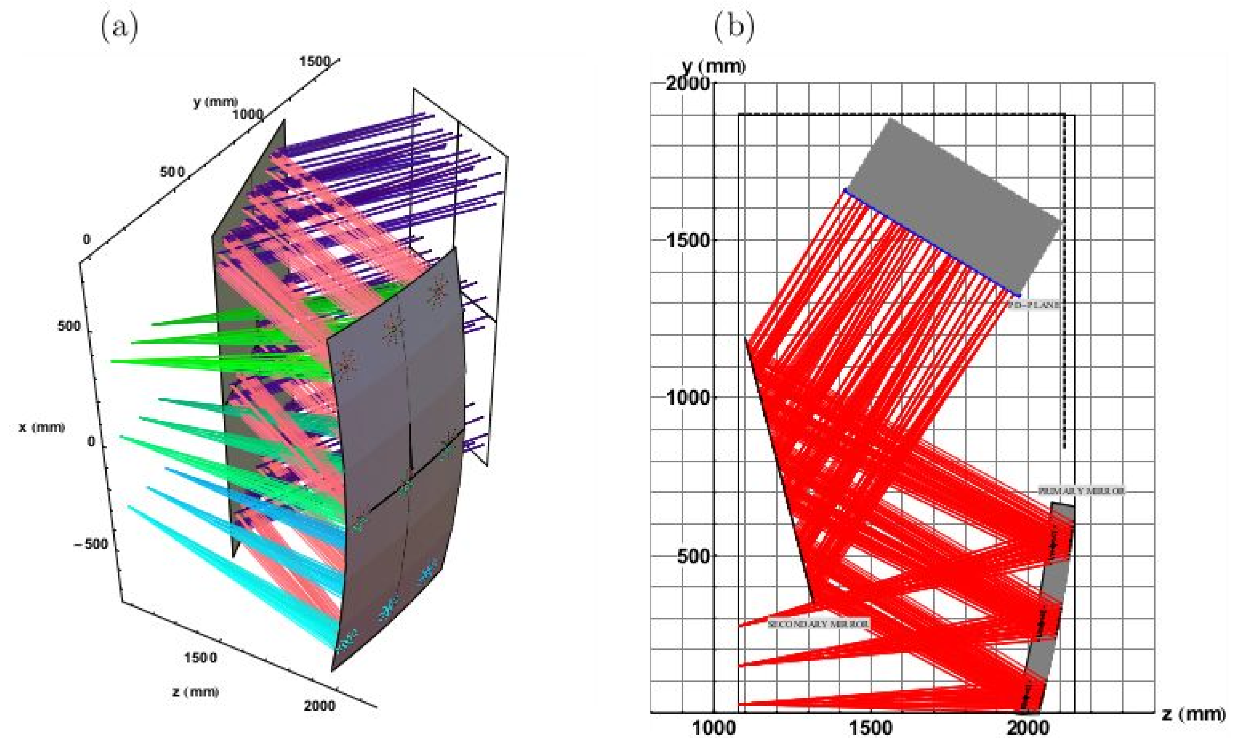
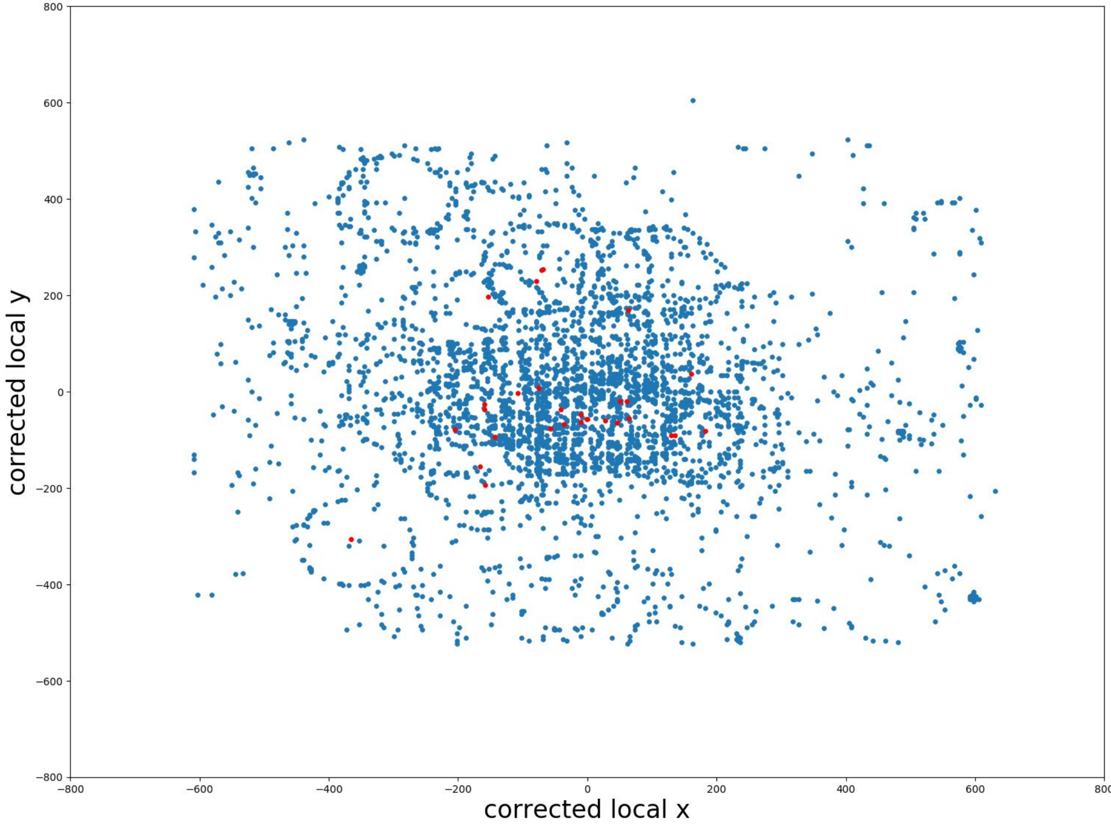


Figure 2.21: Simulated Cherenkov photons in the upper half of the upgraded RICH 1; (a) 3D view, (b) 2D view in the vertical plane.

The RICH classification problem

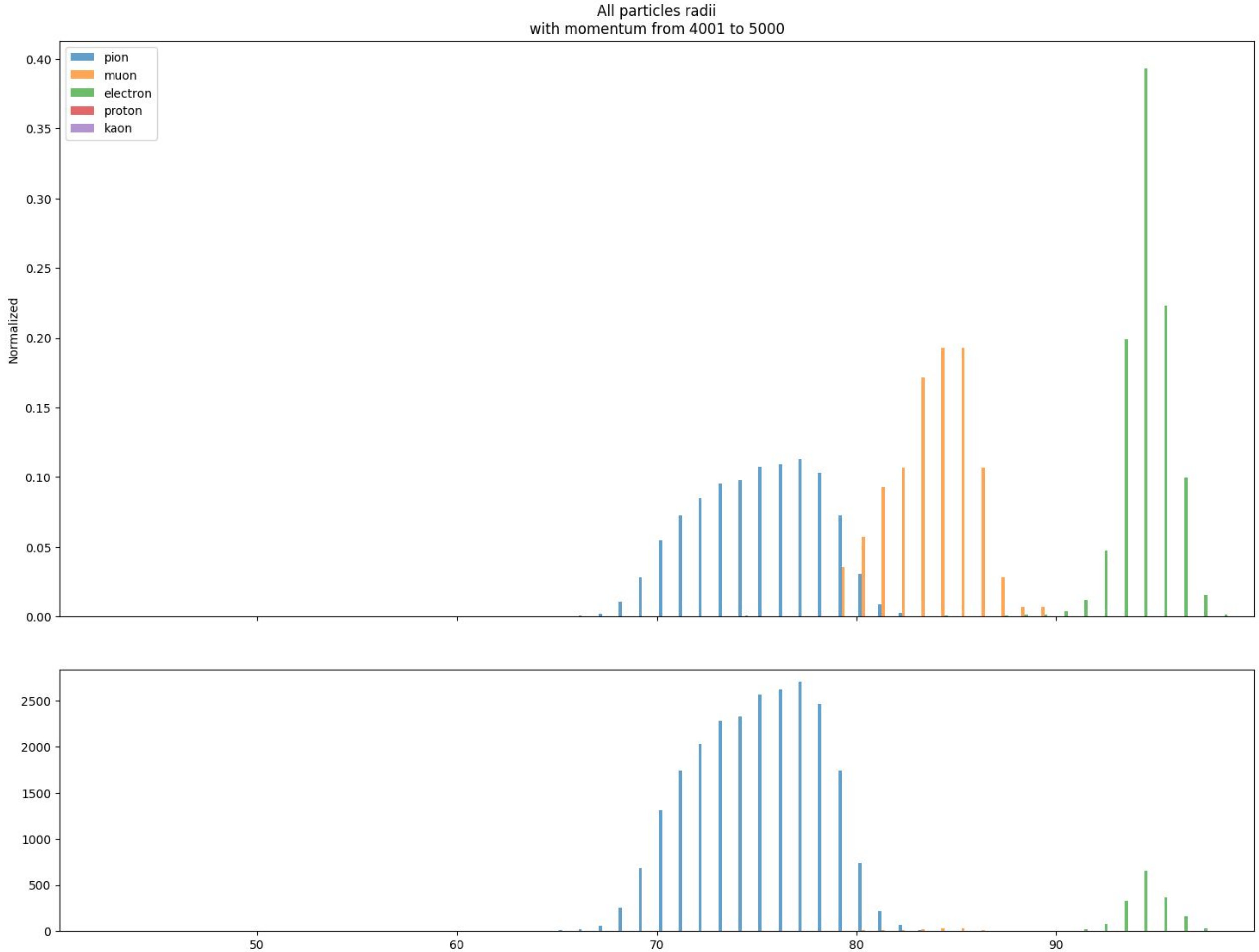
These pixels can then be associated to their corresponding track. The radius of the found circle will determine the particle type (PID).

This problem can then be translated into a classification problem, from the input data (pixels, momenta and track segment “centroid”) into a *particle type* (pion, kaon, muon, proton, electron).



Goal

Over the upcoming year we will study the feasibility of a RICH reconstruction based off AI-driven techniques. A *Convolutional Neural Network* is a good candidate to tackle this problem, separately for each segment, given a momentum cut.



CONCLUSIONS

IBM signed in 2017 the framework agreement for a CERN - IBM collaboration

Future project agreement between IBM, CMS, LHCb and CERN with Openlab as facilitator will be signed soon. It will focus on the 2 main topics:

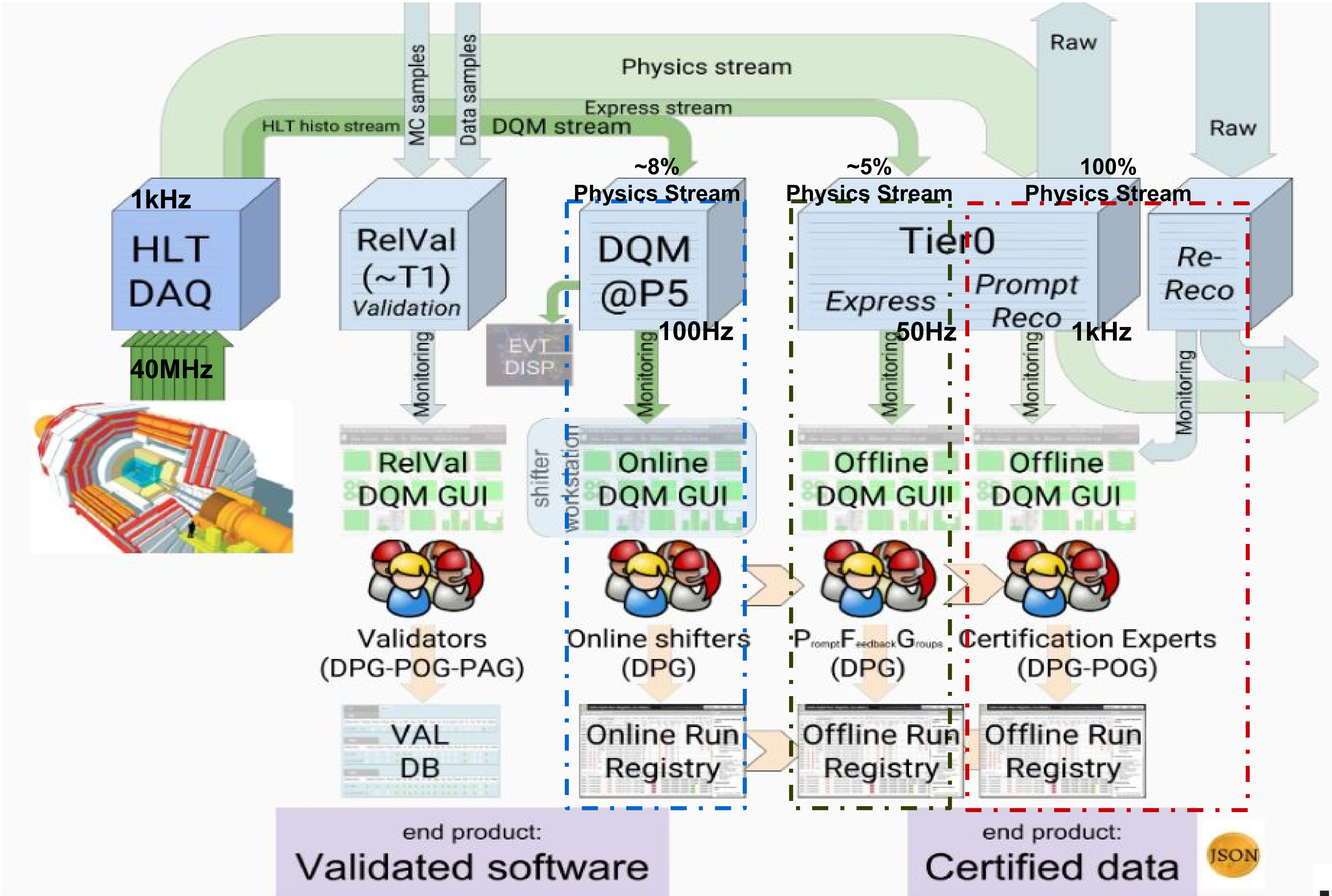
. CMS - IBM: Improve the Data Quality Monitoring operations
.. Looking toward a more performing monitoring:
  . development of a demonstration application
  . inclusion of machine metadata to foreseen hardware failures
… partnership with IBM will benefit on 2 levels:
  . share of powerful hardware for trial use to support CMS ML efforts
  . access to manpower long standing expertise to advise and complement our understandings

. LHCb - IBM: Study the ability to reconstruct RICH with A.I. techniques
.. Installing the server at the moment, just received it
.. Analyzing the data, reading literature, forming a small team
… partnership with IBM will be very beneficial to LHCb as well

Questions?
Contact me: azzolini@cern.ch
Or the e-group: cms-ml4dqm@cern.ch

BACKUP

DQM system used in CMS



Data Quality Assessment

AUTOMATIC QA

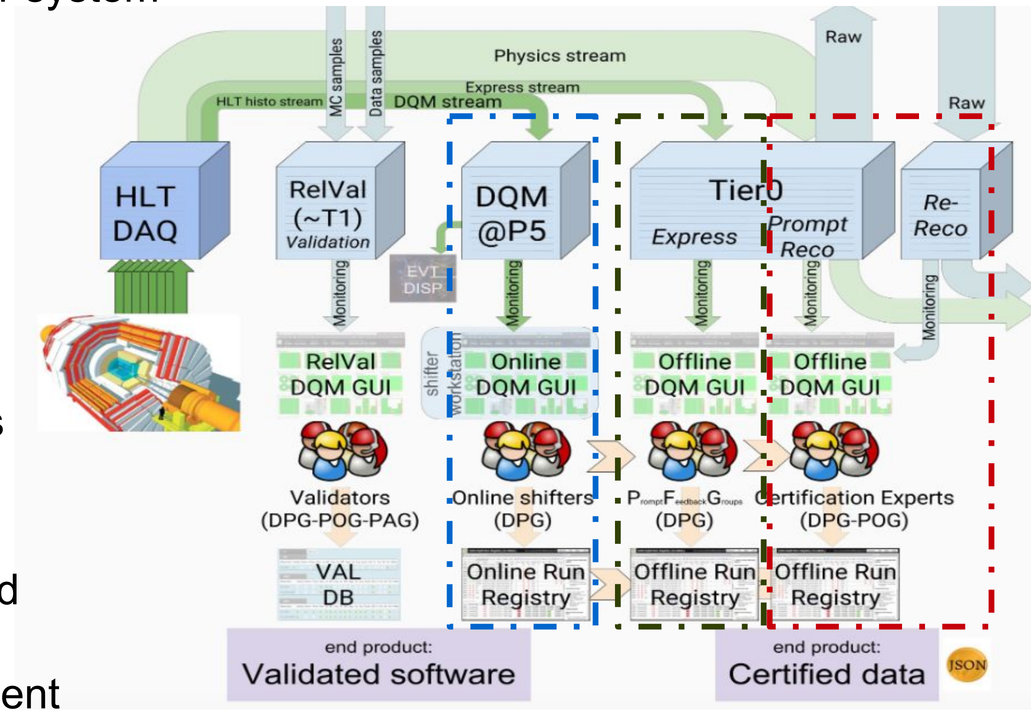
1) near-real-time applications

. fraction of the events with a rate of about 100 Hz
. automatic tests are validated via visual human inspection
. identify problems in the detector and trigger system

OPERATIONAL QA

2) fast reconstruction on a part of data

. subset of the data promptly reconstructed and monitored with ~1h
. goodness of the data regarding also the reconstruction software and the alignment and calibration constants



SCIENCE QA

3) full reconstructed data

. full set of data taken promptly reconstructed and monitored with ~48h latency
. same aim as 2), but typically better alignment and calibration constants are available

3-bis) reprocessed data once per year or at need

. data are again monitored and certified
. same aim as 2) and 3), but typically better reconstruction software and better alignment and calibration constants are available

On the side: release validation on Monte Carlo production,
. validate functionalities and performance of the reconstruction software

What we monitor for Quality Assessment

Online DQM: mostly focused on Hardware level checks

. integrity of the data-format, errors from the read-out electronics

count errors, classify errors, monitor # of errors vs LS

. occupancy of signals (hits) in the various channels

maps and distributions in the detector

presence of noisy/dead read-out channels

. distribution of energy/momentum/time of the signals

. resolution plots, pulls

Offline Data Certification: principally focus on Physics

. detector subsystem:

..Certify the correctness of detector calibration and alignment application,

these conditions are recalculated una tantum, because statistics dependent

Almost same distributions as online

. physics objects (muon, electron, photons, tracks, jets)

.. Monitoring quantities product of the reconstruction, ingredient of future analysis

(# vertices, 3 tracks, energy, typology, topology of the particles, key quantities

Summary and occupancy maps

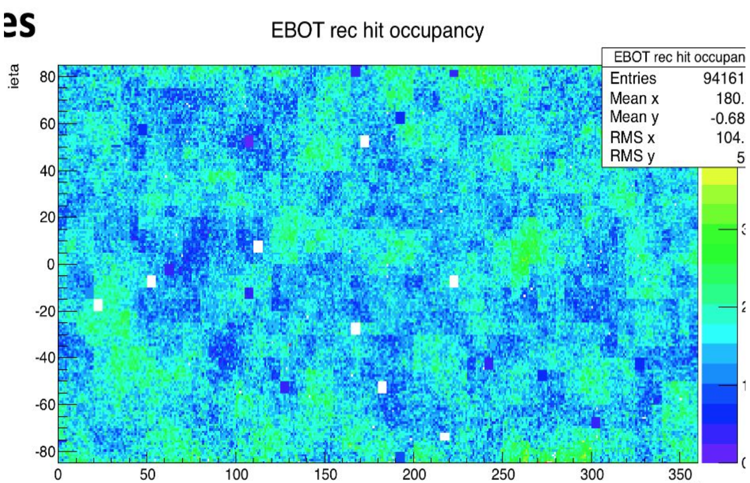
Distribution of quantities used to characterize the candidate particles

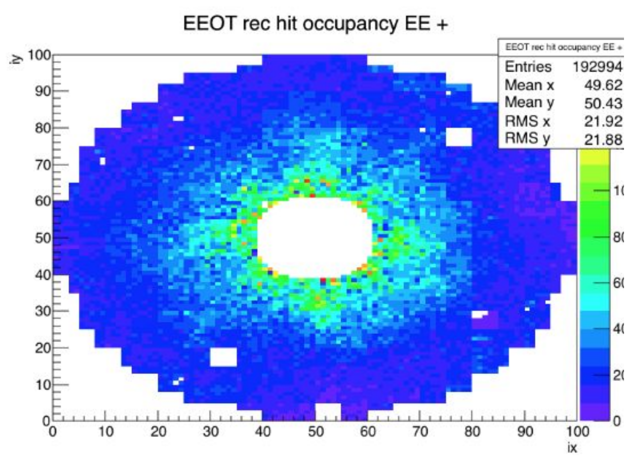
Dataset

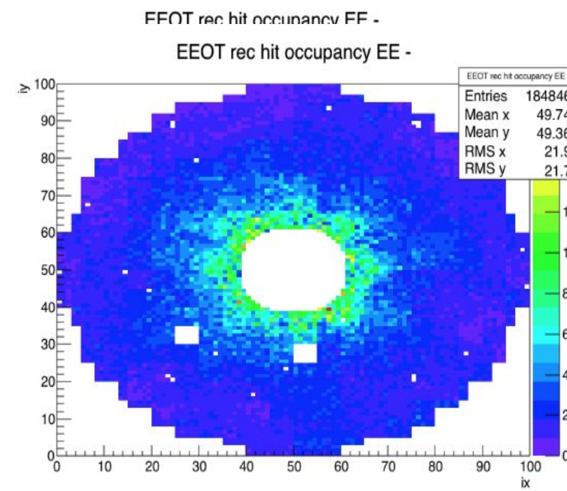
. Current dataset consists of ~40000 samples from 2016 data from lumisections marked as good. (2016 goldenjson , CMSSW_9_2_11) [Thanks to Tanmay & Michael from ECAL DQM team]

. The dataset (SingleElectron/RAW) is processed to emulate the online DQM running conditions and produce one sample (set of images) per lumisection (Most of the RAW data required for this has been moved to TAPE. Was able to acquire only 40k images. Plan to add more images using 2017 data in the near future.)

. The current image set per sample consists of rechit occupancy and timing plots: one for barrel and one each for both endcaps.







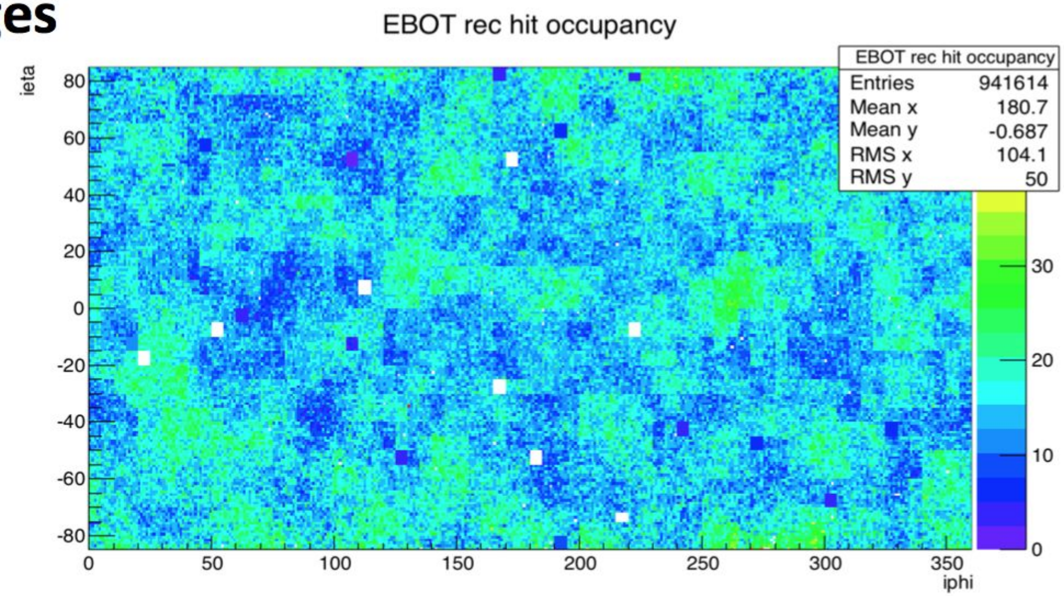
. PREPROCESSING: The only preprocessing that was done was to norma
integral.

The holes in the plot are usually due to permanently masked channels/towers and network can be expected to learn that they are ok.

Input Images

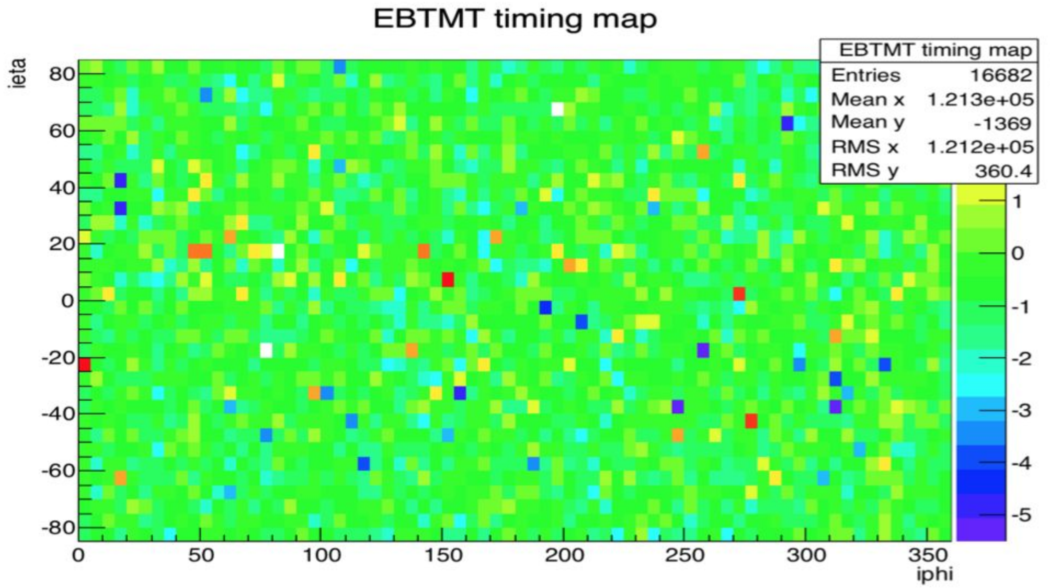
Task histograms

170, 360



Client histograms

34, 72

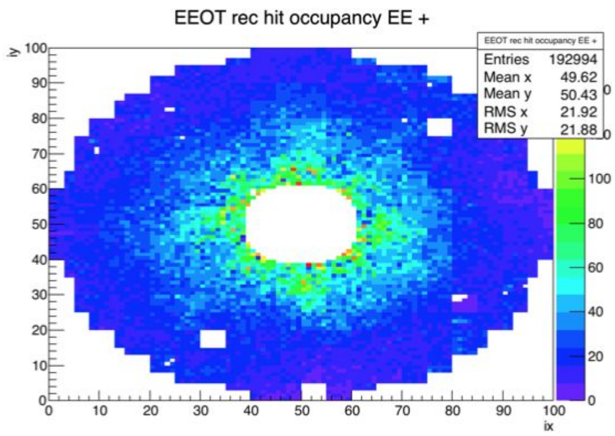


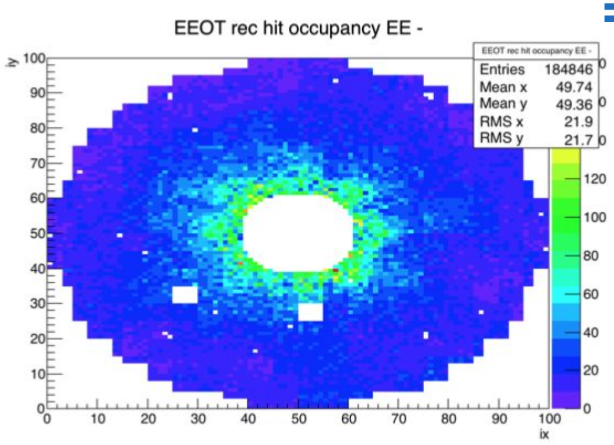
BARREL

Input Images

Task histograms

100,
100

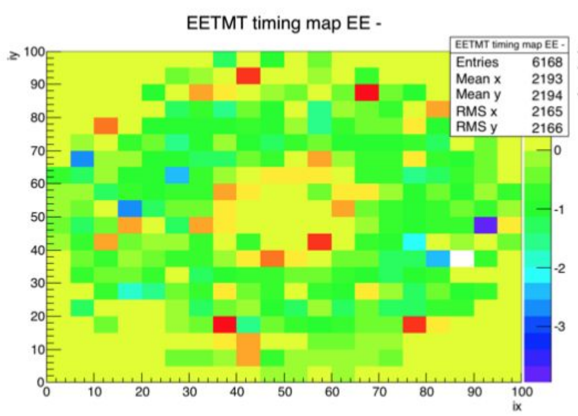


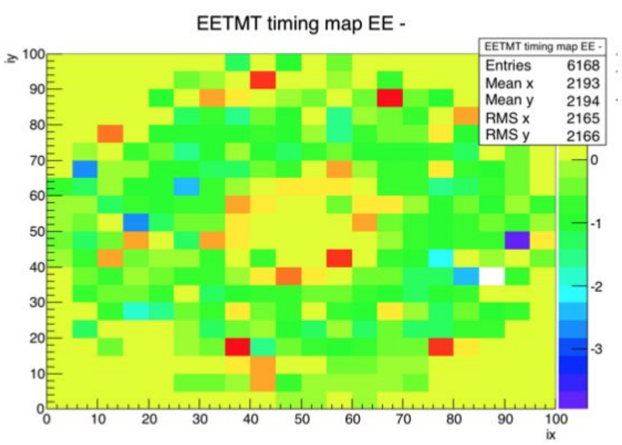


ENDCAPS

Client histograms

20,20

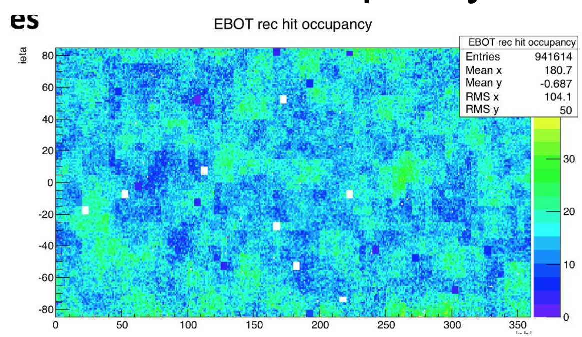


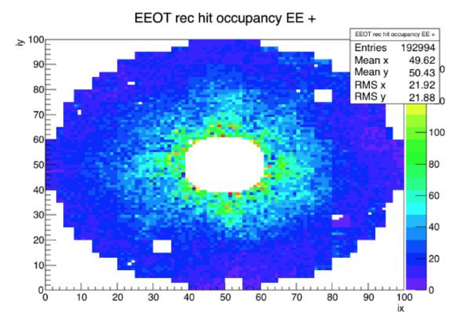


Method, metric and dataset (backup)

Semi-supervised learning:
Assuming that normal instances occur much more frequently than anomalous instances, use only normal instances* to train a convolutional Auto-Encoder (input is mapped to output, the system learns to reconstruct the input with minimum loss) to minimize the loss function.

Monitoring temperature maps of the detector: rechit occupancy and timing plots: one for barrel and one each for both endcaps.





Dataset: emulation of the online DQM running conditions, producing one sample (set of images) per LumiSection**

Loss is the metric :
Good instances should be reconstructed with low loss
Bad instances should be reconstructed with higher loss

*Current dataset : ~40000 samples from 2016 data from lumisections marked as good

** Lumisection: minimum quantum in data taking time

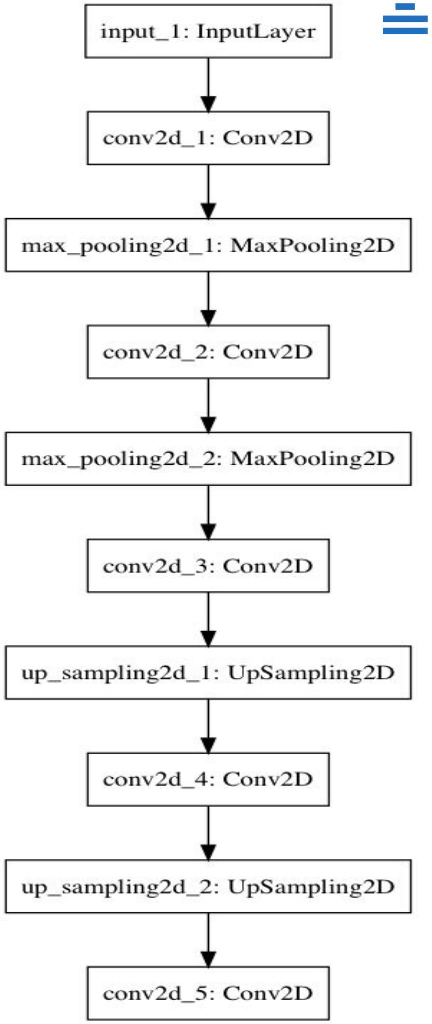
Model 0

- Auto-encoder with convolutional layers

- Conv2D (8 channels,(3x3) patches)→ MaxPooling2D(2, 2)→Conv2D (8 ,(3x3))→ MaxPooling2D(5, 5)→Conv2D (8 ,(3x3))→ UpSampling2D(5, 5)→Conv2D (8 ,(3x3))→ UpSampling2D(2, 2)→Conv2D (8 ,(3x3))

- All Conv layers are ‘padded’ to keep size of output channels same as input.

- FrameworkUsed:
  Keras library using tensorflow backend



Model 0: training performance and Loss

. Auto-encoder with convolutional layers, in framework Keras library using tensorflow backend

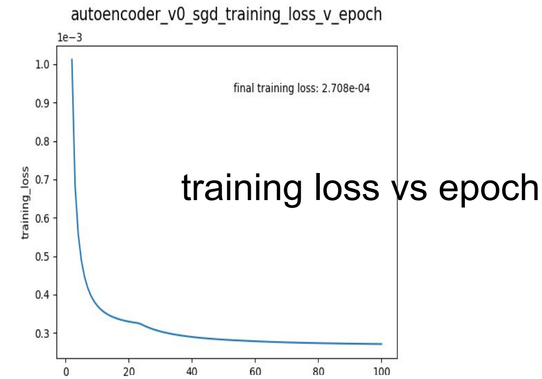
. Trained on gpu, batch size of 30

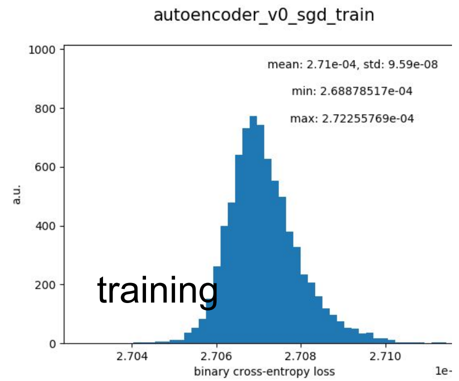
. 60:20:20 split of train:test:validation

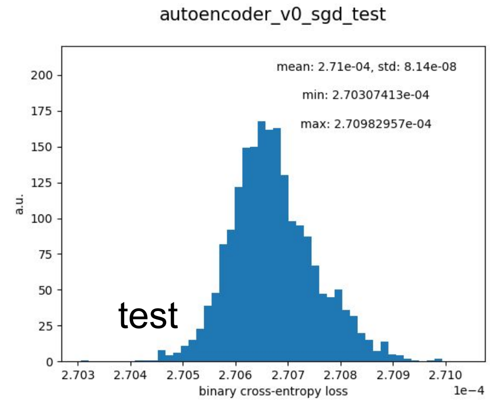
. Patience =5

. Optimizer: Gradient Descent (learning rate 0.01).

. Loss Function: Binary-Cross Entropy loss



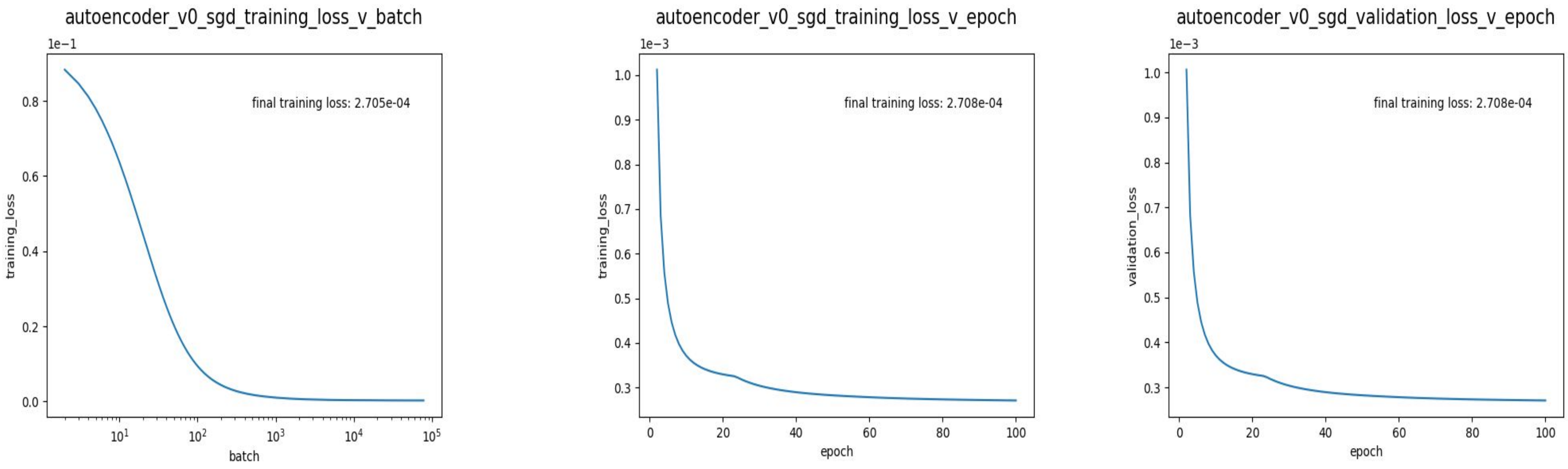




<u>ongoing tests</u>: small batch size, more layers, less pooling, bigger patience
<u>future tests:</u> increase training size, evaluate model on anomalous examples, test more sophisticated networks, use other images besides occupancy(e.g. timing), compare performances with other supervised technique

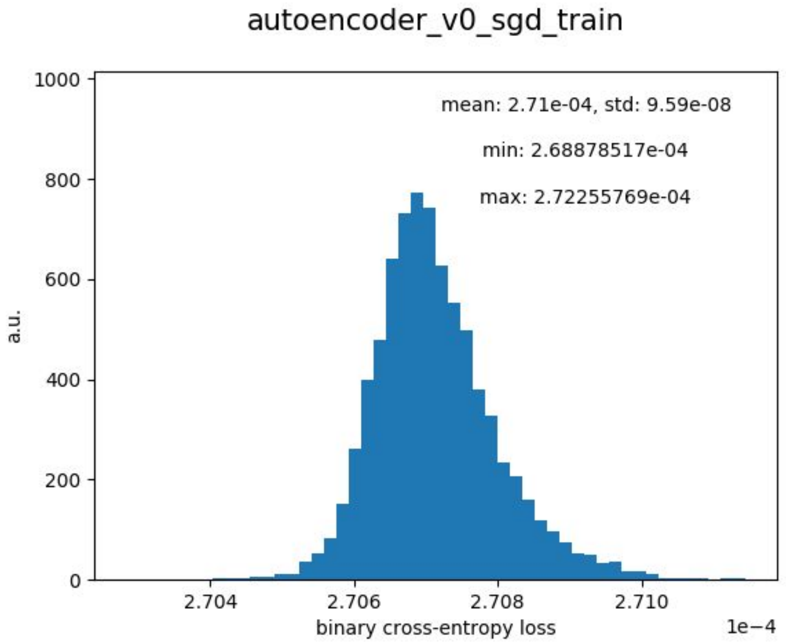
Training performance

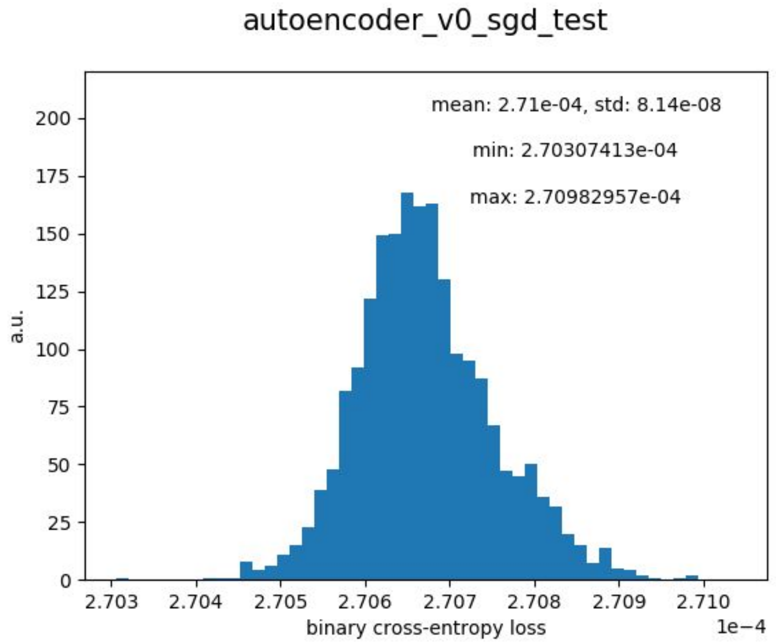
- ➢ Trained on gpu, batch size of 30
- ➢ 60:20:20 split of train:test:validation
- ➢ Patience =5 (Number of epochs to wait in which validation loss doesn't decrease by a minimum threshold (0.05%) before stopping training)
- ➢ Optimizer: Gradient Descent (learning rate 0.01), Loss Function: Binary-Cross Entropy loss



Loss as metric

➢ Evaluate trained model over (each sample) training and test sets and histogram the reconstruction loss

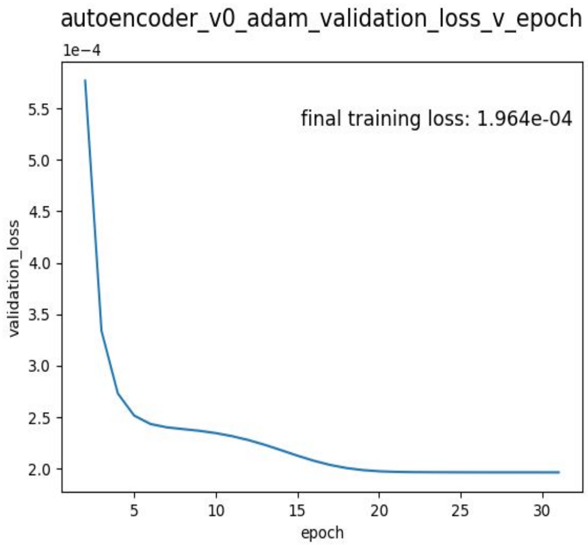


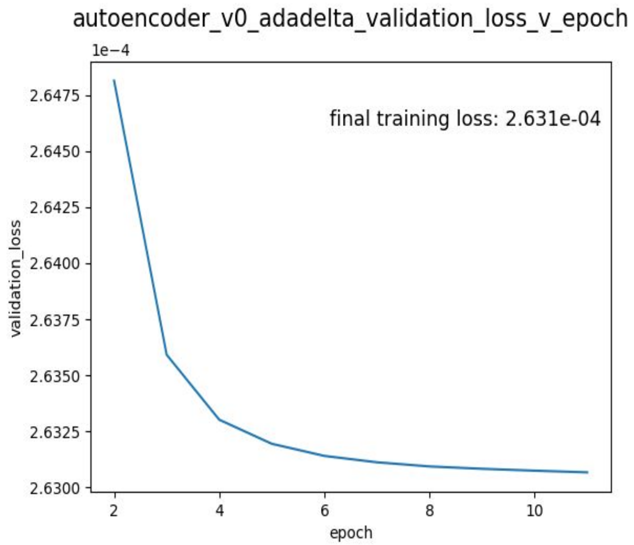


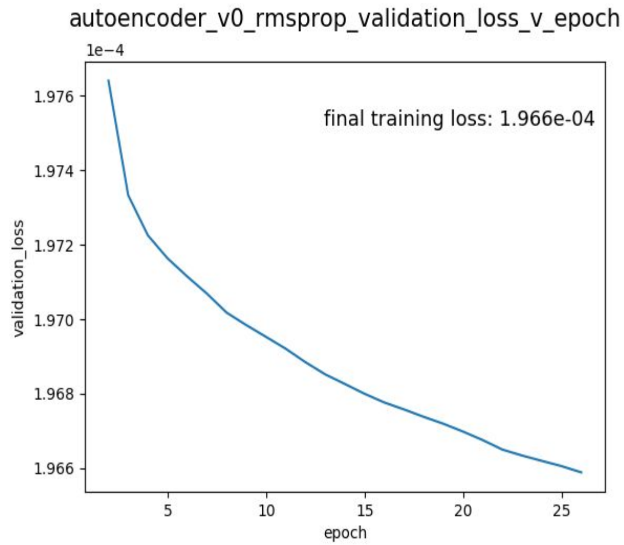
➢ Training and test sets have similar performance.

Choosing a better optimizer

- ➢ Tried several other optimizers available within the KERAS library: ADAM, ADAM with Nesterov, RMSPROP, ADADELTA etc.

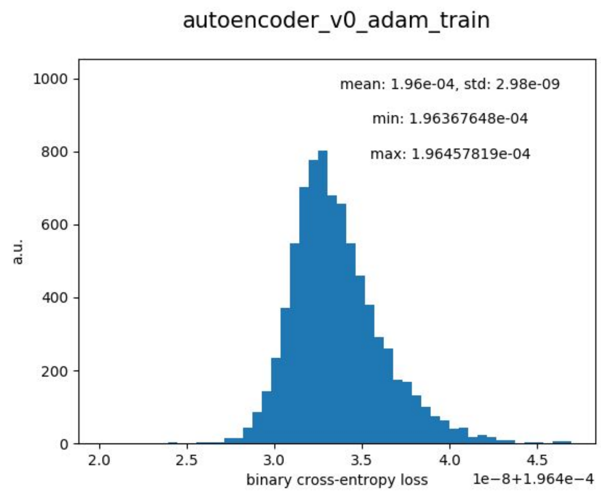


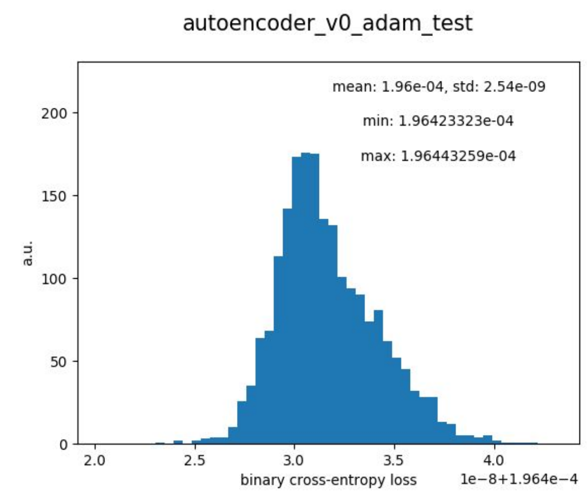




- ➢ Chose ADAM optimizer based on validation loss, rmsprop has similar performance

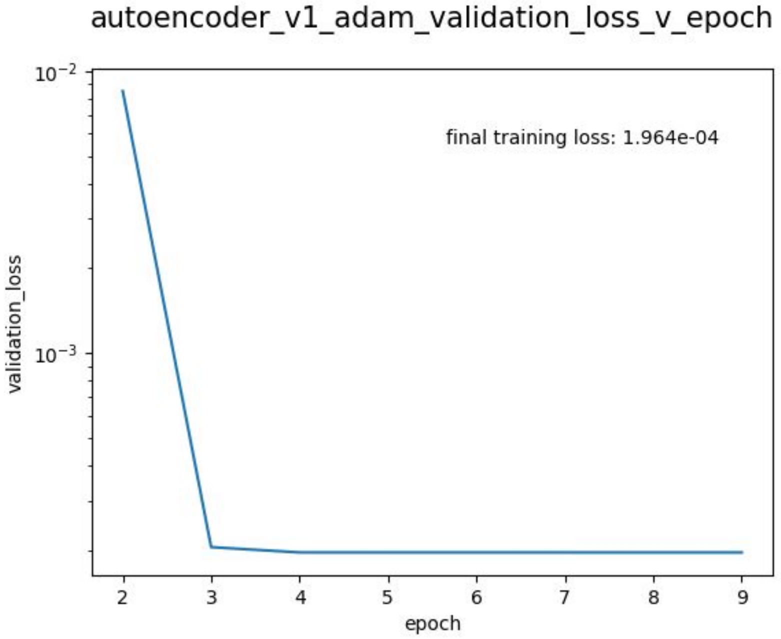
- ➢ Training and test sets have similar performance.

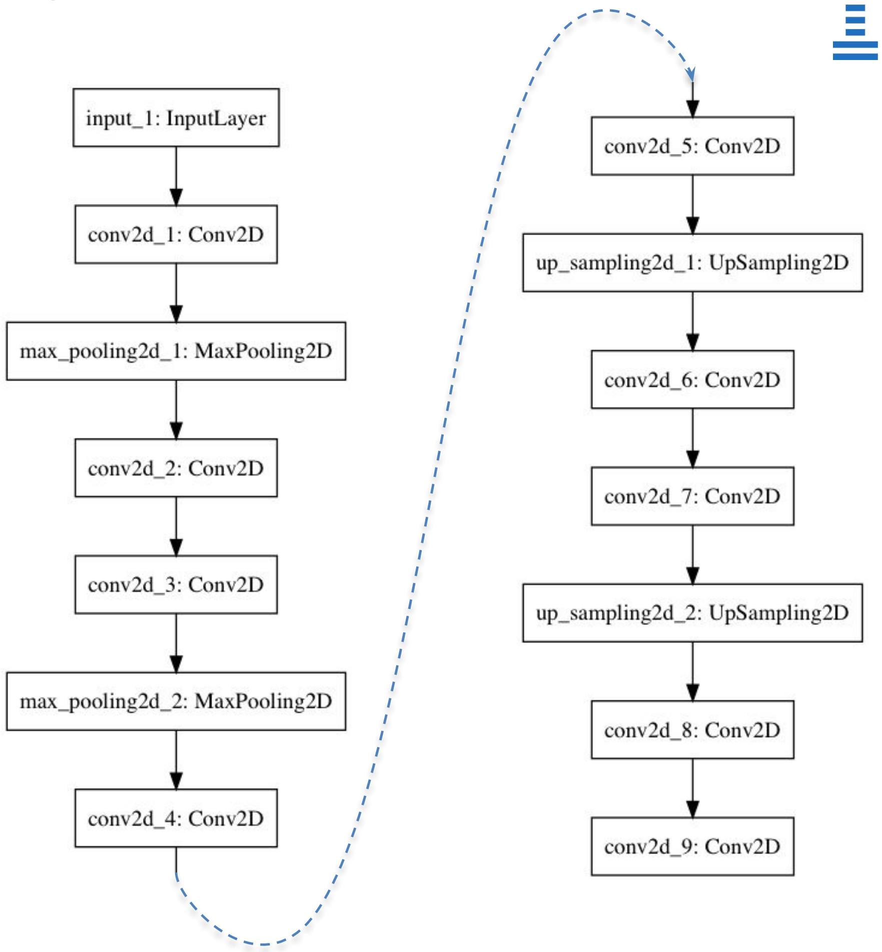




Adding more layers: model 1

- Had to decrease batch size to 20, was hitting gpu memory bottleneck (I think)
- No decrease in loss, trains faster probably due to smaller batch size.

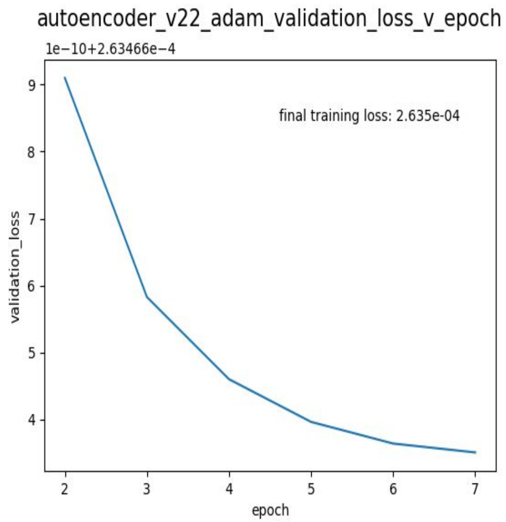




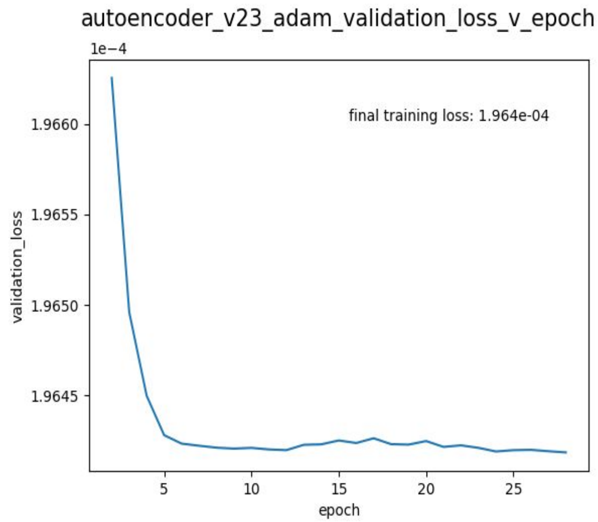
Even more layers, less pooling: model 2

➢ Final train/val loss is worse with respect to model version 1 , with the same patience

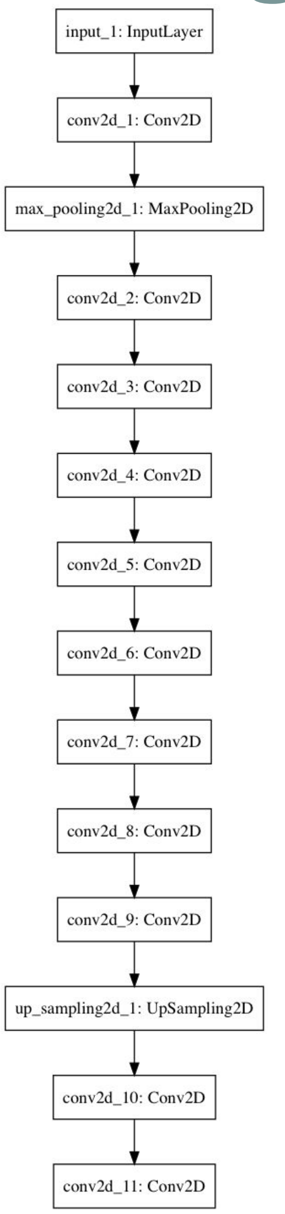
➢ Increasing patience helps decrease loss to similar value as mode v1.



Patience=5



Patience=20



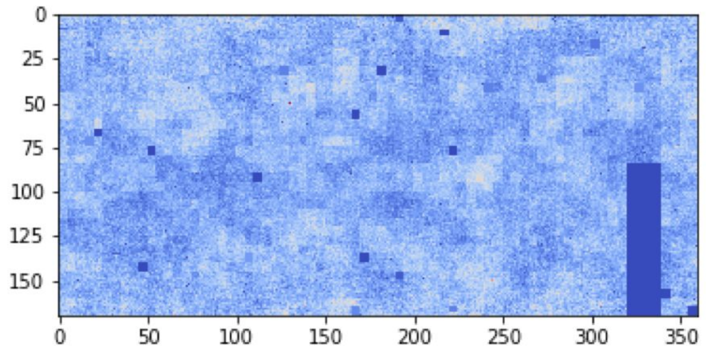
Next Steps

- ➢ Gather some anomalous examples and evaluate the model on them
- ➢ Compare their loss spectrum to that of normal examples
- ➢ Increase training set size.
- ➢ Try more sophisticated networks: bigger autoencoders with sparsity constraints etc.
- ➢ Use other images besides occupancy (e.g. timing) as input.
- ➢ Try other (supervised) learning techniques (e.g.- SVMs) and compare performance.

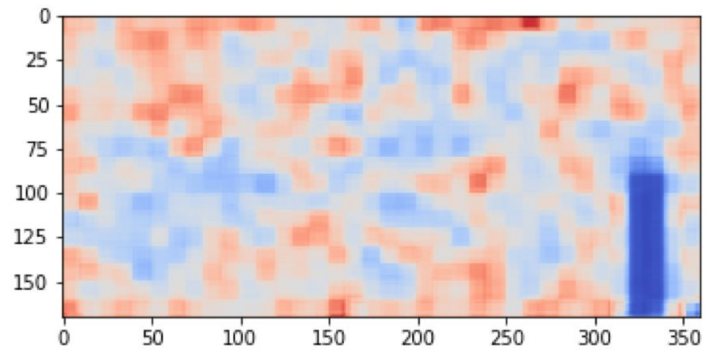
AE Model 0, BAD input and output

ECAL barrel rechit occupancy map:

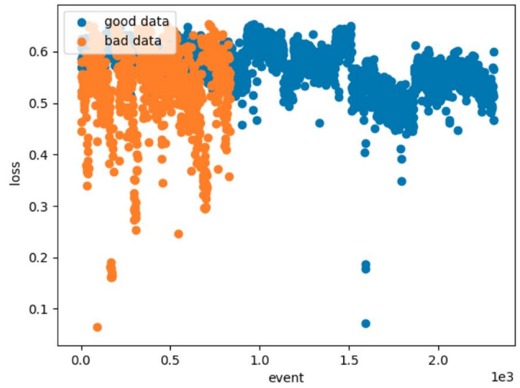
Bad input image (Ecal missing module)



AE model output image reconstruction



loss spectrum over several such images* vs loss spectrum of good test images
reconstructed loss spectrums
are not that different,
doesn't really succeed in catching
the missing modules



Results is not correct:
reconstruction appears to blur out the edges of the missing module but overall does a similar job as when reconstructing good images. investigating different pre-processing techniques and more sophisticated models

CMS - IBM partnership

Maximize the best Quality Data for physics analysis

Improve Data Quality Monitoring (DQM):

Monitors and ensures data quality of each data
Measuring data properties
Anomaly detection