CMS Workflow Failures Recovery Panel Towards AI-assisted Operation

Christian Contreras (DESY), Jean-Roch Vliman (Caltech), Thong Nguyen (Caltech), Matteo Cremonesi (Fermi National Lab), Daniel Abercrombie (MIT), Paola Rozo Bernal (Univ. de los Andes), Allison Hall (Notre Dame) , for the CMS Collaboration

Computing in High-Energy Physics (CHEP)
July 9th - 13th, 2018

Abstract

The central production system of CMS is utilizing the LHC grid and effectively about 200 thousand cores, over about a hundred computing centers worldwide. Such a wide and unique distributed computing system is bound to sustain a certain rate of failures of various types. These are appropriately addressed with site administrators a posteriori. With up to 50 different campaigns ongoing concurrently, the range of diversity of workload is wide and complex, leading to a certain amount of mis-configurations despite all efforts in request preparation. Most of the 2000 to 4000 datasets produced each week are done so in full automation, and datasets are delivered within an agreed level of completion. Despite effort of reducing the rate of failure, there remains a good fraction of workflows that requires non trivial intervention. This work remains for computing operators to do.

Introduction and Motivation

CMS production manages thousands of "workflow" tasks each with thousands of jobs. Common issues are errors in grid jobs may be due to missing, corrupt input files, high memory usage, etc. Workflows that suffered from similar failures are bundled and presented as such to the operator. An operator must look at the error codes and decide on what appropriate actions to take on the workflow. An algorithm that encompasses all possible patterns that can be anticipated would be difficult to program and - most important - impossible to maintain.

Machine learning stands as a very natural solution

Goal and Strategy

- Deliver a CMS workflow failures recovery panel towards AI assisted operations.
  - Move error handling from manual operator intervention into automated actions
- Use supervised learning to build multi-class classifier that predict recovery action (kill, clone, resubmit, recover as appropriate) and number of job splits

Data Processing

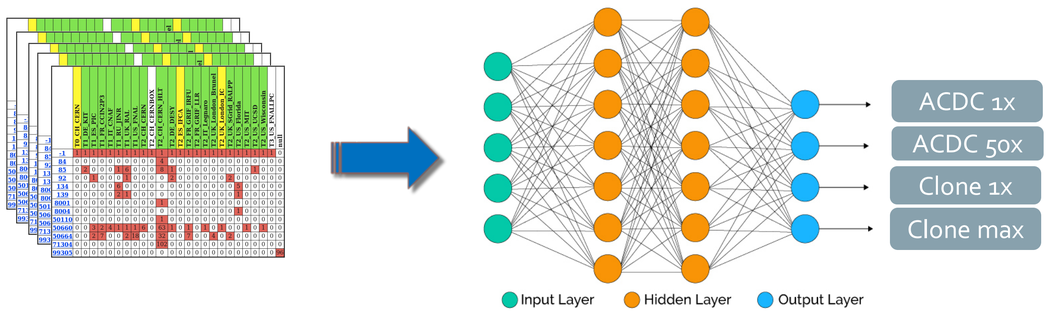
- Data pulled from CMS services using **Workflow Team Web Tools** which extracts information from Site-Readiness report
  - Site availability maintained by the site support team, at the time the workflow was reported as needing assistance
- For each task (*workflow+campaign*), we know the number of times each possible error code is thrown at each site
  - Use the **"exit codes - site status"** as input information
  - This leads to a sparse matrix of numbers of error codes per sites, with each element being the number of matching errors

Method and Pipeline

We consider a simple **feed-forward multilayer perceptron** composed of a series of dense layers with relatively small numbers of nodes. The input represented as sparse matrices of number error codes per sites are used to train the network.

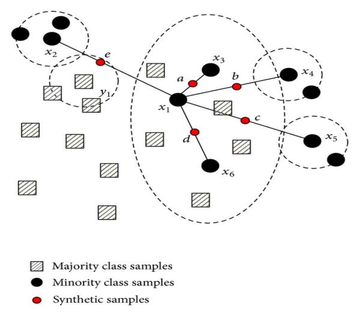
Cascade model training approach

- First model builds binary classification between full and partial recovery targets
- Second model builds a multi-class classification of all class targets



Handling Imbalanced Class Distribution

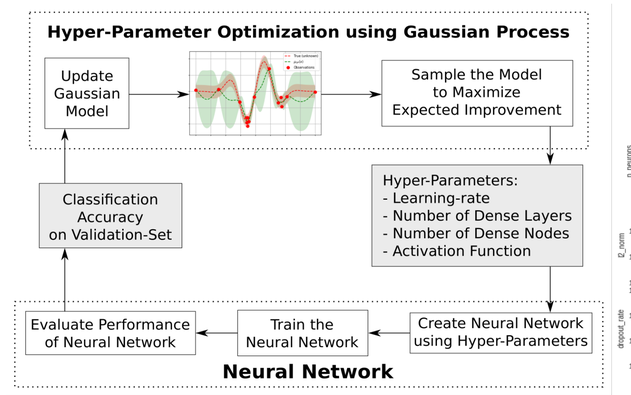
A dataset is imbalanced if the classification categories are not approximately equally represented. We studied the use of **Synthetic Minority Over-sampling** resampling method to deal with highly unbalanced datasets. It consists of under-sampling the majority class and adding more synthetic examples from the minority class.



Model Tuning with Bayesian Optimization

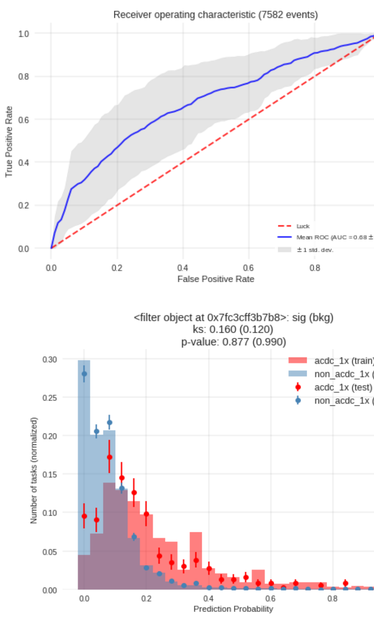
Hyper-parameter selection is crucial for the success of the neural network. We need an optimization method that can search for hyper-parameters as efficiently as possible. **Bayesian optimization (BO)** gives a new suggestion for hyper-parameters in a region of the search-space that we haven't explored yet that brings the most improvement.

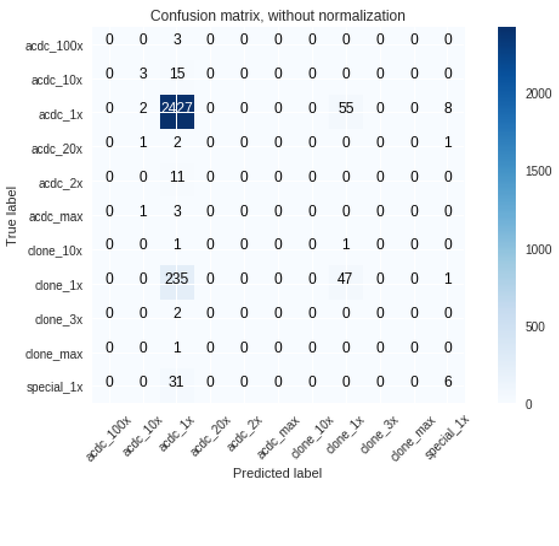
- Applied BO using Gaussian Process to model the surrogate and optimized the Expected to search-space for hyper-parameters



Model Performance

- Data split into 70%/30% for model training and testing, respectively
- Both variable scaling and data resampling with SMOTE applied
- 15-fold **Cross-validated ROC curves** to check the variability of the binary classification prediction given the amount of data used to train the model
- The **confusion matrix** for multiclass classification is a good way of looking at how good our classifier is performing when presented with new data





Model type	accuracy	Recall	Precision	Log-loss	F1-score
Binary	87%	87%	83%	40%	83%
Multi-class	87%	86%	83%	56%	84%

Summary and Future Work

A first pass for the supervised learning in error handling prediction. The operator's procedure will be automatized further by applying the decisions that are predicted with acceptable confidence.

Improve current WTC web interface

- To start using Machine Learning Model
- Include the prediction for recommended action
  - Start recovery from trivial cases
  - Monitor performance for model re-training
- Add GUI display for diagnostic summary reports