3rd CMS Machine Learning Workshop
17-19 June 2019 CERN

OSDroid: A supervised ML application for CMS workflow operation support

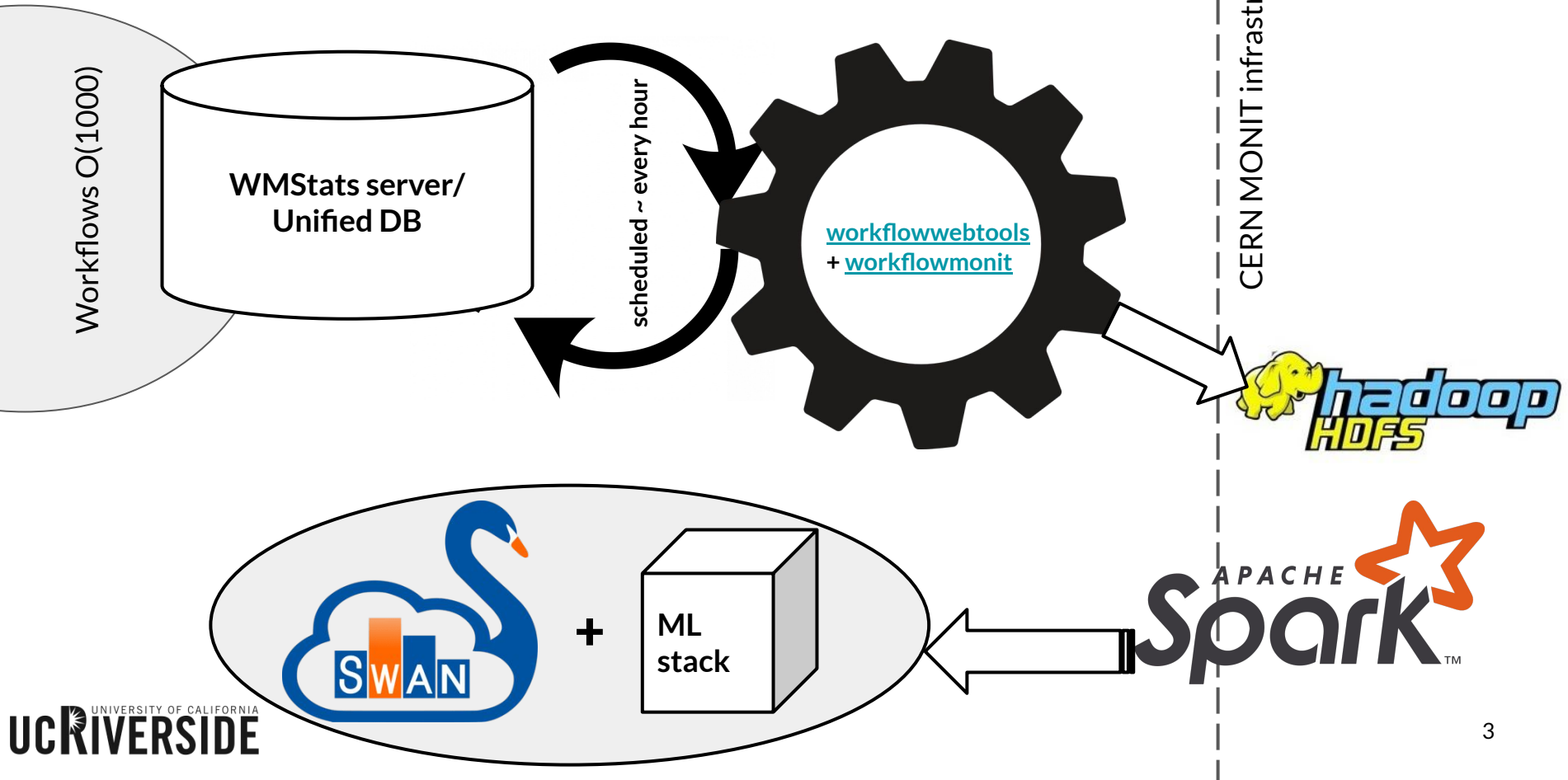
Andres Vargas[1], Daniel Abercrombie[4], Hamed Bakhshiansohi[2], Jennifer Adelman-Mccarthy[3], Lukas Layer[5], Matteo Cremonesi[3], Weinan Si[6]

1: Catholic University of America
2. DESY
3. FNAL
4. MIT
5. Universita e sezione INFN di Napoli
6. University of California, Riverside

Introduction

- In CMS, offline data processing are organized into “**workflows**”.
- Workflows are composed of several **tasks**, which are further splitted into hundreds or thousands of **jobs**, distributed to available sites around the globe.
- Sometimes, jobs/tasks/workflows would fail due to **site problems, job configuration error, job management system glitch…**
- This will cause resource waste, delivery delay, and possibly quality of physics results.
- We have been relying on operators to diligently monitor the system, spot problems, report and remedy the situation.
- As workloads keep increasing, we need a tireless “Jarvis” to assist the operation.

Data collection



Label creation, Data summary

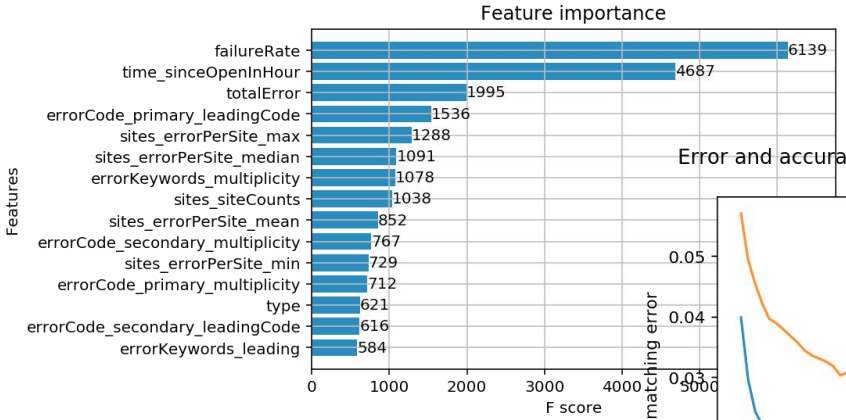
- Depending on the actions operators took after the workflow is done, workflows are categorized with **3** labels: *Good, ACDC-ed, Resubmitted*.
  - source: cmsprodmon
  - query workflows associated with the same PrepId
    - Only 1 ⇒ *Good*
    - >1 and “ACDC” present in one of the names ⇒ *ACDC-ed*
    - >1 and no “ACDC” in any of the names ⇒ *Resubmitted*
- We call a snapshot of a workflow at a timestamp a “**record**”.
- We started collecting data since Feb.2019, so far, we have **17178 records** labelled from **1376 workflows**. The ratio of 3 categories is *Good : ACDC-ed : Resubmitted = 8910 : 4938 : 3330*.
- **20%** of total records are kept as test subset.
- **19** features are extracted for each records. (details in next page)

workflow features

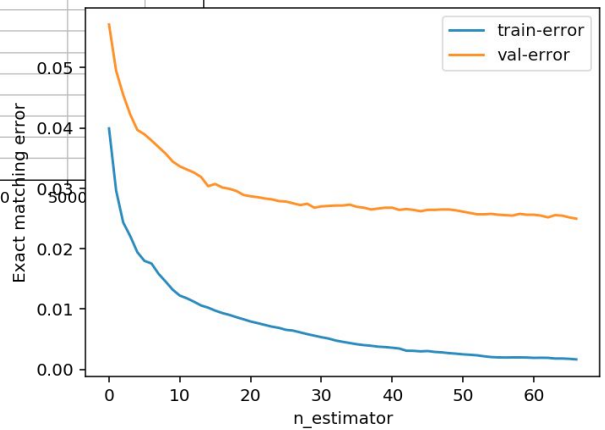
Name	comments
General	
failureRate	-
totalError	total counts of this workflow
type	type of workflow. Label encoded.
time_sinceOpenInHour	Hours since running-open status declared
Site Info	
sites_errorPerSite_max	sum of error counts per site; maximum
sites_errorPerSite_min	sum of error counts per site; minimum
sites_errorPerSite_median	sum of error counts per site; median
sites_errorPerSite_mean	sum of error counts per site; mean value
sites_errorPerSite_stdDev	sum of error counts per site; standard deviation
sites_siteCounts	number of sites reported errors
Error Code Info	
errorCode_primary_multiplicity	multiplicity of primary error codes
errorCode_primary_leadingCode	most frequent primary error code. Label encoded.
errorCode_primary_leadingRatio	fraction of the most frequent primary error code's counts
errorCode_secondary_multiplicity	multiplicity of secondary error codes
errorCode_secondary_leadingCode	most frequent secondary error code. Label encoded.
errorCode_secondary_leadingRatio	fraction of the most frequent secondary error code's counts
Error Keyword Info	
errorKeywords_multiplicity	multiplicity of error keywords
errorKeywords_leading	most frequent error keyword. Label encoded.
errorKeywords_leadingRatio	fraction of the most frequent error keyword's counts

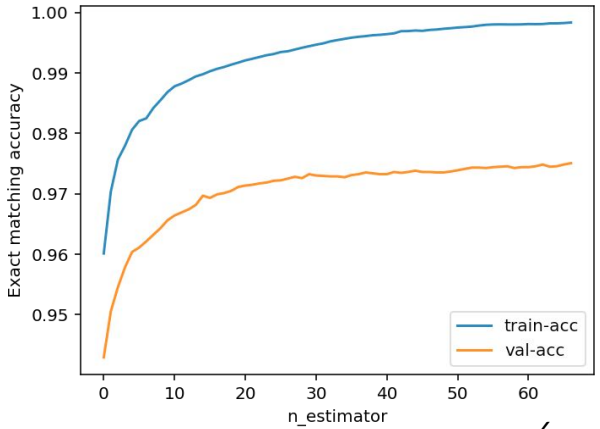
Training #1 BDT

- Trained with XGBoost.
- Bayesian optimization on hyperparameters. (scikit-optimize)
- **97.4%** prediction precision on test subset.



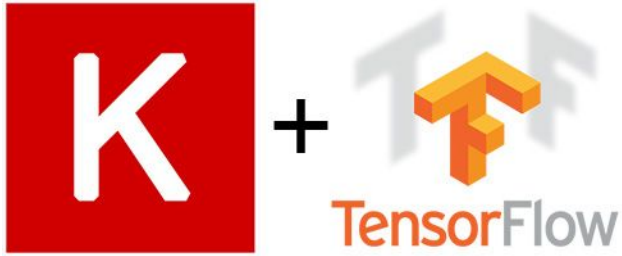
Error and accuracy for training and validation subsets during training process with the best hyperparameters found



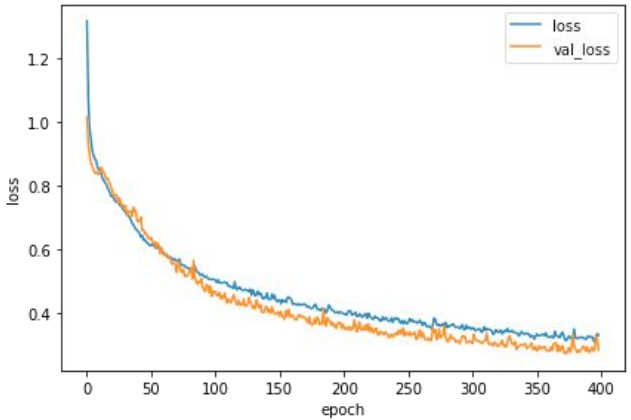


Training #2 DNN

- Trained with Keras (backend TensorFlow).
- Dense deep neural network with dropouts and batch normalization.
- Bayesian optimization on hyperparameters. (scikit-optimize)
- **88.4%** prediction accuracy achieved on test subset.




```
Best parameters:
best_hidden_layers = 2
best_initial_nodes = 85
best_dropout = 0.10919572308469
best_batch_size = 591
best_learning_rate = 0.00056007
```



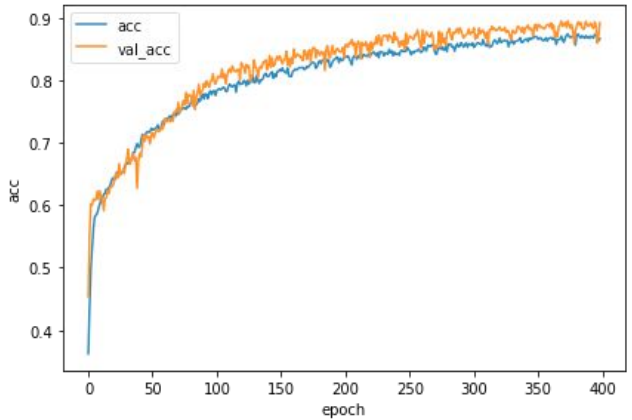


Fig: Loss and accuracy for training and validation subsets during training process with best parameters.

label encoding:
Good: 0, ACDC-ed: 1 resubmitted: 2

Examples

pdmvserv_task_EGM-RunIIWinter19PFCalib17GS-00007__v1_T_190404_133937_5166

(Records collected 36h-267h since running-open declared, with frequency of ~every hour)

resubmitted

BDT

true: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2]

predict: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2]

precision: 229/230

DNN

true: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2]

predict: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 0 0 2 0 2 1 0 0 1 1 2 1 0 0 0 1 2 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2]

precision: 215/230

pdmvserv_task_SUS-RunIIFall17NanoAODv4-00273__v1_T_190412_123605_8505

(Records collected 14h-50h since running-open declared, with frequency of ~every hour)

ACDC-ed

true: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
predict: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

precision: 36/36

true: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
predict: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

precision: 36/36

Summary

- ML is promising to make accurate predictions on the actions that need to be taken for running workflows.
- A supervised approach is taken and seems to be effective.
  - Resubmittd workflows can be predicted several days in advance!
- Two models, BDT and DNN are explored. At first sight, BDT performs better.
- Data is continuously being collected, relying on CERN Monit service. Long term storage is feasible.
- More models and features will be explored.

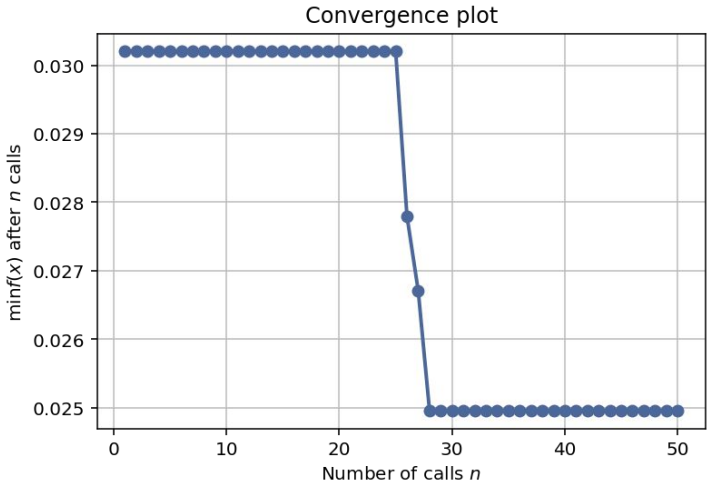
Thanks to Valentin Kuznetsov, Federica Legger and CERN Monit support!

Backup

XGBoost hyperparameter tunning

Best hyperparameters after 50 calls:

learning_rate	0.19219525806919469
min_child_weight	0
max_depth	13
subsample	0.5637243305591189
colsample_bytree	1.0
reg_lambda	6.843491785655634e-06
reg_alpha	1.0
gamma	1e-09
min_child_weight	0
scale_pos_weight	499.99999999999994



Reference of all xgboost hyperparameters:
https://xgboost.readthedocs.io/en/latest/parameter.html