Prototype of Machine Learning "as a Service" for CMS Physics in Signal vs Background discrimination

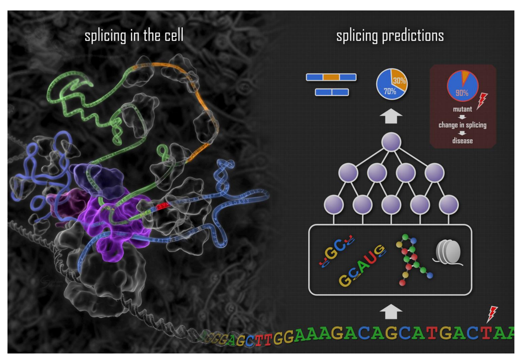
Luca Giommi[1], Daniele Bonacorsi[1], Valentin Kuznetsov[2], on behalf of the CMS collaboration

[1]University of Bologna, INFN-Bologna, Italy
[2]Cornell University, Ithaca, NY 14850, USA

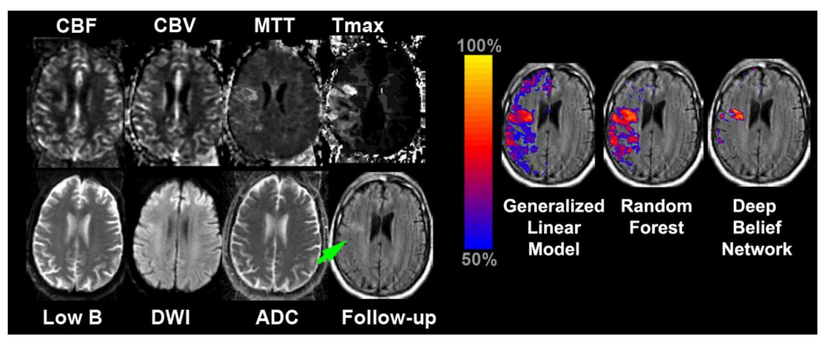
Machine Learning

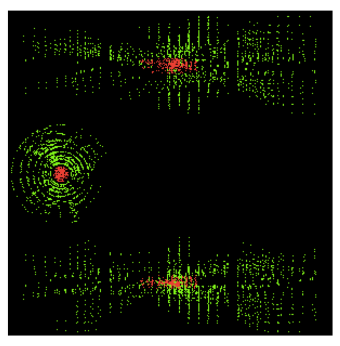
not-HEP

HEP
towards High Luminosity LHC [1]



- Trigger decision
- Tracking
- Detector quality monitoring
- Detector anomaly detection
- Simulation
- Computing resource optimisation [2]
- Physics analysis
- Particle Identification and particle properties



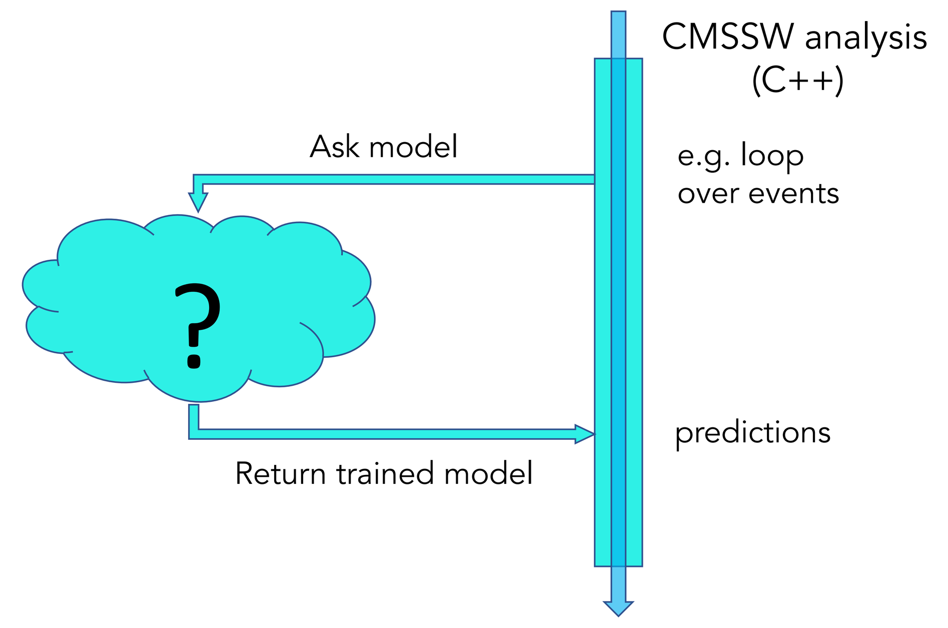


Machine Learning "as a service" for CMS

The CMS experiment at CERN exploits various ML techniques due various physics and computing related projects. The construction and deployment of a **ML project** and its deployment for production use **requires specific skills** and it is a highly time-consuming task. There are no data science teams stably collaborating with CMS physicists and helping them to achieve their ML objectives. At the same time, the CMS physicists themselves rarely have specific data science skills to face such challenges alone. What is needed to design and run successfully a ML project is often not found in a basic CMS physicist expertize (or a HEP physicist, for what matters) whose primary competences are focussed on high energy physics, data analysis (including statistics), and whose ultimate goal is to work towards a physics publication. Facing the need to improve a physics data analysis and understanding that ML might be an interesting exploration is hence just a first step towards actually embracing ML in an analysis.

Goal

The work presented in this poster contributes to build a **ML "as a service"** for CMS Physics needs, namely build an **end-to-end data-service to serve ML trained model to the CMSSW framework**. The basic idea is as simple as this: instead of asking each physicists who wants to exploit ML in their own task to just learn how to do it themselves independently, each user would ultimately build a modified data analysis code where **"calls" to an external service** - as simple as calls to functions - would be added to return a trained ML model output that could be directly used in the analysis code (e.g. in loops over events) in a streamlined manner, thus **hiding all the complexity related to the ML machinery** via outsourcing this to an external service.



Tensorflow "as a service" (TFaaS)

Choice of the Machine Learning framework

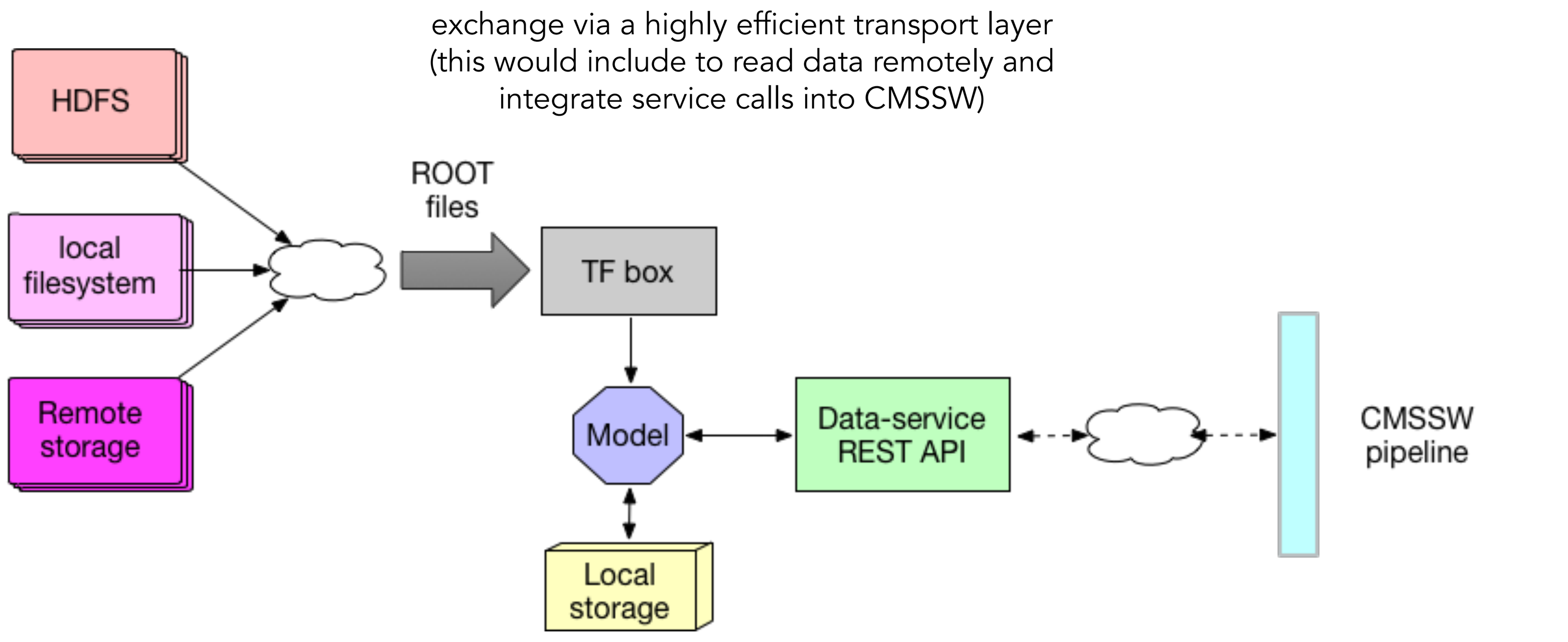
Caffe, Chainer, DL4J, KERAS, CNTK, MatConvNet, MINERVA, mxnet, Purine, TensorFlow, theano, torch

Main functionalities of TFaaS [3]

read ROOT files and convert them into a suitable form as input to ML/DL systems

train the model on a TF box and serve the trained ML/DL model via REST API with data exchange via a highly efficient transport layer (this would include to read data remotely and integrate service calls into CMSSW)

deploy the service to the cloud (e.g. rent GPUs to train the model), and use it for predictions



General process

Select use case

Create a model

Interface with the prototype

In this work

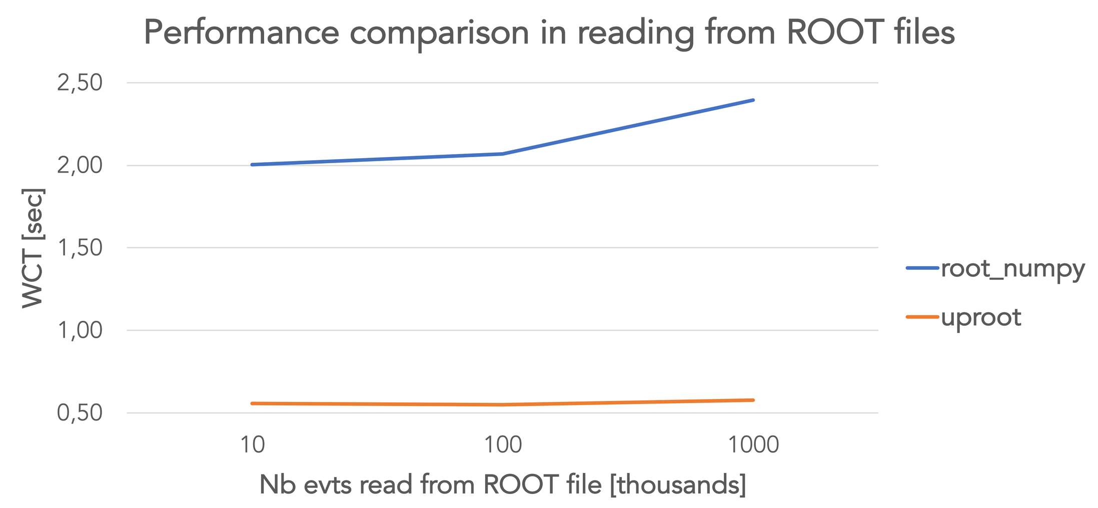
$t\bar{t}$ selection

Scikitlearn-based

1. *Data Preparation* → hardest part to read ROOT file using a new tool ("**uproot**")
2. *Data Validation*
3. *Algorithm selection*
4. *Parameter tuning*

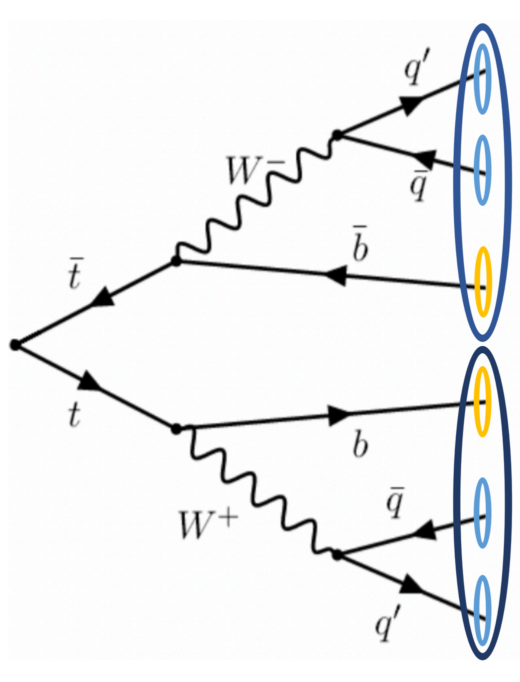
Create a keras-tensorflow model

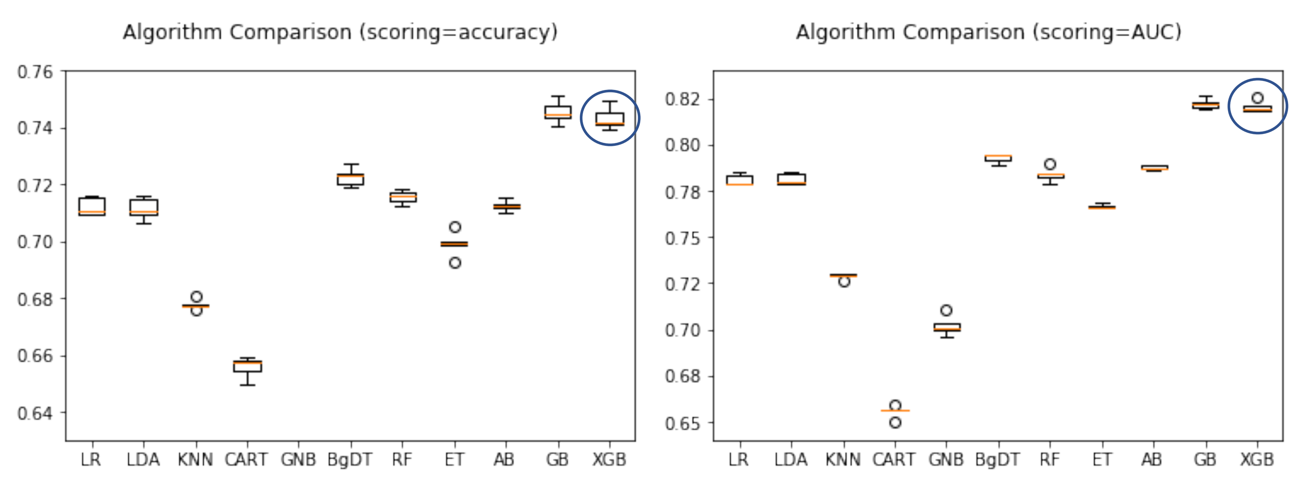
Disclaimer: this is a direct comparison among two approaches, and not a performance benchmark.



$t\bar{t}$ use case

the Signal versus Background discrimination in the selection of $t\bar{t}$ events in the all-hadronic topology



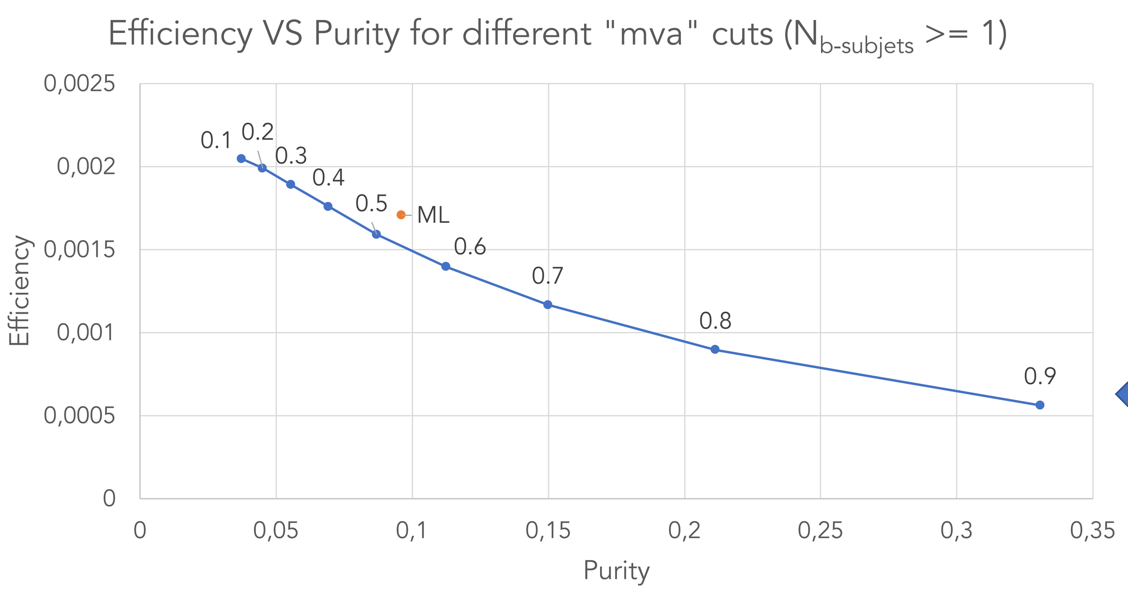


XGBoost (eXtreme Gradient Boosting) is an advanced implementation of gradient boosting algorithm. It offers many improvements with respect to GBM, among which the most relevant are: i) regularization, that helps to reduce overfitting; ii) parallel processing, hence it delivers performances that are demonstratedly better than GBM; iii) high flexibility.

Below a simple use of TFaaS is shown: after the TFaaS server is launched and the model loaded, the prediction for an event in terms of probability is produced.

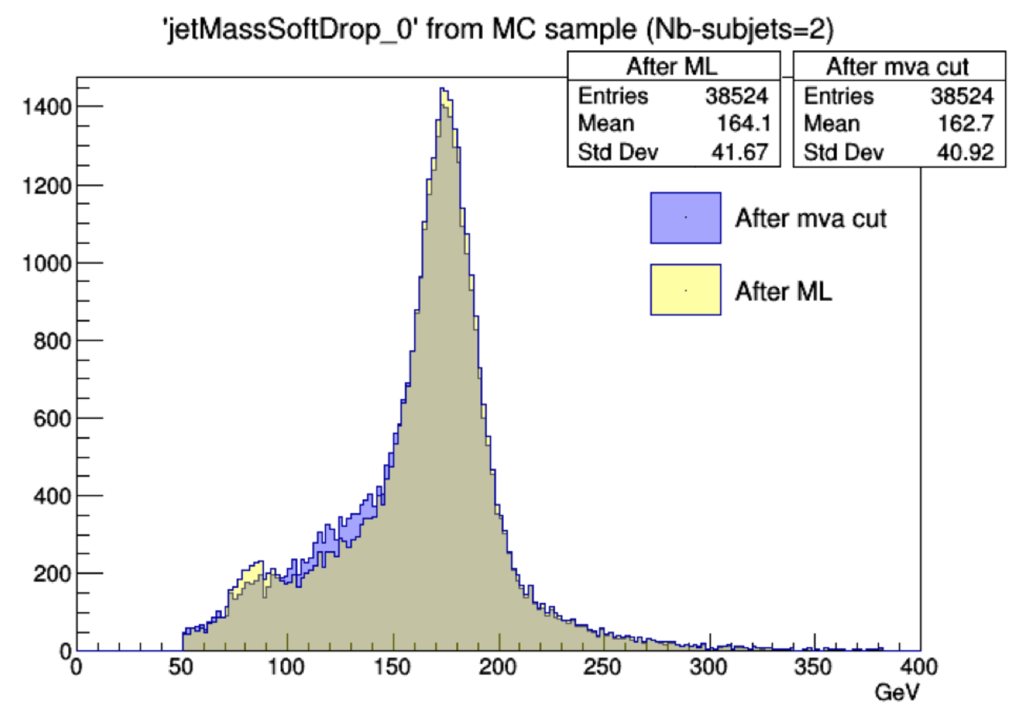
```
str957-135:Demo luca.giommi$ cd TFaaS/src/Go/
str957-135:Go luca.giommi$ ./tfaas -config=config.json &
[1] 8510
str957-135:Go luca.giommi$ INFO[0000] <Config port=8083 dir=/Users/luca.giommi/Demo/keras_to_tensorflow base= auth=false config
Prot= verbose=0 log=text crt= key=>
INFO[0000] Starting HTTP server                          Addr=":8083"

str957-135:Go luca.giommi$ curl -L -k -i -X POST http://localhost:8083/upload -F 'name=model1' -F 'params=@/Users/luca.giommi/D
emo/TFaaS/src/Go/params1.json' -F 'model=@/Users/luca.giommi/Demo/keras_to_tensorflow/11_04_model.h5.pb' -F 'labels=@/Users/luc
a.giommi/Demo/keras_to_tensorflow/sb_labels.csv'
HTTP/1.1 100 Continue

INFO[0160] store as params.json                          FileName=params1.json
INFO[0160] Uploaded                                      File=/Users/luca.giommi/Demo/keras_to_tensorflow/model1/params.json
INFO[0160] Uploaded                                      File=/Users/luca.giommi/Demo/keras_to_tensorflow/model1/11_04_model.h5
.pb
INFO[0160] Uploaded                                      File=/Users/luca.giommi/Demo/keras_to_tensorflow/model1/sb_labels.csv
HTTP/1.1 200 OK
Date: Fri, 25 May 2018 19:05:28 GMT
Content-Length: 0
Content-Type: text/plain; charset=utf-8

str957-135:Go luca.giommi$ curl -s -L -k --key -H "Content-type: application/json" -d '{"keys": ["nJets", "nLeptons", "jetEta_0
", "jetEta_1", "jetEta_2", "jetEta_3", "jetEta_4", "jetMass_0", "jetMass_1", "jetMass_2", "jetMass_3", "jetMass_4", "jetMassSof
tDrop_0", "jetMassSoftDrop_1", "jetMassSoftDrop_2", "jetMassSoftDrop_3", "jetMassSoftDrop_4", "jetPhi_0", "jetPhi_1", "jetPhi_2
", "jetPhi_3", "jetPhi_4", "jetPt_0", "jetPt_1", "jetPt_2", "jetPt_3", "jetPt_4", "jetTau1_0", "jetTau1_1", "jetTau1_2", "jetTa
u1_3", "jetTau1_4", "jetTau2_0", "jetTau2_1", "jetTau2_2", "jetTau2_3", "jetTau2_4", "jetTau3_0", "jetTau3_1", "jetTau3_2", "je
tTau3_3", "jetTau3_4"], "values": [2.0, 0.0, 0.9228423833849999, -1.1428750753399999, 0.0, 0.0, 0.0, 155.239425659, 142.7096099
85, 0.0, 0.0, 0.0, 83.5365982056, 120.549507141, 0.0, 0.0, 0.0, 1.9305502176299998, -1.17742347717, 0.0, 0.0, 0.0, 481.41979980
5, 449.04394531199995, 0.0, 0.0, 0.0, 0.296700358391, 0.286615312099, 0.0, 0.0, 0.0, 0.164555206895, 0.19625715911400002, 0.0,
0.0, 0.0, 0.117722302675, 0.155229091644, 0.0, 0.0, 0.0]}' http://localhost:8083/json
INFO[0193] load to cache                                 Model=model1
2018-05-25 21:06:01.763295: I tensorflow/core/platform/cpu_feature_guard.cc:140] Your CPU supports instructions that this Tenso
rFlow binary was not compiled to use: SSE4.2 AVX AVX2 FMA
INFO[0193] load TF model                                 Labels=/Users/luca.giommi/Demo/keras_to_tensorflow/model1/sb_labels.cs
v Model=/Users/luca.giommi/Demo/keras_to_tensorflow/model1/11_04_model.h5.pb
[0.6046947]
```

Comparison between ML and ROOT MVA



Efficiency and purity for different mva cuts (the values quoted above the dots), compared to the same obtained by ML, in the case of $N_{b\text{-subjets}}$ >= 1.

ML is not worst respect to MVA.



Invariant mass of the leading jet after the "soft drop declustering" algorithm for the MC data with $N_{b\text{-subjets}}$ = 2.

Conclusions

An end-to-end data service has been developed to provide trained ML models to the CMS software framework and in particular the proof-of-concept has been demonstrated in the s/b discrimination in the all-hadronic channel in the $t\bar{t}$ decay. A simple **demo** [4] has been created that shows how a common user can use TFaaS to make predictions.

Next steps: review and improve all the steps done and move the model creation and training on cloud.

References

[1] The HEP Software Foundation (HSF), A Roadmap for HEP Software and Computing R&D for the 2020s. 2017, arXiv:1712.06982
[2] V. Kuznetsov, T. Li, L. Giommi, D. Bonacorsi, T.Wildish, Predicting dataset popularity for the CMS Experiment. Journal of Physics: Conference Series 762.1 (2016), arXiv:1602.07226.
[3] Tensorflow as a Service (TFaaS). URL: https://github.com/vkuznet/TFaaS
[4] URL: https://drive.google.com/file/d/1Ipwt9dOJCCb9EN3lmiYKhExel6dd4baO/view

Contact

luca.giommi2@studio.unibo.it

LHCP Bologna 2018