16. MLaaS4HEP for the Higgs boson ML challenge

Author(s)

Name	Institution	Mail Address	Social Contacts
Luca Giommi	INFN-CNAF	luca.giommi@cnaf.infn.it	N/A
Mattia Paladino	University of Bologna	mattia.paladino2@unibo.it	N/A

How to Obtain Support

Mail	luca.giommi@cnaf.infn.it
Social	N/A
Jira	N/A

General Information

ML/DL Technologies	classification algorithms
Science Fields	High energy Physics
Difficulty	low
Language	English
Type	runnable, fully annotated

Software and Tools

Programming Language	Python
ML Toolset	Keras, Tensorflow, sklearn, PyTorch, XGBoost
Additional libraries	uproot, matplotlib,
Suggested Environments	Google CoLab, Docker, own PC, INFN-Cloud VM

Needed datasets

Data Creator	ATLAS experiment

Data Type	simulation
Data Size	57 MB compressed
Data Source	Kaggle, CERN opendata

Short Description of the Use Case

In this exercise, we use the MLaa4HEP machinery to deal with the Higgs boson ML challenge, a competition held in 2014, organized by a group of ATLAS physicists and data scientists, and hosted by the Kaggle platform.

The Higgs decay channel chosen in this challenge was H->, which is challenging for two main reasons. First, since neutrinos cannot be directly detected, their presence in the final state makes it difficult to evaluate the mass of the Higgs candidate. Second, the Z boson can also decay in two taus, and this happens much more frequently than the Higgs decay. Moreover, since the mass of a Z (~91 GeV) is not very far from the mass of the Higgs (~125 GeV), the two decays produce similar events which are difficult to separate. In the challenge, a specific topology among the many possible ones is chosen, i.e. events where one tau decays into an electron or a muon and two neutrinos, and the other tau decays in hadrons and a neutrino.

The signal and background samples are Monte Carlo simulated events using the official ATLAS full detector simulator. The signal sample contains events in which Higgs bosons were produced. The background sample was generated by other known processes which can produce events that mimic the signal. For simplicity, only three background processes are used in the challenge. The first set of events comes from the decay of the Z boson in two taus. The second set contains events with a pair of top quarks, which can have hadronic tau and lepton in their decay. The third set involves the W boson decay, where one electron or muon and a hadronic tau can appear simultaneously only through imperfections in the procedure of particle identification.

In the challenge, the Approximate Median Significance (AMS) metric is used to evaluate the quality of the classifier. The AMS metric is defined by the equation reported below, where s and b represent the estimated number of signal and background events respectively, whereas b_{reg} is a regularization term set to 10 for the challenge.

$$AMS = \sqrt{2\left(\left(s+b+b_{reg}\right) ln\left(1+\frac{s}{b+b_{reg}}\right)-s\right)}$$

For more details about the physics use case, the meaning of the features used, and the challenge itself, see here, here and here.

As we said, here our goal is to use the MLaaS4HEP machinery applied on the specific use case of the Higgs boson ML challenge. MLaaS4HEP stands for Machine Learning as a Service for High Energy Physics, and it consists of two components. The first one, the MLaaS4HEP framework, covers the data reading, data processing, and ML model training phases, in a completely model-agnostic fashion, directly using ROOT files of arbitrary size from local or distributed data sources. The second one, the TFaaS framework, can be used to host pre-trained Tensor-based ML models and obtain predictions via HTTP calls.

MLaaS4HEP is designed to cover ML use cases of type classification. It has been successfully tested using:

- MLP written in Keras and PyTorch;
- MLP, Gradient Boosting, AdaBoost, Random Forest, Decision Tree, kNN, SVM, and Logistic Regression written in Scikit-learn;
- Gradient Boosting written in XGBoost.

The MLaaS4HEP framework is developed primarily to accept flat ROOT TTrees (e.g. the CMS NANOAOD data format) as input for HEP classification problems. The MLaaS4HEP training workflow is performed running the *workflow.py* script, e.g. in the following way:

```
./workflow.py --files=files.txt --labels=labels.txt --model=model.py --params=params.json --preproc=preproc.json
```

The *workflow.py* script takes several files as arguments:

- *files.txt* stores the path of the input ROOT files;
- *labels.txt* stores the labels of the input ROOT files in case of classification problems;
- *model.py* stores the definition of the custom ML model to use in the training phase, in the user's favorite ML framework;
- *params.json* stores the parameters on which MLaaS4HEP is based, e.g. number of events to use, chunk size, batch size, and redirector path for files located in remote storage;
- *preproc.json* stores the definition of preprocessing operations to be applied to data.

How to execute it

Way #1: Use Googe Colab

You can run this Jupyter notebook using Google Colab, by clicking here. It covers several steps, from inspecting data, to running the MLaaS4HEP framework to obtain the trained ML models, to uploading the submissions file to the Kaggle website.

Way #2 Use the MLaaS4HEP Docker image

If you don't want to use MLaaS4HEP in the Google Colab notebook but you want to use your resources, instead of installing all the dependencies you can use the MLaaS4HEP Docker image, i.e. felixfelicislp/mlaas:xrootd_pip. An example of the command to run is the following:

docker run --name={name} --memory={memory} --cpus={cpus} felixfelicislp/mlaas:xrootd_pip --files={files} --labels={labels} --model={model} --params={params} --fout={fout}

Way #3: Use the MLaaS4HEP server

Another way to use the MLaaS4HEP framework is to interact with the APIs of the MLaaS4HEP server. We implemented a working prototype connecting an OAuth2-Proxy server, a MLaaS4HEP_server, an xrootd proxy-cache server, an X509 proxy renewer, and TFaaS, hosted by a VM of the INFN Cloud.

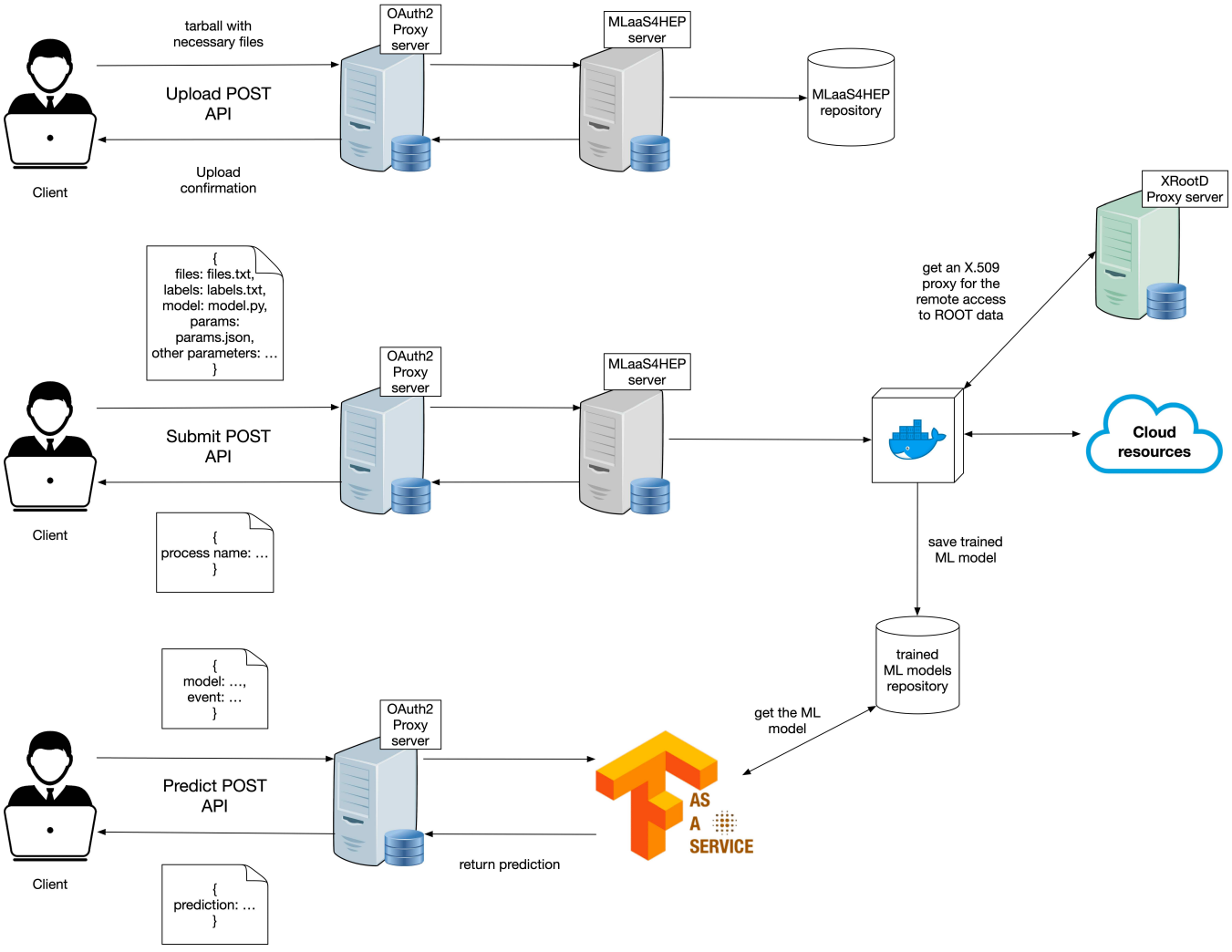
- The OAuth2-Proxy server allows to authenticate and authorize users to use the service. The authorization server we chose is https://cms-auth.web.cern.ch, so only CMS users can access this application.
- A MLaaS4HEP_server allows to submit MLaaS4HEP workflows, i.e. to produce trained ML models starting from ROOT files and ML algorithms defined by the user.
- xrootd proxy-cache server and X509 proxy renewer allow to create an X.509 proxy and renew it in order to make possible the access to remote ROOT files stored in grid sites.
- TFaaS takes the ML models trained and produced by the MLaaS4HEP server to make inference.

The MLaaS4HEP_server APIs can be reached at the following address https://90.147.174.27:4433 while TFaaS at https://90.147.174.27:8081. Once the user obtains an access token from the authorization server, he/she can contact the MLaaS4HEP server or TFaaS using curl, e.g. in the following ways:

curl -L -k -H "Authorization: Bearer ${TOKEN_MLAAS}" -H "Content-Type: application/json" -d @submit.json https://90.147.174.27:4433/submit

curl -L -k -H "Authorization: Bearer ${TOKEN_TFAAS}" -X POST -H "Content-type: application/json" -d @predict_bkg.json https://90.147.174.27:8081/json

The former command allows training a ML model, whereas the latter allows using this model to get the prediction on a given event (stored in the *predict_bkg.json* file). All the instructions about how to use these services can be found here. A demo version of the services can be found here. A pictorial representation of the services is the following:



References

V. Kuznetsov, L. Giommi, D. Bonacorsi, MLaaS4HEP: Machine Learning as a Service for HEP. Comput Softw Big Sci 5, 17 (2021). DOI: 10.1007/s41781-021-00061-3

L. Giommi, D. Spiga, V. Kuznetsov, D. Bonacorsi, Prototype of a cloud native solution of Machine Learning as Service for HEP. PoS ICHEP2022 (2022), 968. DOI: 10.22323/1.414.0968

L. Giommi, D. Spiga, V. Kuznetsov, D. Bonacorsi, M. Paladino, Cloud native approach for Machine Learning as a Service for High Energy Physics. PoS ISGC2022 (2022), 012. DOI:10.22323/1.415.0012

L. Giommi, V. Kuznetsov, D. Bonacorsi, D. Spiga, Machine Learning as a Service for High Energy Physics on heterogeneous computing resources. PoS ISGC2021 (2021), 019. DOI: 10.22323/1.378.0019

L. Giommi, D. Bonacorsi, V. Kuznetsov, Prototype of Machine Learning “as a Service” for CMS Physics in Signal vs Background discrimination. PoS LHCP2018 (2018), 093. DOI: 10.22323/1.321.0093.

L.Giommi, Machine Learning as a Service for High Energy Physics (MLaaS4HEP): a service for ML-based data analyses. Ph.D. thesis, University of Bologna, 2023. http://amsdottorato.unibo.it/id/eprint/10586

M. Paladino, Machine learning “as a service” for high energy physics (mlaas4hep): Evolution of a framework for ml-based physics analyses. M.S. thesis, University of Bologna, 2022. https://amslaurea.unibo.it/26129/

Presentation made on 27 Mar 2023 : https://agenda.infn.it/event/35136/contributions/193692/attachments/103435/144722/Giommi_ML_INFN.pdf