# Machine learning techniques for jet flavour identification at CMS

*Mauro* Verzetti[1,2*]

[1]CERN, 1211 Geneva 23, Switzerland
[2]Flanders Research Foundation - FWO, Egmontstraat 5, 1000 Brussel, Belgium

**Abstract.** Jet flavour identification is a fundamental component of the physics program of the LHC-based experiments. The presence of multiple flavours to be identified leads to a multiclass classification problem. We present results from a realistic simulation of the CMS detector, one of two multi-purpose detectors at the LHC, and the respective performance measured on data. Our tagger, named DeepJet, relies heavily on applying convolutions on lower level physics objects, like individual particles. This approach allows the usage of an unprecedented amount of information with respect to what is found in the literature. DeepJet stands out as the first proposal that can be applied to multi-classification for all jet flavours. We demonstrate significant improvements by the new approach on the classification capabilities of the CMS experiment in simulation in several of the tested classes. At high momentum improvements of nearly 90% less false positives at a standard operation point are reached.

## 1 Introduction

Jet flavour identification has been a longstanding staple of the physics program of HEP experiments. The problem has always been approached as a supervised learning classification problem, leveraging the peculiar features of heavy flavour hadrons embedded inside the parton shower. Heavy-flavour hadrons, which carry a charm or beauty quarks, have a significant lifetime and produce displaced tracks and secondary vertices (SV) within the clustered jet. Additionally, they have a significant semi-leptonic and leptonic branching fraction. It is therefore possible to use this information as well in the classification, even though it is rarely done in practice within the CMS collaboration as such information is used to obtain a data sample enhanced in heavy flavour content to measure the tagger efficiency in real data.

## 2 AK4 Jets

The current default jet flavour classifier in CMS is DeepCSV. DeepCSV was firstly introduced in Ref. [1] and consist of a dense deep neural network taking as input 8 features for the six most displaced tracks in the jet, 8 features from the most displaced secondary vertex, and 12 global variables, totalling 68 input features. Missing features are zero-padded. Tracks and SVs undergo a pre-selection before the feature extraction to reject fake and pile-up tracks and nuclear interaction vertices. The 68 input features are then fed into five layers with 100

[*] On behalf of the CMS Collaboration, e-mail: mauro.verzetti@cern.ch

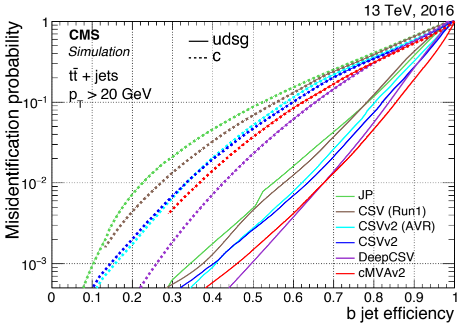

**Figure 1.** Performance of the different heavy flavour tagging algorithms used by the CMS Collaboration since the start of the data taking in 2010. More modern algorithms (in order of appearance: CSVv2, cMVAv2, DeepCSV) show better performance.

nodes each with ReLU activation, and an output layer with SoftMax activation discriminating between four output classes: b, bb (two B-hadrons in the jet), c and light (comprising both quarks and gluons). The model has been trained with the Keras [2] python package using the TensorFlow [3] backend.

DeepCSV outperforms all the previous CMS taggers including cMVAv2, which uses the additional lepton information, as shown in Figure 1.

The big performance gain obtained moving to a deep neural network sparked the interest in more complex models. Convolutional neural networks have been successfully used for image classification and there have been several attempts to use such approach for jet classification. While this approach can be successful for boosted objects, which is mainly focusing on the internal energy distribution of the jet, the flavour identification relies on more features and on quantities that cannot be easily summed in a discretised environment.

The DeepJet algorithm [4, 5] focuses on single particles rather than on images. No preselection is applied to any track, secondary vertex, or neutral candidate before entering the network. The network uses 16 features of up to 25 input tracks (displacement sorted), 8 features of up to 25 neutral candidates, 12 features of up to 4 secondary vertices, and 6 global variables. Each candidate type is then passed through a set of convolutional layers that operate on each single candidate separately. These layers provide an automated form of feature selection and engineering, resulting in 8, 4, and 8 features for each input track, neutral candidate, and secondary vertex, respectively. These layers are then masked for missing inputs and passed to three independent LSTMs, which learn a compact summary of each candidate type. The output dimensionality of these recurrent layers is 150, 50, and 50 for tracks, neutral candidates, and secondary vertices, respectively. Finally, these outputs are combined together with the global variables and fed into seven dense layers with 100 nodes, except the first layer which has 200 nodes. A final output layer provides discrimination between six classes: three b jet types (one B hadron, two B hadrons, one leptonically decaying B hadron), charm jet, light quark jet, and gluon jet. Each node in the network has a RelU activation function, except the output layer which has SoftMax activation. A schematic of the DeepJet network structure can be found in Figure 2.

DeepJet has shown improved performance in b-jet classification, especially at high jet $p_T$, as shown in Figure 3. The multi-classification approach of DeepJet allows to compare the performance of the same model also for quark/gluon discrimination, showing comparable performance to dedicated binary classifiers, as shown in Figure 4.

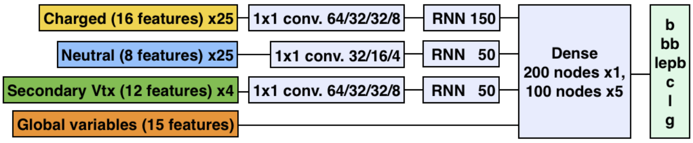

**Figure 2.** DeepJet network architecture.

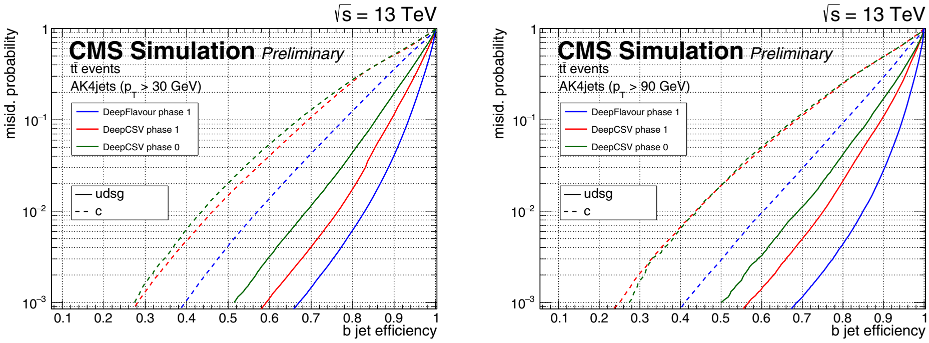

**Figure 3.** Classification performance of the DeepJet algorithm (here labelled “DeepFlavour”) with respect to the DeepCSV for the CMS 2016 and Phase I detectors. The performance is evaluated in a top pair sample with a p$_T$ cut on the jet of 30 GeV and 90 GeV (left and right, respectively)

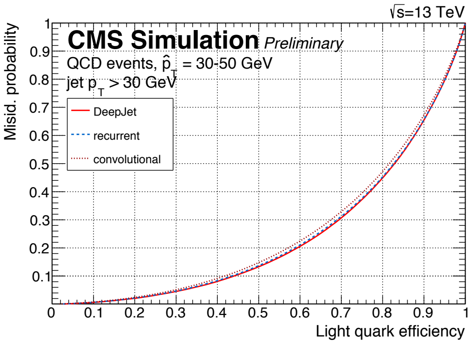

**Figure 4.** Performance of the DeepJet algorithm in the quark/gluon classification task. The algorithm is compared to other two binary approaches exploiting the jet energy deposits in image format (convolutional) and exploiting the single jet constituent kinematics in a list (recurrent).

## 3 Boosted resonances, AK8 Jets

A similar architecture of DeepJet is employed by the DeepDoubleB and DeepDoubleC taggers [6], although neutral candidates are ignored and the number of tracks and SVs is limited by a preselection. These classifiers aim at identifying the decay of a boosted resonance into a pair of b and c jets, respectively. As previously mentioned, the network structure, summarised in Figure 5, is very similar to the one of DeepJet, but in this case GRUs are used in the recurrent units instead of LSTMs. The DeepDoubleB/C taggers are trained as binary taggers aiming at rejecting the QCD background (DeepDoubleBvL and DeepDoubleCvL)

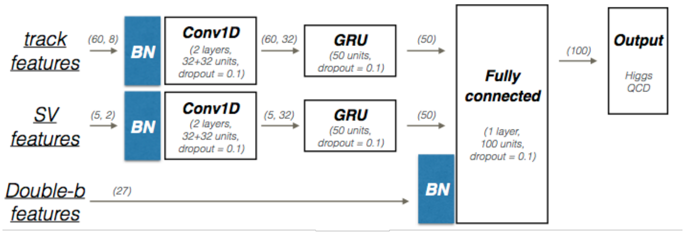
track features
(60, 8)
BN
Conv1D
(2 layers, 32+32 units, dropout = 0.1)
(60, 32)
GRU
(50 units, dropout = 0.1)
(50)
SV features
(5, 2)
BN
Conv1D
(2 layers, 32+32 units, dropout = 0.1)
(5, 32)
GRU
(50 units, dropout = 0.1)
(50)
Double-b features
(27)
BN
Fully connected
(1 layer, 100 units, dropout = 0.1)
(100)
Output
Higgs
QCD

**Figure 5.** The DeepDoubleB and DeepDoubleC architectures.

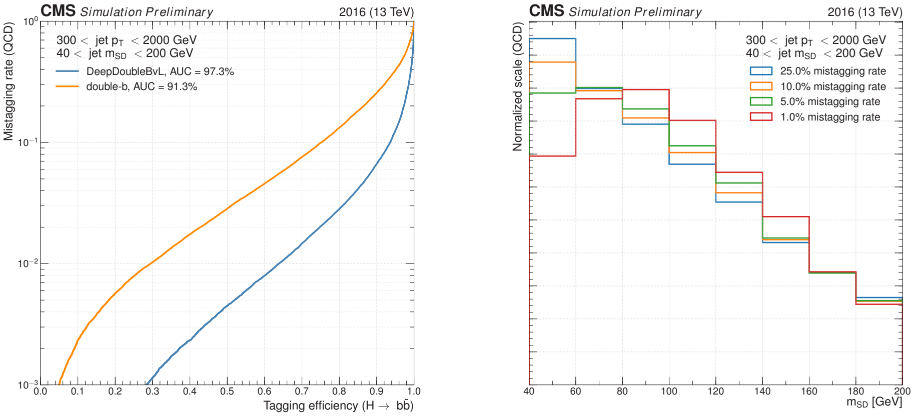

**Figure 6.** Left: performance of the DeepDoubleBvL algorithm in separating boosted H(bb) from QCD jets, compared to the previous double-b algorithm. Right: mass sculpting induced in the background QCD sample by applying different selections on the classifier output.

and at separating boosted double b decays from double charm jets (DeepDoubleCvB). The performance of these new taggers is shown in Figure 6 and 7.

DeepDoubleB significantly outperforms the previous double-b classifier, but also introduces a significant mass sculpting, as shown in Figure 6, which is undesirable for physics analyses. To overcome this issue, two penalty terms proportional to the Kullback-Leibler divergence between the original background and signal mass distributions and the classifier output-weighted distributions are applied per batch. These term ensures that the classifier output is decorrelated from the jet mass. The mass decorrelation comes at negligible cost in classification performance, as shown in Figure 8.

The DeepAK8 [7] tagger applies no preselection to the jets constituents and up to a hundred of them is used. The large amount of candidates makes computationally unfeasible the training of recurrent units. Therefore, a set of convolutional kernels spanning multiple candidates is used instead. Ten features for each charge and neutral candidate are passed to one of these convolutional blocks ordered in candidate $p_T$ to learn the jet sub-strucure. The flavour content of the jet is learned by other two of these blocks, one using only charged jet constituents, sorted by displacement, and one using secondary vertices, sorted by flight distance.

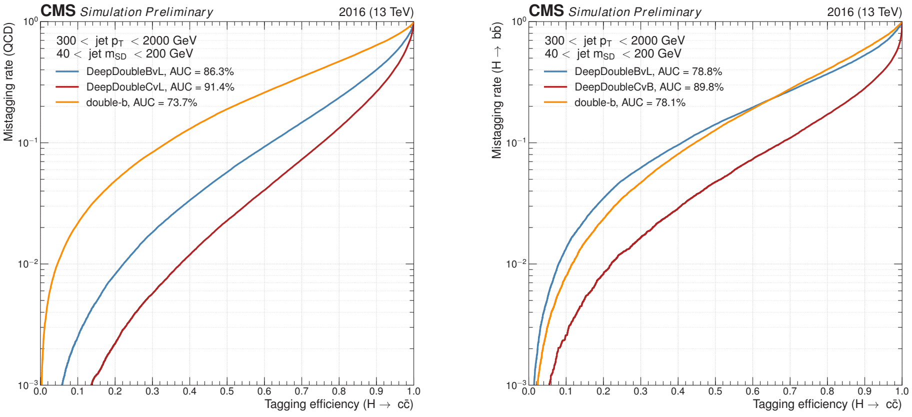

**Figure 7.** Left: discrimination power of the DeepDoubleCvL at correctly classifying boosted H(cc) from QCD jets, compared to the double-b and DeepDoubleBvL classifiers. Right: discrimination power of the DeepDoubleCvB at correctly classifying boosted H(cc) from boosted H(bb) jets, compared to the double-b and DeepDoubleBvL classifiers.

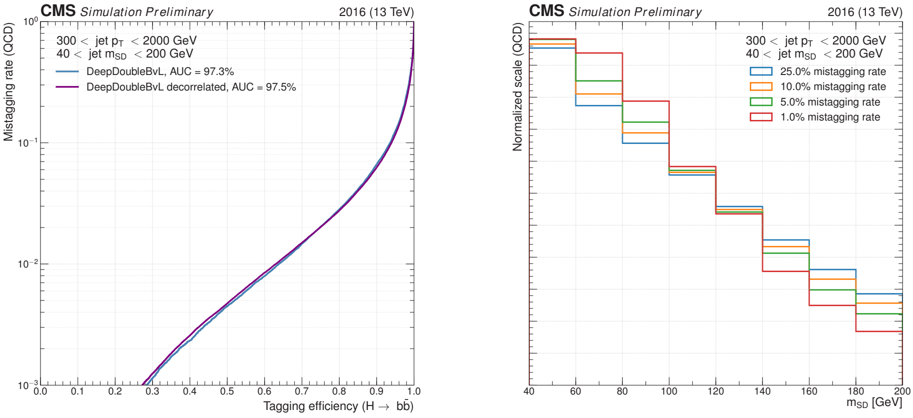

**Figure 8.** Effect of the jet mass decorrelation procedure on the discrimination performance (left) and on the QCD background shape (right) for different selections on the classifier outputs.

These three convolutional nodes are then merged into a single dense layer with 521 nodes before reaching the output layer. A schematic of the DeepAK8 architecture can be found in Figure 9. DeepAK8 aims at classifying a wide variety of resonances in multiple decay modes. DeepAK8 outperforms a simpler BDT approach in the task of top classification as shown in Figure 10.

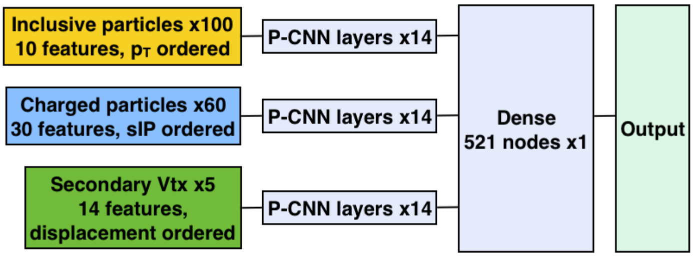

**Figure 9.** The DeepAK8 network architecture.

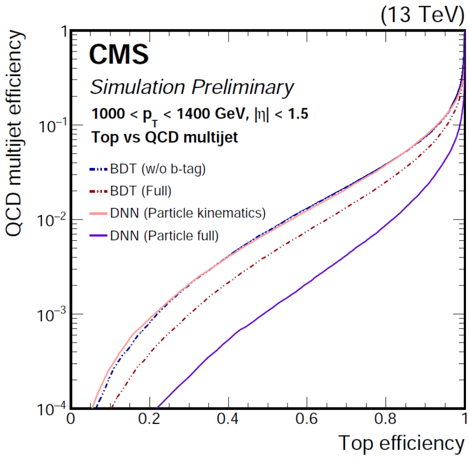

**Figure 10.** Boosted top classification performance of the DeepAK8 algorithm compared to the performance of a BDT and a deep neural network based only on kinematic information of the jet constituents and a BDT based on kinematic information and displacement of the jet constituents.

## 4 Model Deployment in Production Environment

The deployment of these new deep-learning-based models in the production environment of a large HEP experiment poses a significant challenge. The training environment is usually python-based with a custom set of libraries installed and running with minimal memory or CPU usage restrictions. The production environment, instead, is usually based on a custom C++ framework, with tight constraints on memory and running on multiple threads. The harmonisation of the threading pool implementation in TensorFlow and in CMSSW and the optimisation of the model for inference have been the two major issues the collaboration had to overcome before fully integrating a TensorFlow inference engine into our production workflow. Further reduction of the memory footprint and the number of external dependencies may come from model pruning or AOT compilation of the trained model.

## 5 Conclusions

The latest developments in heavy flavour classification from the CMS Collaboration have been reviewed in this contribution. Switching to a deep learning approach have brought significant improvements in this field together with new challenges in deploying these models in the production environment of a large HEP experiment.

## References

[1] CMS Collaboration, “Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV”, JINST **13** (2018) no.05, P05011 doi:10.1088/1748-0221/13/05/P05011 [arXiv:1712.07158 [physics.ins-det]].

[2] F. Chollet et al., Keras, https://github.com/fchollet/keras.

[3] M. Abadi et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, https://arxiv.org/abs/1603.04467.

[4] CMS Collaboration, "CMS Phase 1 heavy flavour identification performance and developments", "CMS-DP-2017-013", "http://cds.cern.ch/record/2263802".

[5] CMS Collaboration, "New Developments for Jet Substructure Reconstruction in CMS", "CMS-DP-2017-027", "https://cds.cern.ch/record/2275226".

[6] CMS Collaboration, "Performance of Deep Tagging Algorithms for Boosted Double Quark Jet Topology in Proton-Proton Collisions at 13 TeV with the Phase-0 CMS Detector", "CMS-DP-2018-046", "http://cds.cern.ch/record/2630438".

[7] CMS Collaboration, "Boosted jet identification using particle candidates and deep neural networks", "CMS-DP-2017-049", "https://cds.cern.ch/record/2295725".