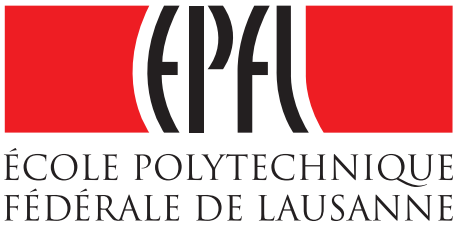
A search for the $B^0 \to D_s^+ D_s^-$ decay using multivariate techniques at LHCb

Master Thesis

Marc Huwiler
marc.huwiler@epfl.ch

Supervisor: Dr. Conor Fitzpatrick
Director: Prof. Olivier Schneider

January 24, 2018



Abstract

A search for the decay $B^0 \to D_s^+ D_s^-$ is conducted using the $3\,\text{fb}^{-1}$ of Run 1 data collected by the LHCb experiment. The use of new DNN classifiers recently added to TMVA is investigated in the selection process. A small excess of 2 sigma is measured in the signal region. The use of Run 2 data was envisaged, but the changes in the dataflow and the absence of Run 2 MC did not allow a precise efficiency calculation on the Run 1+Run 2 data. The measured branching fraction using the LHCb Run 1 data is:

$$\mathcal{B}(B^0 \to D_s^+ D_s^-) = [6.64 \pm 2.15\ (\text{stat}) \pm 2.62\ (\text{syst}) \pm 0.76\ (\text{norm})\ ] \cdot 10^{-5}$$

This is in agreement with previous limits and compatible with the SM prediction.

Contents

1 Introduction

The aim of this report is to describe a search for the decay $B^0 \to D_s^+ D_s^-$, using data from the LHCb (Large Hadron Collider beauty) experiment. LHCb is one of the four main experiments around the LHC (Large Hadron Collider) accelerator at CERN. It is dedicated to the study of b and c flavoured mesons, with as main goal to help understand the difference between matter and antimatter, and to probe the Standard Model parameters for potential New Physics. The study of B mesons is of special interest for investigating the violation of the CP asymmetry, which could explain the differences between matter and antimatter, as well as the processes that led to the universe as we know it. The precise study of flavour-violating decay branching fractions provides an insight into the symmetries of universe, by constraining the Cabibbo-Kobayashi-Maskawa matrix elements. The present study focuses on the search for the $B^0 \to D_s^+ D_s^-$ decay, and aims to measure its branching fraction.

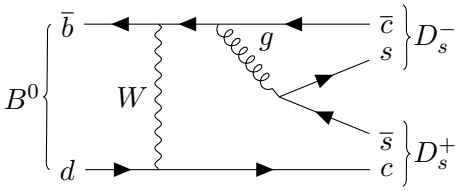


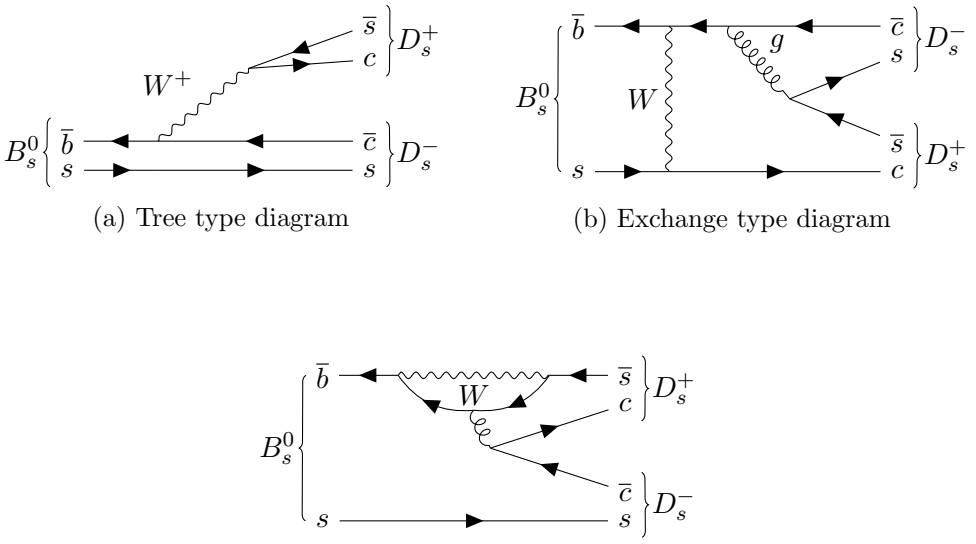
Figure 1: Leading order Feynman diagram for the $B^0 \to D_s^+ D_s^-$ decay.

2 Study of rare B meson decays

The Standard Model of particle physics (SM) is the currently accepted theory describing the fundamental particles and their interactions. Relying on a quantum field theory, it was developed during the 20th century and was successful at describing almost all experimental results so far [1], and found experimental confirmation by accurately predicting new particles, such as the top quark or the Higgs boson. It was also successful at accounting for the violation of the CP symmetry, first observed in the 60s in kaon decays. The three generations of quarks in the Standard model are necessary to explain the CP violation. The couplings of the flavour non-conserving weak decays to the different quarks are described by the Cabibbo-Kobayashi-Maskawa (CKM) matrix [2].

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \approx \begin{pmatrix} 1-\frac{\lambda^2}{2} & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1-\frac{\lambda^2}{2} & A\lambda^2 \\ A\lambda^3(1-\rho-i\eta) & -A\lambda^2 & 1 \end{pmatrix} \tag{1}$$

with $\lambda = 0.2257^{+0.0009}_{-0.0010}$, $A = 0.814^{+0.021}-0.022$, $\rho = 0.135^{+0.031}-0.016$, $\eta = 0.349^{+0.015}_{-0.017}$



(a) Tree type diagram

(b) Exchange type diagram

(c) Penguin diagram

Figure 2: Leading order Feynman diagrams for the $B_s^0 \to D_s^+ D_s^-$ decay.

and

$$\begin{pmatrix} |V_{ud}| & |V_{us}| & |V_{ub}| \\ |V_{cd}| & |V_{cs}| & |V_{cb}| \\ |V_{td}| & |V_{ts}| & |V_{tb}| \end{pmatrix} = \begin{pmatrix} 0.97427 \pm 0.00015 & 0.22534 \pm 0.00065 & 0.00351^{+0.00015}_{-0.00014} \\ 0.22520 \pm 0.00065 & 0.97344 \pm 0.00016 & 0.0412^{+0.000021}_{-0.000046} \\ 0.0.00867^{+0.00029}_{-0.00031} & 0.0404^{+0.0011}_{-0.0005} & 0.999146^{+0.000021}_{-0.000046} \end{pmatrix} \text{[2]}$$

The CKM matrix contains three mixing angles describing the rotation between the eigenstates of the weak interaction and the mass eigenstates, and one CP violating parameter.

The decays of B mesons to open-charm in particular provide a probe for the Cabibbo-Kobayashi-Maskawa matrix elements. Since the CKM matrix incorporates a CP violating phase, the precise measurement of its elements could help understanding the CP violation and in a wider extent contribute to our understanding of the difference between matter and antimatter [3]. Furthermore, deviations from the SM predictions could represent hints for new physics [4].

Flavour changing interactions are only driven by the weak charged currents (carried by the W^+ and W^- bosons). The strong and electromagnetic interactions as well as the neutral weak current (carried by the Z^0 boson) are flavour conserving. Processes changing a D type quark (d, s, b) or a U type (u, c, t) one into another quark of the same type are forbidden at the tree level, and require at least one off diagonal V_{CKM} element. They are suppressed by the Glashow-Illopoulos-Maiani mechanism. According to the Cabibbo-Kobayashi-Maskawa matrix, weak interaction vertices involving quarks from

different generations have a much lower probability, and are called *Cabibbo-suppressed*.

The $B^0 \to D_s^+ D_s^-$ decay involves two flavour violating vertices at leading order (figure 1), and is thus Cabibbo-suppressed. There exists no tree-level diagram. The Feynman diagrams at leading order for the $B_s^0 \to D_s^+ D_s^-$ decay are provided in figure 2, where in addition to the Cabibbo suppressed decay, a tree diagrams and a lower order Penguin diagram contribute also to the decay. The $B^0 \to D_s^+ D_s^-$ has therefore a lower branching fraction predicted to be smaller than $3.6 \cdot 10^{-5}$ at 90% confidence level (PDG), compared to that of $B_s^0 \to D_s^- D_s^+$ $\left((4.40 \pm 0.05) \cdot 10^{-3}\right)$, which is by at least a factor 100 lower.

3 Motivations

The $B^0 \to D_s^+ D_s^-$ decay has not yet been observed experimentally. A limit on the branching fraction is currently set at $3.6 \cdot 10^{-5}$, which is close to a theoretical computation expecting it at $(1.12 \pm 0.15) \cdot 10^{-5}$ [5]. The LHC Run 1 and Run 2 data could represent enough statistics to make an observation possible, according to the number of events obtained for the $B_s^0 \to D_s^+ D_s^-$ decay in Run 1 alone [6].

In addition, the TMVA (Toolkit for MultiVariate Analysis) library dedicated to machine learning in the *ROOT* framework, was lately improved with the addition of a deep learning module. In particular, a new Deep Neural Network classifier was added, as well as a method providing access to the state-of-the-art library TensorFlow, through a Keras backend. The considerable amount of background expected in this analysis provides an opportunity to test these new classifiers.

4 The LHCb detector

The LHCb detector is a single arm forward spectrometer designed for precision study of beauty and charm physics at the LHC (Large Hadron Collider) at CERN. Its main purpose is to record the tracks of decay products of b and c hadrons, to perform precision measurements of their branching fractions, CP-violating parameters and lifetimes.

Since the b quark is heavy, the b hadrons containing it, formed by the pp collisions, have a low transverse momentum p_T and their trajectory form a low angle with respect to the beam axis. The LHCb experiment was designed accordingly, featuring a series of sub-detectors stacked behind each other along the the beam pipe.

A scheme of the full detector is shown in figure 3. A short description of each subdetector and its main task is given below:

VELO The VErtex LOcator is the core part of the detector. Made of 42 silicon strip modules surrounding the beam next to the collision point, it allows to locate the position of b hadron decay vertices to a precision of 10 μm, by tracking their decay products. The b hadrons travel $O(1\,\text{cm})$ in the detector. They are in general not

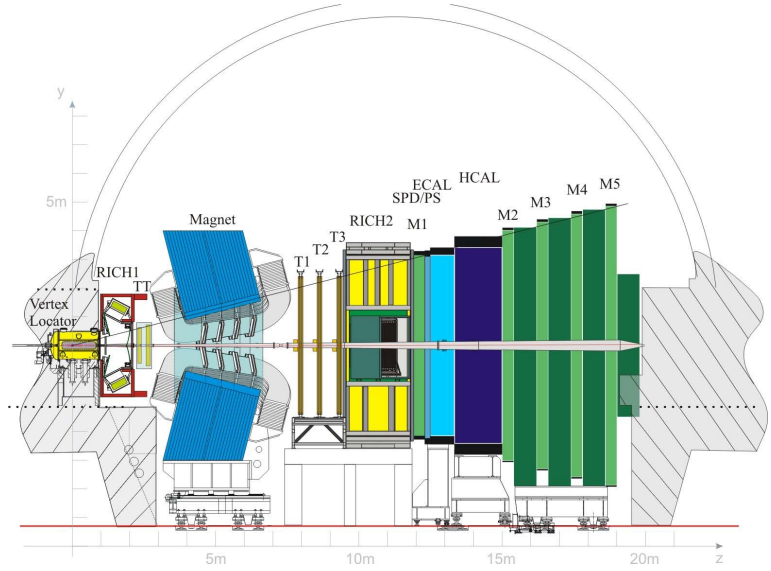


Figure 3: Scheme of the LHCb detector, with its sub-detectors.

directly detected, but their decay point is reconstructed, based on the precise track measurements of their daughters, made possible in the VELO. The modules are divided in two semicircular halves. Due to the close distance of the VELO modules to the beam (about 8 mm under running conditions), the VELO detector is held away from the beam during injection, to avoid damage by the high energy beams. During beam time, the two halves of the detector are moved towards each other until the half-circle modules join and surround the beam.

RICH The Ring Imaging CHerekov subdetectors are designed for particle identification, based on the Cherenkov radiation principle: a particle travelling faster than the speed of light in a given medium (for instance a dense gas) emits a light cone. The light cone is reflected by a mirror system into the hybrid photon detectors (HPDs) held further away from the beam to protect them against radiation. The light cone's opening angle is related to the velocity of the particle and can be determined in the RICH subdetectors. The RICH1 module is located before the bending magnet, and has a two component radiator gas (silica aerogel for low momentum particle of $O(1\,\mathrm{GeV})$ and C_4F_{10} for particles with momentum of $10 - 65\,\mathrm{GeV}$) whereas the RICH2 has a CF_4 radiator gas targeting particles with momenta in the range $15 - 100\,\mathrm{GeV}$, and is located after the bending magnet where supposedly high momentum particles are left.

Tracking System Formed by the **Trigger Tracker (TT)** positioned before the bending magnet, and the **Inner and Outer Tracker (IT & OT)** positioned after the bending magnet, the tracking system's main goal is to reconstruct the trajectories of the particles. The Trigger Tracker is made of silicon microstrip detectors, with a resolution down to 0.05 mm. The tracking stations after the magnet have an innermost part (Inner Tracker) also made of silicon strip detectors, to provide a

higher resolution close to the beam pipe where the density of particles is higher. The part further away from the beam (Outer Tracker) is made of straw tubes filled with a mixture of 70% Ar and 30% CO_2. The gas is ionized by the charged particles and the electrons drift towards the anode wire. Timing provides information on the position of the hit. The resolution is about 0.2 mm.

Combining information of the silicon tracker and outer tracker before and after the bending magnet, the curvature of the trajectory can be measured and thus the momentum of the particle can be determined. Based on this information, mass hypotheses are tested and the match between the expected ring diameter and the measurement in the RICH detector are evaluated to set the mass of the particle, and it can thus be identified.

Calorimeters The Electromagnetic CALorimeter (ECAL) is designed to measure the energy of electrons, positrons and photons. It consists of a sandwich structure of alternating metal and scintillating polystyrene layers. In the metal, the particles produce secondary particles, which will excite the scintillator molecules and produce UV light that is detected. The number of UV photons produced in the scintillator is proportional to the energy of the particle that produced the shower. The electromagnetic calorimeter is the part of the detector able to detect photons.

The Hadron calorimeter (HCAL) is based on the same principle and design as the electromagnetic calorimeter. Hadrons excite scintillating molecules which emit light which is detected and proportional to the energy of the incident particle. It is mainly used to detect neutrons and neutral mesons. In addition a Scintillating Pad Detector (SPD) and Pre-Shower Detector (PS) are positioned before the calorimeters, and are mainly used at the trigger level. The SPD determines whether the incident particle has a charge or not, while the PS differentiates between charged (e) and neutral (γ) electromagnetic particles.

Muon System The muon system is composed of five muon stations, made of multi-wire proportional chambers, containing a mixture of Ar, CO_2 and CF_4. The muon information is used for reconstruction and high transverse momentum muons are used in the low level trigger (L0) already.

5 Multivariate analysis techniques

Several Machine Learning algorithms can be used as Multivariate analysis (MVA) methods, especially for classification problems. In the field of High Energy Physics (HEP), the most common task is to discriminate between signal and background. The signal is the decay of interest, while background contains other physical processes, as well as non-physical background. The algorithms are designed such that the method "learns by itself" patterns and structures inherent to the data to be analysed. Thus, a particularity of machine learning algorithms, is that they undergo a training phase. In other words, the

algorithm has internal degrees of freedom that adjust to the data it has to analyse, trying to extract its particularities.

Classification problems enter the category of *supervised learning* in opposition to *unsupervised learning*, where unlabelled data are provided to the algorithm. In supervised learning, the algorithm is provided with labelled data, which means the expected output of the algorithm is provided. For classification problems, each event is given with the class to which it belongs to. The training then becomes a minimisation problem, where the internal degrees of freedom of the algorithm are adjusted to minimise an *error function*. The error function is commonly based on the difference between the classifier output and the expected output according to a certain metric.

One risk of this procedure is *overtraining* (see sections 5.3.2 and 5.3.3). We speak of overtraining or overfitting, when the algorithm learns too detailed properties of the training set, that are not generally characteristic of the data it was taken from. In the extreme case, the algorithm can even learn the full training set, without being able to generalize (see section 5.3.2).

Overtraining

There exist several techniques to reduce the risk of overtraining, explained in section 5.3.3. There are also a couple of ways to check for overtraining. In particular, the training data are split into a *training set* and a *test set*. The training of the algorithm is performed on the training set essentially, and then its performance is evaluated on the test set. If the algorithm achieved a very low error on the training data, but shows bad classifying performance on the test set, it means it "has learnt" features too specific to the training set, that do not generalise to the test set and the data in general.

Another useful technique to check for overtraining is to superimpose the classifier output distributions for the training set and the test set. Without overtraining, the two distributions are expected to match, there should be very little difference between the histograms. The Kolmogorov-Smirnov test (see section 8.1.3) can provide a quantitative measure of the agreement between two distributions. Although not recommended on binned data, it is frequently used (including in this analysis) as overtraining check.

5.1 Deep Neural Networks

A *Neural Network*, also called *Artificial Neural Network* is a collection of processing units called *(artificial) neurons*, interconnected together [7]. Their functioning was inspired by how neurons in the brain are connected together, and send stimuli to each other (see section 5.2). In most of the architectures (including the ones available in TMVA), the neurons are organised into sequential *layers*, where the neurons in one layer only depend on the outputs of the previous layer. Figure 4 shows a scheme of a simple neural network architecture. The first layer is called *input layer* and the last layer *output layer*. The

model can contain one or more intermediate layers called *hidden layers*, in which case it is called a *Deep Neural Network* (see section 5.3).

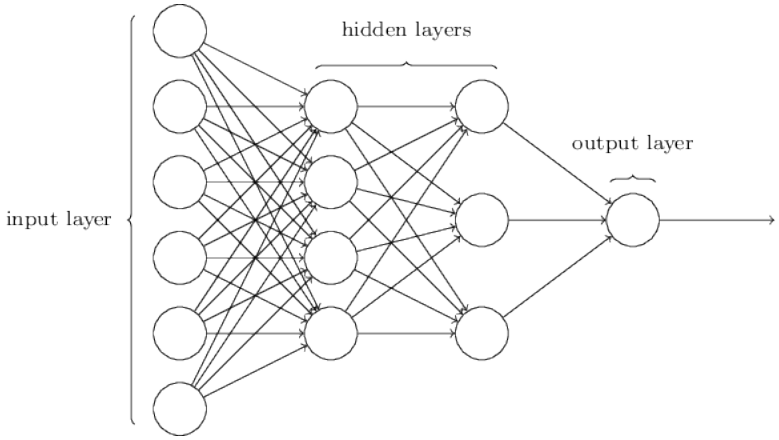


Figure 4: Scheme of a Deep Neural Network (DNN).

Mathematically, a feed-forward multilayer perceptron can be represented as a function $F : \mathbb{R}^n \to \mathbb{R}^m$ that maps the inputs $\{x_i\}$ with $i \in [1, n]$ to the outputs $\{y_k\}$ with $k \in [1, m]$:

$$\{x_i\} \to \{y_k\} = F(\{x_i\}) \tag{2}$$

The input layer $\{x_i\}$ must have the same dimension as the dataset (the number of input neurons must be equal to the number of variables chosen from the dataset for training). The input layer does not really play a role in the learning, it is just a representation of the data vector, and its output values are simply the values of the variables in the dataset. The output layer can be of any wished dimension, and depends on the task the network is aimed to perform. In particular, in case of binary classification, the output layer will consist of two neurons (one for each class), and in case of multiclass classification featuring n classes of n neurons. The output neurons of classifier networks usually have a *softmax* (or *sigmoid*) activation function, and give the probability of the event belonging to each class based on the classifier training. In TMVA, the algorithms are dedicated to classification (binary or multiclass) or regression.

5.2 Artificial neuron

An artificial neuron is a logical unit the output of which is given by a function of its inputs (see scheme 5). Each neuron of a given layer takes as inputs the outputs of every neuron in the previous layer. A given neuron contains a set of weights $\{\omega_i\}$ for each input. Each input (x_i) is multiplied by its weight (ω_i) before being added up together. Eventually, a bias b is added to the sum. The neuron also contains an *activation function* α, which is responsible for the neuron's response. The output is given by the activation function α evaluated for the sum of the inputs multiplied by their respective weight and the bias:

$$y_{output} = \alpha\big(\vec{\omega} \cdot \vec{x} + b\big) \tag{3}$$

The activation function significantly affects the behaviour of the neural network. Especially, if α is linear, the neural network will only be able to perform linear transformations of the input space. The breakthrough neural networks brought about is their ability to model complex and non-linear structures present in data. This is where the activation function plays a crucial role.

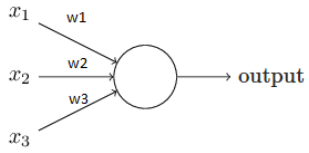


Figure 5: Scheme of an artificial neuron.

When connected together into a network, the neurons receive *stimuli* from the outputs of the previous neurons. The output of the neuron depends on the stimuli it received from its inputs, how each input is weighted, as well as on the activation function. It produces an output response, which can in turn acts as stimuli to other neurons. This behaviour is inspired by the functioning of the brain, and explains the name given to the "processing units".

5.3 Deep learning

With the increasing computing power, the training of larger and larger neural networks became possible, and gave rise to the field known as *deep learning*. Deep learning is, however, not restricted to neural networks, but englobes a larger scope of algorithms featuring non-linear processing units arranged in a large number of layers [8]. The layers are usually *feed forward*, which means that they are connected such that the inputs of each layer are the outputs of the previous one. Each layer is supposed to contain a representation of the data of a different level of abstraction. The higher (the further away from the data) the more abstract the representation is.

5.3.1 Cooking with hidden layers

The number of hidden layers and their number of neurons are generally not established, and depend on the problem to be solved. It is mathematically proven [7] that a network with one single hidden layer of infinite dimension can fit any continuous function F. However, adding several hidden layers of reasonable size can show similar performance, with fewer neurons. There is no general rule on how to design the network architecture, neither clear limits on the number of neurons per layer and the layers themselves. There are however some considerations based on rigourous proofs, that can help avoiding inefficient architectures.

Number of neurons in the layer

It can be shown [9] that some classification tasks need at least one layer larger than the input layer to be able to learn an accurate representation of the data. This is the case when one class is surrounded by another class in the hyperspace of the dataset. Each layer is a homeomorphism; or its weight matrix has a determinant of 0, in which case the data are collapsed along one axis and the classes become entangled. In case it is a homeomorphism, the effect of the layer is to distort the feature space until a hyperplane can separate the classes: it maps each point of the input into the output space, in a continuous way. The two classes, if one surrounded by the other at the input of the layer, thus cannot be disentangled in the output unless one or more additional dimensions are added to the output space (see [9] for more details).

5.3.2 Upper limit in the number of neurons

A naive and effective way to improve the performance of a neural network, is to build it deeper (more hidden layers) and with more neurons per hidden layer. It gives the net the ability to learn more detailed patterns in data, and to model more accurately the structure present in the training sample.

One of the major problems of deep learning is *overtraining* or *overfitting*. In short, it means that the model has as many parameters as to be able to learn the dataset itself, instead of its inherent structure. This becomes a major problem when trying to improve the performance by increasing the number of neurons in the hidden layers or the number of layers itself. It has been shown that a two layer neural network with $p = 2n + d$ parameters can fit perfectly any dataset of size $n \cdot d$ (n entries in d dimensions) [10]. Its hidden layers learn the training set and achieve zero error on it, but on the testing set the classifier accuracy is bad. With a larger number of layers k, the full overtraining happens already with $O(n/k)$ neurons per layer. To push the demonstration further, [10] show that even when replacing the labels by random ones during training, the network achieves zero training error. Of course when testing on the test set the error is catastrophic.

It is essential to bear in mind this limit, especially in high energy physics where the training sets can become very small after the PID selection.

5.3.3 Overtraining

There exist several *regularization* techniques to reduce overtraining. One of the most effective is called *Dropout*, and consists in setting randomly the output of the neurons to zero during the training. Each layer's neurons are given a dropout probability, which is the probability for the neuron's output to be set to zero. This forces the network to learn the representation in a more robust way, since it cannot rely on single neuron connections to model certain features. Usually, a dropout probability between 0 and 0.5 is suggested, but this value needs to be tuned according to the problem.

5.4 Boosted Decision Trees

A *Boosted Decision Tree* (BDT) classifier is a series of single decision trees derived from the same training set. Yet alone, a series of decision trees is called a *Random Forest*, where the output of the classifier is an average of each single tree's decisions. Instantiating multiple trees conveys the classifier a better stability against fluctuations in the training set. *Boosting* is a technique that consists in reweighting the misclassified events, to give them more importance for the next instantiated tree. The main difference it brings about compared to *Random Forests* is that the classifier gains also in accuracy by reducing misclassification. Boosting is a very powerful technique that made BDTs become of the best classifiers for several years. Their simpler functioning than neural networks and their shorter training time make them often the most chosen classifier. The boosted decision tree classifier will serve as a benchmark in this part of the work, and will finally be used as classifier because it showed better performance on the small MC datasets available.

For the used BDT algorithms, *AdaBoost* (adaptative boost) algorithm was used. It consists in multiplying the weights of the previous tree for misclassified events by a *boost weight* α, given by equation 4. The events are renormalized, in order to keep the sum of weights constant.

$$\alpha = \frac{1-\epsilon}{\epsilon} \tag{4}$$

where ϵ is the error of the misclassification rate of the previous tree.

The classifier response of a Boosted classifier is given by:

$$y(\vec{x}) = \frac{1}{N} \cdot \sum_{i}^{N} ln(\alpha_i) \cdot h_i(\vec{x}) \tag{5}$$

A decision tree is made of several *nodes*, which are the points in the tree where a decision splits the dataset in two. A leaf is the end node of a series of decisions, and is categorized into signal or background. The classification result of a BDT can be seen as a split of the phase space in hypercubes of signal and background.

Several parameters can affect the performance of a BDT. The most significant ones are the minimal node size *MinNodeSize* and the maximal depth *MaxDepth*. MaxDepth is the maximal number of successive decisions allowed in the tree. Since boosting works best on weak classifiers, it is recommended to keep the depth small (about 2 or 3) for BDTs. The minimal node size will be the minimal percentage of the training set that is required in a node. Once the MinNodeSize is attained, no further decision is allowed on the branch of the tree. A smaller minimal node size will allow more detailed feature modelling by the MVA, but is also subject to overtraining. Similarly, a too large number of successive decisions (high depth) will also favour overtraining. These parameters will be studied to determine their optimal values as part of this analysis.

6 Data samples

The selection is initially realised on the data collected during 2011 and 2012 in "Run 1" of the LHC. All the selection efficiencies are obtained from Monte-Carlo simulations of both years when available, or using the available year. A consistency check is made on samples where simulations for both years were available (see section 10.1). The MVA methods are trained using normalisation channel (see 6.2) MC samples from 2011 and 2012 simulation as well as Run 1 wrong-sign data.

Eventually, 2015 and 2016 data are added to enhance the statistics. Since there are currently no Monte-Carlo simulations available for Run 2 regarding signal as well as any of the backgrounds, the efficiencies computed using Run 1 are used (2012 which is assumed to be closer to Run 2). Special care was taken on the Particle Identification (PID) requirements (see section 7.2). All PID requirements in the strippings $21r1$, 21, $24r0p1$ and 28 were compared and the strongest requirement among all years is applied to all data in the offline selection. For the rest of the decay reconstruction, the requirements were left as they are in the respective strippings. The trigger underwent significant changes between Run 1 and Run 2. Further studies of the trigger efficiency would require Run 2 MC. For this report we make no assumption regarding the Run 2 trigger efficiency.

The data files from all years are taken from the $B02DDBeauty2CharmLine$ in the $Bhadron$ or $BhadronCompleteEvent$ streams. A summary of their stripping version and lines are provided in the table 1.

Year	Data type	Stream	Stripping version
2011	normal/WS	Bhadron	$21r1$
2012	normal/WS	Bhadron	21
2015	normal	BhadronCompleteEvent	$24r0p1$
2015	WS	Bhadron	$24r0p1$
2015	normal	BhadronCompleteEvent	28
2015	WS	Bhadron	28

Table 1: Data samples analysed. All samples were taken from the $B02DDBeauty2CharmLine$.

The MC files are also taken from the $B02DDBeauty2CharmLine$ in the $Bhadron$ stream. The stripping versions vary between 20, $20r1$, 21 and $21r1$, depending on the decay. The full list of the different MC files used for signal as well as for background is shown in table 2.

6.1 Signal and sidebands definition

The analysis was performed blinded during the whole selection process, until the fit. First, both the invariant mass region of the B^0 and the B_s^0 were blinded, using a mass window

Year	Data type	Stream	Stripping version					
	$B_s^0 \to D_s^+ D_s^-$							
2011	Bs_DsDs=DDALITZ,DecProdCut_pCut1600MeV	13296003	Beam3500GeV-2011-MagDown_Nu2-Pythia6	Sim08a	Digi13	Reco14a	20r1	520499
			Beam3500GeV-2011-MagDown_Nu2-Pythia8	Sim08a	Digi13	Reco14a	20r1	521999
			Beam3500GeV-2011-MagUp_Nu2-Pythia6	Sim08a	Digi13	Reco14a	20r1	511500
			Beam3500GeV-2011-MagUp_Nu2-Pythia8	Sim08a	Digi13	Reco14a	20r1	520498
2012	Bs_DsDs=DDALITZ,DecProdCut_pCut1600MeV	13296003	Beam4000GeV-2012-MagDown-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	387998
			Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	377200
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	385198
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	379999
	Bs_DsDs,KKpi3pi=DDALITZ,DecProdCut_pCut1600MeV	13296011	Beam4000GeV-2012-MagDown-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	378200
			Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	388399
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	384199
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	382798
	Bs_DsDs,KKpiKpipi=DDALITZ,DecProdCut_pCut1600MeV	13296021	Beam4000GeV-2012-MagDown-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	383397
			Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	379798
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	387400
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	383999
	Bs_DsDs,3pi3pi=DDALITZ,DecProdCut_pCut1600MeV	13296031	Beam4000GeV-2012-MagDown-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	377499
			Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	375997
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	427499
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	399499
	$B^0 \to D D_s$							
2011	Bd_DsD=DDALITZ,DecProdCut_pCut1600MeV	11296012	Beam3500GeV-2011-MagDown_Nu2-Pythia8	Sim08h	Digi13	Reco14c	21r1	1343179
			Beam3500GeV-2011-MagUp_Nu2-Pythia8	Sim08h	Digi13	Reco14c	21r2	1316762
2012	Bd_DsD=DDALITZ,DecProdCut_pCut1600MeV		Beam4000GeV-2012-MagDown-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	886996
			Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	907493
			Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08h	Digi13	Reco14c	21	2545792
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia6	Sim08a	Digi13	Reco14a	20	892500
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	911498
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim08h	Digi13	Reco14c	21	2603052
	$\Lambda_b^0 \to \Lambda_c D_s$							
2011	Lb_LcDs,pKpi,Kkpi=DecProdCut	15296003	Beam3500GeV-2011-MagDown-Nu2-Pythia8	Sim09b		Reco14c	21r1	101531
			Beam3500GeV-2011-MagUp-Nu2-Pythia8	Sim09b		Reco14c	21r1	105733
2012	Lb_LcDs,pKpi,Kkpi=DecProdCut	15296003	Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08h	Digi13	Reco14c	20	114232
			Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim09b		Reco14c	20	251349
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	sim08h	Digi13	Reco14c	20	109315
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim09b		Reco14c	20	254110
	$B_s^0 \to D_s K K \pi$							
2011	Bs_DsKKpi,Kkpi=DecProdCut	13166030	Beam3500GeV-2011-MagDown-Nu2_Pythia8	Sim08e	Digi13	Reco14a	20r1	319747
			Beam3500GeV-2011-MagUp-Nu2_Pythia8	Sim08e	Digi13	Reco14a	20r1	352390
	$B_s^0 \to D_s^* D_s$							
2012	Bs_DsstDs=DDALITZ,DecProdCut_pCut1600MeV	13296206	Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	251500
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	250000
	$B_s^0 \to D_s^{*+} D_s^{*-}$							
2012	Bs_DsstDsst=DDALITZ,DecProdCut_pCut1600MeV	13296205	Beam4000GeV-2012-MagDown-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	251998
			Beam4000GeV-2012-MagUp-Nu2.5-Pythia8	Sim08a	Digi13	Reco14a	20	254500

Table 2: Monte-Carlo samples used in the analysis.

around the mass values found in the PDG. The width of the blinded region is taken for both the B_s^0 and the B^0 candidate at $\pm 5\sigma$ from a previous analysis [6]. The value σ taken is 7.9 MeV and the mass values are taken from the 2014 edition of the PDG. The different regions in the B invariant mass are shown in table 3. In the final phase of the selection process, the B_s^0 mass window was dropped and only the B^0 stayed blinded until the final selection was set and applied.

Region	range [MeV]
Lower sideband	$m_B < 5240.08$
B^0 mass window	$\lvert m_B - 5279.58\rvert < 39.5$
Intermediate sideband	$5319.08 < m_B < 5327.27$
B_s^0 mass window	$\lvert m_B - 5366.77\rvert < 39.5$
Upper sideband	$m_B > 5406.27$

Table 3: B candidate invariant mass division and blindings. A blinding of $\pm 5\sigma$ from the B^0 and B_s^0 PDG masses was applied, taking $\sigma = 7.5$ MeV from the B_s peak width in another analysis.

6.2 Normalisation mode

In this analysis we use the $B_s^0 \to D_s^+ D_s^-$ decay as a normalisation channel, where we assume identical selection efficiencies for the B_s^0 and B^0 modes. The branching fraction of the signal decay can be determined as a ratio with respect to the normalisation channel. We reconstruct both the signal and normalisation channel in the same final states, allowing direct comparison between the two modes. The same final state particles ensure the same efficiencies, provided no requirement strongly correlated to properties that differ between the decay modes is made. The branching fraction of both decays are shown below, as well as the b quark hadronisation fractions.

$f(\bar{b} \to B^0)$	0.404 ± 0.006
$f(\bar{b} \to B_s^0)$	0.103 ± 0.013
$\mathcal{B}(B^0 \to D_s^+ D_s^-)$	$3.6 \cdot 10^{-5}$ at 90%
$\mathcal{B}(B_s^0 \to D_s^+ D_s^-)$	$(4.4 \pm 0.5) \cdot 10^{-3}$

Table 4: Branching fractions and b-quark hadronisation fractions for the B_s^0 and B^0 candidates. The values are taken from the PDG.

The MVA methods are also trained and tested using MC from the $B_s^0 \to D_s^+ D_s^-$ decay, since no MC was available for the $B^0 \to D_s^+ D_s^-$ decay.

Name	Decay
B2DsDs1 :	$B^0 \to (D_s^+ \to K^+K^-\pi^+)(D_s^- \to K^-K^+\pi^-)$
B2DsDs2 :	$B^0 \to (D_s^+ \to K^+K^-\pi^+)(D_s^- \to \pi^-\pi^+\pi^-)$ & CC
B2DsDs3 :	$B^0 \to (D_s^+ \to K^+K^-\pi^+)(D_s^- \to K^-\pi^+\pi^-)$ & CC
B2DsDs4 :	$B^0 \to (D_s^+ \to \pi^+\pi^-\pi^+)(D_s^- \to \pi^-\pi^+\pi^-)$

Table 5: B meson decay channels, with their names used in this document

7 Selection

The event selection consists of three distinct parts. The first selection is made by the stripping, where events are categorized into streams and stripping lines, containing a certain type of events. In this case, the stripping line was $B02DDBeauty2CharmLine$ and the stream was $Bhadron$ for the 2011 and 2012 data as well as all the WS data, and $BhadronCompleteEvent$ for the 2015 and 2016 data.

The second step, called preselection, happens during the gathering of the data into tuples, and is mainly aimed at reducing their size. Besides choosing the variables to be present in the tuple, some selections were applied. These selections are covered in section 7.1.

The last step, which is the core part of this analysis is the offline selection. The offline selection consists here in two more parts. A "classic" selection, where Particle Identity (PID) requirements, background vetoes and striking reconstruction quality requirements are applied under the form of cuts (see section 7.2). A second selection is made using MVA (Multivariate Analysis) methods, mainly based on reconstruction quality and vertex separation variables, as well as kinematical variables uncorrelated with the invariant mass of the signal (detailed in section 8). A first method is trained to discriminate against combinatorial background, and another more specific method tackles D_s^* background. In the first steps, the offline selection was not applied and the data were directly fed into the MVA after the preselection. However, the enormous combinatoric yield remaining after stripping was too large to train and test on. Instead, an offline selection similar to that of the previous analysis [6] was applied prior to MVA training.

In this analysis, the decay candidates are taken from the four different final states (or channels) listed in table 5 and referred to with the names in the first column for the rest of the document. In the whole document, charge conjugated final states are included also when referring to a decay.

7.1 Preselection

In the preselection, no strong requirements were made on the B and D candidates. This was done on purpose, in order to make as much information available to the MVA as possible. In particular, track quality variables were planned to be used in the MVA

selections, and thus the decision was taken to not limit the sample by applying a cut at this stage. A PID selection on the final state particles is made, based on the stripping identifications. In addition, a loose D_s candidate mass window of $\pm 100\,\text{MeV}$ from the PDG value is applied.

7.2 Offline selection

In order to select further and to bring down the number of events in the tuples, an offline selection was performed. The offline selection consists in Particle IDentity (PID) requirements, as well as fit quality and vertex separation requirements on the B and D candidates.

To categorise the decay candidates into the four channels listed in table 5, each D_s candidate undergoes a PID selection. The D_s candidates are divided into 3 final states: 1. $K^+K^-\pi^+$, 2. $K^+\pi^-\pi^+$ and 3. $\pi^+\pi^-\pi^+$. The criteria are explained in the following sections and summarized in section 7.2.1. The appropriate combinations of D_s candidates from these three categories are formed to make the decay tuples of table 5.

7.2.1 Summary of the offline selection

All decay candidates are required to satisfy:

- Impact parameter χ^2 of the B candidate $\chi^2_{IP}(B) < 20$
- The vertex fit per degree of freedom for the B candidate $\chi^2/n_{d.o.f} < 8$
- The difference in χ^2 when including or excluding the D candidates in the B vertex is required to satisfy $\Delta\chi^2_{vx} > 100$
- The final products are required to have:
  For K candidates : $DLL_{K\pi} > -5$
  For π candidates : $DLL_{K\pi} < 10$
- Trigger: TOS is required on the HLT2 Topo $2, 3$ or 4 body trigger
- No additional requirement on the B lifetime is made to the one in the stripping, which is $\tau > 0.2\,\text{ps}$

In addition, each final state has its own additional requirements:

1. $D_s \to KK\pi$
   - The π is required to have $ProbNNpi > 0.01$

- D^{*+} veto: the invariant mass difference between the D_s^+ candidate and the kaon pair satisfies : $m(K^+K^-\pi^+) - m(K^+K^-) > 150\,\text{MeV}$

(a) $D_s^+ \to \phi\pi^+$

- The K are required to have $ProbNNk > 0.01$
- The invariant mass of the ϕ candidate satisfies : $m(K^+K^-) < 1040\,\text{MeV}$

– The D candidate is required to decay at $z_D - z_B > -1.0\,\text{mm}$

(b) $D_s^+ \to (\bar{K}^{*0} \to K\pi)\ K$

- Fail (a)
- The K are required to have $ProbNNk > 0.05$
- The invariant mass of the $\bar{K}^*$ candidate satisfies : $|m_{K^-\pi^+} - 892| < 100\,\text{MeV}$
- D^+ veto: **either** $|m(\pi^+K^-\pi^+) - 1869| > 25\,\text{MeV}$ with the K^+ mass swapped to π^+ **or** (**$log(ProbNNk/ProbNNpi) > 0.35$ and $p < 80\,\text{GeV}$** for the K^+)

– The D candidate is required to decay downstream from the B: $z_D - z_B > 0\,\text{mm}$

– The χ^2 vertex separation between the D and B satisfies $\chi^2_{FD}(D-B) > 2$

(c) Other $D_s^+ \to K^+K^-\pi^+$

- Fail (a) and (b)
- The K are required to have $ProbNNk > 0.12$
- Same D^+ veto as for (b) : **either** $|m(\pi^+K^-\pi^+) - 1869| > 25\,\text{MeV}$ with the K^+ mass swapped to π^+ **or** (**$log(ProbNNk/ProbNNpi) > 0.35$ and $p < 80\,\text{GeV}$** for the K^+)

– The D candidate is required to decay downstream from the B: $z_D - z_B > 0\,\text{mm}$

– The χ^2 vertex separation between the D and B satisfies $\chi^2_{FD}(D-B) > 2$

2. $D_s^+ \to K^+\pi^-\pi^+$

- Fail (1) with the π^- mass swapped to K^-
- The K is required to have $ProbNNk > 0.05$
- The π are required to have $ProbNNpi > 0.01$
- A $D^+ \to \pi^+K^-\pi^+$ veto is applied :
  either the D_s^+ invariant mass satisfies $|m(\pi^+K^-\pi^+) - 1869| > 25\,\text{MeV}$ with the K^+ mass swapped to π^+ and the π^- mass swapped to K^-
  or the K^+ is required to have $log(ProbNNk/ProbNNpi) > 0.35$ and $p_{K^+} < 80\,\text{GeV}$ and the π^+ is required to have $log(ProbNNpi/ProbNNk) > 0.35$ and $p_{\pi^-} < 80\,\text{GeV}$

- A D^{*+} veto is applied :
  The D_s^+ invariant mass satisfies $m(K^+K^-\pi^+) - m(K^+K^-) > 150\,\text{MeV}$ with the π^- mass swapped to K^-

– The D candidate is required to decay downstream from the B: $z_D - z_B > 0\,\text{mm}$

– The χ^2 vertex separation between the D and B satisfies $\chi^2_{FD}(D-B) > 2$

3. $D_s^+ \to \pi^+\pi^-\pi^+$

- Fail (1) with the π^- mass swapped to K^- and one of the π^+ masses swapped to K^+
- Fail (2) with one of the π^+ masses swapped to K^+
- The π are required to have $ProbNNpi > 0.01$
- A $D^{*+} \to (D^0 \to \pi^+\pi^-)\pi^+$ is applied:
  The invariant D_s^+ mass satisfies $m(\pi^+\pi^-\pi^+) - m(\pi^+\pi^-) > 150\,\text{MeV}$ for both π^+ taking part in a D^0 candidate.
- A $D^{*+} \to (D^0 \to \pi^+K^-)\pi^+$ veto is applied:
  The invariant mass difference between the D_s^+ satisfies $m(\pi^+K^-\pi^+) - m(\pi^+K^-) > 150\,\text{MeV}$ for both π^+ taking part in a D^0 candidate with the π^- swapped to K^-.

– The D candidate is required to decay downstream from the B: $z_D - z_B > 0\,\text{mm}$

– The χ^2 vertex separation between the D and B satisfies $\chi^2_{FD}(D-B) > 6$

First, the selection was performed with only the PID requirements, marked with a (•) in 7.2.1, and the MVA methods were applied directly after. It turned out that the requirements were not strong enough and a huge amount of background persisted in the B^0 mass region. Thus, the additional fit quality and flight distance criteria on the B and D candidates were added, preceded by a (−) in 7.2.1. In addition, the Delta-log likelihood (DLL) were sharpened to match the strongest stripping requirement among the 2011, 2012, 2015 and 2016 lines.

8 MVA selection

A large amount of background remains in the whole B candidate invariant mass range after the offline selection so far, as shown in figure 6. MVA methods were designed to discriminate against combinatorial background, in order to bring down the combinatorial background yield. The first series of trainings involved a separate method for each channel (B2DsDs1, B2DsDs2, B2DsDs3, B2DsDs4), and featured three different methods to be compared: the new CPU accelerated DNN (DNN CPU) method added to TMVA in summer 2016 and benchmarked during summer 2017, a TMVA deep neural network method

featuring a TensorFlow backend (PyKeras), and a BDT as benchmark. The results of the first training session are briefly presented in section 8.1.2.

A way to compare the classifier performance is the ROC (Receiver Operating Characteristic) curve, which is the background rejection plotted as function of the signal efficiency. The ROC curves of the first trainings on the B2DsDs1 channel are shown in figure 8. It was used as benchmark to choose the classifier with the best separating power. The area under the ROC curve (AUC) represents a numerical benchmark for the classifier, and was used later, for the hyperparameter optimisation (section 8.1.3).

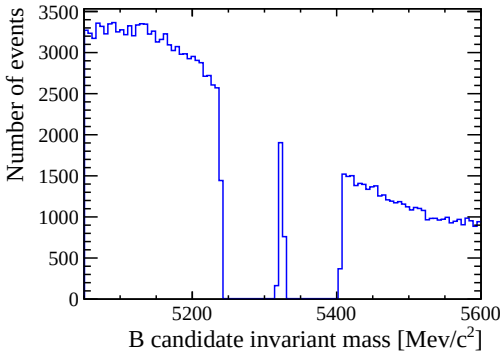


Figure 6: Background left in the B candidate invariant mass after the offline selection presented in 7.2.1 was applied.

After the combinatorial (WS) discriminating MVA, a series of MVAs was launched to discriminate against the sidebands, but with low additional discriminating power and poor performance. A training against background taken from the lower sideband only was carried out, without a clear increase in performance. The results of these trainings are not shown here. Finally, the second MVA method was trained against a $B_s^0 \to D_s^* D_s$ MC sample as background. The tuning is described in section 8.2. Since the MC sample for the D_s^* was produced with the D_s candidates decaying to $KK\pi$, a significant number of events persisted only in the B2DsDs1 channel after the offline selection. Thus, a single MVA for all channels was trained. The final choice of WS MVA used to discriminate against combinatorial background was also featuring a single method, trained on a mix of signal taken from the four channels. It was done in order to simplify the analysis, and because the input variables chosen, described in section 8.1.1 are independent of the final state (namely by taking *min* and *max* of the variables among the final products).

8.1 Combinatorial background MVA

8.1.1 Choice of discriminating variables

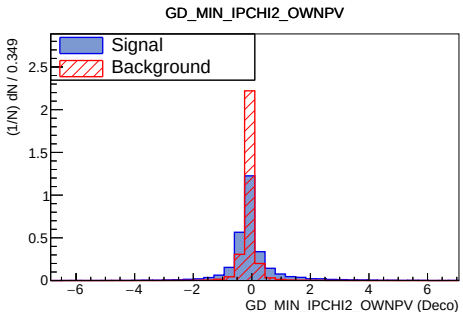
As input to the MVA methods a set of kinematical variables, as well as reconstruction quality indicators was chosen. The full list is shown in table 6. Special care was taken not to use any mass-correlated variable, since the MVA method is trained on MC data of the

$B_s^0 \to D_s^+ D_s^-$ normalisation mode decay. The PID related variables were also left out, as the PID selection is done extensively in the offline selection (section 7.2). The variables shown in table 6 were chosen on their separation power,

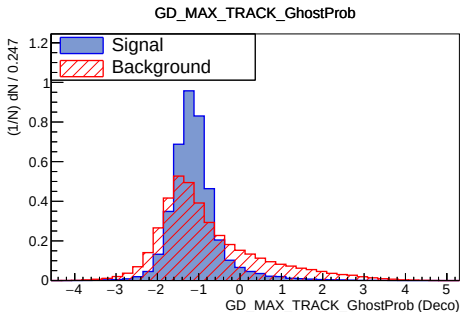
$$V_{sep} = \int_{-\infty}^{+\infty} \frac{1}{2} \left(\frac{s(x) - b(x)}{s(x) + b(x)} \right)^2 \cdot dx \tag{6}$$

where s and b are the signal respectively background densities.

The reason for using the separation between signal and background is that it is method-independent. Indeed, the initial goal was to use the DNN method and to keep the BDT method as a benchmark. More specific variable ranking to the BDT method will be shown in section 8.2. A table of the 20 most separating variables is also shown in Table 7, and the signal and background distributions are shown in figure 7. For the D_s candidates, the *min* and *max* value between the two was used, in order to prevent the MVA method from learning decay specific features to one particular channel. This is specially useful for the B2DsDs2 and B2DsDs3 channels, with different final states for the D_s candidates. For the final state particles, also the *max* and *min* values among the six candidates were taken as inputs to the MVA methods. Again, it reinforces the robustness of the method against learning decay specific features, and also reduces the number of variables for the final state particles.



(a) Minimal IP χ^2 w.r.t. PV among the final products.



(b) Maximal ghost probability among the final products.

Figure 7: Signal drawn from 2011 and 2012 MC superimposed to background taken from 2011 and 2012 WS data for the 2 most separating variables. The full list of histograms is not shown in the document but was used to check the quality of the MVA.

8.1.2 Results of the first training session

A series of DNN classifiers was launched and trained on of each of the channels (B2DsDs1, B2DsDs2, B2DsDs3 and B2DsDs4) against the corresponding wrong-sign (WS) channel from data, to discriminate between signal and combinatorial background. A BDT classifier with the default options was added as a benchmark. First, an architecture of 3 layers with 128 neurons each was chosen. Since the performance was not at the level of the benchmark (BDT classifier), the number of layers was increased to 5. The performance was still not

Variable	LOKI variable name
***B* candidate**	
Momentum	P
Transverse momentum	PT
Flight distance of the B candidate	FD_OWNPV
Vertex χ^2 separation between PV and SV	FDCHI2_OWNPV
Cosine of the angle between momentum and track $cos(\theta(\vec{p},\vec{x}))$	DIRA_OWNPV
χ^2 difference when including or excluding the B in the PV	OWNPV_CHI2
χ^2 difference when including or excluding the B in the SV	ENDVERTEX_CHI2
Number of degrees of freedom of the PV fit	OWNPV_NDOF
DecayTreeFitter $\chi^2/n_{d.o.f}$	DTF_CHI2NDOF
***D* candidates**	
Momentum	P
Transverse momentum	PT
Flight distance of the D candidate (w.r.t. SV)	FD_ORIVX
Flight distance of the D w.r.t. the PV	FD_OWNPV
$cos(\theta(\vec{p},\vec{x}))$ w.r.t. SV	DIRA_ORIVX
$cos(\theta(\vec{p},\vec{x}))$ w.r.t. PV	DIRA_OWNPV
Decay vertex χ^2 separation w.r.t. SV	FDCHI2_ORIVX
Decay vertex χ^2 separation w.r.t. PV	FDCHI2_OWNPV
χ^2 difference when including or excluding the D in the SV	ORIVX_CHI2
χ^2 difference when including or excluding the D in the PV	OWNPV_CHI2
Decay vertex χ^2	ENDVERTEX_CHI2
Number of degrees of freedom of the PV fit	OWNPV_NDOF
Impact parameter (IP) w.r.t. PV	IP_OWNPV
IP χ^2 w.r.t. PV	IPCHI2_OWNPV
Final products	
Momentum	P
Transverse momentum	PT
Origin vertex χ^2	ORIVX_CHI2
PV χ^2	OWNPV_CHI2
Number of degrees of freedom of the PV fit	OWNPV_NDOF
Impact parameter (IP) w.r.t. PV	IP_OWNPV
IP χ^2 w.r.t. PV	IPCHI2_OWNPV
Track fit χ^2 per degree of freedom	TRACK_CHI2NDOF
Track match χ^2	TRACK_MatchCHI2
Track ghost probability	TRACK_GhostProb
Track momentum fit χ^2	TRACK_PCHI2
Track in calorimeter (boolean)	hasCalo

Table 6: Input variables to the wrong-sign discriminating MVA method. For the D candidates and the final products, the *min* and the *max* of the variables among the particles was taken. GD stands for Grand Daughters and refers to the final state particles, whereas DS refers to the D_s candidates.

LOKI variable name	separation power
GD_MIN_IPCHI2_OWNPV	0.143738
GD_MAX_TRACK_GhostProb	0.125618
GD_MIN_hasCalo	0.123947
GD_MIN_TRACK_GhostProb	0.111433
DS_MIN_FDCHI2_ORIVX	0.111251
lab0_ENDVERTEX_CHI2	0.0844772
GD_MIN_P	0.0795826
DS_MAX_ORIVX_CHI2	0.0792568
lab0_FDCHI2_OWNPV	0.0740933
DS_MAX_IPCHI2_OWNPV	0.0708693
GD_MAX_TRACK_CHI2NDOF	0.0703473
DS_MAX_P	0.0668567
DS_MAX_FDCHI2_OWNPV	0.0646456
GD_MIN_PT	0.0641255
GD_MAX_P	0.0614862
DS_MAX_FDCHI2_ORIVX	0.0580403
DS_MIN_FDCHI2_OWNPV	0.0574959
GD_MIN_TRACK_PCHI2	0.0540742
DS_MIN_IPCHI2_OWNPV	0.053397
GD_MIN_OWNPV_NDOF	0.0483562

Table 7: Ranking of the 20 most discriminating variables between MC signal and WS data taken from Run 1.

close to the BDT classifier. Thus, an architecture of 3 hidden layers with 500 neurons per layer was tested. Increasing the number of neurons per layer can help better model the features in the data, if they are rather complex, as discussed in 5.3.1. This model however lead to a huge training time and to errors happening due to the large number of parameters. In the end an architecture of 500 neurons in the first hidden layer and 2 more hidden layers with 128 neurons each was built, and the results on the B2DsDs1 channel are shown in figure 8. On the other channels, similar results were obtained and the order of the classifiers was the same for all.

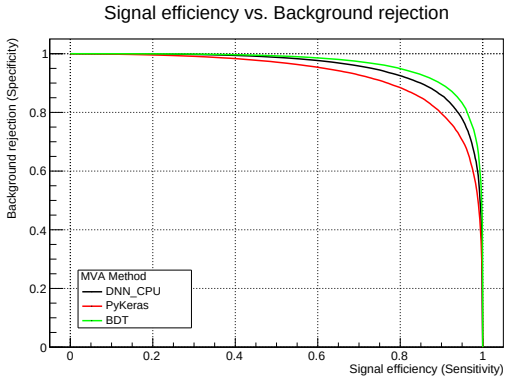


Figure 8: Training result of the TMVA DNN, PyKeras with TensorFlow backend and BDT classifiers, trained on signal MC against wrong-sign data in the B2DsDs1 channel.

The final DNN training was performed with the following classifiers:

- Boosted Decision Tree (BDT)
  - Minimal node size: 2.5%
  - Maximal tree depth: 3
  - Number of trees: 850
  - Number of cuts: 20
  - Boost type: AdaBoost
- CPU accelerated Deep Neural Network (DNN CPU)
  - Number of layers: 3
  - Number of neurons per layer: 500, 128, 128
  - Batch size: 256
  - Learning rate: 0.1
- TensorFlow backend Deep Neural Network (PyKeras)
  - Number of layers: 3
  - Number of neurons per layer: 500, 128, 128

– Batch size: 256

– Learning rate: 0.1

The poor performance of the DNN classifiers can be explained by the low statistics in the tuples. Indeed, it was not possible to train with more than a few thousand events, and a deep architecture with a high number of neurons per layer is likely to overtrain as discussed in 5.3.2. For instance, if the test set contains 5000 events and 60 variables are chosen as input to the classifier, the lower limit in the number of parameters where a 2 layers network with *relu* (rectified-linear) activation function can completely learn the training set is given by $p = 2 \cdot n + d = 2 \cdot 5000 + 60 = 10060$. Now, let us consider a network with two hidden layers, each having x neurons. With a dense architecture, each neuron from a layer is connected to every neuron in the next layer. The first hidden layer has x neurons with each 60 inputs and one bias. Thus the first layer has $61x$ parameters. The second hidden layer will have x inputs, with each $x + 1$ parameters, so $x \cdot (x + 1)$ parameters. Finally, the output layer, consisting of 2 neurons has x inputs and one bias for each neuron, so $2 \cdot (x + 1)$. The total number of parameters is given by: $p = 61x + x(x + 1) + 2 \cdot (x + 1) = x^2 + 64x + 2$. The maximal number of neurons in the layer to fully overtrain is given by $x = 73$. This is for a 2 layer network. Increasing the number of layers makes the maximal number of neurons per layer decrease further.

Since the DNN classifiers did not outperform the BDT used as benchmark, the decision was taken to abandon this classifier. The choice was also motivated by the fact that there was no operating version of the DNN method available on *Lxplus*, and that all the data had to be processed on the laptop computer, downstreaming huge amounts of data. In addition, the training time is much longer for a DNN than for a BDT classifier.

8.1.3 Hyperparameter tuning

For the final choice of MVA, the BDT classifier was chosen, since the DNN classifiers did not outperform the benchmark. A hyperparameter tuning on the BDT was carried out, by testing two of the most performance affecting parameters over the ranges suggested in [7]. The testing was performed using a mix of signal from the four channels, according to their respective proportions. The background was taken randomly from the corresponding wrong-sign channels. The results are shown in figures 9, 10a and 10b. These figures are not straightforward to interpret[1]: the Kolmogorov-Smirnov test values shown in figures 10a and 10b show that the test gives 0 for any tree depth greater than $2-3$. It means that for decision trees with a depth of 3 or more, we loose any measure for overtraining (except visually interpreting the classifier output distributions). Thus, a maximal tree depth of 3 was chosen for the WS BDT. Since boosting works best on weak classifiers, it makes

[1] A Kolmogorov-Smirnov test should not be performed on binned data, unless the variations to be modelled are larger than the binning. If the bins are larger than the distributions to be modelled, the test result stays less accurate than on unbinned data. So a result of zero can partly be explained by the binning, which degrades the test result. In addition, small numbers are rounded to zero.

sense to limit the tree depth. Too deep decision trees might lead to ineffective boosting, by often wrongly identifying events due to a too large number of consecutive decisions.

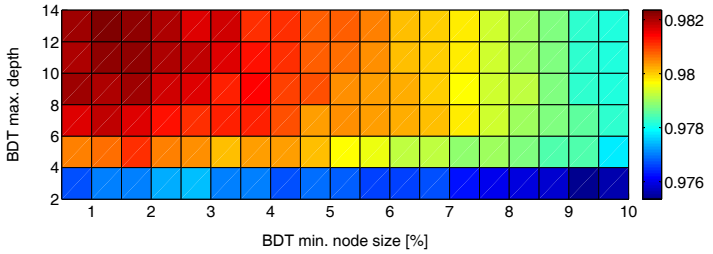


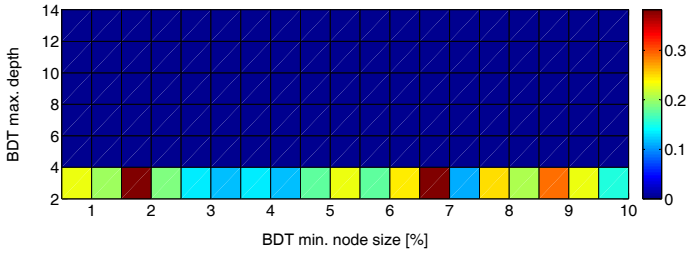
Figure 9: ROC integral values on the test set as function of the $MinNodeSize$ and $Depth$ of the decision trees for the WS BDT classifier.

For the minimal node size, the figure 9 does not provide sufficient information to decide the value. Thus a more detailed study was performed only testing $MinNodeSize$ values between 0.2 and 10% for a fixed maximal tree depth of 3. The results are shown in figures 11 and 12a and 12b. The values are still subject to important statistical fluctuations, but the curve in 11 shows a decreasing trend, starting from about 2.5%. Finally, a value of 2% was chosen for minimal node size of the WS BDT.

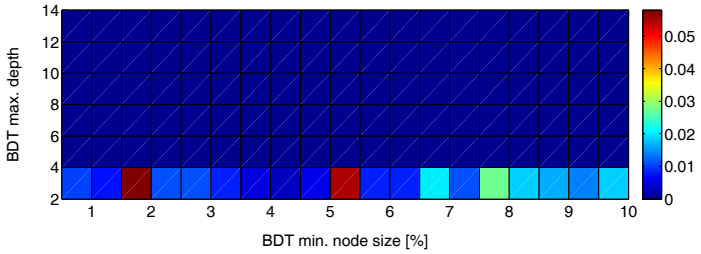
8.2 D_s^* discrimination

During a series of rather unsuccessful training attempts of signal MC against sidebands data as background, a significant amount of background in the lower sideband persisted. It is mostly due to the $B_s^0 \to D_s^* D_s$ decay, where the D_s candidate is genuine, but the photon is not reconstructed. Thus, a MVA method was trained especially against this source of background. The tricky part here is that the identified final products are exactly the same as for $B^0 \to D_s^+ D_s^-$, and the energy lost through the missing photon makes the invariant mass of the reconstructed B candidate peak into the B^0 mass region.

The input variables choice is similar to the one for the WS BDT shown in table 6. The variables $hasCalo$, as well as FD_OWNPV, FD_ORIVX, $FDCHI2\_OWNPV$ and $FDCHI2\_ORIVX$ for all candidates were removed. The ROC curves of the trainings with this initial set of variables do barely show better performance than random guessing (the black line in figure 15 is almost diagonal from point $(0,1)$ to point $(1,0)$, which corresponds to a ROC curve for random guessing).



(a) Kolmogorov-Smirnov test value on signal as function of the $MinNodeSize$ and $Depth$ of the decision trees. The Kolmogorov-Smirnov test is applied between the classifier output distributions of the signal training set and test set for the WS BDT.



(b) Kolmogorov-Smirnov test value on background as function of the $MinNodeSize$ and $Depth$ of the decision trees. The Kolmogorov-Smirnov test is applied between the classifier output distributions of the background training set and test set for the WS BDT.

Here, the low statistics left in the training samples make it impossible to use a DNN classifier. In order to respect approximately the 50% testing sample size, no more than 2000 events could be used for training. According to the discussion in 8.1.2 and 5.3.2, it was impossible to construct an effective DNN architecture for such a low number of events. Thus, a BDT was used, without even comparing with a DNN. The Boosted decision tree was built with the following parameters (when ranges are indicated, the value was tested over the range and the most optimal value was chosen):

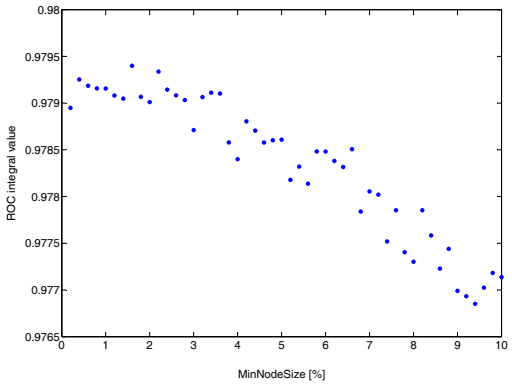
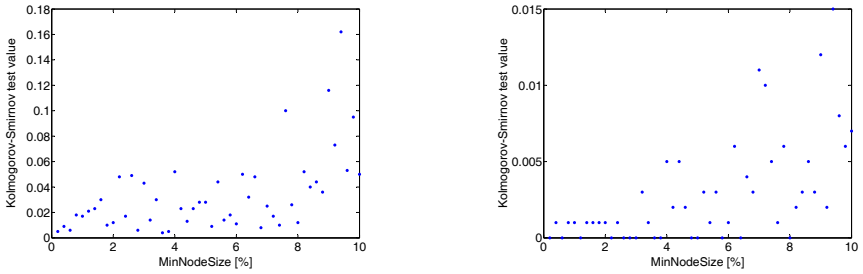


Figure 11: ROC curve integral values as function of the $MinNodeSize$ value of the WS BDT over a larger range. Although subject to large statistical fluctuations, a decreasing trend can be observed for increasing node size.



(a) (b)

Figure 12: Kolmogorov-Smirnov test value on signal (12a) and on background (12b) as function of the $MinNodeSize$ for a maximal tree depth of 3. The Kolmogorov-Smirnov is applied between the classifier output distributions of the signal training set and test set for the WS BDT. The test values fluctuate and an increasing trend is only observable from about 8% for signal and at about 3% for background.

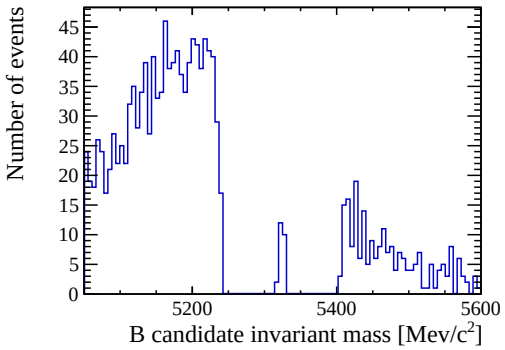


Figure 13: Background left in the B candidate invariant mass after the WS BDT is applied with the optimal cut value suggested by the Punzi F.o.M. (see equation 13).

- Boosted Decision Tree (BDT)

- Minimal node size: $0.5 - 10\%$
- Maximal tree depth: $0 - 10$
- Number of trees: 1000
- Number of cuts: 20
- Boost type: AdaBoost
- Decorrelation pretransformation

8.3 Constructing a new discriminating variable

An additional variable was constructed, with the aim of providing a sense for the missing energy of the photon. The variable used is the difference between the corrected mass m_{corr} as presented in equation 1 of [11] and the reconstructed mass m (see equations 7 and 8).

$$\Delta m = m_{corr} - m \tag{7}$$

$$m_{corr} = \sqrt{m^2 + |p'_{Tmissing}|^2} + |p'_{Tmissing}| \tag{8}$$

where $p'_{Tmissing}$ is the missing transverse momentum to the beam of the decay and m is the mass of the B^0 candidate as reconstructed by the DecayTreeFitter.

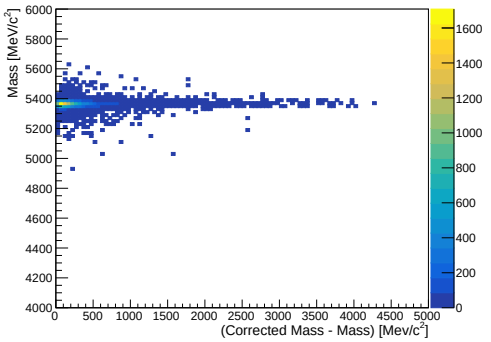
This formula is used to account for massless missing final particles in the decay. The missing transverse momentum $p'_{Tmissing}$ is obtained by projecting the momenta of the two D_s candidates on the flight direction of the B candidate, and taking the magnitude of the transverse vector.

Though being an estimate, formula 8 is described to well model the mass of the B candidate if any daughter is missing, according to [11]. Moreover, it is supposed to produce a narrow mass distribution, shifted towards a higher mean, if there were effectively missing daughters, and to slightly broaden the distribution with only a slight shift if the decay reconstruction already accounted for all daughters.

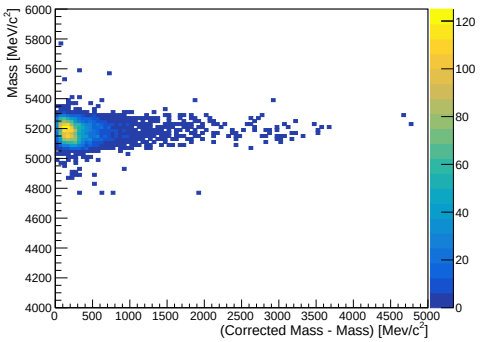
8.3.1 Correlation check

Before using the new variable Δm defined in equation 7, correlation tests were made on both the signal ($B_s^0 \to D_s^+ D_s^-$) and background ($B_s^0 \to D_s^* D_s$) samples, against the reconstructed mass. Indeed, it is essential not to have any variable correlated with the mass in the dataset, since the MC from the normalisation channel is used as training sample. The results, shown in plots 14a and 14b, do not show any correlation. Thus, the variable adds new information that is truly independent of the reconstructed mass, and is likely to account for the missing energy. The mass correction was also performed on the $B_s^0 \to D_s^* D_s^*$ sample, and the results shown in figure 14c, since this sample was also

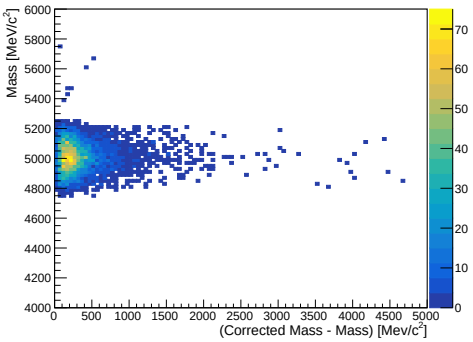
envisaged as training sample. The performance improvement is shown in the ROC curve in figure 15.



(a) Correlation between Δm and m for the $B \to D_s D_s$ decay.



(b) Correlation between Δm and m for the $B \to D_s^* D_s$ decay.



(c) Correlation between Δm and m for the $B \to D_s^* D_s^*$ decay.

Figure 14: Correlation check between the new variable Δm and the mass.

8.4 Decorrelation transformation

In addition to the mass difference Δm between m_{corr} and m, some further tuning had to be done. BDTs are known to perform worse in presence of correlated variables. A decorrelation transformation was thus inserted in front of the BDT.

The decorrelation transformation is a linear transformation performed on the input variables. If $\vec{x}$ is an event, the covariance matrix is given by:

$$C_{ij} = cov(x_i, x_j) \tag{9}$$

The decorrelation transformation such as implemented in TMVA consists in multiplying the input events $\vec{x}$ by the inverse of the square root C' of the correlation matrix C:

$$\vec{x} \to (C')^{-1}\vec{x} \tag{10}$$

where C' is the square root of the covariance matrix C.

The square root matrix C' is obtained by diagonalisation, where the square root of the diagonal matrix $\sqrt{D}$ is the diagonal matrix of square roots of the eigen-values of D.

$$\begin{aligned} D &= S^T C S \\ C' &= S\sqrt{D}S^T \end{aligned} \tag{11}$$

8.5 Improvements to the ROC curve

The training performance shows a dramatic improvement with the additional decorrelation. Figure 15 shows the ROC curve for two BDT classifiers with the same input variables (including Δm) and the same hyperparameters, but one was preceded by a decorrelation transformation on the variables (in red).

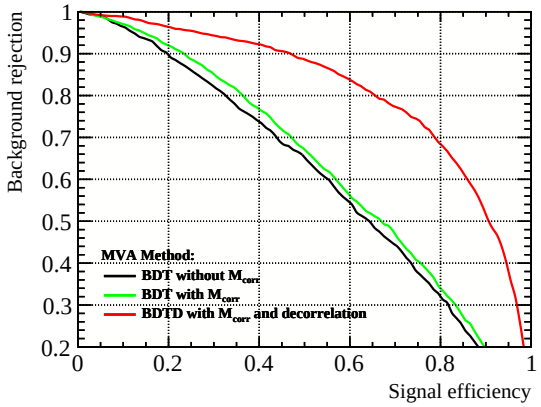


Figure 15: ROC curves for the D_s^* BDT without the difference with the corrected mass m_{corr}, with the difference as input variable, and with the difference and an additional decorrelation transformation. The decorrelation transformation brings about a huge improvement.

The BDT specific variable ranking also changed with the additional decorrelation transformation. A comparison is shown in table 8. The new ranking seems sensible: the momenta of the candidates are now the most discriminating variables.

8.6 Hyperparameter Tuning

Finally a hyperparameter tuning was also performed, looping on the *MinNodeSize* and *Depth* of the trees. Here, the Kolmogorov-Smirnov tests yield zero in all tested cases. The superposition of the classifier output distributions for both testing and training events on

Rank	Variable	Variable Importance	Rank	Variable	Variable Importance
1	lab0_DeltaM	3.26E-02	1	DS_MIN_P	4.284e-02
2	GD_MIN_PT	3.15E-02	2	lab0_P	4.104e-02
3	lab0_DTF_CHI2NDOF	3.05E-02	3	DS_MAX_P	3.833e-02
4	GD_MIN_TRACK_GhostProb	2.91E-02	4	lab0_PT	2.917e-02
5	GD_MIN_TRACK_MatchCHI2	2.88E-02	5	GD_MAX_P	2.860e-02
6	DS_MAX_ENDVERTEX_CHI2	2.82E-02	6	DS_MIN_PT	2.802e-02
7	lab0_ENDVERTEX_CHI2	2.81E-02	7	DS_MAX_DIRA_OWNPV	2.620e-02
8	GD_MIN_OWNPV_NDOF	2.79E-02	8	DS_MAX_PT	2.542e-02
9	GD_MIN_TRACK_CHI2NDOF	2.64E-02	9	DS_MAX_IPCHI2_OWNPV	2.526e-02
10	GD_MAX_TRACK_PCHI2	2.58E-02	10	DS_MIN_IP_OWNPV	2.314e-02
11	GD_MAX_PT	2.58E-02	11	GD_MAX_TRACK_PCHI2	2.266e-02
12	GD_MAX_TRACK_MatchCHI2	2.53E-02	12	GD_MIN_OWNPV_CHI2	2.247e-02
13	GD_MIN_TRACK_PCHI2	2.49E-02	13	lab0_DeltaM	2.212e-02
14	DS_MAX_PT	2.48E-02	14	GD_MIN_P	2.175e-02
15	DS_MIN_ENDVERTEX_CHI2	2.47E-02	15	GD_MIN_TRACK_CHI2NDOF	2.167e-02
16	lab0_OWNPV_NDOF	2.46E-02	16	DS_MAX_ORIVX_CHI2	2.158e-02
17	GD_MAX_OWNPV_NDOF	2.45E-02	17	DS_MAX_ENDVERTEX_CHI2	2.141e-02
18	GD_MAX_TRACK_GhostProb	2.43E-02	18	GD_MIN_IP_OWNPV	2.107e-02
19	GD_MAX_TRACK_CHI2NDOF	2.42E-02	19	GD_MAX_TRACK_GhostProb	2.083e-02
20	GD_MIN_IP_OWNPV	2.41E-02	20	DS_MIN_ENDVERTEX_CHI2	2.075e-02
21	DS_MIN_P	2.34E-02	21	DS_MIN_IPCHI2_OWNPV	2.061e-02
22	GD_MAX_P	2.34E-02	22	GD_MIN_PT	2.044e-02
23	DS_MIN_PT	2.31E-02	23	GD_MIN_TRACK_MatchCHI2	2.019e-02
24	GD_MIN_P	2.30E-02	24	GD_MAX_PT	1.942e-02
25	DS_MAX_DIRA_OWNPV	2.29E-02	25	GD_MAX_TRACK_CHI2NDOF	1.917e-02
26	lab0_PT	2.24E-02	26	GD_MAX_TRACK_MatchCHI2	1.894e-02
27	lab0_P	2.24E-02	27	GD_MIN_TRACK_GhostProb	1.876e-02
28	DS_MIN_DIRA_OWNPV	2.17E-02	28	lab0_DIRA_OWNPV	1.845e-02
29	DS_MAX_P	2.16E-02	29	GD_MIN_IPCHI2_OWNPV	1.816e-02
30	DS_MAX_OWNPV_NDOF	2.15E-02	30	GD_MAX_ORIVX_CHI2	1.810e-02
31	DS_MIN_OWNPV_NDOF	2.11E-02	31	GD_MIN_ORIVX_CHI2	1.729e-02
32	DS_MAX_IP_OWNPV	1.77E-02	32	DS_MIN_OWNPV_CHI2	1.714e-02
33	GD_MAX_IPCHI2_OWNPV	1.67E-02	33	GD_MIN_TRACK_PCHI2	1.689e-02
34	GD_MIN_OWNPV_CHI2	1.65E-02	34	GD_MAX_IPCHI2_OWNPV	1.617e-02
35	GD_MAX_OWNPV_CHI2	1.62E-02	35	GD_MAX_OWNPV_CHI2	1.610e-02
36	DS_MIN_IP_OWNPV	1.61E-02	36	lab0_DTF_CHI2NDOF	1.597e-02
37	GD_MAX_IP_OWNPV	1.59E-02	37	DS_MIN_ORIVX_CHI2	1.510e-02
38	DS_MAX_OWNPV_CHI2	1.59E-02	38	DS_MAX_IP_OWNPV	1.502e-02
39	lab0_OWNPV_CHI2	1.44E-02	39	GD_MAX_IP_OWNPV	1.478e-02
40	GD_MIN_IPCHI2_OWNPV	1.44E-02	40	GD_MIN_OWNPV_NDOF	1.468e-02
41	DS_MIN_IPCHI2_OWNPV	1.43E-02	41	GD_MAX_OWNPV_NDOF	1.429e-02
42	DS_MIN_OWNPV_CHI2	1.42E-02	42	lab0_ENDVERTEX_CHI2	1.420e-02
43	DS_MAX_IPCHI2_OWNPV	1.32E-02	43	DS_MAX_OWNPV_CHI2	1.397e-02
44	lab0_DIRA_OWNPV	1.19E-02	44	DS_MIN_DIRA_ORIVX	1.348e-02
45	DS_MIN_DIRA_ORIVX	1.11E-02	45	lab0_OWNPV_CHI2	1.311e-02
46	DS_MAX_DIRA_ORIVX	8.54E-03	46	DS_MAX_DIRA_ORIVX	1.298e-02
47	DS_MAX_ORIVX_CHI2	0.00E+00	47	DS_MIN_OWNPV_NDOF	1.210e-02
48	DS_MIN_ORIVX_CHI2	0.00E+00	48	DS_MIN_DIRA_OWNPV	1.149e-02
49	GD_MAX_ORIVX_CHI2	0.00E+00	49	DS_MAX_OWNPV_NDOF	9.740e-03
50	GD_MIN_ORIVX_CHI2	0.00E+00	50	lab0_OWNPV_NDOF	8.933e-03

Table 8: Comparison of the BDT specific variable ranking before and after the decorrelation pretransofrmation.

signal and background are shown in figures 17a, 17b, 17c and 17d, where the four most extreme $MinNodeSize$ and $Depth$ parameter combinations were chosen. The distributions in 17b show an utterly overtrained classifier. Thus the combination of small minimal node size together with deep trees is definitely to ban here. Provided the small performance difference the Depth brings about, compared to the huge difference in classifier responses on training and testing set, a tree with a $MinNodeSize$ of 0.5% and a $Depth$ of 2 was chosen. Given the small difference between the two signal and background categories, a small minimal node size makes sense. Deep trees however are not a sensitive choice, especially since boosting is most effective on weak classifiers. This is the main reason behind the choice of limiting the tree depth to 2. The boosting is thus not weakened by too deep trees.

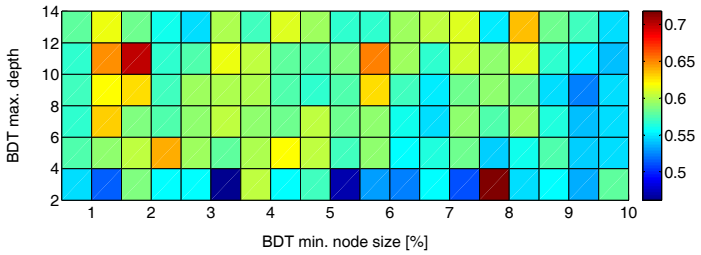


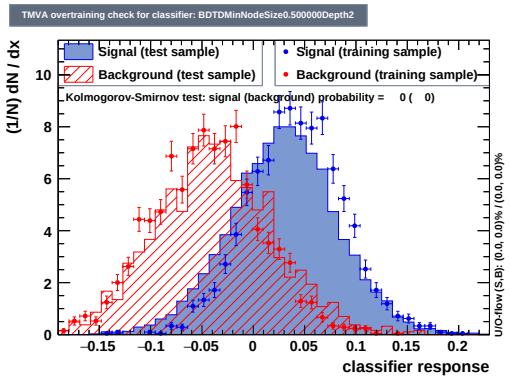
Figure 16: ROC integral values on the test set as function of the $MinNodeSize$ and $Depth$ of the decision trees.

To back up the choice, a study over a larger range of $MinNodeSize$ values was performed, making them vary between 0.1% and 50% (almost the largest possible range). The $Depth$ was fixed to 2. The figure 18 shows large statistical fluctuation in ROC integral values, but an overall decreasing trend. This means the small $MinNodeSize$ value of 0.5% is justified. The statistical fluctuations are probably due to the small number of training/testing events (about 2000/1700) that are available in the $D_s^* D_s$ MC sample.

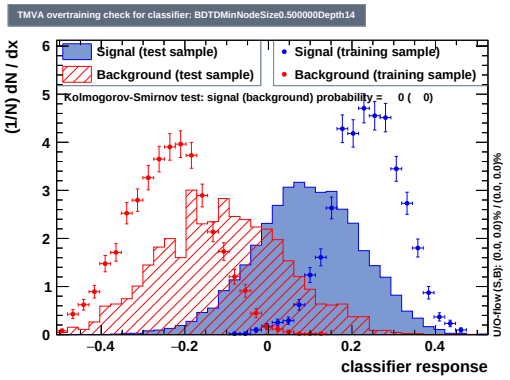
8.7 Choice of the optimal MVA cuts

8.7.1 Figure of Merit

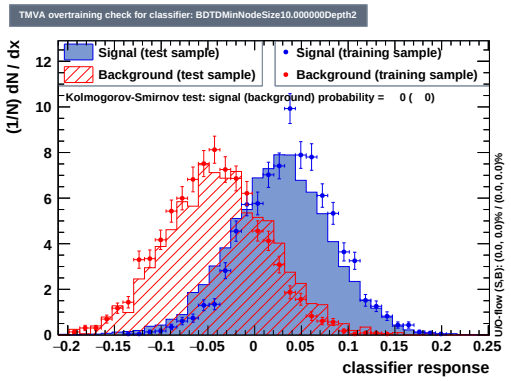
The choice of the optimal cut is based on a figure of merit (F.o.M.) which is maximised. The figure of merit can be seen as a measure of the discrimination capability of a cut between signal and background. In this analysis, two figures of merit are used.



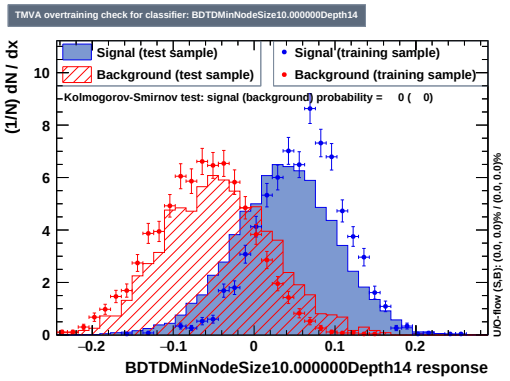
(a) Classifier output distribution of the training set and test set superimposed for signal and background, for a BDTD classifier with $MinNodeSize$ of 0.5% and a $MaxDepth$ of 2.



(b) Classifier output distribution of the training set and test set superimposed for signal and background, for a BDTD classifier with $MinNodeSize$ of 0.5% and a $MaxDepth$ of 14.



(c) Classifier output distribution of the training set and test set superimposed for signal and background, for a BDTD classifier with $MinNodeSize$ of 10% and a $MaxDepth$ of 2.



(d) Classifier output distribution of the training set and test set superimposed for signal and background, for a BDTD classifier with $MinNodeSize$ of 10% and a $MaxDepth$ of 14.

The first one, shown in figure 20a, is the significance defined as:

$$FoM_{significance} = \frac{S}{\sqrt{S+B}} \tag{12}$$

where S is the signal yield in the given mass window, and B is the background yield in the same mass range. It is the most used F.o.M. for optimising cuts. It has the advantage to give the expected significance of the observation, if the hypotheses used in the signal (and background) estimation are correct.

For rare decays, another Figure of Merit (equation 13), also called Punzi F.o.M. can be

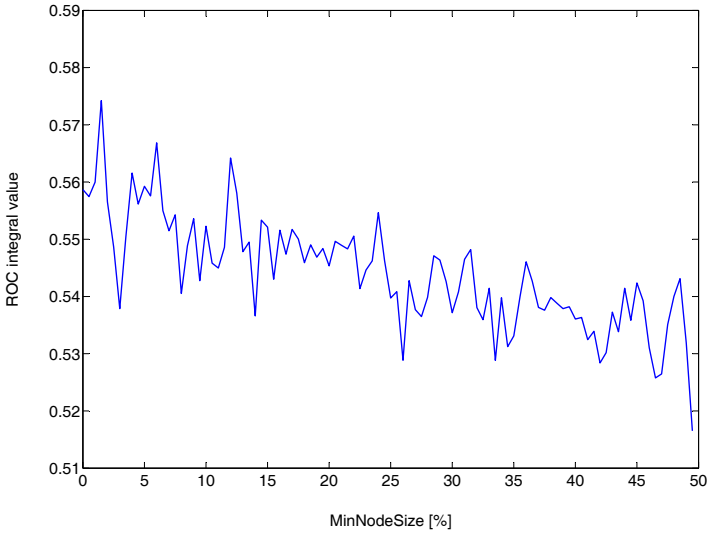


Figure 18: ROC curve integral values as function of the $MinNodeSize$ value of the BDTD MVA. Although subject to large statistical fluctuations, a decreasing trend can be observed for increasing node size.

used, which is derived in [12].

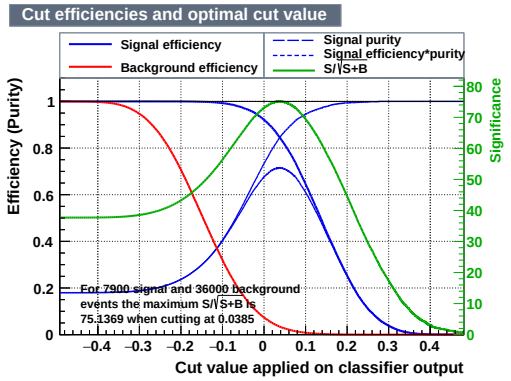
$$FoM_{Punzi} = \frac{\epsilon(S)}{\frac{\sigma}{2} + \sqrt{B}} \tag{13}$$

where $\epsilon(s)$ is the signal efficiency and σ the significance of the discovery aimed (chosen to be 3 here).

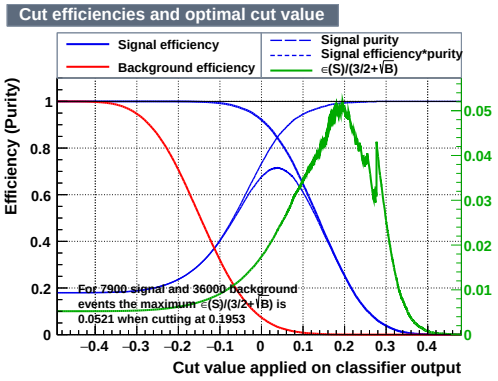
The latter figure of merit was not currently available in TMVA, so the code was modified accordingly and $ROOT$ recompiled in order to be able to use this figure as well. In figures 19a and 19b, both figures of merit are shown for the WS BDT. The number of background events is calculated for the B_s^0 peak, since the combinatorial background (which the WS BDT was designed to discriminate) is the main source of background in the B_s^0 mass region. The fluctuations in the Punzi F.o.M are due to the small values obtained and the discreteness of the cuts. For the final cut, the precision of the cut discretization was set to avoid this effect.

8.7.2 Figure of merit and final cuts

To choose the optimal MVA cuts, a Figure of Merit (F.o.M.) is computed in two dimensions. Since a cut is applied on both MVA responses, the optimisation needs to be made in two dimensions. On the x and y axes are the responses of the combinatorial, respectively D_s^* BDTs, and the z axis shows the value of the F.o.M. (on a colour scale). The two F.o.M. (significance and Punzi) described in 8.7.1 were chosen, and shown in figures 20a and 20b for the classifiers trained with the parameters described in sections 8.1 and 8.2.

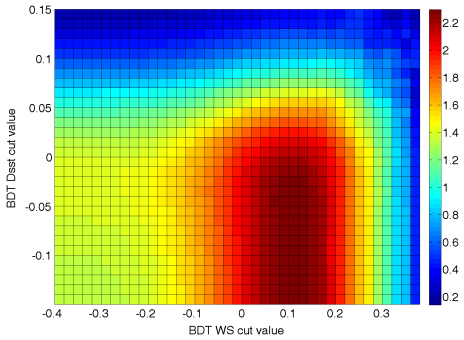


(a) Significance F.o.M. traditionally used in TMVA.

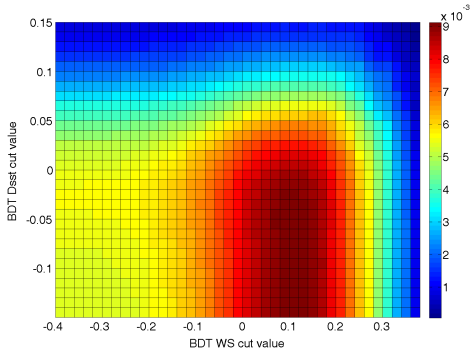


(b) Punzi figure of merit obtained after modifying and recompiling *ROOT*.

Figure 19: Figures of merit used to optimise the cuts. These F.o.M were obtained for the B_s peak with combinatorial background.



(a) Significance F.o.M. as function of the WS BDT and D_s^* BDT classifier responses.



(b) Punzi F.o.M. as function of the WS BDT and D_s^* BDT classifier responses.

Figure 20: Final figure of merit in 2D used to optimise the cuts. The WS BDT is on the x axis and the D_s^* is on the y axis.

Both F.o.M. suggest to cut on the WS BDT at a value of 0.08 and on the D_s^* at a value of -0.05.

9 D_s candidate invariant mass cuts

After both WS and D_s^* MVAs are applied, a significant amount of background persists in the D candidates invariant mass (since the mass window cuts were loosened in the stripping to about 100 MeV from the D_s mass in the PDG). Thus, a tighter cut is applied, after the peaks were fit with a double crystal ball added to a constant polynomial. The fit results are shown in figure 21 and in table 9.

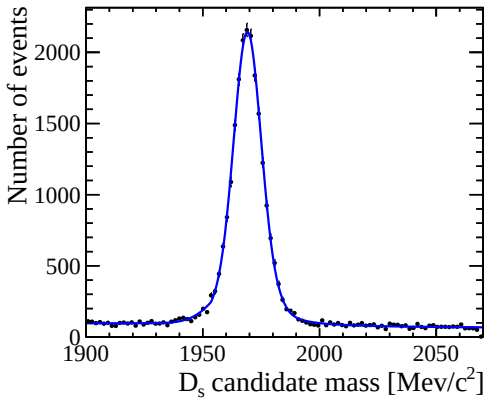


Figure 21: A double crystal-ball is fit to the sum of both D_s candidate invariant mass distributions. The background is modelled by a constant polynomial. The results are shown in table 9 and finally, a window of $\pm 20\,\text{MeV}$ was applied around the mass value from the PDG.

Parameter	Value [MeV]
μ	$1.97 \pm 0.55 \cdot 10^3$
σ_l	5.65 ± 0.13
σ_r	1.13 ± 0.92

Table 9: Results of the double crystal-ball fit to both D_s candidates mass distribution.

Given the fit results, a mass window of $\pm 20\,\text{MeV}$ was kept around the mass value from the PDG. Since the mean obtained from the fit is very close to the PDG value of $1968.3\,\text{MeV}$, and all other cuts in this analysis were made around the values from the PDG, the choice is justifiable.

10 Estimated yields

In this section, a calculation of the expected signal and background yields after all selection is performed for the Run 1 data. The calculation is based on the efficiencies in section 10.1, and the branching fractions from the 2014 edition PDG. It will provide an estimate of the amount of signal and background in the invariant mass distribution after the full selection. In addition, the signal yield after the PID selection is needed in order to decide the MVA cuts (see section 8.7). The signal yield as well as the different background yields after the whole selection can also be useful to fix the yields of the different components in the fit (described in section 11). The obtained predicted yields are summarized in table 16.

The expected number of events after the whole selection is given by equation 14:

$$N = L \cdot \sigma_{b\bar{b}} \cdot f(\bar{b} \to B^0_{(s)}) \cdot 2 \cdot \mathcal{B}(B^0_{(s)} \to final\ products) \cdot \epsilon \tag{14}$$

where $B^0_{(s)}$ stands for B^0 or B^0_s, depending on which B-meson produced the final particles. L is the integrated luminosity, $\sigma_{b\bar{b}}$ the b-quark production cross-section, $f(\bar{b} \to B^0_{(s)})$ the $\bar{b}$ to B^0 or B^0_s hadronisation fraction, $\mathcal{B}(B^0_{(s)} \to final\ products)$ the branching fraction for the B^0 or B^0_s decay to the given $final\ products$ and ϵ is the overall efficiency.

10.0.1 Luminosity

The integrated luminosity values for the years 2011, 2012, 2015 and 2016 were acquired using the $IntegratedLuminosity$ and $IntegratedLuminosityErr$ variables from the $LumiTuple$ in the data files. The values obtained are summarized in table 10[2]:

Year	Integrated luminosity [pb^{-1}]	Uncertainty [pb^{-1}]
2011	978	$\pm$11
2012	1990	$\pm$16
2015	281	$\pm$7
2016	1641	$\pm$45

Table 10: Integrated luminosity per year.

For 2016, the luminosity uncertainty calibration was not presently available, thus the worst relative uncertainty among the magnet polarities of the previous years was used and extrapolated to the 2016 luminosity. (For instance, it was the 2015 Mag Down, which value is $6.26/160.49 = 0.039$. The 2016 uncertainties are computed as $\epsilon_{2016} = L_{2016} \cdot 0.039$ for both magnet polarities.)

10.0.2 b-production cross section

The $b\bar{b}$ production cross sections for Run 1 at centre of mass energy of $\sqrt{s} = 7\,\text{TeV}$ (2011) and Run 2 are taken from [13], and for Run 1 at $\sqrt{s} = 8\,\text{TeV}$ (2012) from [14]. The values in 4π are summarized in table 11:

Year	$\sigma_{b\bar{b}}$ [μb]
2011	295
2012	298
2015	600
2016	600

Table 11: b-quark production cross section per year in 4π.

[2] A more detailed summary with the separate magnet polarities is given in appendix 15.1

10.1 Signal and background efficiencies

Signal efficiencies

The signal efficiencies are computed, based on the available normalisation channel MC samples. The total efficiency is the product of the individual efficiencies of the different selection processes:

$$\epsilon_{total} = \epsilon_{Gen} \cdot \epsilon_{Stripping} \cdot \epsilon_{Offline} \cdot \epsilon_{MVAs} \cdot \epsilon_{Ds\ MW} \tag{15}$$

where ϵ_{Gen} is the generator level acceptance, $\epsilon_{Stripping}$ the stripping efficiency, $\epsilon_{Offline}$ the efficiency of the offline selection and ϵ_{MVAs} the efficiency of the MVA methods. Since the D_s mass windows were applied after all other selections, an additional $\epsilon_{Ds\ MW}$ comes into account.

The efficiencies obtained for each decay channel are listed in table 12 together with the MC sample they were obtained with. A detailed table with all separate efficiencies is shown in appendix 15.2.

channel	simulation condition	ϵ_{total}
B2DsDs1	2011 Pythia8 Sim08a	$(8.22 \pm 0.10) \cdot 10^{-4}$
	2011 Pythia6 Sim08a	$(6.10 \pm 0.08) \cdot 10^{-4}$
	2012 Pythia8 Sim08a	$(7.29 \pm 0.11) \cdot 10^{-4}$
	2012 Pythia6 Sim08a	$(5.26 \pm 0.09) \cdot 10^{-4}$
B2DsDs2	2012 Pythia8 Sim08a	$(2.62 \pm 0.06) \cdot 10^{-4}$
	2012 Pythia6 Sim08a	$(2.13 \pm 0.05) \cdot 10^{-4}$
B2DsDs3	2012 Pythia8 Sim08a	$(3.06 \pm 0.07) \cdot 10^{-4}$
	2012 Pythia6 Sim08a	$(2.35 \pm 0.06) \cdot 10^{-4}$
B2DsDs4	2012 Pythia8 Sim08a	$(4.09 \pm 0.07) \cdot 10^{-4}$
	2012 Pythia6 Sim08a	$(3.19 \pm 0.06) \cdot 10^{-4}$

Table 12: Signal efficiencies calculated from the normalisation channel MC samples. A more detailed list with the separate efficiencies can be found in appendix 15.2.

Background efficiencies

For the different backgrounds, the efficiencies were determined in the same way, taking the available MC samples, and multiplying the efficiencies obtained at each step. A summary is provided in table 13, and a detailed table can be found in appendix 15.3.

decay	simulation condition	ϵ_{total}
$B^0 \to DDs$	2011 Pythia8 Sim08h	$(3.14 \pm 0.36) \cdot 10^{-6}$
	2012 Pythia8 Sim08h	$(3.17 \pm 0.27) \cdot 10^{-6}$
	2012 Pythia8 Sim08a	$(2.55 \pm 0.41) \cdot 10^{-6}$
	2012 Pythia6 Sim08a	$(2.42 \pm 0.39) \cdot 10^{-6}$
$B_s^0 \to D_s^* D_s$	2012 Pythia8 Sim08a	$(3.34 \pm 0.09) \cdot 10^{-4}$
$B_s^0 \to D_s^* D_s^*$	2012 Pythia8 Sim08a	$(9.99 \pm 0.41) \cdot 10^{-5}$

Table 13: Background efficiencies calculated from the MC samples. All efficiencies are taken from the B2DsDs1 channel, since the other channels do not contain a sufficient amount of events after the offline selection. A more detailed list with the separate efficiencies can be found in appendix 15.3.

10.2 Number of expected signal candidates

For the calculation of the expected signal yield (B^0 candidates in the invariant mass distribution), the branching fraction ($B^0 \to final\ products$) is following:

$$
\begin{aligned}
&\mathcal{B}(B^0 \to D_s^+ D_s^-) \cdot \Big(\\
&\quad \mathcal{B}(D_s^+ \to K^+K^-\pi^+) \cdot \mathcal{B}(D_s^- \to K^-K^+\pi^-) + \\
&\quad \big(\mathcal{B}(D_s^+ \to K^+K^-\pi^+) \cdot \mathcal{B}(D_s^- \to \pi^-\pi^+\pi^-) + \mathcal{B}(D_s^- \to K^-K^+\pi^-) \cdot \mathcal{B}(D_s^+ \to \pi^+\pi^-\pi^+)\big) + \\
&\quad \big(\mathcal{B}(D_s^+ \to K^+K^-\pi^+) \cdot \mathcal{B}(D_s^- \to K^-\pi^+\pi^-) + \mathcal{B}(D_s^- \to K^-K^+\pi^-) \cdot \mathcal{B}(D_s^+ \to K^+\pi^-\pi^+) + \\
&\quad \mathcal{B}(D_s^+ \to \pi^+\pi^-\pi^+) \cdot \mathcal{B}(D_s^- \to \pi^-\pi^+\pi^-) \Big)
\end{aligned}
\tag{16}
$$

The obtained yields in each channel for Run 1 are shown in table 14. The selection efficiencies are taken from section 10.1, and the most recent sample was taken (with Pythia8). For 2011, the B2DsDs1 efficiency is obtained from the 2011 MC sample since it was available for this channel, and for the other channels, the 2012 efficiencies were taken. For Run 2, the same calculation was performed, with the efficiencies obtained with the 2012 MC samples used in the formula. Thus, the values are purely indicative and cannot be used for setting a limit or determining a branching fraction. For the normalisation mode, the same calculation was performed, and shown in table 15.

Channel	Run 1 expected yield	Run 2 expected yield	Run 1 + 2 expected yield
B2DsDs1	57.86 ± 5.20	72.69 ± 5.90	130.55 ± 7.86
B2DsDs2	7.97 ± 0.84	10.43 ± 1.10	18.40 ± 1.38
B2DsDs3	5.64 ± 0.64	7.38 ± 0.89	13.02 ± 1.09
B2DsDs4	1.25 ± 0.12	1.63 ± 0.15	2.88 ± 0.20
Total	72.72 ± 5.56	92.13 ± 6.06	164.85 ± 8.06

Table 14: Expected signal yield per channel for Run 1, Run 2 and the total of both.

Channel	Run 1 expected yield	Run 2 expected yield	Run 1 + 2 expected yield
B2DsDs1	1803.05 ± 276.94	2265.04 ± 424.97	4068.10 ± 507.24
B2DsDs2	248.37 ± 40.88	325.06 ± 64.79	573.43 ± 76.61
B2DsDs3	175.70 ± 29.83	229.95 ± 47.58	405.65 ± 56.16
B2DsDs4	77.73 ± 12.50	101.73 ± 19.71	179.46 ± 23.34
Total	2304.85 ± 281.80	2921.78 ± 432.96	5226.63 ± 516.59

Table 15: Expected normalisation mode yield per channel for Run 1, Run 2 and the total of both.

10.3 Background yields

A calculation of the expected number of $B_s^0 \to D^- D_s^+$ is also performed, since this background is peaking under the B_s^0 peak, with a tail going until the B^0 mass region. The 2012 stripping and MC are used to compute the efficiency, and a crosscheck is made with the 2011 MC and stripping. Since there is no 2015 or 2016 MC available, the efficiency of 2012 is used for all years. The expected yields of $B_s^0 \to D_s D_s^*$ and $B_s^0 \to D_s^{*+} D_s^{*-}$ are also determined, to obtain a relative ratio between them. Finally, the expected number of $B_s^0 \to D_s^- K^+ K^- \pi^+$ is computed, mainly to show that it is negligible compared to the B_s^0 peak.

The generator level acceptance, as well as the stripping, offline selection, MVA and D_s mass window cuts efficiencies are listed in the tables of appendix 15.2 and 15.3. The generator level acceptances are taken from [15] and [16]. The expected yields are given in table 16.

Decay	Run 1 expected yield	Run 2 expected yield	Run 1 + 2 expected yield
$B^0 \to D^- D_s^+$	80.73 ± 11.63	106.01 ± 23.94	186.73 ± 26.61
$B_s^0 \to D_s^- K^+ K^- \pi^+$	1.94 ± 1.65	2.54 ± 2.90	4.49 ± 3.33
$B_s^0 \to D_s D_s^*$	2173.22 ± 349.80	2844.23 ± 704.58	5017.46 ± 768.63
$B_s^0 \to D_s^{*+} D_s^{*-}$	634.03 ± 105.30	829.79 ± 210.51	1463.82 ± 235.37

Table 16: Expected yields for each background component. The yields are determined upon the MC efficiencies obtained in 10.1, and are only available in the B2DsDs1 channel. The efficiencies of the other channels could not be determined due to an insufficient or absent number of candidates after the selection, and are assumed to be 0. Note: for the $B_s^0 \to D_s^- K^+ K^- \pi^+$ decay, no MC generator level efficiency was available, so the estimate is based on a generator level efficiency of 0.1 ± 0.1 which explains the huge uncertainty.

10.4 Calculation of the $B^0 \to D^- D_s^+$ fraction

To constrain the yield of the $B^0 \to D^- D_s^+$ decay with respect to the $B_s^0 \to D_s^+ D_s^-$ one, the ratio between the two is computed. The calculation is performed as follows:

The ratio between $B^0 \to D^+ D_s^-$ and $B_s^0 \to D_s^+ D_s^-$ expected in the B2DsDs1 channel is

computed according to the respective branching fractions and hadronisation fractions:

$$r = \frac{f(\bar{b} \to B^0) \cdot \mathcal{B}(B^0 \to D^- D_s^+) \cdot \mathcal{B}(D_s^+ \to K^+K^-\pi^+) \cdot \mathcal{B}(D^- \to \pi^- K^+\pi^-) \cdot \epsilon_{DDs}}{f(\bar{b} \to B_s^0) \cdot \mathcal{B}(B_s^0 \to D_s^+ D_s^-) \cdot \mathcal{B}(D_s^+ \to K^+K^-\pi^+) \cdot \mathcal{B}(D_s^- \to K^-K^+\pi^-) \cdot \epsilon_{Bs\to DsDs}} = 0.037 \pm 0.08 \tag{17}$$

Now, to extrapolate this ratio obtained for the B2DsDs1 channel to all four channels, some considerations are made. The $B^0 \to D^- D_s^+$ pollution mainly occurs through π^+ that are misidentified as K^+ (or π^- as K^-) and enter the B2DsDs1 channel to form a $K^+K^-\pi^+$ $K^-K^+\pi^-$ pair. Both of the D_s candidates can thus be a misidentified D candidate. For the B2DsDs2 and B2DsDs3 channels which contain only one D_s decaying to $KK\pi$, the probability to contain a misidentified D candidate is halved. Indeed, a D decaying to $\pi^-K^+\pi^-$ to be identified as $\pi^-\pi^+\pi^-$ or $K^-\pi^+\pi^-$ would need at least one K misidentified as a π which would mean the reconstructed mass is lower than the one of B^0. These cases are excluded by the mass windows applied to the D_s candidates. For the B2DsDs4 channel, no D_s is likely to be a misidentified D candidate. Thus the correction to the ratio obtained reads:

$$f_{DDs/BsDsDs} = \left(\frac{N_1}{N} + \frac{1}{2} \cdot \frac{N_2}{N} + \frac{1}{2} \cdot \frac{N_3}{N}\right) \cdot r \tag{18}$$

where $\frac{N_i}{N}$ is the fraction of candidates in each channel and $N = N_1 + N_2 + N_3 + N_4$.

10.5 Calculation of the $B_s^0 \to D_s^{*+} D_s^{*-}$ fraction

The fraction of the $B_s^0 \to D_s^{*+} D_s^{*-}$ background is also fixed but to a fraction of the $B_s^0 \to D_s D_s^*$ yield. Here the calculation is a bit more straightforward, since the probability of the B_s^0 decaying to a D_s^* or to D_s is independent of the final state. The fraction is thus obtained by the ratio:

$$r = \frac{\mathcal{B}(B_s^0 \to D^{*+} D_s^{*-}) \cdot \mathcal{B}(D_s^* \to D_s\gamma) \cdot \epsilon_{Ds*Ds*}}{\mathcal{B}(B_s^0 \to D_s^* D_s) \cdot \epsilon_{Bs\to Ds*Ds}} = 0.30 \pm 0.07 \tag{19}$$

11 Fit

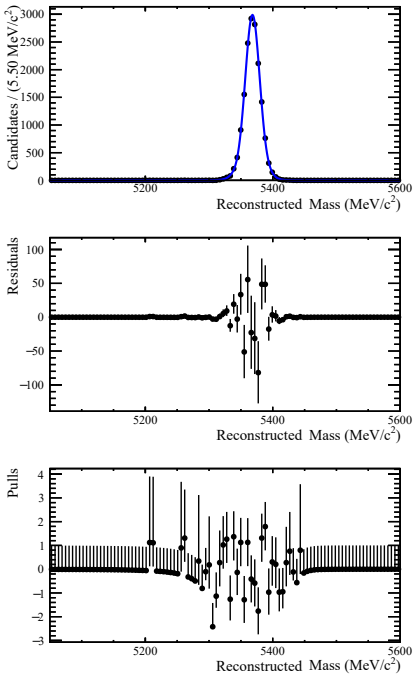
11.1 Run 1 2011 and 2012 data

The fit applied to the B candidate invariant mass is a rather complex shape, made of several components for each background. A description of the backgrounds modelled as well as the shape used is provided in this section. The individual components are fit to the MC and wrong-sign $D_s^+ D_s^+$ samples to get the shape, and once fixed are combined to build the full fit. The signal shapes were produced from the four channels, but for the background, the majority of generated MC backgrounds were produced in the B2DsDs1 channel (with the D_s^+ decaying to $K^+K^-\pi^+$). The shapes were thus taken from this mode. This is justifiable, since the B2DsDs1 channel has the highest branching fraction.

For the signal shape where MC was avilable in all four modes, the B2DsDs1 channel is used in the final fit as for the backgrounds for consistency reasons. Some minor corrections (mass shift and width constraints) are applied to the shapes during the fit to data.

11.2 Signal shapes

The signal shape for the $B^0 \to D_s^+ D_s^-$ is taken from the MC samples for the $B_s^0 \to D_s^+ D_s^-$ from 2011 and 2012 generated with Pythia 8 and Sim08a. A sum of two crystal-ball functions is fit to the B invariant mass for the four channels. The resulting shapes are shown in figure 22a and the parameters for each mode are summarized in figure 22b. The residuals, as well as the pulls are also shown.



(a) Fit shape for the signal and normalisation mode in the B2DsDs1 channel taken from the $B_s^0 \to D_s^+ D_s^-$ 2012 and 2011 MC samples produced with Pythia 8 and Sim08a.

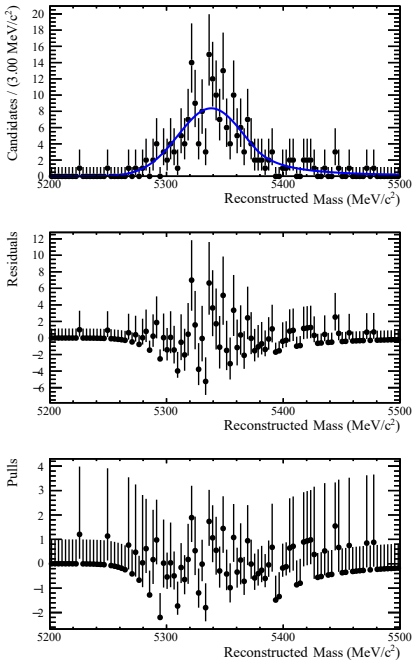
Parameter	value	uncertainty
μ	5368.2	± 0.1
σ_l	10.7	± 0.1
n_l	3.1	± 0.6
a_l	2.4	± 0.1
σ_r	16.3	± 0.3
n_r	15	± 3
a_r	-2.4	± 0.2

(b) Fit parameters for the $B_s^0 \to D_s^+ D_s^-$ decay in the B2DsDs1 channel.

Figure 22

$B^0 \to DD_s$

For the $B^0 \to DD_s$ background, a crystal-ball shape is used, and obtained from the MC samples in the B2DsDs1 channel. To enhance the statistics, the 2012 and 2011 samples produced with Pythia 8 and Sim08h were taken to obtain the shape. Figure 23a shows the fit result. The fit parameters are given in figure 23b.



(a) Fit shape of the $B^0 \to DD_s$ decay taken on the 2012 and 2011 MC samples produced with Pythia 8 and Sim08h.

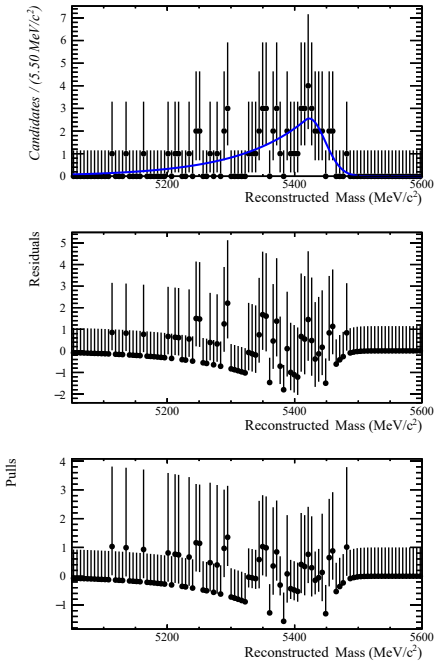
Parameter	value	uncertainty
μ	5338	± 2
σ	28	± 1
n	2	± 2
a	-1.3	± 0.4

(b) Fit parameters for the $B^0 \to DD_s$ shape.

Figure 23

$\Lambda_b \to \Lambda_c D_s$

The $\Lambda_b \to \Lambda_c D_s$ is also modelled by a crystal ball shape, and obtained from the B2DsDs1 channel. A non negligible amount of data was to be observed in the B2DsDs3 channel as well but not enough to produce a good shape. The fit was made on 2012 and 2011 MC samples generated with Pythia 8 and Sim09b. The shape is shown in figure 24a and the fit parameters are shown in figure 24b, and are fixed in the full fit.



(a) Fit shape of the $\Lambda_b \to \Lambda_c D_s$ decay taken on the 2012 and 2011 MC samples produced with Pythia 8 and Sim09b.

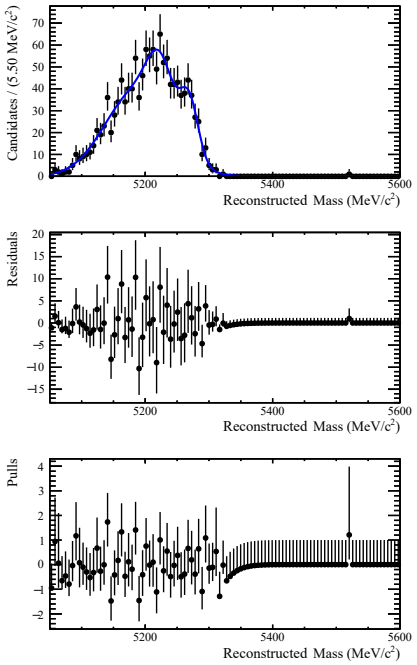
Parameter	value	uncertainty
μ	5422	± 11
σ	25	± 7
n	113	± 129
a	0.23	± 0.08

(b) Fit parameters for the $\Lambda_b \to \Lambda_c D_s$ shape.

Figure 24

$B_s^0 \to D_s D_s^*$

The $B_s^0 \to D_s D_s^*$ background shape was inspired by [17] and is made of three gaussian distributions with different means and standard deviations. Their mean, standard deviation and the ratios between each other are determined on the available MC sample from 2012 with Pythia 8 and Sim08a. The shape obtained is presented in figure 25a and the fit parameters are shown in figure 25b.



(a) Shape of the $B_s^0 \to D_s D_s^*$ background, taken from a fit to the MC sample from 2012 with Pythia 8 and Sim08a. The shape is a sum of three gaussian distributions with different mean and standard deviation.

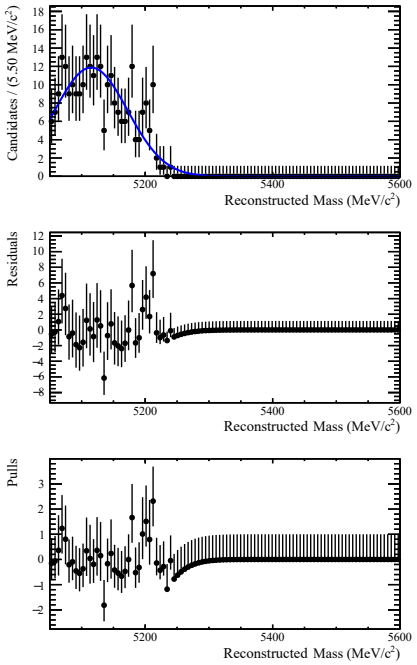
Parameter	value	uncertainty
μ_1	5186	± 4
σ_1	$50e$	± 1
r_1	0.68	± 0.06
μ_2	5223	± 6
σ_2	20	± 8
r_2	$0.19e$	± 0.09
μ_3	5269	± 4
σ_3	14	± 2

(b) Fit parameters for the $B_s^0 \to D_s D_s^*$ shape.

Figure 25

$B_s^0 \to D_s^* D_s^*$

For the $B_s^0 \to D_s^* D_s^*$ background, the shape was made with a gaussian. The shape was also obtained using the MC sample from 2012, generated with Pythia 8 and Sim08a. The results regarding the fit shape are shown in figure 26a and the parameters are given in figure 26c. Part of the events are located outside the mass range.



(a) Shape of the $B_s^0 \to D_s^* D_s^*$ background, taken from a fit to the MC sample from 2012 with Pythia 8 and Sim08a. The shape is a gaussian distribution.

Parameter	value	uncertainty
μ	5115	± 5
σ	57	± 4

(b) Fit parameters for the $B_s^0 \to D_s^* D_s^*$ shape.

(c)

Figure 26

$B_s^0 \to D_s K K \pi$

For the $B_s^0 \to D_s K K \pi$ decay, an insufficient amount (four events) of signal was left after the selection. A plot of the B candidate invariant mass distribution is shown in figure 27. We estimate 4 events should remain after the selection, so we ignore this background in the fit.

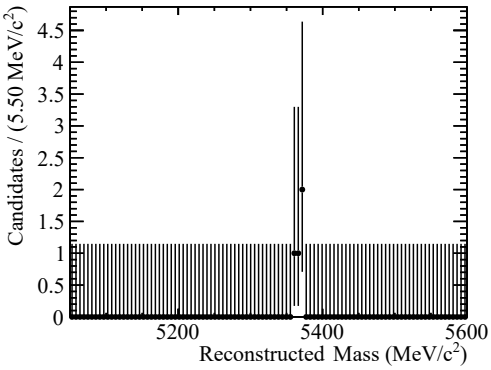


Figure 27: Reconstructed $B_s \to D_s D_s$ invariant mass distribution of true $B_s \to D_s K K \pi$ after the selection. Too few events are left to build a fit shape.

Combinatorial background

The combinatoric background is obtained by fitting the wrong-sign data with an exponential shape. The result is shown in figure 28 and the exponetial parameter α is given by $(-8.6 \pm 6.9) \cdot 10^{-4}$.

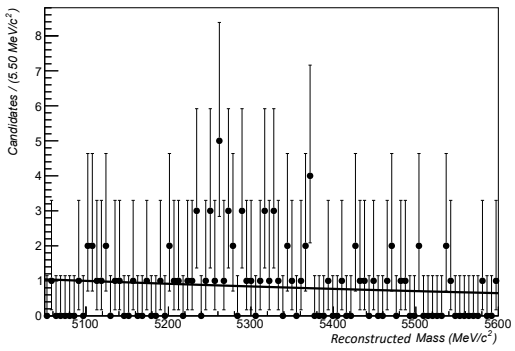


Figure 28: Combinatorial background fit obtained from an exponential fit to the wrong-sign data.

11.3 Fit to 2011 and 2012 data

The full fit to data is the combination of all shapes obtained in the fits to the different MC samples.

Once the fit is applied to data, some more corrections to the shape are made. The distinct steps are explained in the sections below.

11.3.1 Signal shape tweaking

The $B_s^0 \to D_s^+ D_s^-$ shows a slight shift in mass in data with respect to the peak obtained with MC. The distribution also looks slightly broader in the data. Thus, a correction was made to the $B_s^0 \to D_s^+ D_s^-$ and the $B^0 \to D_s^+ D_s^-$ shapes. A shift is applied to the mean value obtained in MC, and a scale factor to the width. The correction values are determined in the fit to data. The shapes for the $B^0 \to D_s^+ D_s^-$ and the $B_s^0 \to D_s^+ D_s^-$ are determined from a shared width and mean (with a shift equal to the mass differences from the PDG), in other words the $B_s^0 \to D_s^+ D_s^-$ shape is used to constrain the $B^0 \to D_s^+ D_s^-$. (This is based on the assumption that the mass values taken from the PDG are accurately determined via other measurements, and that the mismatch between the shapes in MC and data is due either to differences in calibrations or modelling in the MC.) Since the B_s^0 peak has the largest contribution, the correction will be mainly driven by the normalisation channel, and the $B^0 \to D_s^+ D_s^-$ contribution is negligible.

11.3.2 Fixing the fraction of $B^0 \to DD_s$

The $B^0 \to DD_s$ background is not very well constrained in the general fit, since it is not the most important contributing background to the general shape over the full range. Thus, it can easily be misused by the minimisation algorithm to minimise the error on the fit, setting its yield to any non realistic value. To prevent this happening, the yield of the $B^0 \to DD_s$ decay was set to a fraction of the $B_s^0 \to D_s^+ D_s^-$ yield, obtained in the calculation in equation 18. The $B_s^0 \to D_s^+ D_s^-$ shape is rather pure. By fixing the ratio between $B_s^0 \to D_s^+ D_s^-$ and $B^0 \to DD_s$, the fit can determine both yields in their expected proportions. Ideally the fraction should also be fixed for the $\Lambda_b \to \Lambda_c D_s$ decay, but this was unfortunately not possible due to the absence of a generator level efficiency value. Since the $\Lambda_b \to \Lambda_c D_s$ shape is far away from the B^0 in the upper tail of the $B_s^0 \to D_s^+ D_s^-$ distribution, and has a very low contribution to the $B^0 \to D_s^+ D_s^-$ mass region, it is less critical and the yield was left floating.

11.3.3 Fixing the fraction of $B_s^0 \to D_s^* D_s^*$

The fraction of $B_s^0 \to D_s^* D_s^*$ was also fixed, but with respect to the yield of $B_s^0 \to D_s D_s^*$. Indeed, it indirectly affects the contribution of the $B_s^0 \to D_s D_s^*$ background in the $B^0 \to D_s^+ D_s^-$ mass region, by its yield which is anticorrelated to the one of $B_s^0 \to D_s D_s^*$. Thus, fixing the contributions of these two decays to the right fraction with respect to each other obtained in equation 19, ensures the $B_s^0 \to D_s D_s^*$ contribution is accurately modelled in the B^0 mass region.

11.3.4 Constraining the $B_s \to D_s D_s^*$ shape parameters and the fraction of $B^0 \to DD_s$

The fit shapes for the backgrounds were determined using MC samples, but the low statistics remaining in the samples after the selection lead to significant uncertainties on the fit parameters. The yield of the $B^0 \to D_s^+ D_s^-$ largely depends on the shapes of the $B_s^0 \to D_s D_s^*$ and $B^0 \to DD_s$ decays. In particular, a slight change in their widths can lead to significant changes in the yield of the $B^0 \to D_s^+ D_s^-$ decay. To account for this effect, the uncertainties obtained on the fit shape parameters of the $B_s^0 \to D_s D_s^*$ shape are added in the general fit as constraints. The respective parameters of the general fit were constrained with a gaussian with mean value obtained in the fit to MC, and with standard deviation equal to the uncertainty obtained. The uncertainty on the ratio of $B^0 \to DD_s$ with respect to $B_s^0 \to D_s^+ D_s^-$ is also added as gaussian constraint to the ratio it was fixed to before. Here the μ is the expected fraction obtained from the expected yield calculation in eqution 18, and the σ is the uncertainty from the calculation.

11.4 Results

11.4.1 Fit result

The fit to the 2011 and 2012 data is presented in figure 29 and the yields (or relative fractions when applicable) are shown in table 17

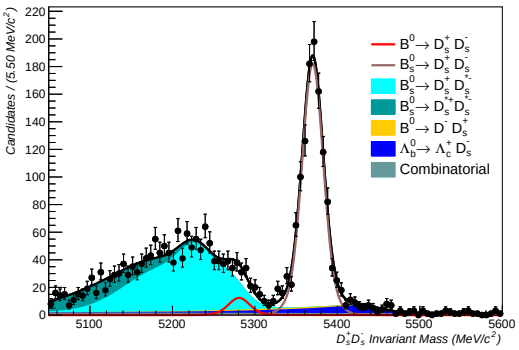


Figure 29: Fit to the 2011 and 2012 data. The $B^0 \to D^- D_s^+$ is fixed to a fraction of the $B_s^0 \to D_s^+ D_s^-$, the other shapes are fixed to the MC values except for the $B_s^0 \to D_s D_s^*$ where the values were fixed to the ones obtained under constraint.

Decay	yield/fraction	uncertainty
$B_s^0 \to D_s^+ D_s^-$	1107.83	37.00
$f_{B0 \to DsDs/Bs \to DsDs}$	$6.80 \cdot 10^{-2}$	$1.63 \cdot 10^{-2}$
$B_s^0 \to D_s D_s^*$	1052.09	33.44
$f_{Lb \to LcDs/Bs \to DsDs}$	0.13	0.03
μ_{shift}	2.05	0.46
width scale	1.11	0.03
combinatorial	158.28	30.85

Table 17: Fit parameters after the fit to 2011 and 2012 data.

An excess can be seen in the B^0 mass region. Interestingly, the yield in the fit is almost of a factor 2 smaller than the ones predicted in equation 14. The cause of this factor two has been investigated, but could not be found to this time. It is unlikely to happen during the selection, since it would also affect the efficiency obtained with the MC and factor out. It could either be a mistake in the calculation, or data missing. It was checked that both magnet polarities were present, and the grid job files were checked again for any potential errors. Such a factor was also present in other analyses, namely [18] where a factor of $1.8 - 2.3$ is reported. This factor is however not a concern for the computation of the branching fraction of the B^0, since it is obtained relative to the B_s^0, and all constraints were determined as fractions between the different backgrounds.

11.4.2 Systematic uncertainties

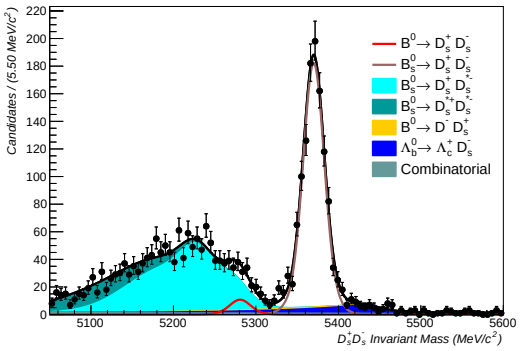


Figure 30: Fit to the 2011 and 2012 data. The $B^0 \to D^- D_s^+$ is fixed to a fraction of the $B_s^0 \to D_s^+ D_s^-$ with a constraint and the mean values and the widths of the gaussians of the $B_s^0 \to D_s D_s^*$ are also floated under constraint.

Decay	yield/fraction	uncertainty
$B_s^0 \to D_s^+ D_s^-$	1107.64	$\pm$36.98
$f_{B0 \to DsDs/Bs \to DsDs}$	$5.92 \cdot 10{-}2$	$2.74 \cdot 10^{-2}$
$f_{B0 \to DDs/Bs \to DsDs}$	$3.90 \cdot 10^{-2}$	$0.80 \cdot 10^{-2}$
$B_s^0 \to D_s D_s^*$	1061.58	40.07
$f_{Lb \to LcDs/Bs \to DsDs}$	0.13	0.03
μ_{shift}	2.05	0.46
width scale	1.11	0.04
combinatorial	156.19	31.11
$\mu 1_{DsDs*}$	5192.71	$\pm$3.01
$\mu 2_{DsDs*}$	5229.02	$\pm$3.70
$\mu 3_{DsDs*}$	5272.42	$\pm$5.12
$\sigma 1_{DsDs*}$	49.16	1.74
$\sigma 1_{DsDs*}$	19.09	3.05
$\sigma 1_{DsDs*}$	18.51	2.12

Table 18: Fit parameters after the fit to 2011 and 2012 data with constraints.

The systematic uncertainty is mainly due to the low number of MC events available to determine the background shapes. To determine the statistical uncertainty only, the fit was performed with the parameters under constraint fixed at their optimal values after the full fit. The statistical uncertainty obtained is $\pm 1.626 \cdot 10^{-2}$. The systematic uncertainty is obtained by removing the statistical uncertainty form the uncertainty of the fit with constraints: $\Delta_{syst} = \sqrt{(2.74 \cdot 10^{-2})^2 - (1.63 \cdot 10^{-2})^2} = 2.21 \cdot 10^{-2}$.

Thus, the fraction of $B^0 \to D_s^+ D_s^-$ compared to $B_s^0 \to D_s^+ D_s^-$ given by the fit result in table 18 becomes $f_{B0 \to DsDs/Bs \to DsDs} = (5.9 \pm 1.6 \pm 2.2) \cdot 10^{-2}$.

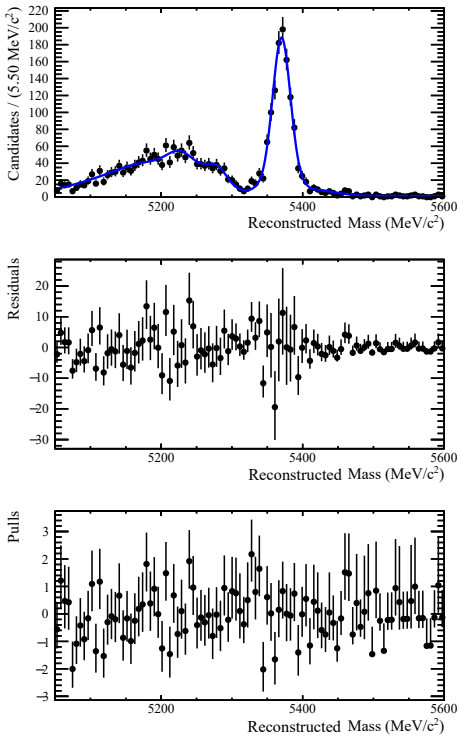


Figure 31: Residuals and pulls of the fit to the 2011 and 2012 data.

The pulls and residual distributions shown in figure 31 show no clear region of the mass range where the fit is not accurate. All pulls are contained within the range $[-2; 2]$, which is the usual fit quality requirement. To increase the precision, one should have more statistics in the MC samples to produce better shapes for the backgrounds.

11.4.3 Calculation of the $B^0 \to D_s^+ D_s^-$ branching fraction

The ratio between the $B^0 \to D_s^+ D_s^-$ and $B_s^0 \to D_s^+ D_s^-$ yield obtained in the fit allows to compute the branching fraction of the $B_s^0 \to D_s^+ D_s^-$ decay. The computation is shown in equation 20:

$$\mathcal{B}(B^0 \to D_s^+ D_s^-) = f_{B0\to DsDs/Bs\to DsDs} \cdot \frac{f(\bar{b} \to B_s^0) \cdot \mathcal{B}(B_s^0 \to D_s^+ D_s^-)}{f(\bar{b} \to B^0)} \tag{20}$$

$$\tag{21}$$

$$= [6.63 \pm 2.14(stat.) \pm 2.61(syst.) \pm 0.76(norm.)] \cdot 10^{-5} \tag{22}$$

where *(stat.)* is the statistical uncertainty, *(syst.)* the systematic one and *(norm.)* the uncertainty due to the one on the $\mathcal{B}(B_s^0 \to D_s^+ D_s^-)$ branching fraction. This result is within two sigma away from the theoretical prediction in [5] which is $(1.12 \pm 0.15) \cdot 10^{-5}$, and in accordance with the current limit in the PDG, which is set at $3.6 \cdot 10^{-5}$ at 90% CL.

11.5 Fit to the Run 1 and Run 2 2011, 2012, 2015 and 2016 data

The fit was also applied to the 2011, 2012, 2015 and 2016 data. Since no MC was available for Run 2, the shapes determined using Run 1 MC in section 11.2 were used. This alone already is very bold, since there are parameters changing between Run 1 and Run 2, starting from the centre of mass energy available in the collision. The trigger also changed, and the stripping changed accordingly. Nevertheless, the fit is shown in figure 32. The first fit results show a small shift in mass of the $B^0 \to D_s^+ D_s^-$ and $B_s^0 \to D_s^+ D_s^-$ decays, in the opposite direction as the 2011 and 2012 data. The shape of the B^0 does not make sense here, since it strongly depends on the shapes of the backgrounds. A slight change in their shape might have a huge impact on the B^0 yield. Indeed, the background shapes are taken on the Run 1 MC, they are not expected to accurately model the shapes of the run 1 and run 2 data together.

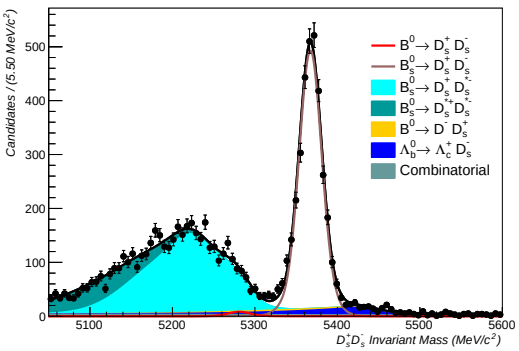


Figure 32: Fit to the 2011, 2012, 2015 and 2016 data. The $B^0 \to D^- D_s^+$ is fixed to a fraction of the $B_s^0 \to D_s^+ D_s^-$ with a constraint and the mean values and the widths of the gaussians of the $B_s^0 \to D_s D_s^*$ are also floated under constraint (same procedure as for the 2011 and 2012 data alone).

Decay/fraction	yield/fraction	uncertainty
$B_s^0 \to D_s^+ D_s^-$	3201.30	62.61
$f_{B0 \to DsDs/Bs \to DsDs}$	$1.68 \cdot 10{-}2$	$2.13 \cdot 10^{-2}$
$f_{B0 \to DDs/Bs \to DsDs}$	$4.23 \cdot 10^{-2}$	$0.79 \cdot 10^{-2}$
$B_s^0 \to D_s D_s^*$	3269.33	80.36
$f_{Lb \to LcDs/Bs \to DsDs}$	0.11	0.02
μ_{shift}	-0.68	0.29
width scale	1.19	0.02
combinatorial	428.06	53.87
$\mu 1_{DsDs*}$	5197.14	2.55
$\mu 2_{DsDs*}$	5227.62	3.41
$\mu 3_{DsDs*}$	5273.65	3.41
$\sigma 1_{DsDs*}$	49.10	1.63
$\sigma 2_{DsDs*}$	23.44	2.74
$\sigma 3_{DsDs*}$	20.13	2.01

Table 19: Fit parameters after the fit to 2011, 2012, 2015 and 2016 data.

The relative fraction between $B^0 \to D_s^+ D_s^-$ and $B_s^0 \to D_s^+ D_s^-$ was dramatically decreased: $f_{B0 \to DsDs/Bs \to DsDs} = (1.679 \pm 2.130) \cdot 10^{-2}$ with an uncertainty larger than the value itself, as shown in table 19.

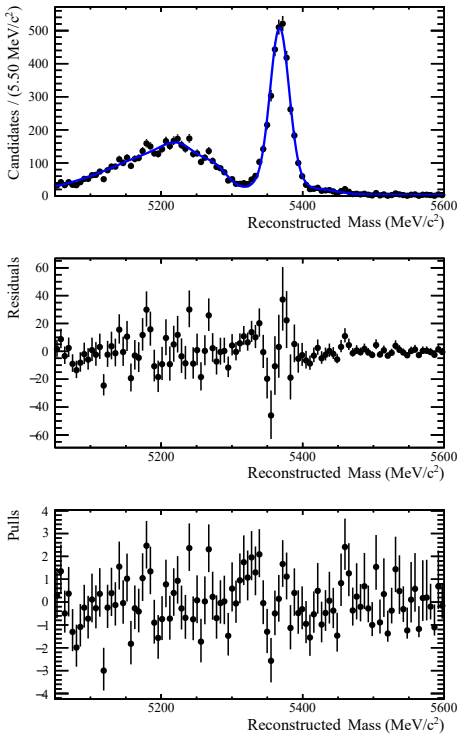


Figure 33: Residuals and pulls of the fit to the 2011, 2012, 2015 and 2016 data.

The mean shift of the B^0 and B_s^0 shapes in the fit to the full 2011, 2012, 2015 and

2016 dataset has a value of $-0.68 \pm 0.29\,\text{MeV}$ against $2.05 \pm 0.46\,\text{MeV}$ in the 2011 and 2012 fit. The width scale factor applied to the same shapes now reads 1.19 ± 0.024 against 1.11 ± 0.04 in the Run 1 fit. The residual and pull distributions in figure 33 also show evidence of mismodelling.

These inconsistencies are all indications that there are differences between the Run 2 and Run 1 data, and that the background cannot be modelled accurately without any Run 2 MC.

One can however estimate the increase in yield the run 2 would bring about, by comparing the yields of the normalisation channel. The B_s^0 yield for Run 1 and Run 2 data is about 3201.30 ± 62.61 against 1107.64 ± 36.98 for Run 1 only. Thus, the yield of Run 2 is expected to be twice the one of Run 1 according to L_{int} and $\sigma_{\bar{b}b}$, which would lead to an overall increase of the statistics by about a factor 3. An observation might become possible, provided we have the MC to obtain accurate efficiencies, and to properly study the background shapes and model them correctly in the fit.

12 Discussion

This analysis allowed to compute a branching fraction for the $B^0 \to D_s^+ D_s^-$ decay, using the 2011 and 2012 data and the $B_s^0 \to D_s^+ D_s^-$ decay as normalisation channel. The result is compatible with zero, and thus an upper limit can be set. A toy study also needs to be carried out, in order to confirm the validity of the fit shape. The computation was not done at the time of submitting the report, but it is envisaged to be done as next step. With the Run 1 alone, no observation was expected, the result is thus not a surprise. Adding the Run 2 data may have improved the statistics by almost a factor 2.9 making an observation possible, assuming systematic uncertainties can also be reduced. Nevertheless, without any MC samples to determine the efficiencies, no quantitative result can be obtained from the run 2 data. The 2015 and 2016 data, even though downloaded and processed until the step of the fit, could not be used in the computation, since no MC samples were available for the Run 2 for signal or any of the backgrounds.

In addition, the shapes of the backgrounds for the Run 2 data may be different, due to changes in the dataflow. The trigger changed between Run 1 and Run 2, and the stripping is also slightly different. Thus MC samples for Run 2 are essential to go further in this analysis. They would enable to obtain the efficiencies for the different physical backgrounds. Their shapes could also be accurately determined for the fit to the run 2 data, which is essential to reduce systematic uncertainties. The $B^0 \to D_s^+ D_s^-$ yield strongly depends on the background shapes, in particular the $B_s^0 \to D_s D_s^*$ and the $B^0 \to D D_s$. The Run 1 result is systematically limited by the MC sample sizes, and could be much more precise with more MC. $B^0 \to D_s^+ D_s^-$ signal MC is also needed to determine any efficiency differences between the $B^0 \to D_s^+ D_s^-$ and $B_s^0 \to D_s^+ D_s^-$ control channel. The analysis assumes these are similar as no $B^0 \to D_s^+ D_s^-$ MC was available, but the decay

time cut will lead to a small efficiency difference as the lifetimes of the B^0 and B_s^0 are different.

Run 2 MC samples are also required in order to train the MVA methods on Run 2 data. Here, the training performed using Run 1 MC and Run 1 wrong-sign data was used for Run 2 data. For the combinatoric background MVA method (WS BDT in section 8.1), a training against a mix of Run 1 and Run 2 wrong-sign data could have been envisaged, but the choice was made not to risk a training that would learn to discriminate based on differences between Run 1 and Run 2 data. For the D_s^* BDT (section 8.2), the MC samples were only available for 2012. The above limitations clearly bias towards an optimal selection for Run 1 data.

Even without the sensitivity improvement from the Run 2 statistics, improvements to the Run 1 fit can also be envisaged. More statistics in the MC samples is required to more accurately model the background shapes. Since the selection is very effective, the low remaining background yields in the samples lead to significant uncertainties on the fit parameters obtained on the MC. In addition, the figure of merit (section 8.7) was chosen to maximise the significance based on the Run 1 and Run 2 expected yields. For a Run 1 only analysis, it could improve the precision to chose cuts optimised for the Run 1 expected yields.

Regarding the MVA selection, a huge effort was put into using the new DNN classifier. The method is still at an early stage, and needs *ROOT* to be compiled against the *BLAS* library. No such version was available on *Lxplus*. The other envisaged classifier was the PyKeras method, which is a wrapper for Keras (with TensorFlow backend). This very nice method allows to use the state of the art deep learning library TensorFlow, through the architectures available in Keras. Unfortunately, *ROOT* needs in this case again to be compiled against the Keras library with TensorFlow backend, which was not available on *Lxplus*. The processing of the data had to be done on the laptop, with all inconveniences arising from this situation (slow processing time, connection instability, etc...). These struggles, together with the much faster training time and the better classification accuracy shown by the BDT classifier lead to abandon the idea of using DNN classifiers.

The poorer performance of the DNN classifiers might although be caused by the low statistics in the MC samples. Indeed, the calculation in section 8.1.2 shows that deep classifiers with a large number of neurons would overtrain given the small training sets. A deep architecture is in turn needed to model complex patterns in data. So a higher statistics in the MC samples might bring a DNN classifier to outperform the BDT one. The DNN classifiers have the ability to generalise, and to learn abstract features present in data. They are also more robust against correlations among the input variables. These nice advantages might have better overcome the differences between Run one and Run two.

The TMVA library is currently being improved with a new deep learning framework, and this analysis tried to make use of the newest additions. They are not yet largely

used and some early drawbacks persist. In addition, it can be very involving in TMVA to realise tasks that other libraries allow to do very easily. For example the *scikit-learn* library provides a *grid optimisation search*, which performs automatically a hyperparameters optimisation, without the pain of running manually a classifier for each combination of parameters. The use of scikit-learn could also be envisaged as an option for a future analysis. The PyKeras method with tensorflow backend in TMVA is yet a very promising classifier, since the training could be done completely in TensorFlow. All tools of the library might be used without restriction, at the price of converting the training sets to a format supported by TensorFlow. The classifier responses could then in turn be applied to the data through the TMVA interface.

13 Conclusion

This analysis enabled to compute a branching fraction for the $B^0 \to D_s^+ D_s^-$ decay, which is compatible with both the theoretical prediction in [5] as well as with the current limit in the PDG booklet. Since the significance is not enough to claim an observation, the computation of a limit is envisaged, once the fit consistency is thoroughly confirmed with a toy study. Unfortunately, the Run two data, which underwent the same selection process as the Run one data throughout the whole analysis, could not be used for the final result because no MC samples were available for 2015 or 2016. If the promising result obtained with the Run one data is accurate, the additional statistics added by the Run two data could enable an observation. This analysis also features very new DNN classifier methods, that were unfortunately not used in the end due to the drawbacks listed in the discussion. However, the experience made can help for future analyses to make a better use of these DNN classifiers.

14 Acknowledgments

This work would not have been possible without the help and backup of the persons who supported me. Therefore I would like to thank and express my deepest gratitude to:

Prof. Olivier Schneider, who accepted me as a Master student in the LPHE and accepted to be my thesis director. I am glad to have had the chance to write my thesis under the direction of one of the best Professors I had during my studies.

My family who had to endure my rather stressed mood during a significant part of the semester, but still stayed supportive until the end.

Biljana Mitreska, who accepted to review the rather complex selection code, and ensured me I should observe a bump with it.

The members of the LPHE, and in particular Brice and Vincenzo who gave me very useful and time saving advice.

My office mates, for answering all primary school level questions I was asking on the fly, and for cheerfully eating the cookies I made.

And last but not least, Dr. Conor Fitzpatrick, my supervisor without whom presence this project would in any respect not be as it is now. Thank you for taking time to answer my questions, whether stupid or thorny, and for being very present at any time of the semester, especially around Christmas and new year when the project really needed it.

References

[1] https://home.cern/about/physics/standard-model Consulted in January 2018

[2] https://en.wikipedia.org/wiki/Cabibbo-Kobayashi-Maskawa_matrix#cite_note-PDG2010-6 consulted in January 2018

[3] Robert Fleischer, Exploring CP Violation and Penguin Effects through $B_d^0 \to D^+D^-$ and $B_s^0 \to D_s^+D_s^-$, arXiv 0705.4421v1, 2007

[4] Thomas Blake, Gaia Lanfranchi, David M. Straub, Rare B Decays as Tests of the Standard Model, arXiv 1606.00916v2, 2016

[5] Martin Jung, Stefan Schacht, Standard Model Predictions and New Physics Sensitivity in $B \to DD$ Decays, Phys.Rev.D.91.034027, arXiv 1410.8396, 2015

[6] LHCb collaboration, Measurement of the $\bar{B}_s^0 \to D_s^+D_s^-$ and $\bar{B}_s^0 \to D_s^+D_s^-$ effective lifetimes, Physics review Letters, 10.1103/PhysRevLett.112.111802, 2014

[7] A.Hoecker & Al., "TMVA Users Guide", CERN, March 2017

[8] https://en.wikipedia.org/wiki/Deep_learning consulted in December 2017

[9] https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/ Consulted in October 2017

[10] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals, Understanding Deep Learning Requires Rethinking Generalization, ICLR 2012, https://openreview.net/pdf?id=Sy8gdB9xx

[11] Mike Williams, Vladimir Gligorov, Chris Thomas, Hans Dijkstra, Jacopo Nardulli, and Patrick Spradlin, HLT2 Topological Lines, LHCb-PUB-2011-002, 2010

[12] Giovanni Punzi, Sensitivity of searches for new signals and its optimization, PHYSTAT2003, SLAC, September 2003, arXiv physics/0308063v2

[13] LHCb collaboration, Measurement of the b-quark production cross-section in 7 and 13 TeV pp collisions, Phys.Rev.Lett., arXiv 1612.0514v7, 2017

[14] LHCb collaboration, Production of J/Ψ and Υ mesons in pp collisions at $\sqrt{s} =$ 8 TeV, JHEP, arXiv 1304.6977v3, 2013

[15] http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/STATISTICS/SIM08STAT/index.shtml consulted in December 2017

[16] http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/STATISTICS/MC11STAT/index.shtml consulted in December 2017

[17] LHCb collaboration, First observations of $\bar{B}_s^0 \to D^+D^-$, $D_s^+D^-$ and $D^0\bar{D}^0$ decays, Physical Review D 87, 10.1103/PhysRevD.87.092007, 2013

[18] S. Gianì, F. Blanc, O. Schneider, K. Trabelsi, Search for $B_s^0 \Rightarrow \eta'\phi$ decays, LHCb-ANA-2016-031, August 2017

[19] Olivier Schneider, "Introduction à la physique nucléaire et corpusculaire", EPFL, 2003

[20] Aurelio Bay, Particules Elémentaires, EPFL

[21] https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721 Consulted in December 2017

[22] https://medium.com/@tifa2up/image-classification-using-deep-neural-networks-a-beginner-friendly-approach-using-tensorflow-94b0a090ccd4 Consulted in December 2017

[23] https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f Consulted in December 2017

[24] https://en.wikipedia.org/wiki/Artificial_neural_network Consulted in December 2017

[25] https://colah.github.io/posts/2015-01-Visualizing-Representations/ Consulted in October 2017

[26] Michael Gronau, David London, Jonathan L. Rosner, Rescattering contributions to rare B-meson decays, Phys Rev D.87.036008, arXiv 1211.5785

[27] P. Baldi, P. Sadowski and D. Whiteson, Searching for Exotic Particles in High-Energy Physics with Deep Learning, arXiv 1402.4735v2, 2014

[28] https://twiki.cern.ch/twiki/bin/view/Main/LHCb-Facts consulted in December 2017

[29] http://lhcb-vd.web.cern.ch/lhcb-vd/html/project.htm consulted in January 2018

[30] http://lhcb-public.web.cern.ch/lhcb-public/en/Detector/VELO-en.html consulted in January 2018

[31] https://lhcb-public.web.cern.ch/lhcb-public/en/detector/Detector-en.html consulted in January 2018

[32] http://www.scholarpedia.org/article/CP_violation_in_electroweak_interactions consulted in January 2018

15 Appendix

15.1 Integrated luminosity obtained from the tuple variables

Year	Magnet polarity	Integrated luminosity [pb^{-1}]	Uncertainty [pb^{-1}]
2011	md	559.86	±9.57
2011	mu	418.22	±7.15
2012	md	991.26	±11.50
2012	mu	999.42	±11.59
2015	md	160.49	±6.26
2015	mu	121.08	±4.72
2016	md	848.63	±33.10
2016	mu	793.15	±30.93

Table 20: Integrated luminosity per year and magnet polarity

Note: md stands for MagDown and mu for MagUp.

15.2 Detailed summary of the signal efficiencies

channel	simulation condition	ϵ_{Gen}	$\epsilon_{Stripping}$	$\epsilon_{Offline}$	ϵ_{MVAs}	$\epsilon_{D_s\ MW}$	ϵ_{total}
B2DsDs1	2011 Pythia8 Sim08a	0.1271 ± 0.0002	0.0154 ± 0.0001	0.612 ± 0.004	0.713 ± 0.005	0.963 ± 0.002	$(8.22 \pm 0.10) \cdot 10^{-4}$
	2011 Pythia6 Sim08a	0.1184 ± 0.0002	0.0124 ± 0.0001	0.604 ± 0.004	0.716 ± 0.005	0.959 ± 0.003	$(6.10 \pm 0.08) \cdot 10^{-4}$
	2012 Pythia8 Sim08a	0.1291 ± 0.0004	0.0139 ± 0.0001	0.607 ± 0.005	0.693 ± 0.006	0.964 ± 0.003	$(7.29 \pm 0.11) \cdot 10^{-4}$
	2012 Pythia6 Sim08a	0.1217 ± 0.0004	0.0113 ± 0.0001	0.598 ± 0.005	0.669 ± 0.007	0.958 ± 0.003	$(5.26 \pm 0.09) \cdot 10^{-4}$
B2DsDs2	2012 Pythia8 Sim08a	0.1112 ± 0.0003	0.0078 ± 0.0001	0.494 ± 0.007	0.666 ± 0.009	0.920 ± 0.006	$(2.62 \pm 0.06) \cdot 10^{-4}$
	2012 Pythia6 Sim08a	0.1036 ± 0.0003	0.00639 ± 0.00009	0.513 ± 0.007	0.679 ± 0.009	0.923 ± 0.007	$(2.13 \pm 0.05) \cdot 10^{-4}$
B2DsDs3	2012 Pythia8 Sim08a	0.1183 ± 0.0004	0.0073 ± 0.0001	0.574 ± 0.007	0.654 ± 0.008	0.944 ± 0.005	$(3.06 \pm 0.07) \cdot 10^{-4}$
	2012 Pythia6 Sim08a	0.1096 ± 0.0003	0.00601 ± 0.00009	0.578 ± 0.007	0.650 ± 0.009	0.948 ± 0.005	$(2.35 \pm 0.06) \cdot 10^{-4}$
B2DsDs4	2012 Pythia8 Sim08a	0.0963 ± 0.0002	0.0173 ± 0.0002	0.410 ± 0.004	0.653 ± 0.006	0.916 ± 0.005	$(4.09 \pm 0.07) \cdot 10^{-4}$
	2012 Pythia6 Sim08a	0.0892 ± 0.0002	0.0146 ± 0.0001	0.422 ± 0.005	0.658 ± 0.007	0.883 ± 0.006	$(3.19 \pm 0.06) \cdot 10^{-4}$

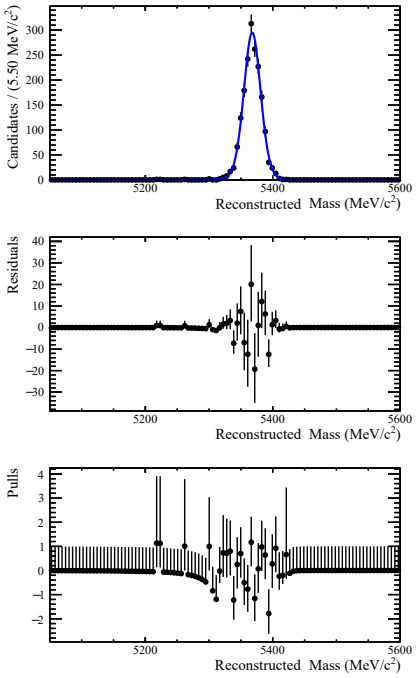
Table 21: Signal efficiencies determined with the normalisation channel MC, detailed view.

15.3 Detailed summary of the background efficiencies

Decay	simulation condition	ϵ_{Gen}	$\epsilon_{Stripping}$	$\epsilon_{Offline}$	ϵ_{MVAs}	$\epsilon_{D_s\ MW}$	ϵ_{total}
$B^0 \to D^- D_s^+$	2011 Pythia8 Sim08h	0.1071 ± 0.0002	0.00604 ± 0.00005	0.031 ± 0.001	0.50 ± 0.02	0.31 ± 0.03	$3.14E-06 \pm 3.6E-07$
	2012 Pythia8 Sim08h	0.1184 ± 0.0002	0.00634 ± 0.00003	0.034 ± 0.001	0.41 ± 0.01	0.30 ± 0.02	$3.17E-06 \pm 2.7E-07$
	2012 Pythia8 Sim08a	0.1187 ± 0.0002	0.00611 ± 0.00006	0.034 ± 0.002	0.44 ± 0.03	0.23 ± 0.03	$2.54E-06 \pm 4.1E-07$
	2012 Pythia6 Sim08a	0.1103 ± 0.0002	0.00499 ± 0.00005	0.036 ± 0.002	0.45 ± 0.03	0.27 ± 0.04	$2.41E-06 \pm 3.9E-07$
$B_s^0 \to D_s D_s^*$	2012 Pythia8 Sim08a	0.1275 ± 0.0003	0.0167 ± 0.0002	0.498 ± 0.006	0.34 ± 0.01	0.942 ± 0.006	$3.34E-04 \pm 9E-06$
$B_s^0 \to D_s^{*+} D_s^{*-}$	2012 Pythia8 Sim08a	0.1265 ± 0.0000	0.0155 ± 0.0002	0.478 ± 0.006	0.119 ± 0.006	0.61 ± 0.02	$6.74E-05 \pm 4.1E-06$

Table 22: Background efficiencies determined with the MC samples in the B2DsDs1 channel, detailed view.

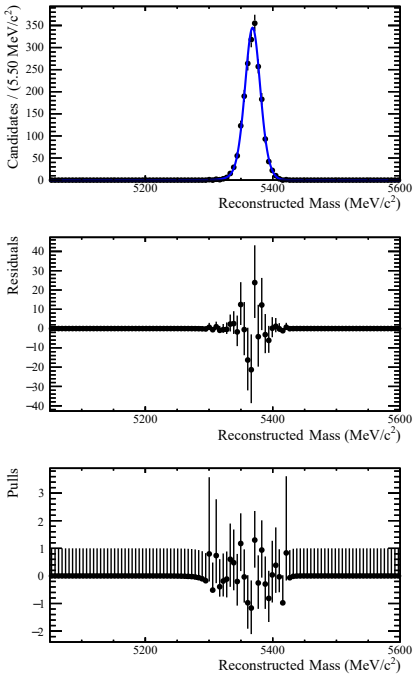
15.4 Signal shapes from the other channels



(a) Fit shape for the signal and normalisation mode in the B2DsDs2 channel taken from the $B_s^0 \to D_s^+ D_s^-$ 2012 and 2011 MC samples produced with Pythia 8 and Sim08a.

Parameter	value	uncertainty
μ	5367.78	± 0.34
σ_l	15.03	± 0.59
n_l	2.38	± 0.98
a_l	2.39	± 0.31
σ_r	10.72	± 1.09
n_r	14.10	± 7.50
a_r	-2.20	± 0.74

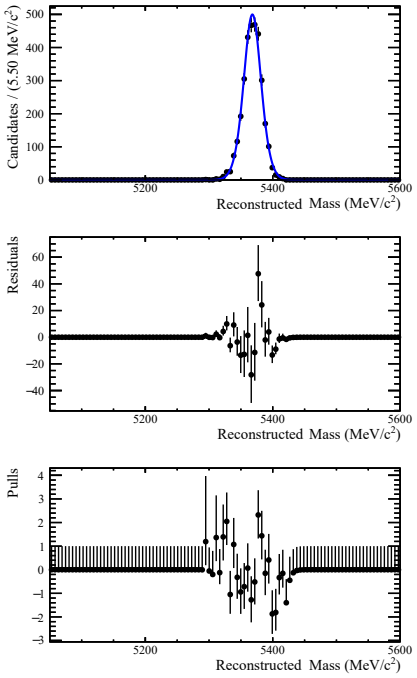
(b) Fit parameters for the $B_s^0 \to D_s^+ D_s^-$ decay in the B2DsDs2 channel.



(a) Fit shape for the signal and normalisation mode in the B2DsDs3 channel taken from the $B_s^0 \to D_s^+ D_s^-$ 2012 and 2011 MC samples produced with Pythia 8 and Sim08a.

Parameter	value	uncertainty
μ	5368.38	± 0.34
σ_l	14.38	± 0.61
n_l	15.00	± 3.17
a_l	1.92	± 0.21
σ_r	9.55	± 0.83
n_r	14.90	± 9.72
a_r	-2.66	± 1.74

(b) Fit parameters for the $B_s^0 \to D_s^+ D_s^-$ decay in the B2DsDs3 channel.



(a) Fit shape for the signal and normalisation mode in the B2DsDs4 channel taken from the $B_s^0 \to D_s^+ D_s^-$ 2012 and 2011 MC samples produced with Pythia 8 and Sim08a.

Parameter	value	uncertainty
μ	5368.04	± 0.26
σ_l	16.23	± 0.29
n_l	14.64	± 0.74
a_l	5.17	± 4.21
σ_r	10.66	± 0.60
n_r	14.64	± 0.73
a_r	-5.83	± 1.58

(b) Fit parameters for the $B_s^0 \to D_s^+ D_s^-$ decay in the B2DsDs4 channel.