Chapter 5

$\mathrm{D_s^+}$ and $\mathrm{D^+}$ signal extraction

The main ingredient for the evaluation of the $\mathrm{D_s^+}/\mathrm{D^+}$ production-yield ratio are the $\mathrm{D_s^+}$- and $\mathrm{D^+}$-meson raw yields, i.e., the number of reconstructed and selected $\mathrm{D_s^+}$ and $\mathrm{D^+}$ mesons. Due to the vast amount of combinatorial background, the extraction of the raw yields through a simple candidate counting method is not feasible. Instead, the raw yield is obtained on a statistical basis by fitting the invariant-mass distribution of the $\mathrm{D_s^+}$- and $\mathrm{D^+}$-meson candidates passing tight selection criteria. To reduce the combinatorial background and preserve the highest possible efficiency of D-meson selection, Machine Learning algorithms have been employed. The following sections describe the procedure for the extraction of the raw yield of $\mathrm{D_s^+}$ and $\mathrm{D^+}$ mesons.

5.1 Machine Learning

The term *Machine Learning* (ML) is a broad and versatile concept, encompassing a wide range of algorithms that grant computers the capacity to learn and adapt without being explicitly programmed to do so [1]. A more comprehensive definition characterises ML as the study of algorithms that enhance their performance at a specific task through the accumulation of experience [2]. In recent years, ML techniques have witnessed widespread adoption across diverse fields, with significant impacts realised especially with the emergence of generative models such as GPT [3]. ML algorithms have found extensive applications in the high-energy physics field, primarily for the task of distinguishing interesting signals from the vast background present in particle-collision data. Furthermore, these algorithms have been employed as triggers, aiding in the rapid identification of events of interest, and have also been instrumental in event reconstruction. Notably, ML algorithms were used in the discovery of the Higgs boson [4], one of the most significant achievements in the field of particle physics of the last decades.

5.1.1 Supervised learning

Supervised learning is one of the main branches of machine learning, along with unsupervised and reinforcement learning. Machine learning tasks are usually described in terms of how the machine learning system should process an example, which is a collection of features $\mathbf{x}$ that have been quantitatively measured from some object or event that one wants the machine learning system to process. In the case of supervised learning, each example is coupled with a corresponding label or target, $\mathbf{y}$. The objective of supervised learning is to learn to predict or infer $\mathbf{y}$ based on the associated features, $\mathbf{x}$, assuming that there exists a functional relationship $\mathbf{y} = f(\mathbf{x})$ between the two. The goal of the ML system is to produce an approximation $\widehat{f}(\mathbf{x})$ of the true function $f(\mathbf{x})$ by minimising a given loss function, which quantifies the discrepancy between the predicted and true labels. Supervised learning problems are further segmented into two distinct sub-categories: classification and regression. In the former, the label $\mathbf{y}$ assumes values from a finite and discrete set of categories, often representing distinct classes or groups. In the latter, the label $\mathbf{y}$ takes the form of one or more continuous variables, necessitating the learning system to deduce a continuous function or mapping between $\mathbf{x}$ and $\mathbf{y}$.

The usage of ML algorithms in classification problems, such as the one presented in this Thesis, allows for the definition of multi-dimensional non-linear decision boundaries, which are not available with traditional selection methods based on linear selections on some cut variables. This is particularly important as it provides more efficient selections and a larger purity of the selected data sample.

The application of a supervised learning algorithm to a dataset involves the following steps: i) the model is trained on a set of labelled data, i.e., the value of $\mathbf{y}$ is known for each example in the training set; ii) the model is tested on a separate set of labelled data, known as the test set, to evaluate its performance; iii) the model is then used to make predictions on new, unseen data.

Training

During the training process, the model learns (i.e., adjusts its internal parameters) to map the input features $\mathbf{x}$ to the corresponding labels $\mathbf{y}$ by minimizing a given loss function. Typically used loss functions include the Mean Squared Error (MSE) for regression tasks and the Cross-Entropy loss [5] for classification tasks. The loss function is minimised through an optimisation algorithm, usually stochastic gradient descent [6], which iteratively updates the model parameters to reduce the loss. Since an over-optimisation of the model on the training data can lead to poor generalisation on unseen data (the model is said to be *overfitting*), a regularisation term is often added to the loss function to penalise overly complex models. The training process continues until the model reaches a satisfactory level of performance on the training data, or until its performance does not improve further.

Before the training phase, hyperparameter tuning is performed to optimise the model's performance. *Hyperparameters* are parameters that are not learned during the training process, but rather define the model's architecture and the training process itself. Hyperparameter tuning is usually performed through a grid search, random search, or with a more efficient bayesian optimisation [7, 8, 9]. Several combinations of hyperparameters are tested on a dedicated labelled dataset: the validation set. Models with different hyperparameter sets are trained with a reduced training phase, and the one yielding the best performance is then selected for a full training phase.

Testing

After the model has been trained, its performance is evaluated on a dataset that was not used during the training process, known as the test set. Like the training and validation sets, also the test set contains labelled examples. While during the training the model is optimised to minimise the loss function, the test set is used to estimate the model's generalisation error, i.e., how well the model performs on unseen data. The model's performance is evaluated using metrics that are specific to the task at hand, such as accuracy for classification tasks, or Mean Squared Error (MSE) for regression tasks. Once the model achieves satisfactory performance on the test set, it is ready to be used for making predictions on unlabelled data.

Cross-validation

With the strategy defined above to optimise the hyperparameters, train the model, and validate its performance, the dataset is divided into three subsets: the validation set, the training set, and the test set, which are used for hyperparameter tuning, training, and testing, respectively. When small datasets are involved, this division can lead to a suboptimal model, as the model's performance can be highly dependent on the specific examples in the training, validation, and test sets. Furthermore, this approach limits the amount of data available for training the model, which can lead to poor generalisation. To mitigate this issue, cross-validation [10] is often employed. This term refers to a set of techniques that allow for a more robust estimate of the model's generalisation performance by using the entire dataset for training and validation. The most common cross-validation technique is the k-fold cross-validation. It consists in dividing the training sample into k subsets of equal size, called *folds*. Then, the ML algorithm is trained k times, each time using $k-1$ folds as training set, while the remaining fold is used as validation set. The model's performance is then averaged over the k folds to obtain a more robust estimate of this quantity. This operation is repeated for each hyperparameter configuration to be considered. The hyperparameter configuration minimising the loss function or other metrics used to evaluate the model's performance is then chosen as the optimal configuration.

5.2 $\mathrm{D_s^+}$ and $\mathrm{D^+}$ selection using Machine Learning

The task of extracting $\mathrm{D_s^+}$- and $\mathrm{D^+}$-meson signals from the vast combinatorial background is a challenging one, due to the large amount of background compared to signal. It is however an excellent example of classification problem, and ML algorithms can therefore be exploited to enhance the efficiency of the selection.

5.2.1 Data preparation

In order to train a ML model, a labelled dataset with a well-defined set of features is required. The dataset used for training the ML algorithms employed in this Thesis is composed of a number of signal and background examples. To obtain a pure sample of signal candidates, Monte Carlo (MC) techniques are used to generate $\mathrm{D_s^+}$

and D^+ mesons and transport their decay products through a simulation of the ALICE apparatus. Proton-proton collisions are simulated using the PYTHIA 8 event generator [11] with colour-reconnection Mode 2 [12], and the generated particles are propagated through the ALICE detector using the GEANT4 transport simulation toolkit [13]. A dataset enriched of heavy-flavour hadrons is obtained by only selecting (*triggering*) events where at least a $c\bar{c}$ or $b\bar{b}$ pair is produced. The produced D_s^+ and D^+ mesons are then forced to decay into the studied decay channels, i.e.: $D_s^+ \to \phi\pi \to K^+K^-\pi^+$, $D_s^+ \to K^+K^{*0} \to K^+K^-\pi^+$, characterised by the same $K^+K^-\pi^+$ final state, $D^+ \to \pi^+K^-\pi^+$, and $D^+ \to \phi\pi \to K^+K^-\pi^+$.

Only prompt and non-prompt $D_s^+ \to \phi\pi \to K^+K^-\pi^+$ decays are used to train the model, as D^+ mesons decay into the same final state as D_s^+ mesons, and selections optimised to reconstruct D_s^+ mesons are also effective for D^+ mesons.

Background candidates are obtained from real data, as MC simulations may not be able to reproduce the complexity of the jet fragmentation and soft processes occurring in the underlying event. Background examples are obtained by selecting candidates from a subsample of the full data sample (corresponding to its 3%) in an invariant-mass region away from both the D_s^+- and D^+-meson mass peaks, where $1.7 < M < 1.75$ GeV/c^2 or $2.1 < M < 2.15$ GeV/c^2, as shown in Fig. 5.1.

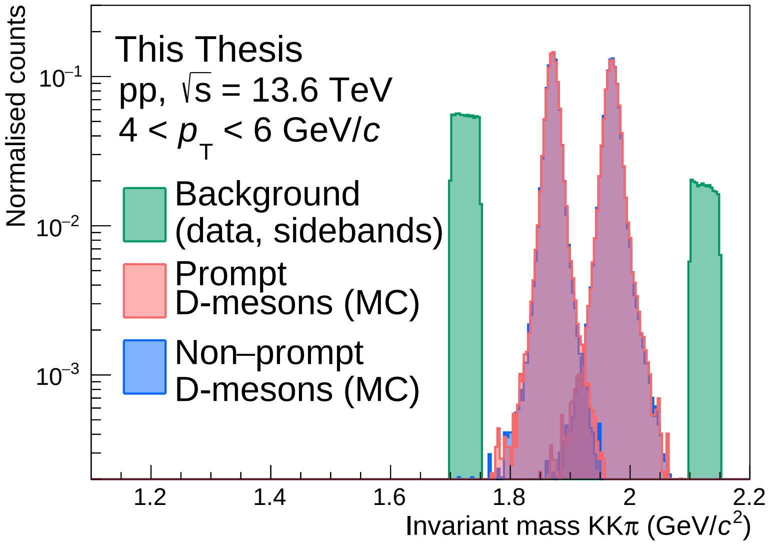


Figure 5.1: Invariant mass distribution of prompt and non-prompt D mesons (red and blue, respectively), taken from MC simulations, and of the background candidates taken from real data used to train the ML model (green) in the $4 < p_T < 6$ GeV/c interval. Background candidates are selected in the $1.7 < M < 1.75$ GeV/c^2 or $2.1 < M < 2.15$ GeV/c^2 invariant-mass interval.

To reduce the amount of combinatorial background in the data samples prior to the application of ML-based selections, some loose selections were applied. These pre-selections are very effective at rejecting the combinatorial background, while preserving the selection efficiency for D_s^+ and D^+ mesons, and are reported in Table 5.1. These criteria include selections on the invariant mass of the KKπ triplet to reduce the number of candidates to those necessary for the invariant mass analysis.

To separate signal and background, topological variables like the decay length L and the cosine of the pointing angle $\cos\theta_{\rm p}$, introduced in Sec. **??** and PID variables, introduced in Sec. **??**. A selection is also applied to the $\chi^2_{\rm PCA}$ variable, quantifying the dispersion of the decay tracks around the secondary vertex, normalised to their uncertainty. The selection criterion on the $n\sigma_{\rm TOF}$ variable is applied only when a TOF cluster is matched to the track, i.e., tracks with no TOF signal are not rejected. In addition, a logical `OR` is applied to the conditions on the PID variables, meaning that a candidate is accepted if at least one of the conditions is satisfied.

Table 5.1: Topological and kinematic pre-selections applied to the $\mathrm{D_s^+}$- and $\mathrm{D^+}$-mesons candidates. The selection criterion on the $n\sigma_{\rm TOF}$ variable is applied only when a TOF cluster is matched to the track, i.e., tracks with no TOF signal are not rejected. In addition, a logical `OR` is applied to the conditions on the PID variables, meaning that a candidate is accepted if at least one of the conditions is satisfied.

	$p_{\rm T}$ interval (GeV/c)		
Variable	0.0−1.5	1.5−12	12−24
$\lvert M - M^{\rm PDG}_{\mathrm{D_s^+}}\rvert$ (MeV/c^2) <	400	400	400
$p_{\rm T}(\pi, \mathrm{K})$ (GeV/c) >	0.3	0.4	0.4
Decay length (cm) >	0.02	0.02	0.03
Normalized decay length XY >	2	2	2
$\cos\theta_{\rm p}$ >	0.85	0.85	0.85
$\cos\theta_{\rm p}^{xy}$ >	0.85	0.85	0.85
$M^{\phi}_{\rm inv}$ - $M^{\phi\ \rm PDG}_{\rm inv}$ (MeV/c^2) <	20	20	20
$\chi^2_{\rm PCA}$ <	10	10	10
$\lvert n\sigma_{\rm TPC}\rvert$ <	5	5	5
$\lvert n\sigma_{\rm TOF}\rvert$ <	5	5	5

Labels are assigned as a numerical value to each candidate, with 0 indicating a background candidate, 1 a prompt $\mathrm{D_s^+}$ meson, and 2 a non-prompt $\mathrm{D_s^+}$ meson.

The dataset is then divided into two different subsamples. The first comprehends 80% of the data, and is used to train the model, while the remaining 20% is used to test its performance. In addition, since the D-meson decay topology can significantly differ depending on the $p_{\rm T}$ of the meson due to different Lorentz boosts, the dataset is divided into several $p_{\rm T}$ intervals, and the model is trained and tested separately for each of them. To achieve a better performance of the ML models, they are trained in broader $p_{\rm T}$ intervals than those used for the analysis, to ensure enough data is available to train a well-performing model. The total number of candidates available for training and testing the model is reported in Table 5.2 for the considered $p_{\rm T}$ intervals.

To produce a balanced dataset, the number of candidates in each class is equalised to the number of examples in the minority class. This is achieved by randomly selecting a subset of the majority classes. The balanced dataset is then used to train the model.

Table 5.2: Number of candidates within the p_T intervals used to train and test the model.

p_T (GeV/c)	Candidates
0–1.5	$\sim 4.6 \times 10^3$
1.5–2	$\sim 6.1 \times 10^3$
2–3	$\sim 26 \times 10^3$
3–4	$\sim 34 \times 10^3$
4–5	$\sim 31 \times 10^3$
5–6	$\sim 24 \times 10^3$
6–8	$\sim 22 \times 10^3$
8–12	$\sim 10 \times 10^3$
12–24	$\sim 2.6 \times 10^3$

The choice of features used to separate signal from background is crucial, as they must be able to discriminate between signal and background candidates, and must be chosen in such a way that no bias is introduced in the final result. The variables used to train the model were introduced in Chapter **??**, and are a mixture of topological, kinematic, and PID variables. The key idea is to exploit the displaced topology of the D-meson decay, which is a distinctive feature of the signal candidates, the kinematic properties of the D-meson decay, and the PID information of the daughter tracks to discriminate between signal and background candidates. The features used to train the model are reported in Table 5.3. The number in parenthesis after $n\sigma$ indicates the prong identification number. The tracks are ordered based on their charge sign, so that the first and third prongs (prongs 0 and 2) are the like-sign tracks (i.e., their charge sign is the same as that of the D meson), while the second prong (prong 1) is the opposite-sign track.

The invariant mass of the candidate and its p_T are excluded from the model training to prevent bias in the final results. Including these variables would cause the model to preferentially select candidates within a specific invariant mass range (corresponding to D_s^+ and D^+ mesons) or p_T, thereby skewing both candidate selection and the p_T distribution of the final sample. This would result in a biased p_T-differential yield. While it is possible to correct for a p_T-dependent selection bias using MC simulations, this approach is not taken here to avoid biases from any inaccuracies in the p_T distribution of the D mesons in the simulations.

Some of the variables used to train the model may be correlated with the invariant mass of the candidate, and the ML may learn to discriminate the signal from the background by exploiting this correlation with the D_s^+ meson mass and transverse momentum, rather than the physical properties of the signal and background. To exclude this possibility, the correlation between the features used to train the model is studied. To quantitatively describe the correlation between the variables, the Pearson correlation coefficient ρ is evaluated for each pair of variables. It is defined as the ratio between the covariance of two variables x and y and the product of their standard deviations, $\rho(x, y) = \text{cov}(x, y)/(\sigma_x \sigma_y)$. It expresses the strength and

Table 5.3: Candidate features used to train the ML model.

Variable
Cosine of pointing angle ($\cos\theta_\mathrm{p}$)
Cosine of pointing angle XY ($\cos\theta_\mathrm{p}^{xy}$)
Decay length (L)
Decay length XY (L^{xy})
Candidate impact parameter XY (d^{xy})
$\lvert\cos^3\theta'(\mathrm{K})\rvert$
Prong 0 impact parameter XY (d_0^{xy})
Prong 1 impact parameter XY (d_1^{xy})
Prong 2 impact parameter XY (d_2^{xy})
$n\sigma_\mathrm{comb}^{\pi}(0)$
$n\sigma_\mathrm{comb}^{\pi}(1)$
$n\sigma_\mathrm{comb}^{\pi}(2)$
$n\sigma_\mathrm{comb}^{\mathrm{K}}(0)$
$n\sigma_\mathrm{comb}^{\mathrm{K}}(1)$
$n\sigma_\mathrm{comb}^{\mathrm{K}}(2)$

direction of a linear correlation between two variables, ranging from $\rho = 1$ (perfect positive linear correlation) to $\rho = -1$ (perfect negative linear relationship). $\rho = 0$ indicates the absence of linear correlation.

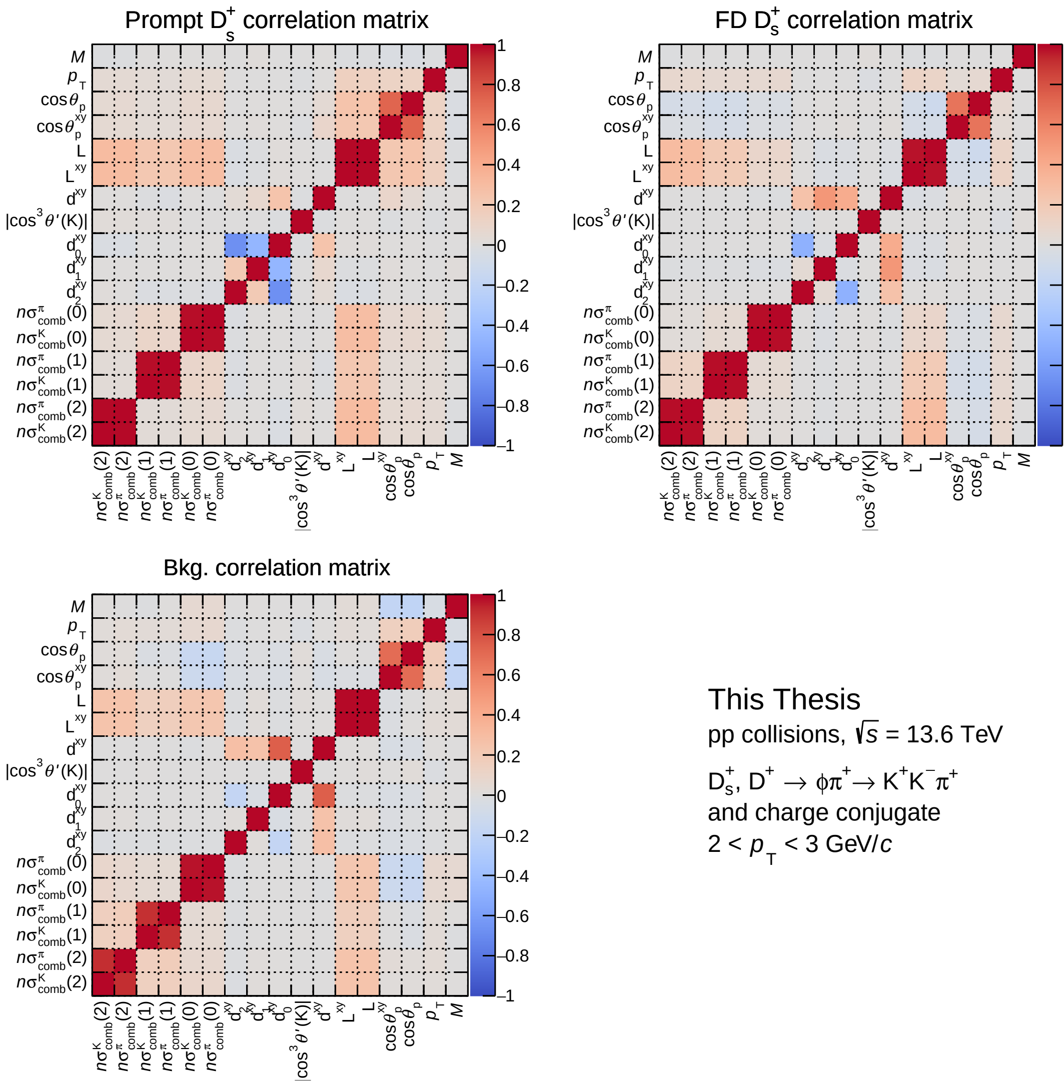
The correlation matrix of the features used to train the model is shown in Fig. 5.2 for the prompt $\mathrm{D_s^+}$, non-prompt $\mathrm{D_s^+}$ and background classes, in the $2 < p_\mathrm{T} < 3$ GeV/c interval of D-meson tranverse momentum. The correlation with the invariant mass and the transverse momentum is also reported. The Pearson coefficient is encoded in the colour of the cell, with red indicating a positive correlation, blue a negative correlation, and grey no correlation. The correlation matrix shows that the variables used to train the model are not correlated with the invariant mass of the candidate, suggesting that a ML model should not modify the invariant-mass distribution of the selected candidates.

Variables carrying similar physical information, such as those related to the candidate decay length, pointing angle, and impact parameter, are strongly correlated among each other, as expected. Different degrees of correlation between the same variable pairs are observed for the different classes. The ML model can exploit these differences to discriminate between the three classes of candidates.

5.2.2 Boosted Decision Trees

Once the training dataset has been composed and the features have been selected, the ML architecture has to be chosen. Several algorithms are available, each with its own strengths and weaknesses. The choice of the algorithm depends on the specific problem to solve, the size of the dataset, and the computational resources available.

Boosted decision trees [14, 15] (BDTs) are a family of machine learning algo-


Prompt D_s^+ correlation matrix

FD D_s^+ correlation matrix

Bkg. correlation matrix

This Thesis

pp collisions, $\sqrt{s}$ = 13.6 TeV

D_s^+, $D^+ \rightarrow \phi\pi^+ \rightarrow K^+K^-\pi^+$

and charge conjugate

$2 < p_T < 3$ GeV/c

Figure 5.2: Correlation matrix of the features used to train the ML model for prompt D_s^+ (top-left), non-prompt D_s^+ (top-right), and background (bottom-left) candidates in the $2 < p_T < 3$ GeV/c interval. The correlation with the invariant mass and the transverse momentum is also reported. The Pearson coefficient is encoded in the colour of the cell, with red indicating a positive correlation, blue a negative correlation, and grey no linear correlation.

rithms employed in different fields, including high-energy physics. Their building blocks are decision trees, which are a versatile type of supervised learning algorithm that can be used for both classification and regression tasks. A decision tree is made of many *nodes*, each containing conditions that split the data into two [16] or more [17] children nodes. The first node of the tree, which receives all the data, is called *root* node, while nodes that do not further split the data are called *leaves*, and contain the output of the tree. The model is trained by considering the Gini index, which measures the impurity of the node:

$$G = 1 - \sum_{i=1}^{n} p_i^2 \quad ,$$

where p_i is the fraction of samples in the node that belong to the class i. The Gini index therefore provides an indication of the quality of the split, with $G = 0$ indicating a perfect split. A commonly used algorithm to build *binary* decision trees (i.e., each node contains binary-output conditions, and is split into two children nodes) is the *Classification And Regression Tree* (CART) algorithm [16], which recursively splits the dataset into subsets based on a single feature k and a threshold t_k that minimises the impurity of the subsets (weighted by their size). The cost function that the algorithm tries to minimise is given by

$$J(k, t_k) = \frac{m_{\mathrm{left}}}{m} G_{\mathrm{left}} + \frac{m_{\mathrm{right}}}{m} G_{\mathrm{right}} \quad ,$$

where m_{left} and m_{right} are the number of samples in the left and right nodes, respectively, summing up to the total number of samples m, and G_{left} and G_{right} are the Gini indices of the left and right nodes. The tree is grown until a stopping criterion is met, such as a maximum depth, a minimum number of samples in a node, or a minimum impurity decrease. These are all hyperparameters that can be tuned to optimise the model's performance.

Given their simplicity, decision trees are fairly easy to interpret, and are often called *white-box* models (in contrast to BDTs and neural networks, where the decision-making process is less transparent, therefore called *black-box* models). An additional strength of decision trees is that they require very little data preparation, e.g., they do not require feature scaling or centering, making them a very powerful yet simple tool for data analysis. However, they are prone to overfitting, as they can grow to a large depth, capturing the noise in the training data. To mitigate this issue, their depth can be constrained, but this may lead to a model with limited discrimination power. To build a robust model with a good discrimination power, ensemble methods may be used. Several decision trees can be trained, and the final prediction is obtained by combining the outcome of all the trees.

XGBoost

In this work, the Extreme Gradient Boosting [18] (XGBoost) Boosted Decision Trees (BDT) algorithm is used. It has achieved state-of-the-art results in a number of machine learning and data mining challenges (for example in Ref. [19]). In addition, this algorithm, which is available as an open-source package, can be easily parallelised on CPUs and GPUs [20], thereby reducing the training and application time.

The term *boosting* refers to any ensemble method combining several weak learners into a strong learner. The general idea of most boosting methods is to train many predictors sequentially, each trying to correct its predecessor [21]. The function estimate $\widehat{f}(x)$ is parametrised with an additive functional form:

$$\widehat{f}(x) = \sum_{\mathrm{k}=1}^{\mathrm{M}} \widehat{f}_{\mathrm{k}}(x) \quad ,$$

where M is the number of iterations, $\widehat{f}_0(x)$ is the initial prediction, and $\widehat{f}_{\mathrm{i}}(x)$ is the function increment at the *i*-th iteration, also called *boost*. To reduce the loss function, a new weak learner, whose functional form is parametrised as $h(x, \theta)$, can be added to the ensemble:

$$\widehat{f}_{\mathrm{t}}(x) \leftarrow \widehat{f}_{\mathrm{t}-1}(x) + \rho_{\mathrm{t}} h(x, \theta_{\mathrm{t}}) \quad .$$

ρ_t is the step size, which is optimised for each iteration t, together with the parameters θ_{t} of the weak learner:

$$(\rho_{\mathrm{t}}, \theta_{\mathrm{t}}) = \arg\min_{\rho,\theta} \sum_{i=1}^{\mathrm{N}} L\left(y_i, \widehat{f}_{\mathrm{t}-1}(x_i) + \rho h(x_i, \theta)\right) \quad ,$$

where L is the loss function, and y_i is the true label of the i-th example. Despite having a well-defined set of equations for minimising the loss function, the optimisation of the parameters is not trivial, as the loss function is non-convex and the search space is high-dimensional. Therefore, the optimisation is usually performed using a gradient-based algorithm [14, 22], where $h(x, \theta_{\mathrm{t}})$ is chosen as the most parallel function to the negative gradient of the loss function with respect to the previous prediction $g_{\mathrm{t}}(x)$:

$$g_{\mathrm{t}}(x) = E_{\mathbf{y}}\left[\left. \frac{\partial L(\mathbf{y}, \widehat{f}_{\mathrm{t}-1}(x))}{\partial \widehat{f}_{\mathrm{t}-1}(x)} \right| x \right] \quad ,$$

where $E_{\mathbf{y}}$ is the expectation over the true labels. The parameters are then optimised by minimising the difference between the negative gradient and the weak learner prediction:

$$(\rho_{\mathrm{t}}, \theta_{\mathrm{t}}) = \arg\min_{\rho,\theta} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left[-g_{\mathrm{t}} - \rho h(x_{\mathrm{i}}, \theta)\right]^2 \quad .$$

Through the iterative addition of weak learners, the loss function is reduced and the model learns the complex patterns of data. The final prediction is obtained by summing the predictions of all the weak learners. In the XGBoost algorithm, the weak learners are decision trees. The output consists of a numerical score for each class, ranging from 0 to 1 and summing up to unity. Each score represents the confidence of the model in the prediction, which can be interpreted as the probability of the example belonging to that class.

5.2.3 Tuning the model's hyperparameters

The XGBoost algorithm has several hyperparameters [23] that can be tuned to optimise the model's performance. The most important hyperparameters are:

- `eta` or `learning_rate`, which is the step size shrinkage of the gradient descent algorithm. To reduce the risk of overfitting, this factor multiplies the weak-learner prediction ($\rho_\mathrm{t} h(x_\mathrm{i}, \theta) \rightarrow \texttt{eta} \cdot \rho_\mathrm{t} h(x_\mathrm{i}, \theta)$), and is usually set to a small value, such as 0.3;

- `max_depth`, which is the maximum depth of a single decision tree. A large depth can lead to overfitting, while a small depth can lead to a model with limited discrimination power. Usually, this parameter is set to around 5;

- `n_estimators`, which defines the number of trees to train. A large number of weak learners can lead to overfitting, while a small number can lead to a model with limited discrimination power. Usually, the number of weak learners is set to around 1000;

- `subsample`, which is the fraction of the training data to be used to train each tree at each iteration;

- `min_child_weight`, which is the minimum sum of instance weight needed in a child. It is related to the purity in a node, and it is used to stop the tree growth;

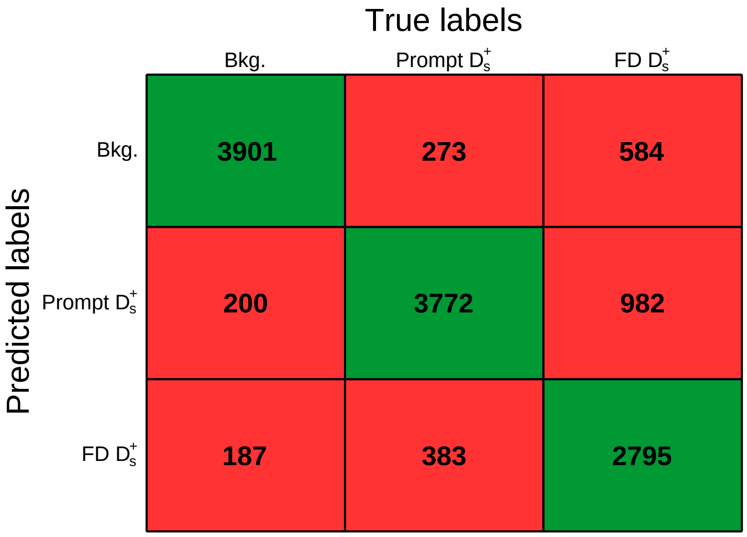
- `colsample_bytree`, which is the fraction of features to be used to train each tree at each iteration;

- `tree_method`, which defines the algorithm used to build the trees. The `hist` option uses an optimised histogram-based algorithm and is usually the fastest.

The hyperparameters are optimised using the Optuna framework [24], which proved to be a powerful tool thanks to its state-of-the-art algorithms for sampling the hyperparameter space and for efficiently pruning unpromising trials. The Tree-Structured Parzen Estimator [25], is used in this Thesis. It is a Bayesian optimisation algorithm able to explore the hyperparameter space efficiently. The aim of a Bayesian optimisation is to maximise (or minimise, depending on the task) an objective function $f(\mathbf{x})$ by iteratively sampling a bounded hyperparameter space, χ. The algorithm builds a probabilistic model of the objective function, and uses it to decide which hyperparameters to sample next. The model is updated at each iteration, and the hyperparameters that are most likely to improve the model's performance are sampled. The Optuna algorithm is also able to prune unpromising trials, reducing the computational cost of the optimisation. The optimisation is performed using a 5-fold cross-validation, and the hyperparameters that maximise the macro-averaged one-vs-one Receiver Operating Characteristic Area Under the Curve (ROC AUC) metric (described in detail in Sec. 5.2.4) are chosen as the optimal configuration. The hyperparameters optimised for the XGBoost model are reported in Table 5.4. An additional hyperparameter, `lambda`, which is the L_2 regularisation term, is also optimised. It helps to prevent overfitting by penalising overly

Table 5.4: Optimised hyperparameter configuration for the p_T bins considered in the model training.

Hyper-parameter	p_T interval (GeV/c)								
	0–1.5	1.5–2	2–3	3–4	4–5	5–6	6–8	8–12	12–24
`max_depth`	3	3	3	3	3	3	3	3	3
`learning_rate`	0.04	0.068	0.065	0.10	0.091	0.84	0.070	0.046	0.030
`n_estimators`	473	339	1352	909	1256	1392	1142	1437	1188
`min_child_weight`	1	3	10	10	10	9	3	7	5
`subsample`	0.87	0.95	0.84	0.85	0.95	0.85	0.81	0.94	0.88
`colsample_bytree`	0.91	0.98	0.90	0.98	0.96	0.95	0.88	0.96	0.89
`lambda`	8.0×10^{-4}	4.8×10^{-4}	9.1×10^{-4}	1.4×10^{-4}	3.0×10^{-4}	3.2×10^{-4}	1.9×10^{-4}	9.8×10^{-4}	6.7×10^{-4}
`tree_method`	`hist`	`hist`	`hist`	`hist`	`hist`	`hist`	`hist`	`hist`	`hist`



Predicted labels \ True labels	Bkg.	Prompt D_s^+	FD D_s^+
Bkg.	3901	273	584
Prompt D_s^+	200	3772	982
FD D_s^+	187	383	2795

Figure 5.3: Confusion matrix for the BDT model trained in the $2 < p_T < 3$ GeV/c interval. Candidates are classified as the class with the highest score.

complex models. The optimal hyperparameters are then used to train the model on the full training dataset.

5.2.4 Evaluation of the model's performance

After training the model, its performance is evaluated on the test dataset. The model's performance can be assessed using a *confusion matrix*, which summarises the number of examples for a given class (the true label) that are classified by the model as belonging to any of the available classes (the predicted label). A good model should provide a high number of correctly-classified examples (reported on the diagonal of the confusion matrix), and a low number of misclassified examples (off-diagonal elements of the confusion matrix). The confusion matrix also allows an understanding of which classes are more difficult to classify, and which classes are more likely to be confused with each other. An example of a confusion matrix is shown in Fig. 5.3 for the XGBoost model trained on the $2 < p_T < 3$ GeV/c interval.

Despite providing a lot of information on the model's performance, more concise metrics of the model's performance are usually used, for a more direct comparison between different models. In addition, the confusion matrix provides a threshold-dependent measure of the model's performance, as the classification threshold (i.e., the threshold on the model output that defines the separation between the different classes) can be varied to increase the number of correctly classified signal candidates at the expense of the number of correctly classified background candidates, and vice versa.

In binary classification tasks, where only two classes are available (a positive and a negative class), several metrics can be defined from the elements of the confusion matrix. The 2×2 confusion matrix contains four entries: the true positives (TP), which are the number of correctly classified positive candidates, the false positives (FP), which are the number of negative candidates being mistakenly classified as positives, and the analogously defined true negatives (TN) and false negatives (FN). One of the most used tools for binary classifiers is the *Receiver Operating Characteristic* (ROC) curve, which represents the true positive rate (TPR) against the false positive rate (FPR) for different threshold values. The TPR is the fraction of correctly classified positive candidates ($\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$), while the FPR is the fraction of incorrectly classified negative candidates ($\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$).

In a binary classification task, the output of the ML model is a single value ranging from 0 to 1, and can be interpreted as the probability of the candidate belonging to the positive class. If positive candidates are selected as those with a score greater than a certain threshold t, then when $t = 0$ all candidates are classified as positive, and both the TPR and FPR will be equal to 1. On the other hand, if $t = 1$, no candidate is classified as positive, and the TPR and FPR will both equal 0. Therefore, the different values of t between 0 and 1, which yield different TPR and FPR values, will trace the ROC curve, going from the point (1,1) to (0,0) as t increases. The ROC *Area Under the Curve* (AUC), is used to measure the model's ability to discriminate between positive and negative candidates, for any given threshold. The ROC AUC ranges from 0 to 1. A random classifier has a ROC AUC of 0.5, while a perfect classifier has a ROC AUC of 1. The ROC AUC is a threshold-independent measure of the model's performance, and is often used to compare different models.

In a multiclass classification task, where more than two classes are available, a generalisation of the ROC curve and the ROC AUC metric is required. In this case, the *One-vs-One* ROC curve [26] can be defined as a plot of the TPR against the FPR for a given pair of classes. The One-vs-One ROC AUC can be averaged to the *macro-averaged* One-vs-One ROC AUC, which is the average of the ROC AUC for each pair of classes and can provide a measurement of the model's ability to discriminate between all the classes. For a classification problem with N classes, there are $\mathrm{N}(\mathrm{N} - 1)$ possible pairs of ordered classes, and therefore of One-vs-One ROC curves.

The One-vs-One ROC curves for the model trained on the $2 < p_{\mathrm{T}} < 3$ GeV/c interval are shown in Fig. 5.4. The ROC AUC is calculated for each class pair, and is reported in the legend. Given that the classification task is a three-class problem, the One-vs-One ROC AUC is calculated for the three possible pairs of classes. Furthermore, since the ROC curve for the A-vs-B pair differs from the

ROC curve of the B-vs-A pair when more than two classes are considered, since the positive and negative classes are swapped, two curves are shown for each pair of classes. This can also be deduced from the fact that the ROC AUC is not the same for the two curves of the same pair of classes. The metric is evaluated on both the training and test sets to test the model's generalisation power. The model's performance is excellent, with a macro-averaged One-vs-One ROC AUC value of 0.965, very close to that of an ideal classifier of 1. In addition, little overfitting is observed, as the ROC AUC values for the training and test sets are similar. The ROC AUC presents an almost flat trend with p_{T}, indicating that the model's performance is stable across the different p_{T} intervals. A slight increase in the ROC AUC of a few percent is observed for increasing p_{T}, although in the highest p_{T} region it decreases due to the limited number of candidates in the training sample, which can lead to a less performant model. The model is then used to select D_s^+- and D^+-meson candidates from the full dataset.

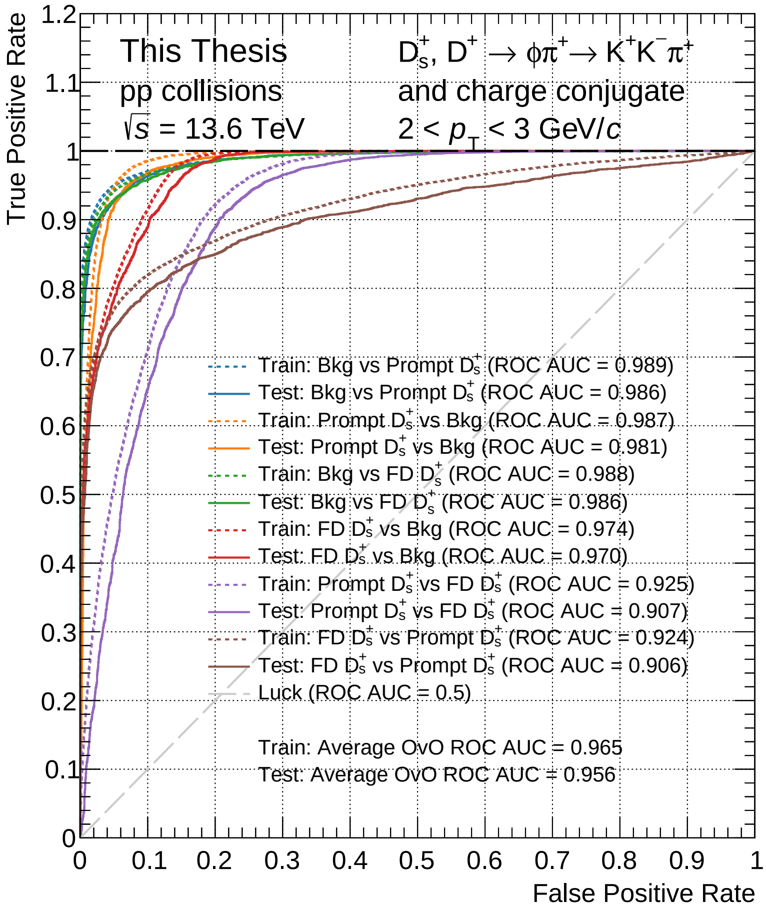


Figure 5.4: ROC curves for the model trained on the $2 < p_{\mathrm{T}} < 3$ GeV/c interval. The One-vs-One ROC AUC metric is calculated for each class pair and reported in the legend.

In addition to the ROC AUC, the model's performance can be evaluated by studying the distribution of the probability of belonging to a given class assigned to labelleled candidates. The score distributions for the model trained on the $2 < p_T < 3$ GeV/c interval are shown in the three panels of Fig. 5.5 for the background, prompt D_s^+-meson, and non-prompt D_s^+-meson classes. For each class, the score distribution is shown for candidates belonging to the different classes and to both the training and test sets. The distribution of the true background score provides interesting information on the model's performance. The score distribution for the background candidates peaks at high values, while the score distribution for the signal candidates (both prompt and non-prompt D_s^+ meson) peaks at low values. This highlights that the model has effectively learned to discriminate between signal and background candidates, with good separation power. Furthermore, the score distributions for the training and test sets are fairly similar, indicating that the model generalises well to unseen data. Since non-prompt D_s^+ mesons present a more displaced topology than promptly produced ones, due to the large lifetime of beauty-hadrons, the separation between non-prompt D_s^+ mesons and background is noticeable in the non-prompt D_s^+-meson score distribution, where the score of true background candidates peaks at zero and the one of non-prompt D_s^+-meson candidates peaks at one. The distribution of prompt D_s^+ mesons, which have a smaller displacement as compared to non-prompt ones falls in between those of background and non-prompt D_s^+ mesons. Lastly, the separation between the three classes is less pronounced in the prompt D_s^+-meson score distribution, where the prompt D_s^+-meson distribution peaks at values significantly lower than one. Similar trends are observed in the other studied p_T intervals.

5.2.5 Interpretation of the model's output: Feature importance

The usage of ML algorithms usually provides a better performance in terms of signal-to-background separation as compared to approaches based on "rectangular" cuts, but it also introduces a level of complexity in the selection process. One of the most difficult aspects of using ML models is the interpretation of their output. To understand how the model makes its decisions, the feature importance can be studied. This allows the understanding of which features are more important for the model's decision-making process, and the optimisation of the feature selection. In addition, the feature importance can be used to check whether the model is learning on the correct features in terms of the physics of the problem.

One of the most used algorithms for feature importance studies is the SHapley Additive exPlanations [27] (SHAP) algorithm. SHAP is a game-theoretic approach to explain the output of any machine learning model. It is based on the Shapley value [28] from cooperative game theory, which requires retraining the model on all feature subsets $\mathcal{S} \subseteq \mathcal{F}$, where $\mathcal{F}$ is the set of all features. An importance value is assigned to each feature, representing the effect on the model prediction of including that feature. To compute this effect, a model $\widehat{f}_{\mathcal{S}\cup\{i\}}$ is trained with that feature present, and another model $\widehat{f}_{\mathcal{S}}$ is trained with the feature withheld. Then, predictions from the two models are compared on the current input $\widehat{f}_{\mathcal{S}\cup\{i\}}(x) - \widehat{f}_{\mathcal{S}}$. The

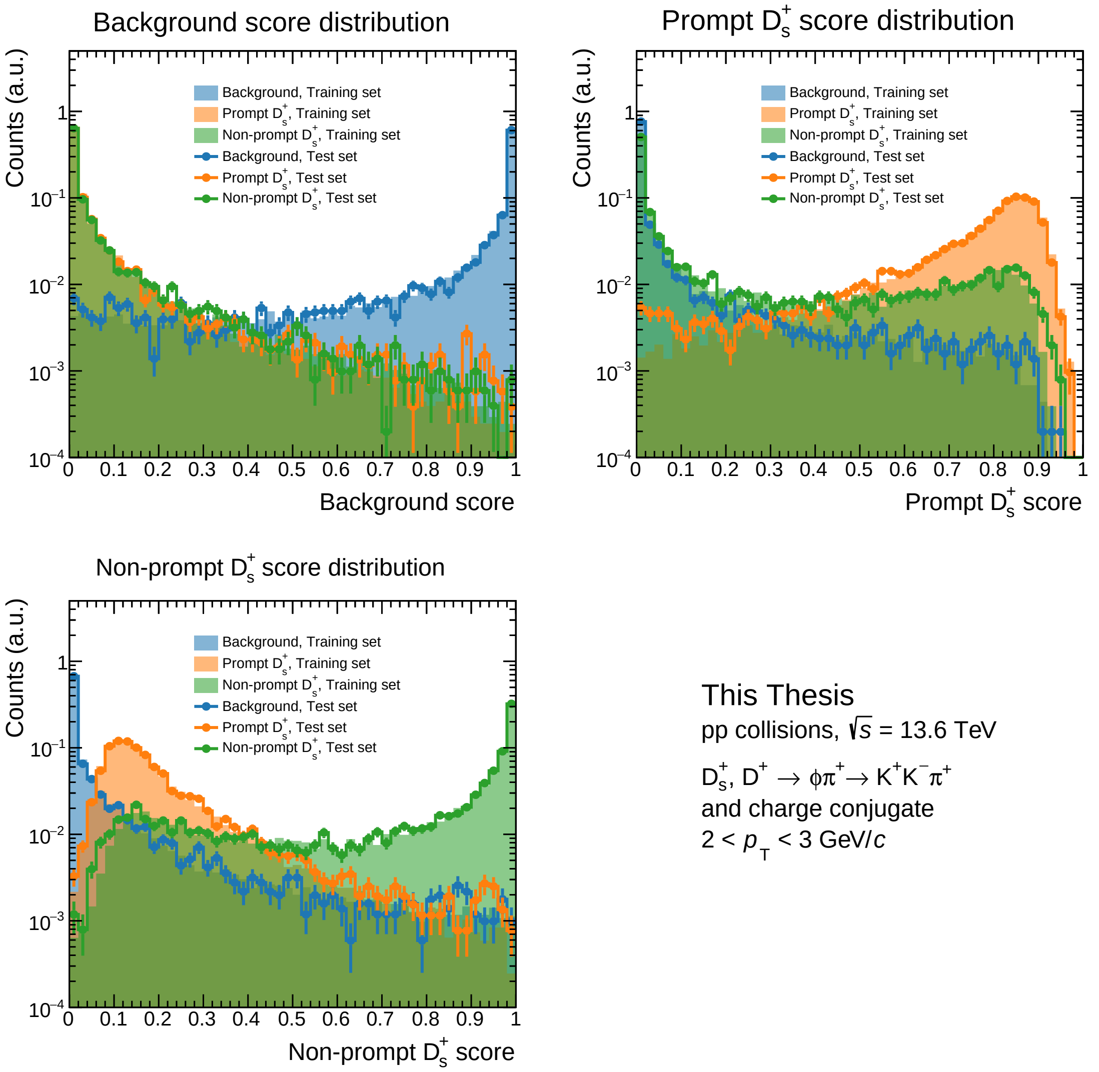


Figure 5.5: Score distributions for the model trained on the $2 < p_{\mathrm{T}} < 3$ GeV/c interval. The score distributions related to the probability of belonging to the background (top-left panel), prompt D_s^+-meson (top-right panel), and non-prompt D_s^+-meson (bottom-left panel) classes are shown. For each class, the score distribution is shown for candidates belonging to the different classes and to both the training (filled area) and test sets (markers).

Shapley values are then computed as the weighted average of all possible differences:

$$\phi_{\mathrm{i}} = \sum_{\mathcal{S}\subseteq\mathcal{F}\setminus\{i\}} \frac{|\mathcal{S}|!(|\mathcal{F}|-|\mathcal{S}|-1)!}{|\mathcal{F}|!}\left[\widehat{f}_{\mathcal{S}\cup\{i\}}(x) - \widehat{f}_{\mathcal{S}}(x)\right] \quad .$$

Since most models cannot handle arbitrary patterns of missing input values, $\widehat{f}(z_{\mathcal{S}})$ is approximated with $E[\widehat{f}(z)|z_{\mathcal{S}}]$, where $z_{\mathcal{S}}$ is the input missing the features in $\mathcal{S}$. SHAP values therefore explain how to get from the base value $E[\widehat{f}(z)]$, which would be predicted if no features were known, to the output $\widehat{f}(x)$.

A beeswarm-style SHAP feature importance plot for the prompt $\mathrm{D_s^+}$-meson probability predicted by the model trained in the $2 < p_{\mathrm{T}} < 3$ GeV/c interval is shown in Fig. 5.6. In this plot, each instance is represented by a single dot on each feature row. The position of the dot along the horizontal axis is determined by the SHAP value of that feature, and the dots "pile up" along each feature row, to provide information on the distribution of the SHAP values. The colour of the dot represents the value of the feature, with blue indicating a low value and red a high value. Positive values indicate that the feature is pushing the model's prediction towards the prompt $\mathrm{D_s^+}$-meson class, while negative values indicate that the model is less likely to classify the candidate as a prompt $\mathrm{D_s^+}$ meson. Feature rows are ordered based on the mean absolute value of the SHAP values for that feature, with the most important features, i.e., those with the highest impact on the model's decision, at the top. The most important features are the cosine of pointing angle $\cos\theta_{\mathrm{p}}$, the decay length L, the decay length in the XY plane L^{xy}, the absolute value of the cosine cubed of the K-π angle in the KK rest frame $|\cos^3\theta'(\mathrm{K})|$, and the PID information on the prong 1. As discussed in Chapter **??**, the first three features are related to the displaced decay topology of $\mathrm{D_s^+}$ mesons, and are therefore expected to be the most important variables in the model decisions. It is also expected that the prong 1 PID information resulted as the most important PID variable, as this is the opposite sign track, which is always a kaon in the considered decay channel. On the contrary, prongs 0 and 2 could be either kaons or pions, resulting in a lower importance of the PID information for these prongs. This check provides a good indication that the model is making decisions based on features that are expected to be relevant for the analysed physics process.

For each p_{T} interval, the SHAP values for the different features employed in the model can be evaluated for the three classes. The overall feature impact on the model's decision can be evaluated by computing the average absolute SHAP value for each feature. The average |SHAP value| feature importance for the BDT models trained on the $2 < p_{\mathrm{T}} < 3$ GeV/c and $8 < p_{\mathrm{T}} < 12$ GeV/c intervals are shown in Fig. 5.7, providing insights on the evolution of the feature importance across the different p_{T} intervals. The features are ordered based on their importance, evaluated as the sum of the mean absolute SHAP values for the three classes. Consistently with Fig. 5.6, in the $2 < p_{\mathrm{T}} < 3$ GeV/c interval the overall most impactful features are the cosine of pointing angle $\cos\theta_{\mathrm{p}}$, the decay length L and the PID information on the prong 1. Differences in the feature importance order are due to the different importance for the classes other than the prompt $\mathrm{D_s^+}$ meson reported in Fig. 5.6. In the $8 < p_{\mathrm{T}} < 12$ GeV/c interval, the decay length L and its projection on the

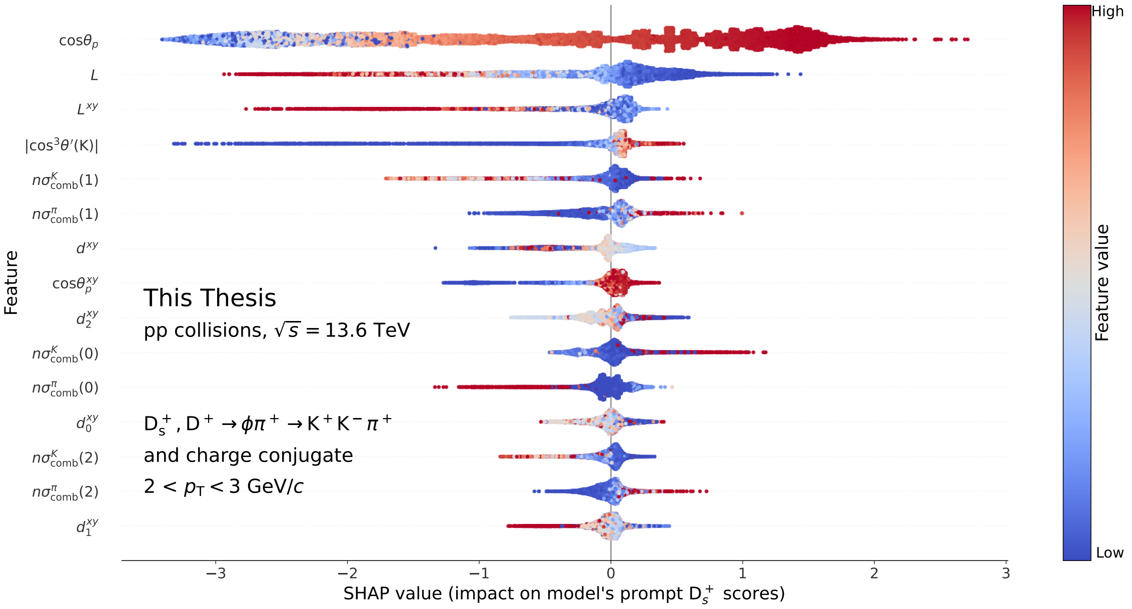


Figure 5.6: Beeswarm-style SHAP feature importance plot for the XGBoost model trained on the $2 < p_T < 3$ GeV/c interval.

transverse plane acquire a significantly higher importance. This is due to the fact that at higher p_T, the decay length of the D_s^+ mesons is larger owing to the higher Lorentz boost, and therefore the selection of both prompt and non-prompt based on their decay length becomes more effective.

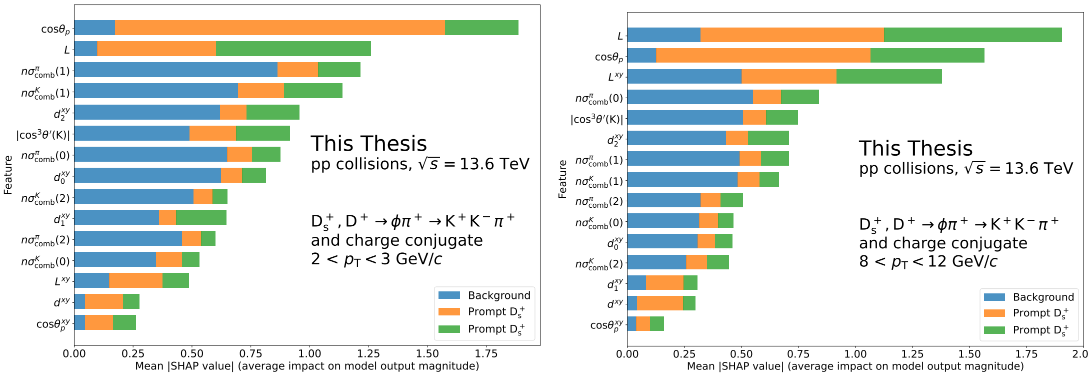


Figure 5.7: Average |SHAP value| feature importance for the XGBoost model trained on the $2 < p_T < 3$ GeV/c (left panel) and $8 < p_T < 12$ GeV/c (right panel) intervals. The feature importance for the background, prompt D_s^+-meson, and non-prompt D_s^+-meson classes are shown with blue, orange, and green bars, respectively.

5.2.6 Optimisation of the selection thresholds on the BDT scores

Once the model performance has been validated, a set of selection criteria on the BDT output scores has to be chosen to select the candidates. This is a crucial step of the analysis, as it will define the signal selection efficiency and the background contamination. Since the model's output consists of a score related to the probability of belonging to each class, and the three probabilities sum up to unity, the selection criteria have a total of two degrees of freedom. A first selection is applied on the maximum probability to be a background candidate, and rejects most of the contamination from the combinatorial background. The second one is applied on the minimum probability of being a prompt $\mathrm{D_s^+}$-meson candidate, and suppresses the signal contribution arising from non-prompt $\mathrm{D_s^+}$-meson candidates.

A first indication of the optimal selection criteria can be obtained by studying the model's output distributions. A good working point can be chosen as the point where a good separation between the three classes is achieved. For this analysis, however, the statistical significance of the extracted inclusive signal (i.e. of both prompt and non-prompt D mesons) is used to define the optimal selection criteria. For each p_T interval of the analysis, the signal S and the background B are evaluated by fitting the invariant mass distribution of candidates from the data passing the different ML selections considered in the optimisation process. Only a subset of the full dataset, corresponding to approximately 3% of the available data sample is used in this process. The working point is chosen using different considerations. Firstly, the selection criteria are chosen to maximise the statistical significance of the signal, defined as $S/\sqrt{S+B}$. This definition of statistical significance is equivalent to the inverse of the relative statistical uncertainty on the signal, for a Poisson distribution of candidate counts. Thus, maximising the statistical significance corresponds to minimising the relative statistical uncertainty on the signal. In order to avoid an over-optimisation of the model to enhance the statistical significance (which could bias the results if statistical fluctuations are not properly taken into account), the considered threshold values on the BDT output scores are multiples of 0.05. Additionally, the efficiency of the selection is checked to ensure that it is kept at sufficiently high levels, to reduce possible biases in the final results due to possible imperfections in the MC description of the data. A smooth increasing trend with p_T of the efficiency is also ensured to provide consistency in the tightness of the selection criteria across the different p_T intervals.

Usually, the maximisation of the statistical significance is avoided in the choice of the selection criteria, as it can lead to a bias in the final results. However, in this case, the optimisation is performed on a very small fraction of the data thanks to the large dataset available, and the bias is expected to be negligible. A different approach to select an optimal working point consists in the optimisation of a pseudo-significance, defined as the ratio of the expected signal and the square root of the sum of the expected signal and the expected background. For each considered selection criterion, the selection efficiency is evaluated on MC simulations, and the expected signal is estimated from the D-meson production cross section provided by FONLL [29] theoretical predictions. The expected background is estimated from a fit of the invariant mass distribution of the candidates in the sidebands of the signal region. This approach avoids the introduction of a bias in the final results, as it is blind to the candidates in the signal region. However, it relies on FONLL predictions and MC simulations. To avoid the introduction of biases in the optimisation due to shortcomings of FONLL predictions or MC simulations, the optimisation of the

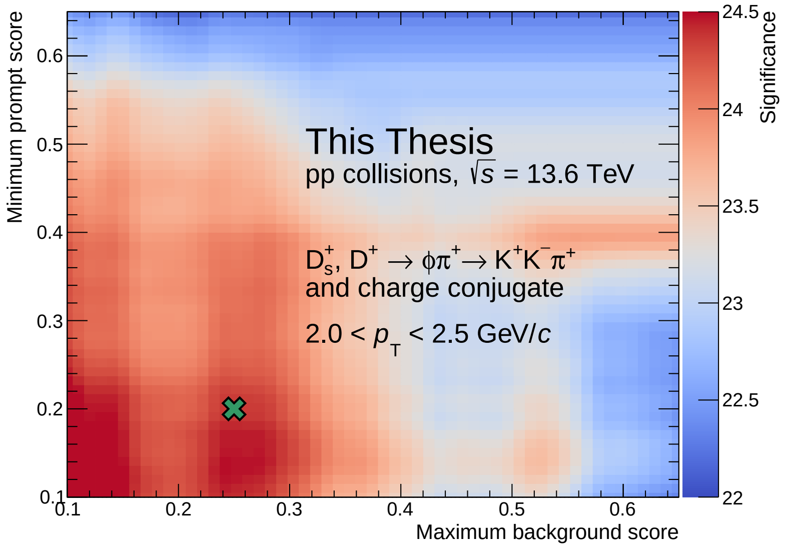


Figure 5.8: Statistical significance of the signal as a function of the selection criteria applied to the model output for the $2.0 < p_T < 2.5$ GeV/c interval. The chosen set of selection criteria is shown with a green cross.

pseudo-significance is not performed in this analysis.

A result of the working point optimisation is presented in Fig. 5.8 for the $2.0 < p_T < 2.5$ GeV/c interval. The statistical significance of the signal is shown as a function of the BDT output score threshold for the probability of being a prompt D_s^+ meson and a background candidate. The chosen set of selection criteria is shown with a green cross.

The optimal selection criteria for each p_T interval considered in the analysis are reported in Table 5.5. Because of the large combinatorial background and the small Lorentz boost of the D_s^+ mesons, which results in a small decay length and a less effective selection based on the displaced topology, the selection criteria in the lowest p_T intervals are chosen to be stringent to extract the signal with a large enough signal-to-background ratio. At higher p_T, the selection criteria are loosened, as the larger decay length of the D_s^+ mesons results in a more effective separation between signal and background candidates. In addition, at higher p_T the combinatorial background is reduced, and the selection criteria can be loosened to increase the signal efficiency. The threshold on the prompt D_s^+-meson probability is kept at around 0.2 for all p_T intervals. This selection allows a significant reduction of the contamination from non-prompt D mesons in the selected sample, while it does not considerably influence the statistical significance of the signal. In the $5.5 < p_T < 6.0$ GeV/c interval, the threshold is increased to 0.25 to guarantee an increasing trend of the efficiency with p_T.

5.3 D_s^+ and D^+ raw-yield extraction

After the working point for the BDT algorithm has been defined, the raw yields of D_s^+ and D^+ mesons are extracted in each p_T interval. They are defined as the sum of particles and antiparticles and are measured in 14 p_T intervals in the range $0.5 < p_T < 24$ GeV/c. The raw yield is extracted by fitting the invariant mass distribution of the selected candidates.

In several analyses performed by the ALICE Collaboration during the Run 2 data-taking period [30, 31, 32], the raw yield of D_s^+ mesons was extracted by fitting the invariant mass distribution of selected candidates with a probability density function constructed as the sum of a function describing the shape of the combinatorial background (usually an exponential function or a low-order (< 3) polynomial) and of two Gaussian distributions to model the D_s^+- and D^+-meson peaks. The raw yields for the two D-meson species are then obtained by integrating the signal function.

Figure 5.9 shows the fit to the invariant mass distribution of the selected candidates in the $1.5 < p_T < 2.0$ GeV/c interval using the approach described above. Due to the concavity-changing shape of the background, the function chosen to describe the background is a third-order polynomial. This change in the concavity of the background invariant-mass distribution was not observed in previous analyses performed on pp collisions data collected during the LHC Run 2 data-taking period. With the upgrade of the ALICE experimental apparatus, the amount of data collected during the ongoing LHC Run 3 data-taking period is significantly larger than that collected during Run 2, as described in Chapter **??**. With the increased number of candidates available, the statistical precision of the data sample is higher, and the concavity-changing shape of the background becomes more evident. The small number of candidates available in the Run 2 data sample did not allow for the observation of this feature, as statistical uncertainties covered the shape of the background. On top of the previously-unobserved peculiar shape of the background, the fitting function is not able to describe the data accurately between the two peaks, overestimating the data in this invariant mass region.

These two features can be understood as due to the fact that the background does

Table 5.5: Selection criteria applied to enhance the significance of the D_s^+ meson signal in the p_T intervals considered for the analysis.

p_T interval (GeV/c)	Probability to be background <	Probability to be prompt D_s^+ meson >
0.5–1.0	0.01	0.20
1.0–1.5	0.05	0.20
1.5–2.0	0.15	0.20
2.0–2.5	0.25	0.20
2.5–3.0	0.3	0.20
3.0–3.5	0.2	0.20
3.5–4.0	0.2	0.20
4.0–4.5	0.2	0.20
4.5–5.0	0.2	0.20
5.0–5.5	0.3	0.20
5.5–6.0	0.3	0.25
6–8	0.45	0.20
8–12	0.5	0.20
12–24	0.55	0.20

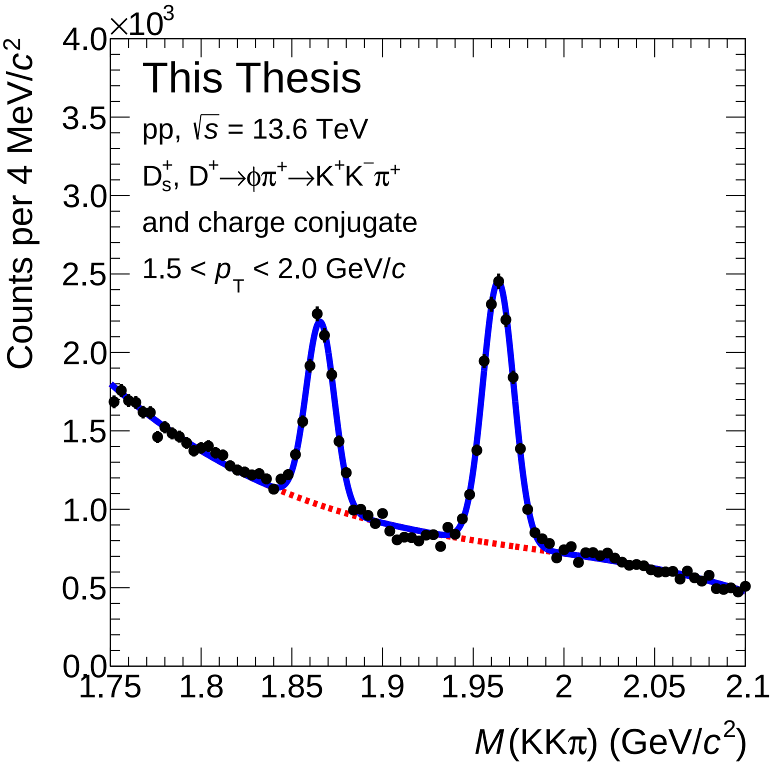


Figure 5.9: Fit to the invariant mass distribution of selected candidates in the $1.5 < p_{\mathrm{T}} < 2.0$ GeV/c interval. The fit function is shown as a solid line, while the signal and background components are shown as dashed lines. The background is modelled with a third-order polynomial function.

not solely arise from the combination of independent tracks, i.e. the combinatorial background. Other physics processes can contribute to the contamination of the data sample, giving rise to a *correlated background.* One such process is the decay of D^+ mesons into the $D^+ \rightarrow \pi^+K^-\pi^+$ decay channel, where one of the pions is misidentified as a kaon. Despite being suppressed by the applied PID and ML selections, this contribution can give rise to a noticeable contribution due to the large BR of 9.38% [33].

To validate this hypothesis, a simulation of D^+-meson decays into $D^+ \rightarrow \pi^+K^-\pi^+$ was run using PYTHIA 8 [11]. Ten billion D^+ mesons were produced with a uniformly distributed p_{T} spectrum in the $0 < p_{\mathrm{T}} < 24$ GeV/c interval. The D^+ mesons were then forced to decay into the $D^+ \rightarrow \pi^+K^-\pi^+$ decay channel. For each decay, two invariant masses were evaluated, by assigning the kaon mass to one of the two pions to simulate the misidentification of one pion, while the correct masses were assigned to the other two prongs. The invariant mass distributions obtained from the simulation are shown in Fig. 5.10 for different p_{T} intervals. The distribution is characterised by a peak at ~ 2 GeV/c^2, well inside the fit range used to extract the raw yield of D_s^+ and D^+ mesons in Fig 5.9. In addition, the invariant mass distribution evolves with the p_{T} of the D^+ meson, with a tail towards higher invariant masses for higher p_{T} values. This follows naturally from the kinematic properties of the decay.

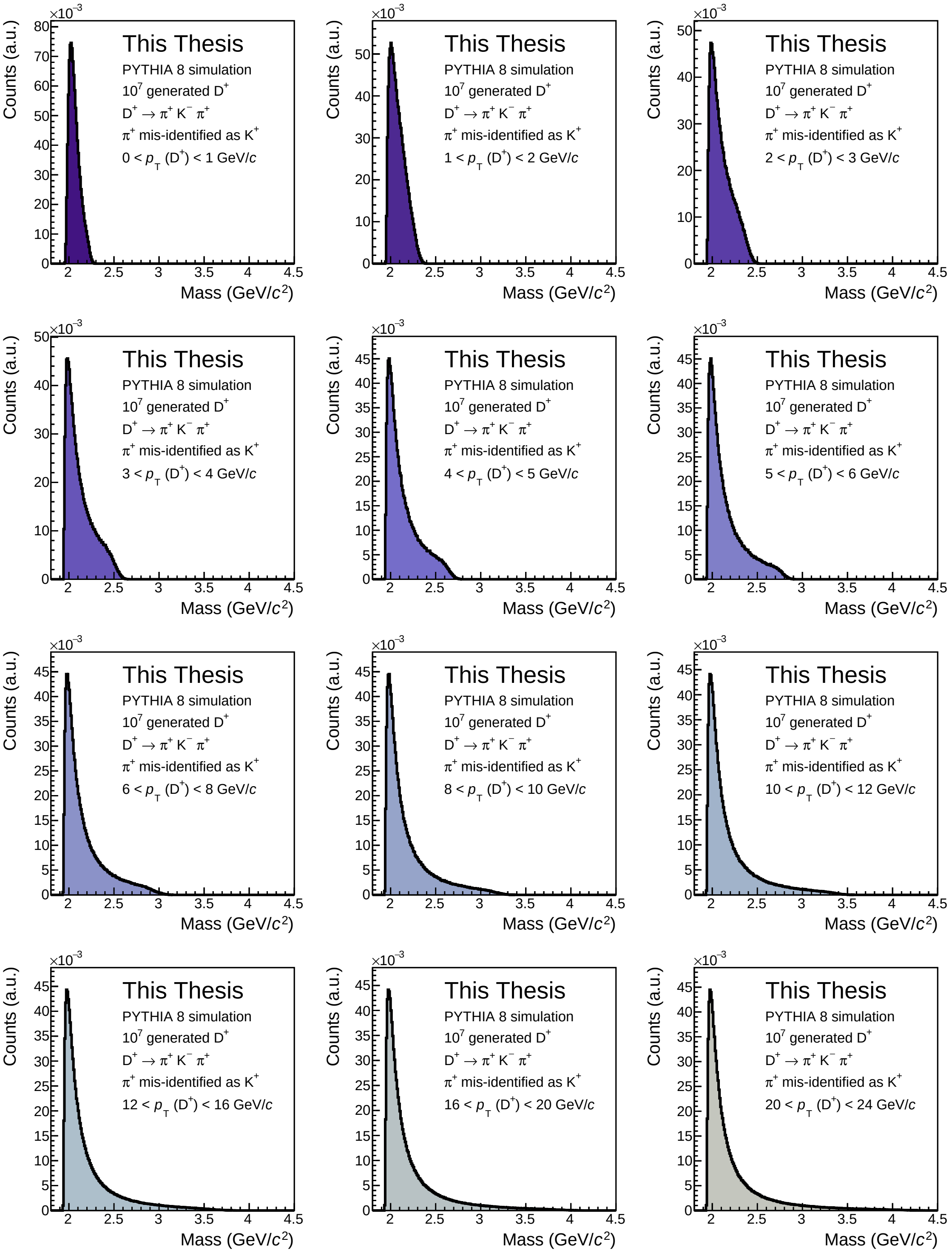


Figure 5.10: Invariant-mass distribution of simulated decays of D^+ mesons into $D^+ \to \pi^+K^-\pi^+$, where one of the pions produced in the decay is misidentified as a kaon. Distributions for different p_T intervals are shown.

To account for this contribution, a template contribution is included in the fit function, in addition to the combinatorial background and the two signal peaks. This method involves fitting the invariant mass distribution using a predefined template to model the misidentified $D^+ \to \pi^+K^-\pi^+$ decay, whose shape is fixed, and the only adjustable parameter is the normalisation. The shape of the template is taken from the same simulation used to train the BDT model, as described in Sec. 5.2.1, where $D^+ \to \pi^+K^-\pi^+$ mesons reconstructed as $D^+ \to K^+K^-\pi^+$, passing PID selections as $K^+K^-\pi^+$, are selected. The distribution is fixed to that obtained before applying any ML models, as it was studied that the applied ML selections do not affect the shape of the correlated background. This allows for reducing statistical fluctuations in the template shape. The final fit function is then constructed as the sum of a parabolic function to describe the combinatorial background, the template function described above for the correlated $D^+ \to \pi^+K^-\pi^+$ background source, and two Gaussian functions to describe the D_s^+- and D^+-meson peaks. The signal parameters (mean, width, normalisation), as well as the parameters of the combinatorial background and the normalisation of the template fit are left free in the fit in the p_T intervals below 8 GeV/c. Because of the observed momentum resolution worsening at higher p_T, and because of the smaller number of candidates available in the high-p_T intervals, the width of the D^+-meson signal peak is fixed to that of the D_s^+ meson divided by a factor 1.2, which is the observed ratio of the peak widths at low p_T, with a flat trend with p_T, as illustrated in Fig. 5.11. In the left panel, the evolution of the peak widths of the D_s^+ and D^+ mesons as a function of p_T is shown for both data and MC simulations. A clear increasing trend of the peak widths with p_T is observed because of the degrading momentum resolution due to the decreasing track curvature. MC simulations underestimate the peak widths extracted from the data for both D_s^+ and D^+ mesons, across the studied p_T range. In the right panel of Fig. 5.11, the ratio between the peak widths of the D_s^+ and D^+ mesons is shown as a function of p_T, for both data and MC simulations. The ratio is observed to be almost constant across the analysed p_T range, with a value of ~ 1.2. The ratio is underestimated by the MC simulations, which predict a value of ~ 1.1 across the studied p_T range.

In the p_T interval above 8 GeV/c, where the peak width of the D^+ meson is fixed to that of the D_s^+ meson divided by a factor of 1.2, a first fit is performed by keeping both peak widths as free parameters. Then, the D^+-meson peak width is fixed to that of the D_s^+ divided by 1.2, and the fit is repeated, keeping the remaining parameters free. Additionally, in order to correctly describe the background shape, the fit is performed in the invariant mass range $1.73 < M(KK\pi) < 2.15$ GeV/c^2 for $p_T > 6$ GeV/c, while a narrower invariant mass window, $1.75 < M(KK\pi) < 2.1$ GeV/c^2, is used at lower p_T. The invariant-mass bin width has been fixed to 2 MeV/c^2. The fits to the invariant mass distributions of selected D-meson candidates are performed using the `flarefly` package [34], which provides a flexible Python interface for performing fits.

The fit to the invariant mass distribution of candidates passing the ML selections is shown in Fig. 5.12 for the two representative p_T intervals of $2.0 < p_T < 2.5$ GeV/c in the left panel, and $5.0 < p_T < 5.5$ GeV/c in the right panel. The total fit function is shown as a solid blue line, the signal contributions are shown as filled green and

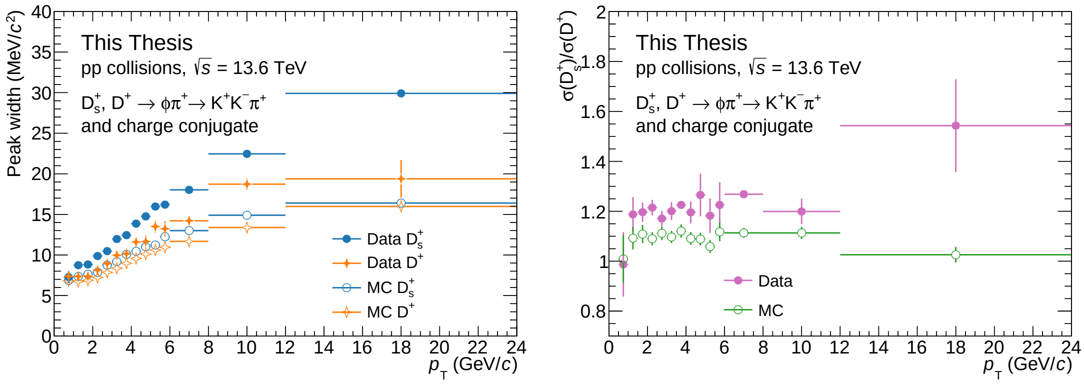


Figure 5.11: Left panel: peak width of the $\mathrm{D_s^+}$ and $\mathrm{D^+}$ mesons as a function of p_T, before fixing the width of the $\mathrm{D^+}$ meson to that of the $\mathrm{D_s^+}$ meson divided by a factor of 1.2 for $p_\mathrm{T} > 8$ GeV/c. Both the peak widths from fits to data and MC simulations are reported. Right panel: ratio of the peak widths of the $\mathrm{D_s^+}$ and $\mathrm{D^+}$ mesons as a function of p_T, for both data and MC simulations.

azure areas for $\mathrm{D^+}$ and $\mathrm{D_s^+}$ mesons, respectively, the combinatorial background is represented with a solid red line, while the correlated background is shown as a dashed violet line. The fit is able to describe the data accurately, as can be deduced from the distribution of the difference between the data and the background fit function, shown in the bottom panels for the two p_T bins. Figure 5.12 also shows that the contribution of the correlated $\mathrm{D^+} \to \pi^+\mathrm{K}^-\pi^+$ background evolves with p_T, with a larger contribution at lower p_T values. The raw yield of $\mathrm{D_s^+}$ and $\mathrm{D^+}$ mesons is then extracted by integrating the signal function.

The evolution of the mean and width of the $\mathrm{D_s^+}$ and $\mathrm{D^+}$ meson peaks as a function of p_T is shown in Fig. 5.13. In the left panel, the means of the Gaussian functions used to describe the $\mathrm{D_s^+}$- and $\mathrm{D^+}$-meson peaks are shown as a function of p_T. They present an increasing trend with p_T lying below the values reported in the PDG [33], represented with dashed lines. The shift of the mass values observed in data from those reported in the PDG is common to all analyses of D mesons.why? In the right panel, the peak widths of the $\mathrm{D_s^+}$ and $\mathrm{D^+}$ mesons, defined as the standard deviation of the Gaussian functions, are shown as a function of p_T. An increasing trend with p_T is observed for both mesons, as detailed above for Fig. 5.11, with the peak width of the $\mathrm{D_s^+}$ meson being larger than that of the $\mathrm{D^+}$ meson across the studied p_T range.

The relative contribution of the $\mathrm{D^+} \to \pi^+\mathrm{K}^-\pi^+$ correlated background to the total fitting function is shown in the left panel of Fig. 5.14 as a function of p_T. At low p_T, where the PID is more effective, the contribution of the correlated background is observed to be smaller. It then increases up to 2 GeV/c and then decreases due to the kinematic properties of the decay as expected from Fig. 5.10. Some residual contribution is observed at higher p_T, related to fluctuations in the fitting procedure. In the right panel of Fig. 5.14, the statistical significance of the extracted $\mathrm{D_s^+}$- and $\mathrm{D^+}$-meson signals is shown as a function of p_T. Due to the large combinatorial background at low p_T, the statistical significance presents a decreasing trend with decreasing p_T for $p_\mathrm{T} \lesssim 2$ GeV/c, for both D mesons. The maximum statistical significance is observed in the $2.5 < p_\mathrm{T} < 3.0$ GeV/c interval for the $\mathrm{D_s^+}$ meson and

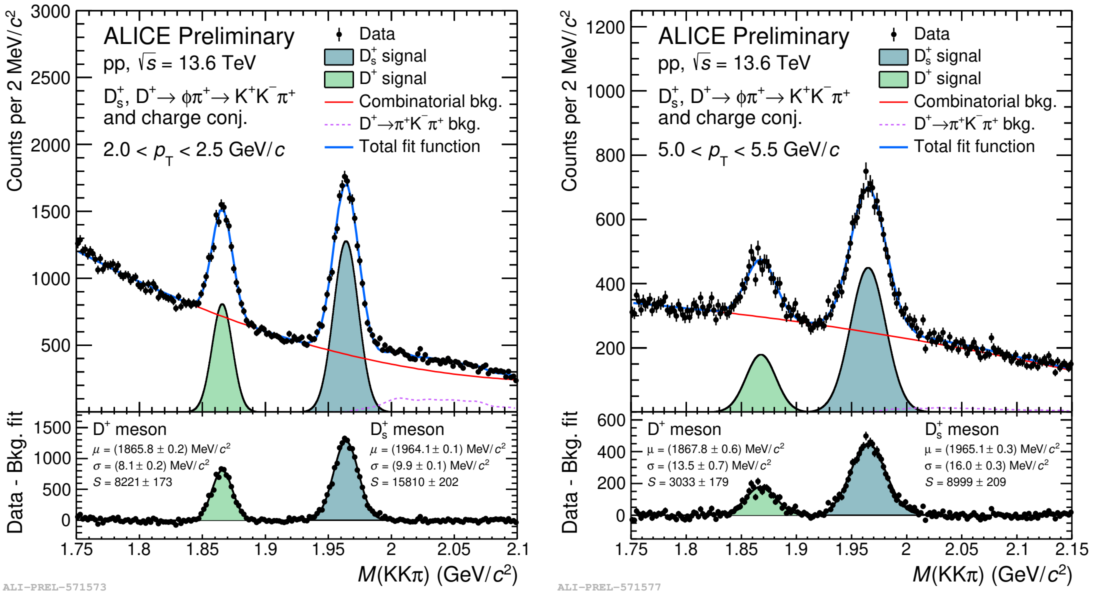


Figure 5.12: Fit to the invariant mass distribution of selected candidate in the $2.0 < p_T < 2.5$ GeV/c (left) and $5.0 < p_T < 5.5$ GeV/c (right) intervals. The total fit function is shown as a solid blue line, while background components are shown as solid red lines (correlated background) and dashed violet lines (dashed violet lines). The signal contributions are shown as filled green and azure areas for D^+ and D_s^+ mesons, respectively. The bottom panels show the distribution of the difference between the data and the background fit function.

in the $2.0 < p_T < 2.5$ GeV/c range for the D^+ meson, where significances of 99 and 50 are reached for the two meson species, respectively. At higher p_T, the statistical significance decreases due to the smaller number of produced D mesons, and the larger peak widths of the D_s^+ and D^+ mesons, which result in a larger background contribution in the signal region. A sudden increase in the statistical significance is observed in the $6 < p_T < 8$ GeV/c interval, and is due to an increase of the p_T interval for the signal extraction, from 0.5 GeV/c to 2 GeV/c, which results in a larger number of candidates available in the signal region.

The extraction of the raw yields is affected by several arbitrary choices, for example, the functional description of the background, the choice of the fit range, the choice of fixing the D^+ width at high p_T, and the invariant-mass bin width. Changes in these choices can lead to variations in the extracted raw yields. To estimate the effect of such arbitrary choices in the final observable (the D_s^+/D^+ production-yield ratio), and estimate a systematic uncertainty associated with the raw yield extraction procedure, the fit is repeated several times by varying the fit range, the bin width, and the functional form of the background. In the low p_T region ($p_T < 4$ GeV/c), the lower fit limit is varied between 1.71 and 1.77 GeV/c^2, while the upper limit is varied between 2.08 and 2.14 GeV/c^2. The functions considered to describe the background are a second-order polynomial and an exponential function. At higher p_T (up to 8 GeV/c), where the signal peaks become broader, the lower limit is varied between 1.71 and 1.75 GeV/c^2, while the maximum mass is varied

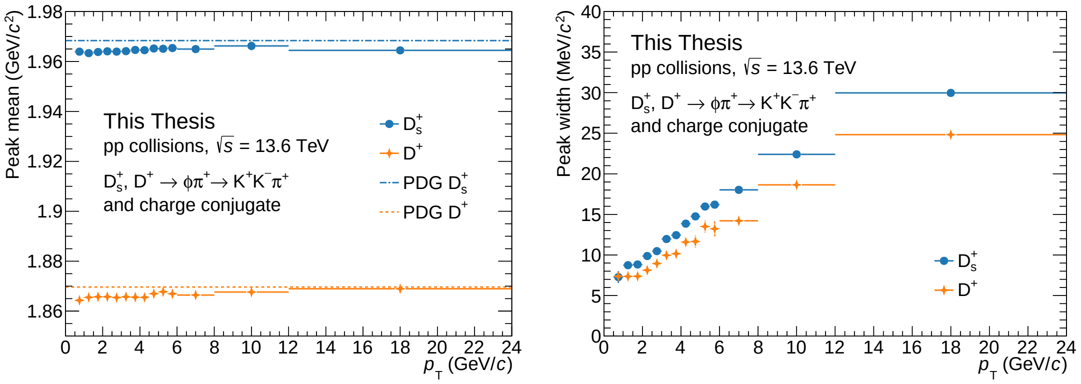


Figure 5.13: Evolution of the mean (left panel) and width (right panel) of the Gaussian functions used to describe the D_s^+ and D^+ meson peaks as a function of p_T. The dashed lines in the left panel represent the mass values reported in the PDG [33] for the two D-meson species.

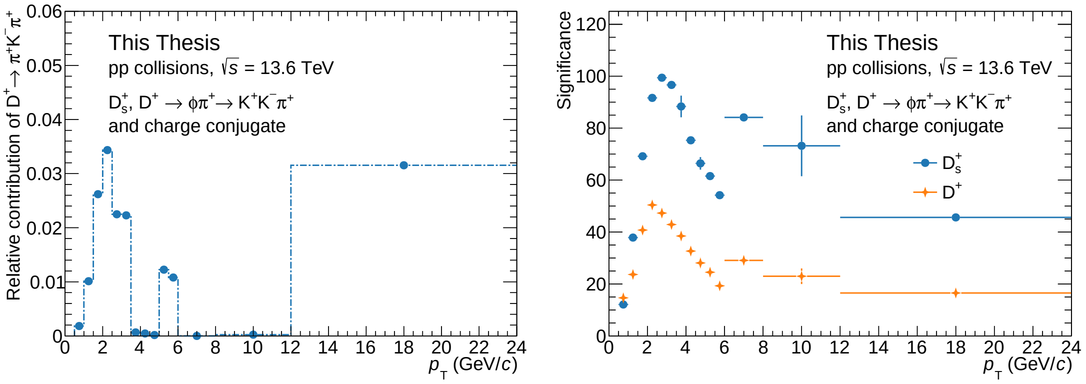


Figure 5.14: Evolution of the $D^+ \to \pi^+K^-\pi^+$ relative contribution to the fitting function (left panel) and statistical significance of the extracted D_s^+- and D^+-meson signal (right panel) as a function of p_T.

between 2.13 and 2.19 GeV/c^2. The bin width is varied between 1 and 4 MeV/c^2 across the studied p_T interval. At $p_T > 8$ GeV/c^2, where the D^+-meson peak width is fixed to that of the D_s^+-meson divided by 1.2, the peak width of the D^+ meson is changed by varying the dividing factor by ±10%. For each possible combination of these variations, the signal is extracted for the two D-meson species, and the ratio between them is calculated. This results in 128, 96, and 288 trials for the $0.5 < p_T < 4$ GeV/c, $4 < p_T < 8$ GeV/c, and $8 < p_T < 24$ GeV/c intervals, respectively. As a quality check, trials with a χ^2/ndf greater than 2 are discarded. The distribution of the D_s^+/D^+ raw-yield ratio is then obtained for the different trials, and the mean and standard deviation are calculated. The difference Δ between the mean of this distribution and the ratio extracted with the default configuration is calculated. The systematic uncertainty is then evaluated as the sum in quadrature of the standard deviation and Δ: $\sqrt{\text{RMS}^2 + \Delta^2}$.

The result of this multi-trial approach for the evaluation of the systematic on the raw-yields extraction is shown in Fig. 5.15 for the $1.0 < p_T < 1.5$ GeV/c interval.

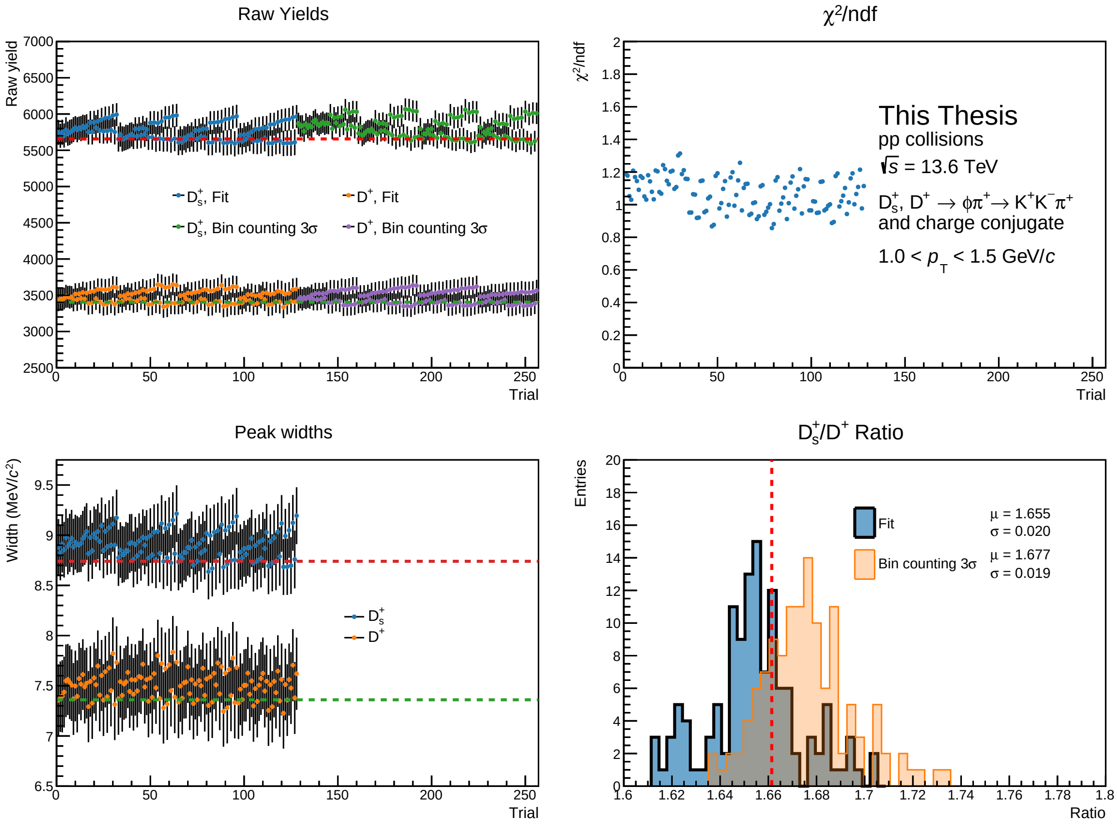


Figure 5.15: Results from the multi-trial approach employed for estimating the systematic uncertainty related to the raw yield extraction in the $1.0 < p_T < 1.5$ GeV/c interval.

In the top-left panel, the raw yields extracted from the fit to the invariant mass distribution are reported for the D_s^+ and D^+ mesons for the different trials. As a cross-check, the raw yields are also extracted by summing the counts of a distribution obtained by subtracting the background fit function (composed of a second-order polynomial function for the combinatorial background and of a template function for the $D^+ \rightarrow \pi^+K^-\pi^+$ correlated background) from the invariant mass distribution of the candidates passing the ML selections. The bin contents are summed within 3 standard deviations from the peak mean. The extracted raw yields are stable within uncertainty across all the trials for the D^+ meson. For the D_s^+ meson, the 3σ bin counting method presents higher raw yields when compared to the ones obtained by integrating the signal function. However, this is considered as related to fluctuations in the invariant mass distribution rather than a systematic shift of the raw yields due to a different definition of the observable. The raw yields extracted with the default configuration described above are shown as dashed lines. In the top-right panel, the χ^2/ndf of the fit to the invariant mass distribution is shown for the different trials. Only trials where the signal is extracted by integration of the fitted signal function are shown in this panel. These results illustrate that the fit function is able to describe the data accurately, independently of the configuration of the fit parameters. In the bottom-left panel, the peak widths for D_s^+ and D^+ mesons are reported, and present a stable behaviour across the different trials. The peak widths for the default configuration are also reported as dashed lines. The bottom-right panel shows the distribution of the D_s^+/D^+ raw-yield ratio for the different trials, obtained using the two methods previously introduced (namely, integration of the signal fit function and bin counting). The 3σ bin counting method presents a higher raw-yield ratio in this p_T interval when compared to the other method, because of the higher D_s^+-meson yields. The systematic uncertainty is defined as the sum in quadrature of the standard deviation of the distribution of D_s^+/D^+ raw-yield ratio obtained by integrating the signal function and the difference Δ between the default value reported with a dashed red line and the mean of this distribution. This quantity ranges from 1% to 10% of the central D_s^+/D^+ raw-yield ratio, depending on the considered p_T interval. The systematic uncertainty is then assigned after smoothing the p_T dependence of the obtained values. The assigned systematic uncertainty is reported in Table 5.6.

Table 5.6: Systematic uncertainty on the raw-yield extraction for the ratio of the D_s^+ and D^+ mesons yields.

p_T (GeV/c)	$\sqrt{\text{RMS}^2 + \Delta^2}$/(central D_s^+/D^+) (%)	Assigned systematic uncertainty (%)
0.5–1	4	3
1–1.5	1	3
1.5–2	2	3
2–2.5	3	3
2.5–3	3	3
3–3.5	3	3
3.5–4	3	3
4–4.5	6	5
4.5–5	7	5
5–5.5	3	5
5.5–6	5	5
6–8	8	8
8–12	9	9
12–24	10	10

Bibliography

[1] A. L. Samuel, “Some studies in machine learning using the game of checkers”, *IBM Journal of Research and Development* **3** (1959) 210–229.

[2] T. M. Mitchell, “Machine learning”, 1997.

[3] OpenAI, “GPT-4 Technical Report”, *arXiv e-prints* (Mar., 2023) arXiv:2303.08774, `arXiv:2303.08774 [cs.CL]`.

[4] **CMS** Collaboration, S. Chatrchyan *et al.*, “Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC”, *Phys. Lett. B* **716** (2012) 30–61, `arXiv:1207.7235 [hep-ex]`.

[5] A. Mao, M. Mohri, and Y. Zhong, “Cross-entropy loss functions: Theoretical analysis and applications”, in *International Conference on Machine Learning*, pp. 23803–23828, PMLR. 2023.

[6] J. Kiefer and J. Wolfowitz, “Stochastic Estimation of the Maximum of a Regression Function”, *The Annals of Mathematical Statistics* **23** (1952) 462 – 466. `https://doi.org/10.1214/aoms/1177729392`.

[7] P. I. Frazier, “A tutorial on bayesian optimization”, *arXiv preprint arXiv:1807.02811* (2018) .

[8] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms”, *Advances in neural information processing systems* **25** (2012) .

[9] J. Mockus, “The bayesian approach to global optimization”, in *System Modeling and Optimization: Proceedings of the 10th IFIP Conference New York City, USA, August 31–September 4, 1981*, pp. 473–481, Springer. 2005.

[10] M. Stone, “Cross-validatory choice and assessment of statistical predictions”, *Journal of the royal statistical society: Series B (Methodological)* **36** (1974) 111–133.

[11] C. Bierlich *et al.*, “A comprehensive guide to the physics and usage of PYTHIA 8.3”, *SciPost Phys. Codeb.* **2022** (2022) 8, `arXiv:2203.11601 [hep-ph]`.

[12] J. R. Christiansen and P. Z. Skands, “String Formation Beyond Leading Colour”, *JHEP* **08** (2015) 003, `arXiv:1505.01681 [hep-ph]`.

[13] **GEANT4** Collaboration, S. Agostinelli *et al.*, "GEANT4–a simulation toolkit", *Nucl. Instrum. Meth. A* **506** (2003) 250–303.

[14] J. H. Friedman, "Greedy function approximation: a gradient boosting machine", *Annals of statistics* (2001) 1189–1232.

[15] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of computer and system sciences* **55** (1997) 119–139.

[16] L. Breiman, *Classification and regression trees.* Routledge, 2017.

[17] J. R. Quinlan, "Induction of decision trees", *Machine learning* **1** (1986) 81–106.

[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", *CoRR* **abs/1603.02754** (2016) , `1603.02754`. `http://arxiv.org/abs/1603.02754`.

[19] B. Kegl, CecileGermain, ChallengeAdmin, ClaireAdam, D. Rousseau, Djabbz, fradav, G. Cowan, Isabelle, and joycenv, "Higgs boson machine learning challenge", 2014. `https://kaggle.com/competitions/higgs-boson`.

[20] R. Mitchell and E. Frank, "Accelerating the xgboost algorithm using gpu computing", *PeerJ Computer Science* **3** (2017) e127.

[21] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* " O'Reilly Media, Inc.", 2022.

[22] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial", *Frontiers in neurorobotics* **7** (2013) 21.

[23] XGBoost Documentation, "Xgboost parameters." `https://xgboost.readthedocs.io/en/stable/parameter.html`.

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework", in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631. 2019.

[25] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization", *Advances in neural information processing systems* **24** (2011) .

[26] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems", *Machine learning* **45** (2001) 171–186.

[27] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions", *Advances in neural information processing systems* **30** (2017) .

[28] S. Lipovetsky and M. Conklin, “Analysis of regression in game theory approach”, *Applied stochastic models in business and industry* **17** (2001) 319–330.

[29] M. Cacciari, M. Greco, and P. Nason, “The p_T spectrum in heavy-flavour hadroproduction.”, *JHEP* **05** (1998) 007, `arXiv:hep-ph/9803400`.

[30] **ALICE** Collaboration, S. Acharya *et al.*, “Measurement of beauty and charm production in pp collisions at $\sqrt{s} = 5.02$ TeV via non-prompt and prompt D mesons”, *JHEP* **05** (2021) 220, `arXiv:2102.13601 [nucl-ex]`.

[31] **ALICE** Collaboration, S. Acharya *et al.*, “Charm production and fragmentation fractions at midrapidity in pp collisions at $\sqrt{\mathrm{s}} = 13$ TeV”, *JHEP* **12** (2023) 086, `arXiv:2308.04877 [hep-ex]`.

[32] **ALICE** Collaboration, S. Acharya *et al.*, “Measurement of prompt $\mathrm{D_s^+}$-meson production and azimuthal anisotropy in Pb–Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV”, *Phys. Lett. B* **827** (2022) 136986, `arXiv:2110.10006 [nucl-ex]`.

[33] **Particle Data Group** Collaboration, R. L. Workman and Others, “Review of Particle Physics”, *PTEP* **2022** (2022) 083C01.

[34] F. Grosa, S. Politanò, and A. Bigot, “flarefly”, Jan., 2023. `https://doi.org/10.5281/zenodo.7579657`.