CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITECNICO NACIONAL

UNIT ZACATENCO
PHYSICS DEPARTMENT

"Measurement of the Branching Fraction of the $B^+ \to K^{*+}\mu^+\mu^-(K^{*+} \to K_s^0\pi^+)$ Decay at the CMS Experiment"

Thesis submitted by

Erick García Rodríguez

In order to obtain the

Master of Science

degree, speciality in

Physics

Supervisor: **Dr. Heriberto Castilla Valdez**

Mexico City November, 2024

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional

UNIDAD ZACATENCO
DEPARTAMENTO DE FÍSICA

"Medición de la "Branching Fraction" del Decaimiento $B^+ \to K^{*+}\mu^+\mu^-(K^{*+} \to K_s^0\pi^+)$ en el experimento CMS"

Tesis que presenta

Erick García Rodríguez

para obtener el Grado de

Maestro en Ciencias

en la Especialidad de

Física

Director de tesis: **Dr. Heriberto Castilla Valdez**

Ciudad de México Noviembre, 2024

Agradecimientos

Quiero agradecer al Doctor Heriberto Castilla Valdez por haberme permitido formar parte de su grupo de investigación, en el cual él junto a Horacio Crotte, Jhovanny Mejía y Rogelio Reyes resolvieron todas mis dudas y me proporcionaron las herramientas para poder desarrollar este trabajo. Gracias por el invaluable tiempo que me dedicaron.

A mi madre Erika y mis tías Araceli y Adriana, por siempre apoyarme en todas mis decisiones, de ellas he recibido el apoyo más grande para concluir mis objetivos.

A mis amigos del Cinvestav, Iván Pérez que desde ESFM me ha apoyado muchísimo, aligerando mi camino; Noe Tepec y Oscar Pérez, definitivamente el intercambio de ideas, conocimiento y experiencias que nos encontrábamos en el camino nos ayudó a progresar más rápido; los colombianos Edwin Pérez y Sebastián Jerez con quienes hicimos mancuerna a través de nuestros cursos de maestría. Todos ellos además del apoyo en el ámbito académico me brindaron su amistad y momentos recreativos.

Al Doctor Iván Heredia, quien me impartió un grandioso curso, me orientó en mi proceso de elegir asesor y además me brindó palabras de aliento cuando me sentía abrumado.

Al Doctor Rodrigo Pelayo Ramos, porque gracias a él me convencí de que mi pasión era la física. Sin su apoyo no hubiera tenido el coraje de cambiarme de carrera y empezar de cero. Sin duda sus cursos han sido los mejores que he tenido y formaron una sólida estructura que me ha ayudado a lo largo de toda mi trayectoria.

A Mariana del Castillo Sánchez, quien siempre nos recibe con una sonrisa y nos apoya eficientemente en los distintos trámites dentro del Cinvestav.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCyT), por el apoyo económico que me permite hacer este posgrado.

Por último quiero agradecer a Yuren Silva, Josué Magaña, Braulio Chamorro, Miguel Trejo y Jesús Rodríguez, con los cuales he tenido una inquebrantable amistad

desde la vocacional.

Abstract

This thesis presents the measurement of the Branching Fraction by q^2 bin of the $B^+ \to K^{*+}\mu^+\mu^-$ rare decay using data collected by the CMS experiment at the Large Hadron Collider (LHC) at CERN.
The channel $B^+ \to K^{*+}J/\psi$ was chosen as the normalization channel due to its high statistics and its ability to cancel out common terms when applying the formula $N_{evts} = \left(\int \mathcal{L}dt\right) \times \varepsilon \times \sigma \times f_u \times BF$ to both resonant and non-resonant channels.
The analysis involved the reconstruction of the B^+ meson candidates, applying filters and using Machine Learning techniques, specifically XGBoost, to efficiently separate signal from background.

The mass distributions were modeled using the Python's Zfit package, where the signal models were obtained by fitting the generated signal from Monte Carlo simulations, while the background was modeled using different models depending on its behavior throughout the q^2 bins.
Efficiencies were computed using simulations, and the data were processed using ROOT and the Python's Pandas package.

This analysis followed the B Physics Blinding Policy, which in this case implies that the obtained results are multiplied by a random number between 0 and 1. However, this work reports the value of the ratio $BF(B^+ \to K^{*+}\psi(2S))/BF(B^+ \to K^{*+}J/\psi) = 0.505 \pm 0.39$, which is consistent within less than one sigma with the Particle Data Group (PDG) reported value.

Resumen

En esta tesis se presenta la medición de la Branching Fraction por bin de q^2 del decaimiento raro $B^+ \to K^{*+}\mu^+\mu^-$ usando datos recolectados por el experimento CMS en el gran colisionador de hadrones, LHC por sus siglas en inglés, en el CERN.
El canal $B^+ \to K^{*+}J/\psi$ fue elegido como canal de normalización debido a su alta estadística y su habilidad de cancelar términos cuando aplicamos la fórmula $N_{evts} = \left(\int \mathcal{L}dt\right) \times \varepsilon \times \sigma \times f_u \times BF$ a ambos canales resonante y no resonante.
El análisis consistió en reconstruir los candidatos a mesón B^+, aplicar filtros y usar técnicas de Machine Learning, específicamente XGBoost, para separar eficientemente la señal del ruido.

Las distribuciones de masa fueron modeladas usando la paquetería Zfit de Python, donde los modelos de señal fueron obtenidos mediante un ajuste a la señal generada en simulaciones Monte Carlo, mientras que el ruido fue modelado con diferentes modelos dependiendo del comportamiento de éste en los distintos bines de q^2.
Las eficiencias fueron calculadas usando simulaciones, estos datos fueron procesados haciendo uso de ROOT y la paquetería Pandas de Python.

Este análisis fue realizado siguiendo las políticas de Blinding del grupo de B Physics, lo cual implica en este caso que los resultados obtenidos se encuentran multiplicados por un número aleatorio entre 0 y 1. Sin embargo, en este trabajo se reporta el valor del cociente $BF(B^+ \to K^{*+}\psi(2S))/BF(B^+ \to K^{*+}J/\psi) = 0.505 \pm 0.39$, el cual es consistente a menos de un sigma con el valor reportado en el Particle Data Group (PDG).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Standard Model

The Standard Model (SM) of particle physics is a theoretical framework that describes the fundamental particles and their interactions, except for gravity. It contemplates three of the four know fundamental forces (electromagnetic, weak and strong interactions) and it has been shown to be consistent with numerous high-energy physics experiments.

The SM uses principles of quantum field theory (QFT) and gauge symmetry, specifically the gauge symmetry group $SU(3)_C \times SU(2)_L \times U(1)_Y$. This symmetry structure allows us to model interactions between particles through the exchange of force carriers: photons for the electromagnetic, $W^{\pm}$ and Z^0 bosons for the weak force and gluons for the strong force [1].

The basic building blocks of the matter are fundamental particles. In the SM, these particles are fermions (particles with half-odd-integer spins and obey the Pauli exclusion principle) and are divided into two categories [2]:

- **Quarks:** There are six flavors of quarks, organized into three generations: up, down, charm, strange, top and bottom quarks.
- **Leptons:** Similarly, there are six leptons, also arranged in three generations: the electron, muon, tau and their corresponding neutrinos.

An important component of the SM is the Higgs mechanism, which explain how

particles acquire mass [3]. In order to explain this a scalar field is introduced, it permeates the space and it is called the Higgs field. Particles gain mass when they interact with this field, and the strength of their interaction determines the magnitude of the mass. The excitation of the Higgs fields manifests as the Higgs boson, whose experimental confirmation was achieved at the Large Hadron Collider (LHC) in 2012 by the Atlas and CMS collaborations [4, 5]. In Figure 1.1 [6] shows a graphic representation of all the fundamental particles of the SM.

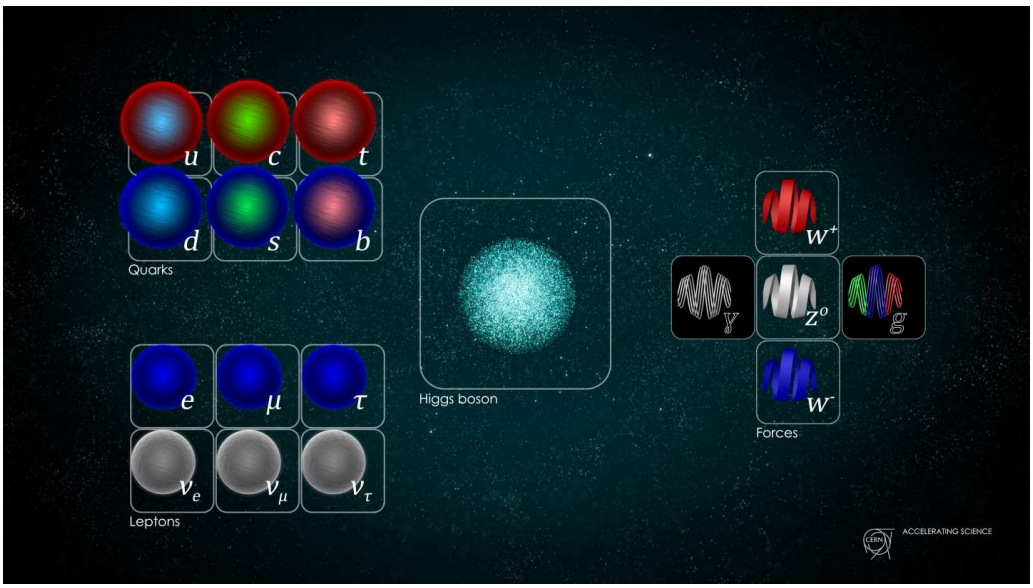


Figure 1.1: Particles of the Standard Model of particle physics.

Some interactions of the SM can occur straightforward, meaning that particles directly interact or decay without intermediate processes. These are said to occur at tree-level, and the corresponding Feynman diagrams are the simplest ones. An example of a tree-level process is the beta decay of a neutron, where a down quark transforms into an up quark via the emission of a W^- boson (Figure 1.2). These processes are directly allowed by the SM and, due to their simplicity, they occur with relatively high probability.

On the other hand, there are loop-level processes that involve one or more closed loops in the Feynman diagram, representing more complex interactions where virtual particles contribute to the process. These processes can occur in scenarios where a tree-level interaction is forbidden by the SM.

The Glashow-Iliopoulos-Maiani (GIM) mechanism tells us that $\Delta S = 2$ transitions, where S is a quantum number called strangeness, and flavor-changing neutral cur-

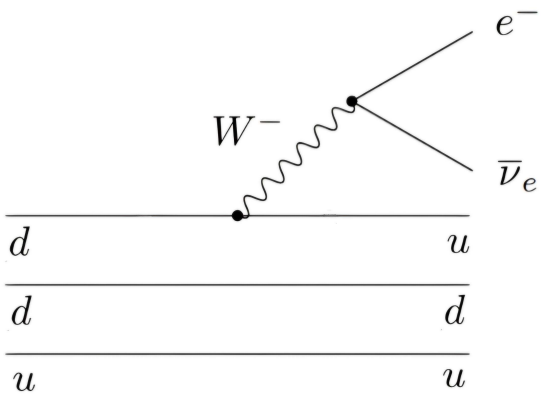
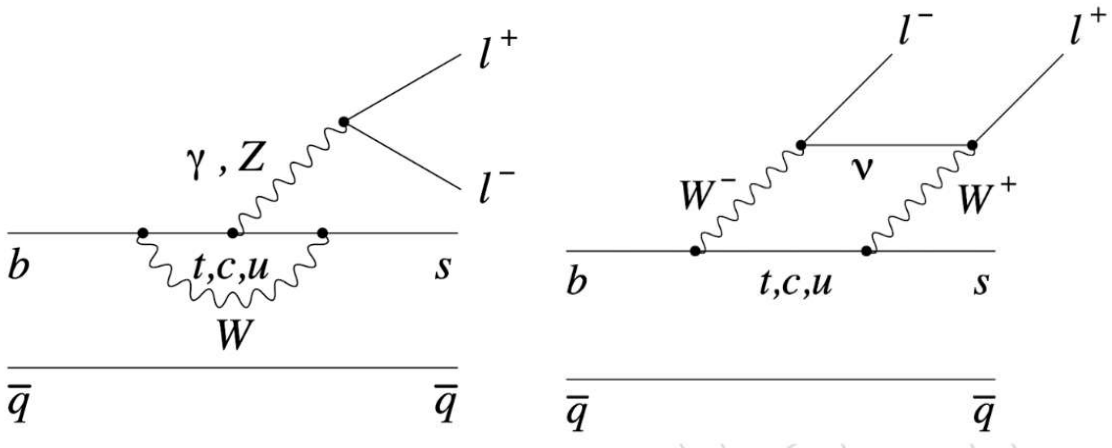


Figure 1.2: Feynman diagram of the beta decay.

rent (FCNC) processes must occur only at second order in the weak interactions [7], meaning they cannot occur at tree-level.

FCNCs are transitions between different quark flavors without changing the electric charge, they are highly suppressed due to the GIM mechanism. The decay $B^+ \to K^{*+}\mu^+\mu^-$ is an example of a FCNC process. Since it is forbidden at tree-level, it must occur at loop-level, specifically through penguin and box diagrams (Figure 1.3).



(a) Penguin diagram (b) Box diagram

Figure 1.3: Feynman diagrams of the process $b \to sll$ involved in the $B^+ \to K^{*+}\mu^+\mu^-$ decay.

1.2 LHC

The LHC, located at CERN (European Organization for Nuclear Research) is the world's most powerful circular particle collider, situated near Geneva in the France-Switzerland border. Since the collisions are between two counter-rotating beams, it has two rings with superconducting magnets to bend the particle trajectories. The rings have a circumference of 26.7 km and they are buried between 45 m and 170 m underground. By accelerating particles to speeds near c and colliding them, scientists can probe the fundamental interactions described by the SM and search for new physics beyond the SM.

To achieve the high energies, the LHC uses a series of smaller accelerators that gradually increase the energy of particles before injecting them into the main collider ring. The process is divided into the following stages [8] (Figure 1.4 [9]):

- **Linear Accelerator (LINAC 4):** Here is where everything begins, LINAC 4 takes negative hydrogen ions (H^-, consisting of a hydrogen atom with an additional electron) and accelerates them to energies around 160 MeV. Before the particles are sent into the next stage of the acceleration chain, the ions are stripped of their two electrons.

- **Proton Synchrotron Booster (PSB):** After exiting the LINAC, the protons are injected into this circular accelerator where they reach energies about 2 GeV.

- **Proton Synchrotron (PS):** The PS first accelerated protons on 24 November 1959, and for a brief period was the world's highest energy particle accelerator. Now it is a section of the acceleration chain. In addition to protons, it can also accelerate heavy ions provided by the Low Energy Ion Ring (LEIR) and serves as a connection to other experiments. The final energy at this stage is about 25 GeV.

- **Super Proton Synchrotron (SPS):** The SPS, with a circumference of 7 km, is the second largest machine in CERN's accelerator complex and it is the final step before entering the LHC. In the SPS the protons are accelerated up to 450 GeV.

- **Large Hadron Collider (LHC):** In the LHC ring, protons are accelerated in two opposite directions in separate beam pipes. Superconducting magnets

guide the protons around the circular ring while several accelerating structures boost their energy. The total center-of-mass energy of proton-proton collisions in the LHC for our data was 13.6 TeV. The beams inside the LHC are made to collide at four interaction points, corresponding to the positions of four particle detectors: ATLAS, ALICE, LHCb and CMS. ALICE focuses on heavy-ion physics, LHCb is a dedicated B physics experiment, while ATLAS and CMS use general-purpose detectors for research across a wide variety of high-energy physics.

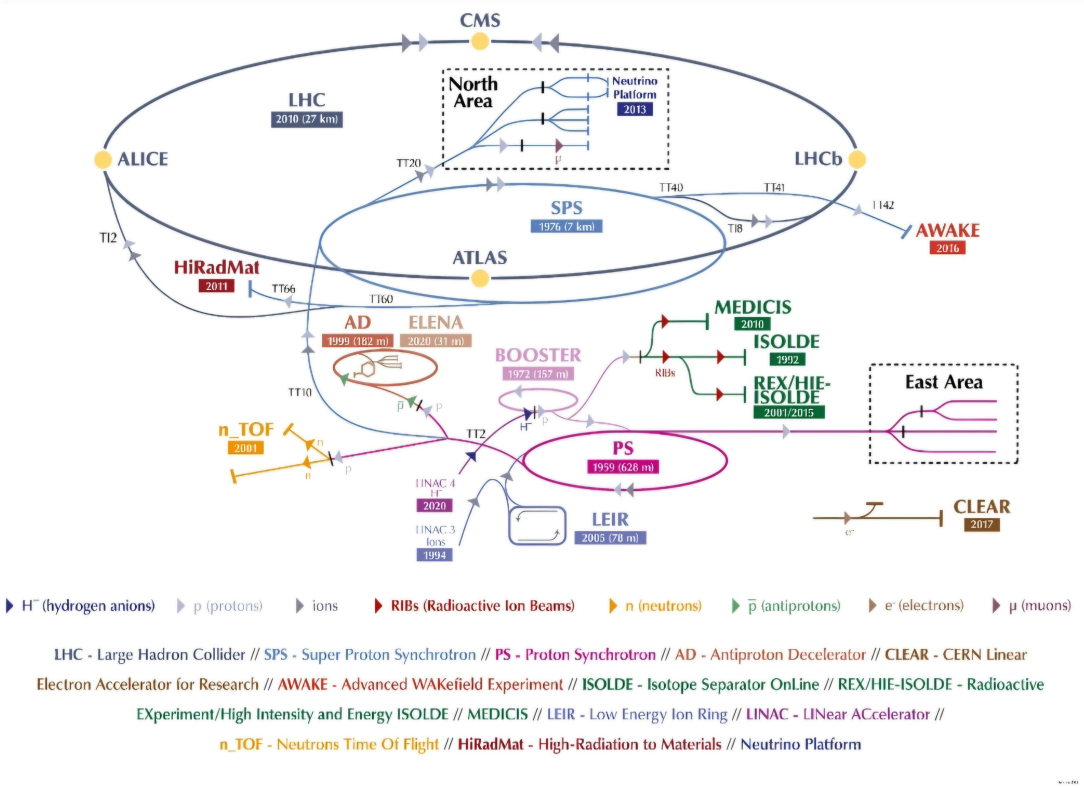


Figure 1.4: Accelerator complex at CERN.

1.2.1 CMS detector

The Compact Muon Solenoid (CMS) is one of the general-purpuse detectors at the LHC. It gets its name from the fact that it is really compact considering all the detector material it contains, it is designed to detect muons very accurately and is has the most powerful solenoid magnet ever made.

CMS is built as a cylindrical detector, surrounding the interaction point. It is divided into layers that measure the properties of the particles emerging from the collision. The CMS components and their functions are as follows [10]:

- **Superconducting Solenoid Magnet:** It provides a magnetic field of 4 Tesla which bends the paths of the electrically charged particles. Its high strength was designed to bend the particles' trajectories more significantly, making it easier to measure their momentum and distinguish their electric charge.

- **Silicon Tracker:** The first component the particles pass through is the tracking system. It is divided into layers and has two sections. The innermost section consist of four layers of pixel detectors and three disks at each end, totaling about 124 million pixels. The size of these pixels is 100 µm by 150 µm.
  The second section is the silicon strip detector, which has an inner barrel consisting of four layers and two endcaps composed of three small disks. The outer barrel consists of six concentric layers and two endcaps with nine disks. This section contains about 10 million detector strips, each with an area of the order of 100 cm^2. Figure 1.5 [11] shows the distribution of the detectors in the tracking system.

  This part of CMS provides a precise measurement the momentum of the electrical charged particles by measuring the curvature of their paths. By using several 2D layers of detectors, it is able to reconstruct the 3D paths of the particles.

- **Calorimeters:** CMS uses two types of calorimeters to measure the energies of most particles. The first one is the Electromagnetic Calorimeter (ECAL), which measures the energy of electrons and photons by stopping them completely. The next layer is the Hadron Calorimeter (HCAL) which stops particles made of quarks. These particles pass through the ECAL with a minimal interaction and deposit nearly all of their primary energy in the HCAL.

- **Muon Detection System:** This is the outermost component of CMS and takes advantage of the fact that muons can penetrate several meters of mate-

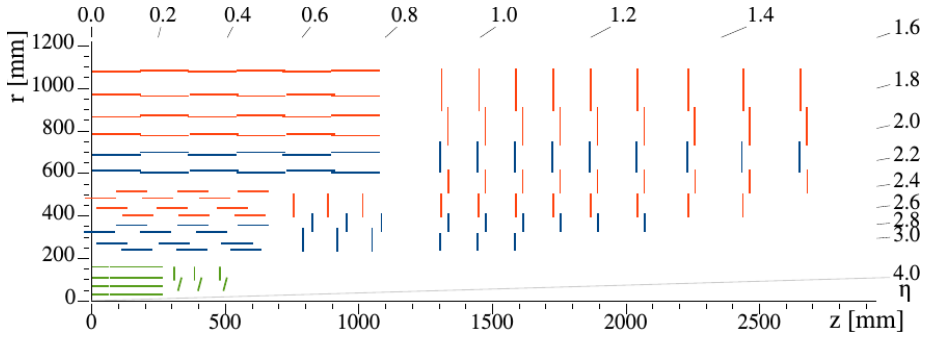


Figure 1.5: Tracking System. Green represents the pixel detectors, while blue and orange we represent doubled-sided and single-sided strip detectors, respectively.

rial while losing little energy. Since Muons are not stopped by the calorimeters, they are the only particles likely to reach this section and leave a signal.
There are four muon stations interleaved with the iron return yoke plates of the magnetic system. The thickness of these plates adds another layer of certainty that the detected particles are muons.

The system consists of 1400 muon chambers, divided into 250 drift tubes and 540 cathode strip chambers for tracking and triggering, as well as 610 resistive plate chambers and 72 gas electron multiplier chambers for a redundant trigger system.

- **Trigger System:** While not a physical component, the trigger system is a critical component of the detection system of CMS. The trigger and Data Acquisition System (TriDAS) is designed to select the most interesting and relevant fraction of events from the billions of interactions per second.
  The first level of triggering, Level 1 (L1), selects or rejects events based on the information from the ECAL and HCAL. The High Level Trigger (HLT) is a software-based trigger that performs more sophisticated analysis, such as pattern recognition and event reconstruction.

A cutaway view of the CMS detector showing the arrangement of the components is shown in Figure 1.6 [10].

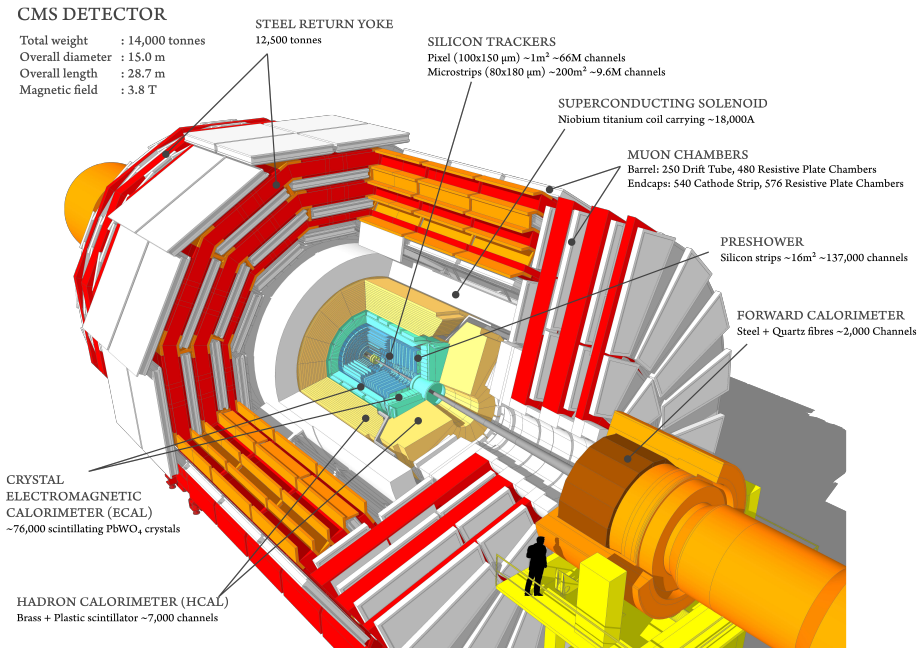


Figure 1.6: Overall view of the CMS detector.

1.3 Differential Branching Fraction

The number of events of a decay is given by:

$$N_{evts} = \left(\int \mathcal{L} dt \right) \times \varepsilon \times \sigma \times f_u \times BF \tag{1.1}$$

where $\int \mathcal{L} dt$ is the integrated luminosity, ε is the total efficiency of the whole selection process, σ is the cross section, f_u is the fragmentation fraction and BF is the branching fraction.

By taking this into account, we can get the Normalized Differential Branching Fraction as:

$$\begin{aligned} &\frac{1}{BF(B^+ \to K^{*+} J/\psi)} \frac{dBF(B^+ \to K^{*+} \mu^+ \mu^-)}{dq^2} = \\ &= \frac{N_{evts}(B^+ \to K^{*+} \mu^+ \mu^-)}{N_{evts}(B^+ \to K^{*+} J/\psi)} \cdot \frac{\varepsilon(B^+ \to K^{*+} J/\psi)}{\varepsilon(B^+ \to K^{*+} \mu^+ \mu^-)} \cdot \frac{BF(J/\psi \to \mu^+ \mu^-)}{q^2_{max} - q^2_{min}} \end{aligned} \tag{1.2}$$

where $N_{evts}(B^+ \to K^{*+} \mu^+ \mu^-)$ is the number of signal candidates in the studied bin, $N_{evts}(B^+ \to K^{*+} J/\psi)$ is the number of signal candidates in the resonant bin, the product

$BF(B^+ \to K^{*+}J/\psi)BF(J/\psi \to \mu^+\mu^-)$ is the branching fraction of the channel of normalization, and $\varepsilon(B^+ \to K^{*+}J/\psi)/\varepsilon(B^+ \to K^{*+}\mu^+\mu^-)$ is the relative efficiency between the signal and the channel of normalization.

1.4 q^2 Binning

In Equation 1.2, we can see that our measurement considered a q^2 binning, which is defined in Table 1.1. This binning was chosen following a LHCb recommendation, with the intention of ensuring consistency between different analyses. The low q^2 range aligns with LHCb, while the high q^2 range is extended due to the higher energy of the run we analyzed.

Bin Number	q^2 Range $(\text{GeV})^2$
1	[1.1, 2]
2	[2, 4]
3	[4, 6]
4	[6, 8]
5 (J/ψ)	[8, 11]
6	[11, 12.5]
7 ($\psi(2S)$)	[12.5, 15]
8	[15, 17]
9	[17, 23]
Low q^2	[1.1, 6]
High q^2	[15, 23]

Table 1.1: Binning in the dimuon invariant mass.

In Figure 1.7, we see how the simulated data distribution looked across the different bins. It is worth mentioning that the initial binning scheme included two bins in the interval [17, 23] GeV^2: [17, 19] GeV^2 and [19, 23] GeV^2. This proposal was abandoned when we realized that the resolution effect caused by the detector simulation led to events migrating from one bin to another—a phenomenon known as *bin migration*. In the case of these two bins, we observed that the last bin gained more events than it lost, as shown in Table 1.2; specifically, the number of events in that bin increased by 46.21%. Additionally, the statistics in this bin were low compared to the other bins, so merging the two bins addressed these issues. In Table 1.2, the columns "EventsGen" and "EventsReco" correspond to the number of events in each bin

before and after applying the detector simulation, respectively. With the chosen binning scheme, the bin migration effects are balanced across the bins, with none changing its size by more than 0.6%.

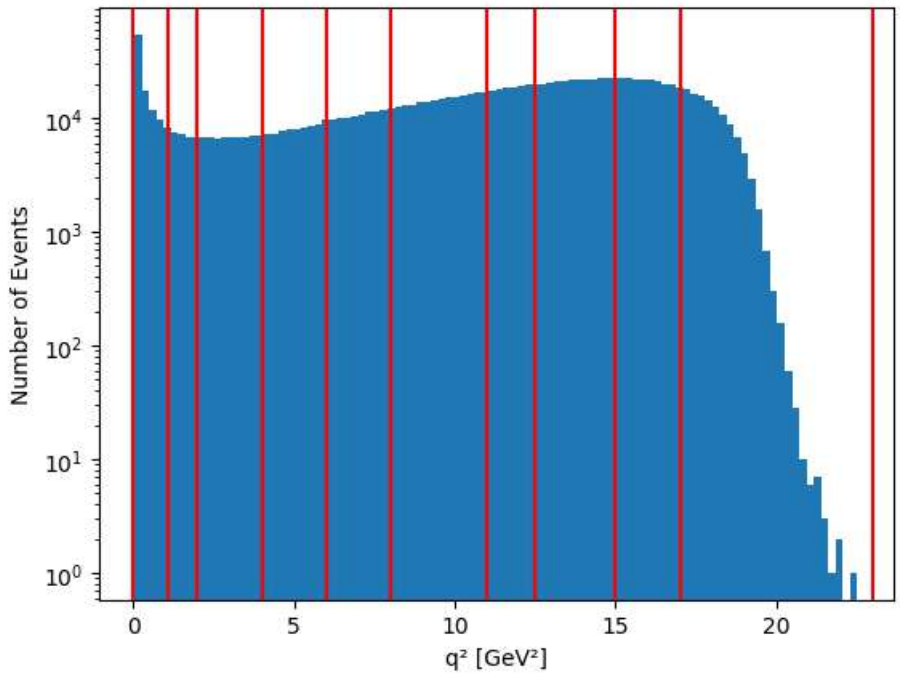


Figure 1.7: q^2 distribution of the simulated data and the division lines of the bins.

1.5 Previous Measurements

The Branching Fraction measurements considered by the PDG [12] include those conducted by LHCb (2014), BaBar (2009), and Belle (2009) [13, 14, 15]. LHCb used data from proton-proton collisions collected with the LHCb detector at center-of-mass energies of 7 TeV and 8 TeV in 2011 and 2012, respectively. BaBar utilized a sample of 384 million $B\bar{B}$ events collected with the BaBar detector at the PEP-II e^+e^- collider. Lastly, Belle used a sample of 657 million $B\bar{B}$ pairs collected with the Belle detector at the KEKB e^+e^- collider. Since LHCb's measurement has a much lower uncertainty compared to the other two, it carries more weight in the average value provided by the PDG. For this reason, we only took LHCb's measurement as a reference for comparison. The values published by LHCb are shown in Table 1.3 [13].

q^2 Bin Range (GeV^2)	EventsGen	EventsReco	LostEvents	InmigrantEvents	100*Reco/Gen
[0, 1.1]	99324	99288	321	285	99.96
[1.1, 2]	28793	28809	702	718	100.06
[2, 4]	60530	60481	1312	1263	99.92
[4, 6]	71140	71285	2520	2665	100.20
[6, 8]	94979	94752	4786	4559	99.76
[8, 11]	194292	194253	8701	8662	99.98
[11, 12.5]	123979	124039	13245	13311	100.05
[12.5, 15]	238624	238357	18067	17800	99.88
[15, 17]	188999	188707	20144	19482	99.84
[17, 19]	116332	114350	13502	11520	98.30
[19, 23]	5770	8436	1539	4205	146.21
[17, 23]	122102	122786	9321	10005	100.56

Table 1.2: Effect of Bin Migration

q^2 range (GeV^2/c^4)	central value	stat	syst
$0.1 < q^2 < 2.0$	59.2	+14.4 -13.0	4.0
$2.0 < q^2 < 4.0$	55.9	+15.9 -14.4	3.8
$4.0 < q^2 < 6.0$	24.9	+11.0 - 9.6	1.7
$6.0 < q^2 < 8.0$	33.0	+11.3 -10.0	2.3
$11.0 < q^2 < 12.5$	82.8	+15.8 -14.1	5.6
$15.0 < q^2 < 17.0$	64.4	+12.9 -11.5	4.4
$17.0 < q^2 < 19.0$	11.6	+ 9.1 - 7.6	0.8
$1.1 < q^2 < 6.0$	36.6	+ 8.3 - 7.6	2.6
$15 < q^2 < 19.0$	39.5	+ 8.0 - 7.3	2.8

Table 1.3: LHCb (2014) Differential Branching Fractions results ($10^{-9} \times c^4/GeV^2$) for the $B^+ \to K^{*+}\mu^+\mu^-$ decay.

Chapter 2

Data and Simulation Samples

2.1 Data

The studies presented here utilized data from Run 3 of pp collisions at $\sqrt{s} = 13.6$ TeV, gathered in 2022. The events were selected using the `HTL_DoubleMu4_3_LowMass` trigger, which necessitated two oppositely charged muons, a vertex-fit probability greater than 0.5%, a dimuon mass between 0.2 and 8.5 GeV, a leading muon transverse momentum p_T exceeding 4 GeV, a trailing muon p_T over 3 GeV, and a muon pseudorapidity $|\eta|$ less than 2.5. For offline reconstruction, two oppositely charged muons were required to match those that activated the detector readout, with some criteria being more stringent than those at the trigger level.

In Table 2.1, we show the datasets from the 2022 ParkingBPH Eras and their recorded integrated luminosities ($\mathcal{L}_{\text{int}}$), which were computed with the `brilcalc` tool [16].

Era	Dataset	Recorded Luminosity (fb^{-1})
Run2022C	/ParkingDoubleMuonLowMassX/Run2022C-PromptReco-v1/MINIAOD	5.07
Run2022D	/ParkingDoubleMuonLowMassX/Run2022D-PromptReco-v1/MINIAOD	3.01
Run2022D	/ParkingDoubleMuonLowMassX/Run2022D-PromptReco-v2/MINIAOD	
Run2022E	/ParkingDoubleMuonLowMassX/Run2022E-PromptReco-v1/MINIAOD	5.88
Run2022F	/ParkingDoubleMuonLowMassX/Run2022F-PromptReco-v1/MINIAOD	18.01
Run2022G	/ParkingDoubleMuonLowMassX/Run2022G-PromptReco-v1/MINIAOD	3.12

Table 2.1: Used datasets and their corresponding luminosities. The X in the dataset names runs from 0 to 7.

2.2 MC Simulation Samples

Throughout this analysis, Monte Carlo (MC) simulation samples were used. They were employed for efficiency studies in the differential branching fraction measurement, training multivariable classifiers, modeling the B^+ invariant mass distribution free of background, and studying possible background sources from other channels.

We utilized Official Monte Carlo Samples, which were generated using `PYTHIA` 8 [17] for production and hadronization processes, `EVTGEN` [18] for the b hadron decays, and `PHOTOS` [19] for the QED final state radiation. The generated events underwent a detailed simulation using the CMS detector, which was based on the GEANT 4 package [20]. This simulation employed the same trigger and reconstruction algorithms as those used for the 2018 collision data. Simulated events accounted for multiple proton-proton interactions occurring in the same or nearby beam crossings (pileup), with a distribution that matched the observed data.

The decay generation model employed for these simulations was the BTOSLLBALL `EVTGEN` model. This model simulated $b \to sll$ decays following the approach of P. Ball et al. [21], resulting in angular distributions consistent with Standard Model predictions.

We utilized two types of samples. The first one was at the Gen level, meaning that the events were only generated and not processed through the detector simulation. Additionally, this sample had no filters applied. The second sample consisted of generated events with some filters applied (Table 2.2), and finally, they underwent the detector simulation.

Particle	p_T (GeV)	η
$\mu_{1,2}$	[2.5, ∞)	[-2.5, 2.5]
K^{*+}	[0.0, ∞)	(-∞, ∞)
π_3	[0.5, ∞)	[-2.5, 2.5]
K^0_s	[0.5, ∞)	[-3.0, 3.0]
$\pi_{1,2}$	[0.3, ∞)	[-3.0, 3.0]

Table 2.2: Filters for p_T and η of the particles involved in the decay.

The MC Data (with filters) was acquired from the next samples:

Non-Resonant:

/BuToKstarMuMu_Filter_MuFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

JPsi Resonant:

/BuToJpsiKstar_JpsiFilter_MuFilter_KstarFilter_K0sFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v1/MINIAODSIM

Psi2S Resonant:

/BuToPsi2SKstar_PsiFilter_MuFilter_KstarFilter_K0sFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

The MC Data (without filters) was acquired from the next samples:

Non-Resonant:

/BuToKstarMuMu_Unbiased_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

JPsi Resonant:

/BuToJpsiKstar_Unbiased_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

Psi2S Resonant:

/BuToPsi2SKstar_Unbiased_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

Chapter 3

Event Selection

3.1 Reconstruction

The $B^+ \to K^{*+}\mu^-\mu^+$ decay process (as well as the $B^+ \to K^{*+}J/\psi$ and $B^+ \to K^{*+}\psi(2S)$ resonance channels) was fully reconstructed in its final charged states. This means that we required two muons and three pions. The specific decay path that we reconstructed was:

$$B^+ \to K^{*+}\mu^+\mu^- \tag{3.1}$$
$$K^{*+} \to K^0_s\pi^+ \tag{3.2}$$
$$K^0_s \to \pi^+\pi^- \tag{3.3}$$

In the case of the resonant channels, we reconstructed the J/ψ quarkonium states through their decay to a muon-antimuon pair:

$$J/\psi \to \mu^+\mu^- \tag{3.4}$$
$$\psi(2S) \to \mu^+\mu^- \tag{3.5}$$

Unless otherwise specified, charge conjugation was implied throughout this analysis. For the reconstruction of J/ψ quarkonium states, we utilized the standard CMS muon reconstruction. The Muon-POG-approved *Soft Muon* selection [22] included the following criteria: the track identified by the tracker had to match with at least

one segment in any muon station in both x and y coordinates (*TMOneStationTight*), and it had to be arbitrated. The track had to qualify as *high-purity*, including more than five hits in the tracker layers, with at least one hit in a pixel layer. The impact parameter of the muons relative to the primary vertex had to be less than 0.3 cm in the transverse plane and less than 20 cm along the beam axis. For offline reconstruction, two oppositely charged muons matching those that triggered the detector readout were required. These muons also had to have $|\eta| < 2.4$, a minimum transverse momentum of $p_T > 4$ GeV, and a dimuon vertex fit χ^2 probability greater than 1%.

Candidate K_s^0 mesons were reconstructed from pairs of oppositely charged tracks, each assigned with the pion mass assumption. These candidates were selected from the general V^0CandidatesKshort list as reconstructed by the `RecoVertex/V0Producer` package (more details in [23]). Following the fitting of the K_s^0 vertex, a series of cuts were applied to reduce background events and enhance the purity of the V^0 candidates:

- Vertex normalized $\chi^2 < 6.63$
- Significance of radial vertex position > 15
- The inner hit of both daughter tracks must not lie within 4σ (radially) from the reconstructed V^0 vertex
- The K_s^0 mass must lie within ± 0.07 GeV of the PDG K_s^0 mass value

These criteria were designed to ensure that the reconstructed K_s^0 mesons were of high quality, minimizing the contamination from other sources. By applying these cuts, we aimed to improve the signal-to-background ratio.

Additionally, Λ^0 contamination was removed from the K_s^0 sample by discarding candidates within the $p\pi$ mass region of $1.1096 < M(p\pi) < 1.1216$ GeV. This was achieved by assigning the proton mass to the leading pion from the K_s^0 decay.

Candidate K^{*+} mesons are reconstructed from the reconstructed K_s^0 and a charged track with pion mass assumption took from the PackedCandidate container, this track needed to be a high-purity track with at least five hits in the tracker layers one

of which had to be a pixel hit. The track also needed to have a minimum transverse momentum of $p_T > 0.95$ GeV and we verified this track did not correspond with any of the muon tracks.

To produce a B^+ candidate, we had to consider each combination of a dimuon with the K^{*+} candidate. We applied to each B^+ candidate a cinematic fit, which contemplates a common vertex fit and the appropiate masses for the charged tracks and the K_s^0 invariant mass, which were constrained to the world-average masses given by the PDG [12]. The intermediate candidate states K^{*+} and K_s^0 were selected if they lied within $0.6 < M(K_s^0\pi) < 1.6$ GeV and $\left|M(\pi^+\pi^-) - m_{K_s^0}^{PDG}\right| < 10$ MeV respectively. The B^+ mesons had to have a vertex χ^2 probability greater than 1%. Primary vertices (PV) were determined by fitting the reconstructed tracks. The vertex candidate with the highest sum of physics-object p_T^2 values was selected as the primary pp interaction vertex. The decay length l of the B^+ hadron was calculated as the distance between the PV and the B^+ hadron vertex, which were considered to be the production and decay vertices of the B^+ meson, respectively. Because of the short lifetime of the K^{*+} meson and the resolution of the CMS detector, we considered that the production and decay vertices of this meson lied in the same point and were the same with the B^+'s decay vertex.

3.2 Resonance and Antiradiation Veto

As we said in at the beginning of this section, all the decays $B^+ \to K^{*+}\mu^-\mu^+$, $B^+ \to K^{*+}J/\psi$, $B^+ \to K^{*+}\psi(2S)$ are reconstructed through the same final state, that's why our reconstruction algorithm can not distinguish between them. In order to discriminate them, two fixed windows were taken in the dimuon invariant mass, $q = m(\mu^+\mu^-)$. These windows were taken to follow other analysis as was shown in the binning in Table 1.1.

However, some events were still able to escape this veto, to solve this problematic we used another veto that is efficient removing these kind of events. This veto

involved the combination of the next conditions:

$$P_1(X,i) : \left|(m_{B^+PDG} - m_{reco}(K^{*+}\mu^+\mu^-)) - (m_{X\ PDG} - m_{reco}(\mu^+\mu^-))\right| < \Delta m_{X,i} \tag{3.6}$$

$$P_2(X) : (m_{reco}(\mu^+\mu^-))^2 < Q_X^{left} \tag{3.7}$$

$$P_3(X) : (m_{reco}(\mu^+\mu^-))^2 > Q_X^{right} \tag{3.8}$$

$$P_4(X) : m_{reco}(\mu^+\mu^-) < c_X \tag{3.9}$$

where $X = \{J/\psi, \psi(2S)\}$, $i = \{\text{left, right}\}$, $[Q_{J/\psi}^{left}, Q_{J/\psi}^{right}] = [8, 11]$, $[Q_{\psi(2S)}^{left}, Q_{\psi(2S)}^{right}] = [12.5, 15]$, and Δm and c are parameters to obtain. Δm is the width of the diagonal stripes and c is an upper threshold on the $m(\mu^+\mu^-)$ of the right stripes of each resonance.
The condition for each resonant window is then:

$$P(X) = [P_1(X, left) \wedge P_2(X)] \vee [P_1(X, right) \wedge P_3(X) \wedge P_4(X)] \tag{3.10}$$

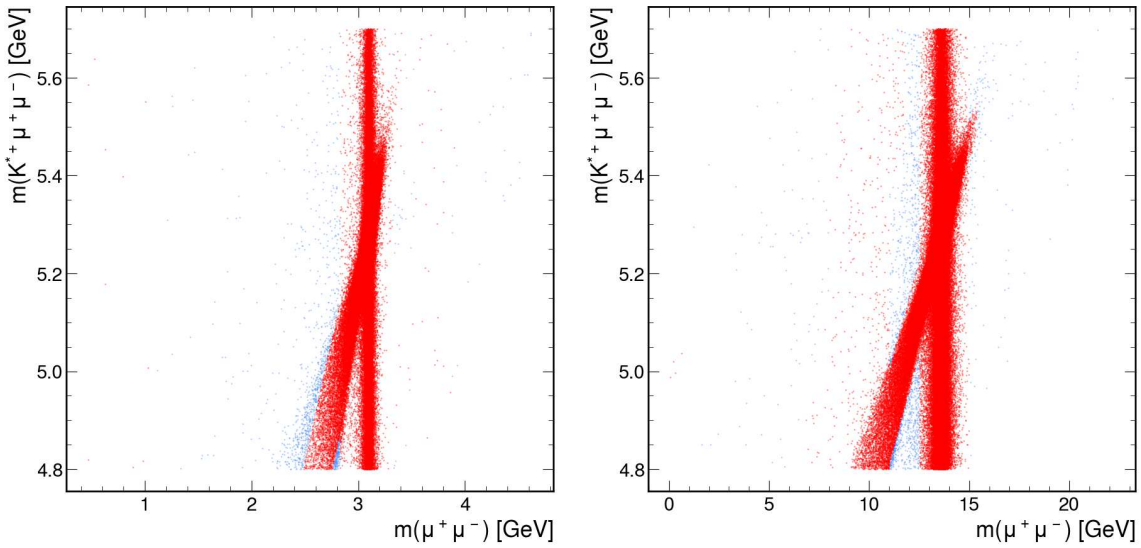
Since some of these events escape their resonant window due to the decay $\mu \to \mu\gamma$, this veto is sometimes called *antiradiation veto*
The values for the Δm and c were taken from a previous analysis [24], the way they were obtained was basically optimizing those values so that about the 90% of $B^+ \to K^{*+}X$ events that escaped the resonance veto got rejected.
In Table 3.1 we can see the values we used for this veto.

X	c_X (GeV)	$\Delta m_{X,left}$ (MeV)	$\Delta m_{X,right}$ (MeV)
J/ψ	3.43	137	134
$\psi(2S)$	3.92	97	44

Table 3.1: Optimized values for the Δm and c

To verify that the two vetos were working correctly, they were applied to the $B^+ \to K^{*+}J/\psi$ and $B^+ \to K^{*+}\psi(2S)$ MC samples, in Figure 3.1 we can see scatter plots $m(\mu^+\mu^-)$ vs $m(K^{*+}\mu^+\mu^-)$ of the events, where the red dots represent rejected events and the blue ones are those that still could escape the vetos. For the J/ψ channel we got a 99.81% of rejection while the $\psi(2S)$ channel got 99.82%.

Finally, in Figure 3.2 we can see the effect of the resonance and antiradiation vetos



(a) J/ψ simulated data

(b) $\psi(2S)$ simulated data

Figure 3.1: Vetos described in this section applied to the simulated data of the resonant channels.

on the real data, additionally we imposed the condition $q^2 > 1.1$ GeV since the first bin began from there.

3.3 Machine Learning based Classification

3.3.1 Generalities

What we had got so far was not helpful yet; if we looked at the B^+ mass distribution, we saw that a bump barely appeared and the background was so high (Figure 3.3). This indicated that our data had too much background. At that point, it was almost impossible to separate our signal from the background with just simple cuts; we needed something more sophisticated. In our data, there were two classes, one being signal (positive class) and the other being background (negative class). Just by simple inspection, we could not tell one from another. To achieve better classification, we used a machine learning algorithm. For now, let's consider it as a black box, and in the next section, we will delve into the mathematical formalities.

To *train* it, we needed to provide the black box with two datasets: the positive and the negative class. Then it would learn to distinguish them using some features of the data. After the training was done, we could take an event and ask the box

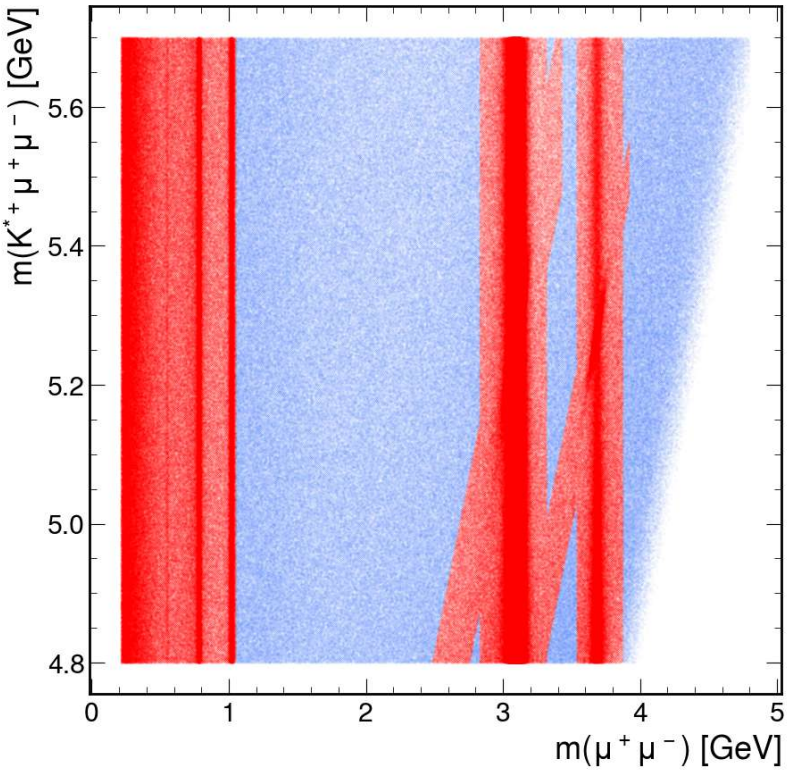


Figure 3.2: Vetos applied to the reconstructed dataset.

whether it was signal or background. The answer we received would be a number in the interval $[0, 1]$, which we could interpret as the probability of the event being signal, the closer the number was from 1 the more likely to be a signal event, and the closer to 0 the more likely to be a background event.

Now let's discuss the steps we took to make this black box work. First of all, we needed to conduct a study that informed us which of the features of the data were more important, this means, which of them contributed more to the prediction.
When we provide the black box with the two datasets we commonly use 70% for training and 30% for testing, then when we train the box it will tune what is inside of it to give us what we call a *model*, and using the fact that we know which events belong to the positive class and which to the negative, we can use *metrics* to quantify the *performance* of the trained box. But this does not end here, because we can modify the characteristics of the box and if we train this new box we obtain a different model and so a new performance. These "characteristics" of the box are called *hyperparams*. So the second step was to navigate in the hyperparam space in

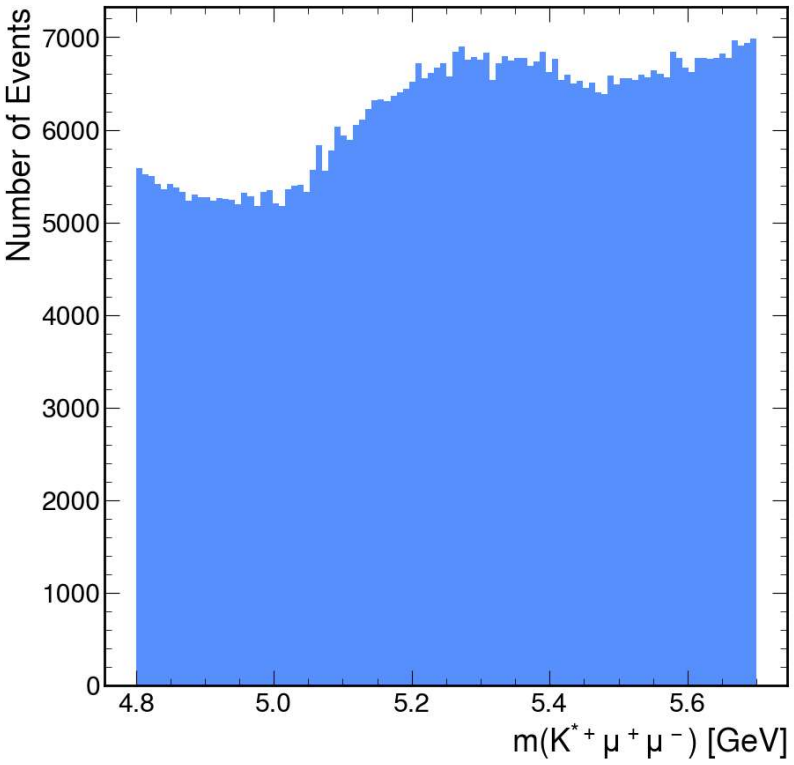


Figure 3.3: Reconstructed B mesons after vetos.

order to get the values that gave us the highest performance.

After getting the model with the best performance, we needed to be sure that it performs well with new data, in this case the testing data. When the performance on testing data is not as good as it is with the training data, we say that we *over-trained* the box, this means that the box learned very well to work with the training data but it can not do the same with new data, so that box is basically useless.
Even if the box did a nice perform with the training data we still needed to do one overtraining test more, due to the fact that both the training and the testing data came from the same sample and were split randomly, we would expect that the distributions of the prediction values given by the box were similar because the box should work with new data as it did with the training data. To have a quantification of this we did a statistical test between the training and the testing output of the box distributions, that told us how likely was that they followed the same distribution.

Once we had a nice trained box that passed all our requirements, we needed to

choose a cut in the output of the box because, as mentioned before, it is a number from 0 to 1. The most natural way to start is to take the cut to be 0.5, so if the output is greater than it we would say that event is signal, otherwise we would classify it as background. But as we know there is no perfect algorithm, and in our case it was far from making a perfect classification, implying that if we took this 0.5 cut we would classify background events as signal events. The problem is that the background and the signal do not have radical differences, which signifies that basically with any cut we take we will have background events identified as signal, those events are called *false positives* (FP).
Solving this is not as simple as just increasing the value of the cut, because in doing so we also start to classify signal events as background, *false negatives* (FN). The events that are correctly classified are called *true positives* (TP) and *true negatives* (TN) for the signal and background events respectively. To solve this problem we maximized a function of the number of TP, TN, FP and FN that fitted better our needs.

This is the essence of the whole algorithm used to create our classifier, now let's move to the formalities.

3.3.2 XGBoost

XGBoost, short for eXtreme Gradient Boosting, is a machine learning method that combines the usage of gradient boosting algorithms and, systems and algorithmic optimizations [25]. XGBoost has showed to have a good performance on a wide range of problems, in this case this machine learning method was chosen for its ability to handle large datasets, and because the robust classification results it has given in other analysis of our research group.

Now we will explain the XGBoost algorithm for classification without going too deep into the details since it is out the scope of this writing.
XGBoost gives its prediction as a sum of outputs from decision trees, scaled by a factor called learning rate, plus a initial value. The main column of the algorithm is its loss function $L(y, F(x))$, were y and $F(x)$ are the real value and the prediction of the training event respectively, and L is differentiable everywhere. The algorithm tries to minimize that loss function. One important thing that this loss function has

is the usage of a L2 regularization parameter λ and a tree complexity parameter γ, the first one prevents overfitting by shrinking the feature weights, making the model more rubust, and the second one controls the minimum loss reduction required to make a further partition on a leaf node of the tree, which affects the tree's complexity by limiting the number of slipts.

Let's assume we want to create a model with M trees, each of them with J_m leaves, for a training dataset of N events whose true value is $y \in \{0, 1\}$ and the n features are encoded in $x \in \mathbb{R}^n$.
Step 1: Initialize the model with a constant value such that:

$$F_0(x) = \underset{\sigma}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \sigma)$$

Step 2: For $m = 1$ to M
a) Compute the residuals from previous prediction $F_{m-1}(x)$ as:

$$r_{mi} = y_i - F_{m-1}(x_i)$$

b) Fit a regression tree to the r_{mi} values and create terminal regions R_{mj} for $j = 1, ..., J_m$, this terminal regions are also called leaves of the tree.
To make the splits, in theory, we need to search the best one among all the possible ones in every variable, XGBoost has some optimization algorithms that make this search less computional demanding but we will not discuss them here.
We start by taking one leaf (the mother), in case it is the first leaf we call it root, and compute the loss function value evaluated at the optimal output value, in XGBoost terminology this is called *Similarity Score* (SS). Then we split the dataset using some threshold in some variable and calculate the SS of each of the two leaves (daughters) that were created by the split. To see how good the split is we take the difference between the SS of the left and right daughter leaves and the mother leaf, this is called $Gain$.

$$Gain = SS_{Left} + SS_{Right} - SS_{Mother}$$

We now just have to take the cut with the highest gain and keep splitting the leaves till one of the following conditions is met: we reach the maximum number of leaves, we reach the maximum depth or the resulting leaf does not have a minimum of ele-

ments.
To avoid overfitting we may prune some of the leaves, given some γ we compute $Gain - \gamma$, if it is larger or equal to zero we keep it, otherwise we prune it. This way we have created the m-th tree with J_m leaves.

c) For $j = 1, ..., J_m$ compute:

$$\sigma_{mj} = \underset{\sigma}{\operatorname{argmin}} \sum_{x_i \in R_{mj}} L(y_i, F_{m-1} + \sigma)$$

d) Update:

$$F_m(x) = F_{m-1}(x) + \eta \sum_{j=1}^{J_m} \sigma_{jm} I_{jm}(x)$$

where η is the learning rate and

$$I_{jm}(x) = \begin{cases} 0 \quad if \quad x \notin R_{jm} \\ \\ 1 \quad if \quad x \in R_{jm} \end{cases}$$

Step 3: Compute the output $F_M(x)$

3.3.3 Feature selection

To select the features we worked with, we conducted a correlation and a feature importance studies, all the invariant masses of the resonances $(B^+, K^{*+}, \mu\mu, K_s^0)$ were immediately ignored since those variables could highly bias the classification.
In Figure 3.4 we can see the correlation values of the candidate variables to be chosen for training, the correlation threshold was 0.75, meaning that the Pi2pt was not consider due to the high correlation it presented with Pi1pt. To quantify the feature importance we used the SHAP (SHapley Additive exPanations) values. These values quantify the impact of each feature on the prediction, they are based on game theory and offer consistent and interpretable feature attributions [26]. We also used XGBoost feature importances metrics, these were[27]:

- **Weight**: The number of times a feature is used to split the data across all trees.

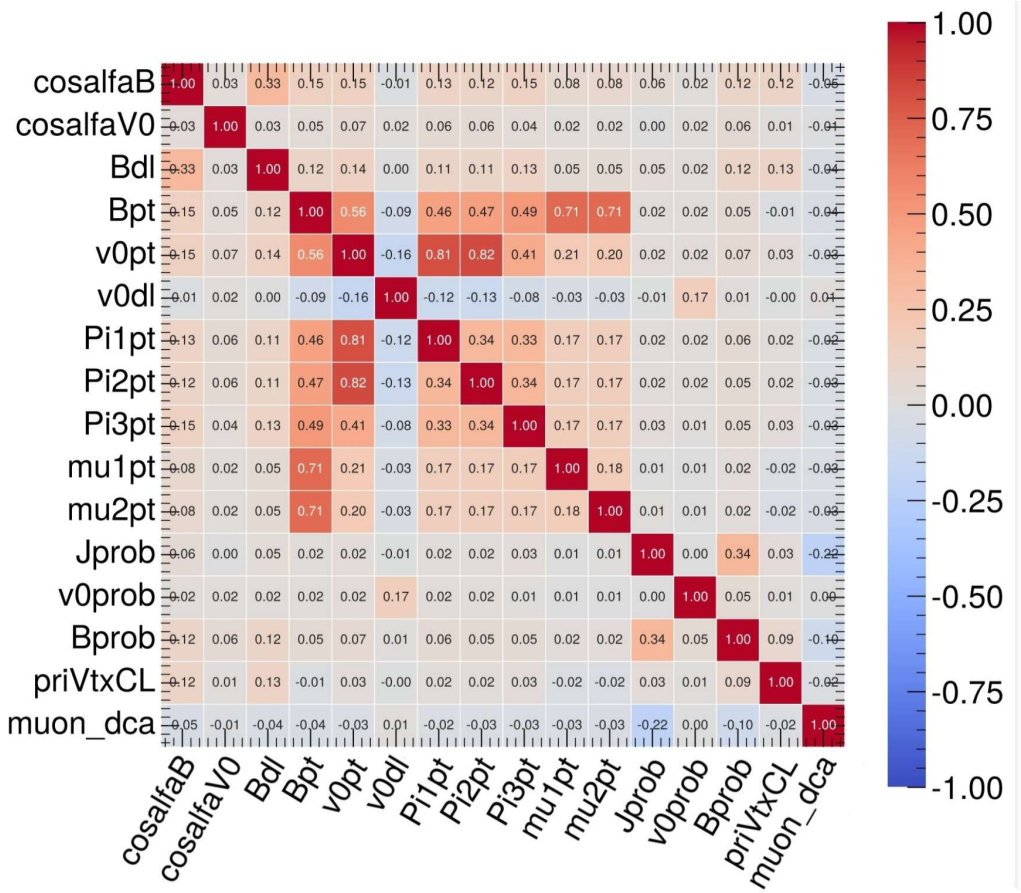


Figure 3.4: Heatmap of the correlation of the studied features.

- **Gain**: The average gain value across all splits that the feature is used in. The gain value is a measurement of the improvement of a tree when an additional split is done.
- **Cover**: The average coverage across all splits that the feature is used in. The coverage is a metric to measure the number of observations affected by a split.
- **Total Gain**: The total gain across all splits that the feature is used in.
- **Total Cover**: The total coverage across all splits that the feature is used in.

Since all these values go from 0 to 1, we added them up and divided by 6, the features that obtained an average greater than 0.2 were used for the training, in Table 3.2 we can see the numerical result of the study. Finally, the selected features based on these two procedures were:

Feature	Weight	Gain	Cover	Total Gain	Total Cover	SHAP	Total	Average
cosalfaB	0.91	1.00	1.00	1.00	0.91	1.00	5.82	0.97
Bdl	1.00	0.21	1.00	0.23	1.00	0.82	4.25	0.71
muon_dca	0.81	0.02	0.76	0.02	0.61	0.33	2.55	0.43
Jprob	0.67	0.02	0.90	0.01	0.60	0.24	2.45	0.41
Bprob	0.71	0.08	0.69	0.06	0.49	0.28	2.29	0.38
priVtxCL	0.81	0.05	0.61	0.04	0.49	0.26	2.26	0.38
Pi3pt	0.62	0.09	0.70	0.06	0.44	0.24	2.15	0.36
v0pt	0.56	0.11	0.64	0.06	0.35	0.19	1.91	0.32
mu2pt	0.54	0.01	0.66	0.00	0.36	0.05	1.61	0.27
Bpt	0.52	0.02	0.63	0.01	0.33	0.05	1.56	0.26
v0dl	0.63	0.01	0.48	0.01	0.30	0.07	1.51	0.25
mu1pt	0.49	0.01	0.62	0.00	0.30	0.04	1.47	0.24
cosalfaV0	0.35	0.02	0.71	0.01	0.25	0.06	1.38	0.23
Pi1pt	0.35	0.00	0.60	0.00	0.21	0.03	1.19	0.20
Pi2pt	0.37	0.00	0.50	0.00	0.19	0.02	1.08	0.18
v0prob	0.35	0.00	0.27	0.00	0.09	0.01	0.73	0.12

Table 3.2: Feature Importance Metrics

- **cosalfaB** and **cosalfaV0**: The cosine of the 3D pointing angle of the B and K_s^0.

$$\cos\alpha_X = \frac{(\mathbf{V}_{Decay} - \mathbf{V}_{Production}) \cdot \mathbf{P}_X}{|\mathbf{V}_{Decay} - \mathbf{V}_{Production}||\mathbf{P}_X|} \tag{3.11}$$

  where $\mathbf{V}_{Decay}$ and $\mathbf{V}_{Production}$ are the coordinates of the vertices and $\mathbf{P}_X$ is the 3-momentum of the X particle.

- **Bdl** and **v0dl**: The decay length of the B and the K_s^0.

- **muon_dca**: The distance of closest approach of the muon.

- **Jprob**: The probability of the vertex reconstructed using the two muons.

- **Bprob**: The probability of the primary vertex.

- **priVtxCL**: Confidence level of the primary vertex.

- **Pi3pt, v0pt, mu2pt, Bpt, mu1pt, Pi1pt**: The transverse momentum of the particles.

3.3.4 Hyperparam search

The hyperparams we used for our training are [27]:

- **max_leaves:** Maximum number of nodes to be added.
- **max_depth:** Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit.
- **learning_rate:** Step size shrinkage used in update to prevent overfitting. After each boosting step, we can directly get the weights of new features, and the learning_rate shrinks the feature weights to make the boosting process more conservative.
- **reg_alpha:** L1 regularization term on weights. Increasing this value will make model more conservative.
- **reg_lambda:** L2 regularization term on weights. Increasing this value will make model more conservative.
- **gamma:** Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be. Note that a tree where no splits were made might still contain a single terminal node with a non-zero score.
- **min_child_weight:** Minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than min_child_weight, then the building process will give up further partitioning. In linear regression task, this simply corresponds to minimum number of instances needed to be in each node. The larger min_child_weight is, the more conservative the algorithm will be.
- **colsample_bytree, colsample_bylevel, colsample_bynode:** This is a family of parameters for subsampling of columns, these hyperparams help to prevent overfitting.

  All colsample_by* parameters have a range of (0, 1], the default value of 1, and specify the fraction of columns to be subsampled.

  colsample_bytree is the subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed.

colsample_bylevel is the subsample ratio of columns for each level. Subsampling occurs once for every new depth level reached in a tree. Columns are subsampled from the set of columns chosen for the current tree.

colsample_bynode is the subsample ratio of columns for each node (split). Subsampling occurs once every time a new split is evaluated. Columns are subsampled from the set of columns chosen for the current level.

In order to get the best hyperparameters we used the library hyperopt, it is a powerful optimization tool that uses Bayesian optimization, the conceptual components are [28]:

i) Null distribution specification language: Specify the hyperparameters you want to optimize and the ranges or distributions they can take.
In our case, we chose the hyperparameters that described earlier and the distributions and ranges were chosen based on the experience of previous analysis of the group and also using some intuition of how to avoid overfitting.

ii) Loss Function: Define a loss function (also called objective function) that Hyperopt will try to minimize (or maximize). This function typically involves training an XGBoost model with a specific set of hyperparameters and then evaluating its performance using some metric.
We tested three different objective funtions, before we talk about them we need to give some definitions:

$$Precision = \frac{TP}{TP + FP} \tag{3.12}$$

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN} \tag{3.13}$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN} \tag{3.14}$$

- **ROC-AUC:** The ROC (Receiver Operating Charasteristic) curve is a plot of the True Positive Rate vs False Positive Rate when different classification thresholds are taken. The ROC-AUC is the Area Under the Curve made by the ROC curve.

- **PC-AUC:** The PR (Precision-Recall) curve is similar with the ROC curve, the difference is that it plots the precision vs the recall (also know as True

Positive Rate). Once again, the PC-AUC consist in taking the area under the curve.

- **F1 Score:** This is a weighted harmonic mean of precision and recall:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.15}$$

This kind of mean punishes when one of the two quantities involved is low. In order to have a more global performance study using the F1 score, we took the highest F1 score among the possible classification thresholds for each set of values of hyperparameters tested.

iii) Hyperparameter Optimization Algorithm (HOA): The HOA is an algorithm which takes as inputs the the search space and the history of values from previous evaluations, and returns suggestions for which configuration to try next. Hyperopt uses a technique called Tree of Parzen Estimators, it builds a probabilistic model that estimates the performance of the different hyperparameter configurations and then it chooses some point that maximizes a ratio that helps to identify regions in the hyperparameter space where the likelihood of finding better-performing configurations is higher.

iv) Database: This approach tends to focus the search in regions that have been historically more promising since the history of configurations and the value of the loss function at each one is stored.

We finally chose the model that was obtained using the ROC-AUC function objective and the hyperparameter values were:
max_leaves=57, max_depth=9, learning_rate=0.0592, reg_alpha=5.4707, reg_lambda=5.5836, gamma=4.0054 min_child_weight=28.7328, colsample_bytree=0.8007, colsample_bylevel=0.7967, colsample_bynode=0.9549.

3.3.5 Performance and Overfitting tests

In Figure 3.5 we show a slightly different version of the ROC curve for our model, the differences are that in the y axis we have the background rejection, it is defined

in terms of the FPR (Equation 3.14) as:

$$Background\ Rejection = 1 - FPR = \frac{TN}{FP + TN}, \tag{3.16}$$

and in the x axis we have:

$$Signal\ Efficiency = TPR \tag{3.17}$$

where TPR is given in Equation 3.13. We can see that the values of the AUC for

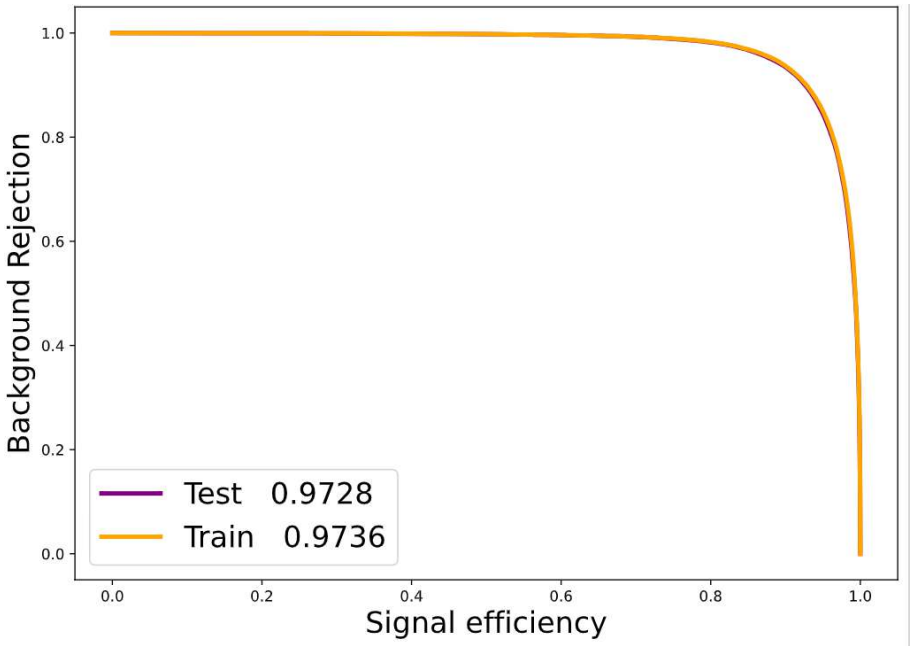


Figure 3.5: ROC curve for our training. The Test and Train curves are so similar that the Test curve is barely visible.

the training and testing datasets were very close, the relative change between the AUC of the model applied on each of the two datasets was only of the 0.08%. This showed us that the performance of the model on new data was very similar with the obtained with the training data.

In order to have an extra validation that tells us that our model was not overfitted we did a Kolmogorov-Smirnov test between the normalized distributions of the output values of the model applied to the training and testing datasets. In the case of this test, the null hypothesis $H0$ is that the distribution of the training dataset

is similar/consistent with the testing dataset one, our criteria to not reject the null hypothesis was that the corresponding p-value were greater than 10%.
In Figure 3.6 we show the obtained distributions from our model as well as the ks-pvalues.

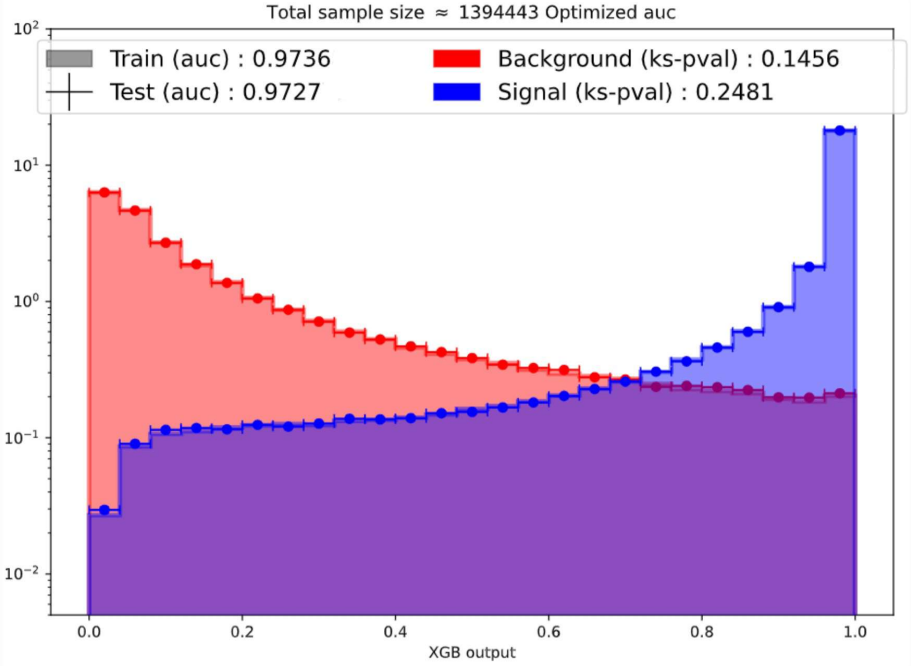


Figure 3.6: Normalized distributions of the XGBoost output for training and testing datasets.

3.3.6 Choosing the optimal cut using a Figure Of Merit (FOM)

In order to decide at which output value of the XGBoost was the best we used the next FOM:

$$FOM = \frac{S}{\sqrt{S+B}} \tag{3.18}$$

where S and B were the number of signal and background events in the signal window (defined in section 3.5) obtained from a Global Fit of all the data, i.e. it was not divided into bins. It is worth mention that at any point the value of the signal yield was known by us, but instead we only knew the values of the whole FOM (Figure 3.7), the point of this was to get a value that gives us the best relation between the signal and background in the signal window.

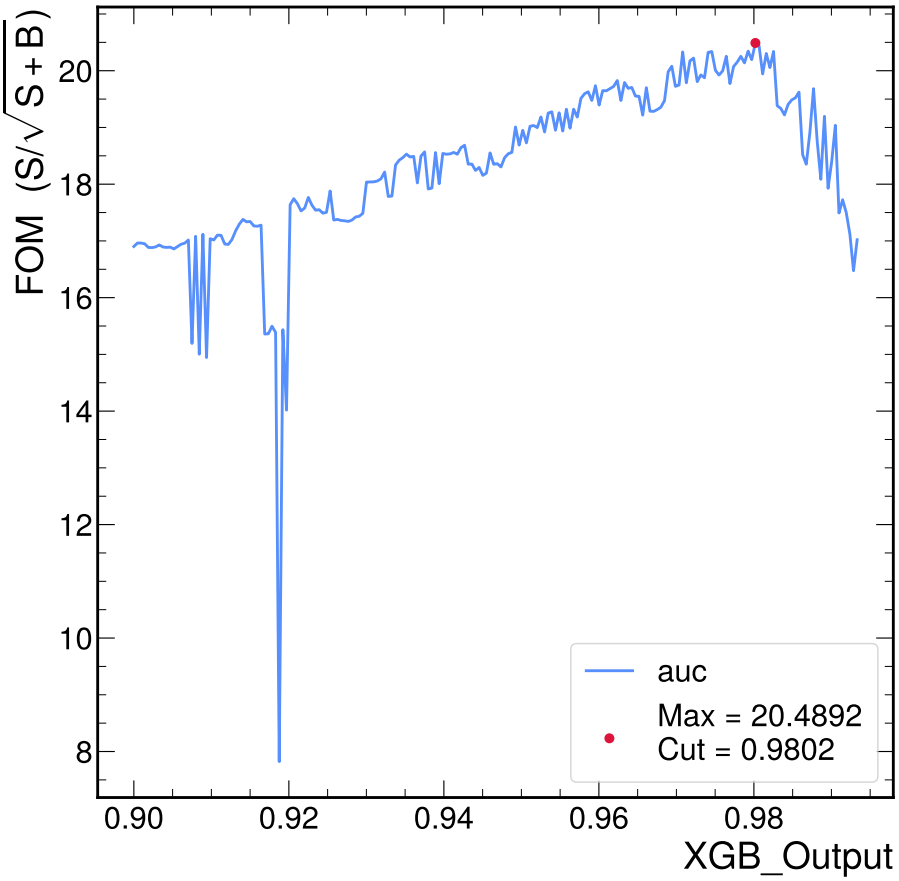


Figure 3.7: Optimization of the FOM

3.4 Study of Possible Sources of Background

There are some decays whose topology is somehow similar with our decay, we wanted to see how our whole selection process treated them. To do this we took some MC samples and passed them through all the steps that have been described in this note, as if they were real data from the non-resonant channel.
The decays that were chosen and their datasets are:

$B^+ \to J/\psi K^{*+}$:

`/BuToJpsiKstar_JpsiFilter_MuFilter_KstarFilter_K0sFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v1/MINIAODSIM`

$B^+ \to \psi(2S) K^{*+}$:

/BuToPsi2SKstar_PsiFilter_MuFilter_KstarFilter_K0sFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

$B^0 \to \mu^+\mu^- K_s^0$:

/B0ToK0sMuMu_MuFilter_K0sFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

$B^0 \to J/\psi K_s^0$:

/B0ToJpsiK0s_JpsiFilter_MuFilter_K0sFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v3/MINIAODSIM

$B^0 \to \psi(2S) K_s^0$:

/B0ToPsi2SK0s_PsiFilter_MuFilter_K0sFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v1/MINIAODSIM

$\Lambda_b \to \mu^+\mu^- \Lambda^0$:

/LambdaBToLambdaMuMu_MuFilter_LambdaFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

$\Lambda_b \to J/\psi \Lambda^0$:

/LambdaBToJpsiLambda_JpsiFilter_MuFilter_LambdaFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v1/MINIAODSIM

$\Lambda_b \to \psi(2S) \Lambda^0$:

/LambdaBToPsi2SLambda_PsiFilter_MuFilter_LambdaFilter_TuneCP5_13p6TeV_pythia8-evtgen/Run3Summer22EEMiniAODv3-124X_mcRun3_2022_realistic_postEE_v1-v2/MINIAODSIM

The reconstruction efficiencies of these samples were all at least 3 orders of mag-

nitude smaller than the one we obtained for our channel of interest, furthermore the events did not accumulate in our signal window, because of that we did not have to take any action like modeling a specific shape of background due to these channels.

3.5 Final Selection

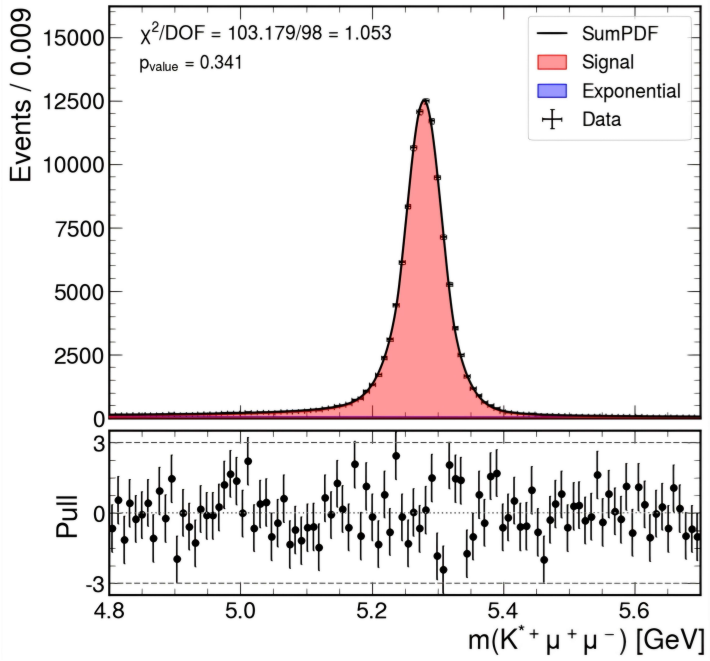
After the machine learning classification of our complete dataset, we had our selected events. From these events we had to obtain the yield of signal from each of the channels by doing a fit of the invariant mass of the B meson, the fit was performed using an extended maximum likelihood fit. This distribution had two main components, the signal and the background.

The signal was modeled using a sum of two *probability distribution functions* (pdfs), a Gaussian plus a Double Crystal Ball (DCB) with common mean, since this total pdf had 7 free parameters (2 of the Gaussian, 6 of the DCB minus 1 of the common mean) we first did a fit on the MC data for each channel (Figure 3.8) to obtain the parameters of the signal distribution to subsequently left only 1 free parameter in the the component of signal of the data distribution fit.

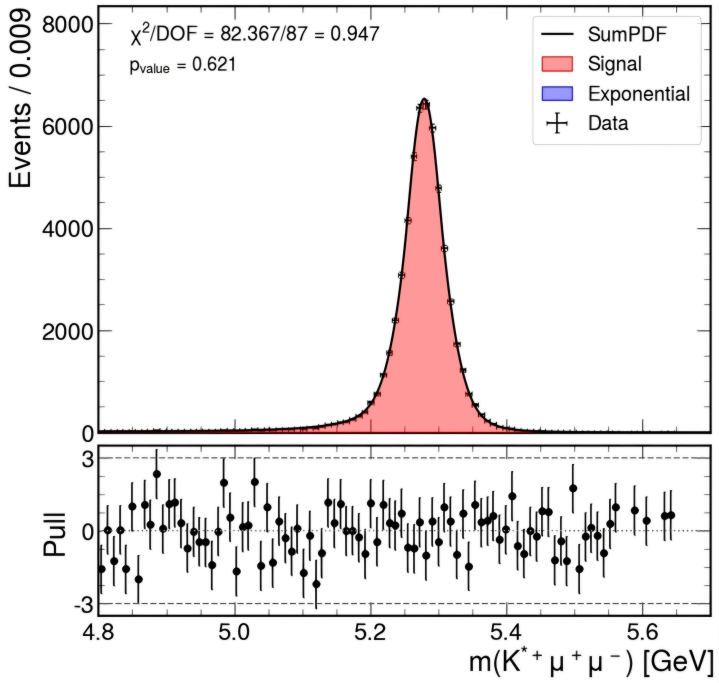
Besides of this, the fit on MC data was used to define the Signal Window, it was obtain by getting a effective sigma from the fitted pdf that was done with MC data that passed preselection filters and had the MC matching, then we took 5 sigmas to each side of the mean of the pdf so that the Window was defined as [5.106, 5.452]. We can visualize this in Figure 3.9.

The background component of the total pdf was not the same across the different fits that had to be done, instead it was modeled depending on each case. All of the background contributions were the sum of an exponential distribution plus another pdf, which could have been a gaussian, the error function distribution or a Chebyshev polinomial.

Finally, the fit to the selected data was performed with 1 free parameter for the signal component, all the background component ones (1-4), and since the total fit was an extended one, 2 for the number of events that corresponded to signal and

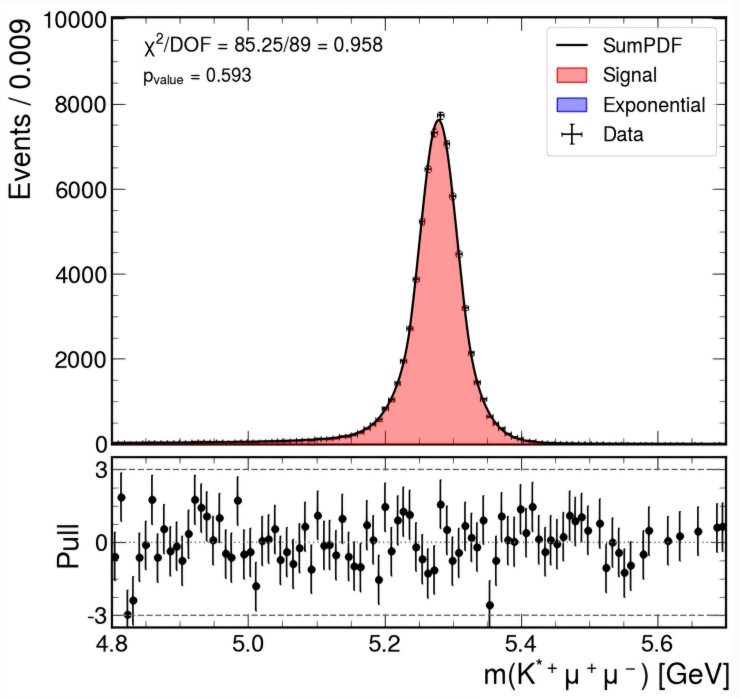


(a) Non Resonant Channel



(b) J/ψ Resonant Channel

Figure 3.8: Fits of the MC data of each channel.



(c) $\psi(2S)$ Resonant Channel

Figure 3.8: Fits of the MC data of each channel.

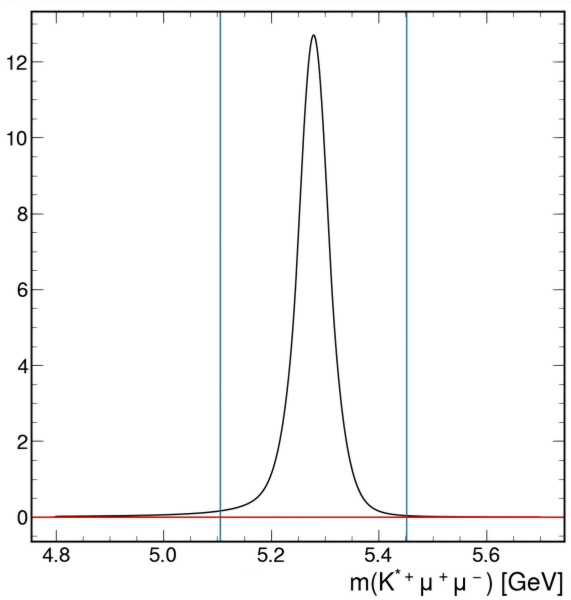


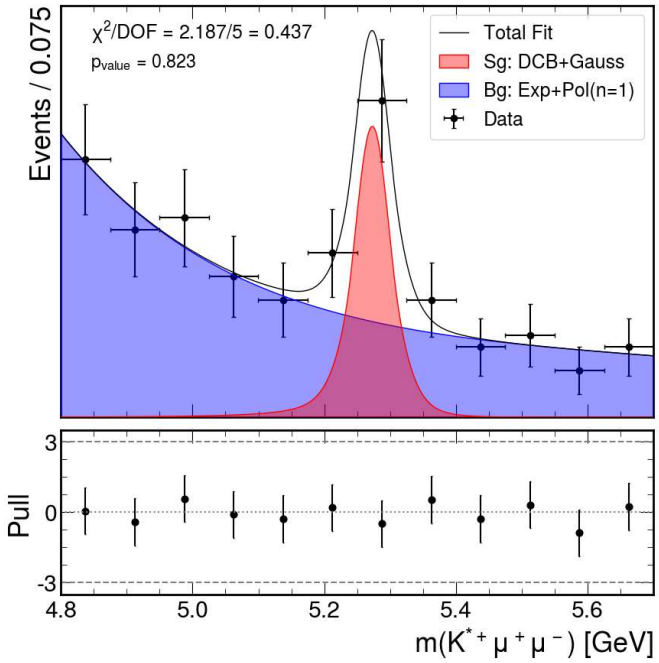
Figure 3.9: Visualization of the Signal Window with respect of the MC distribution of the invariant mass of the reconstructed B mesons.

background.

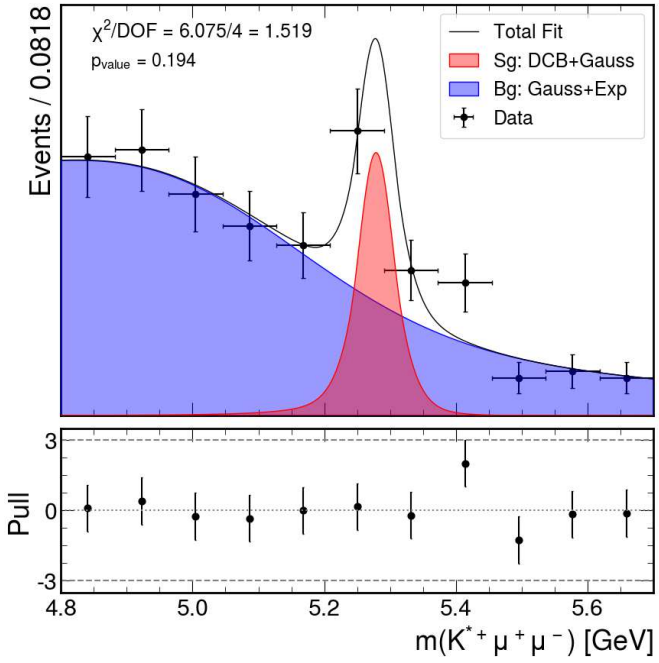
In the case of the resonant channels, the fact that the statistic was larger caused us some troubles and the fits were not being good, because of this we treated these fits with a slightly different approach. Instead of fixing the parameters of the signal we constrained them to be able to vary following a Gaussian Constrain, it means that the values of the parameters prefer to be close the mean and penalizes the further it gets with respect the sigma. In this case the mean were the values we obtain from the fit in MC data and the sigma were the uncertainties of those parameters. So, even if by doing this we got more free parameters, it was justified because it was expected more events in these channels than the non resonant one.

This analysis followed the recommendations of the "BPH Analysis Blinding Policy", to do so we performed the fits with mass distribution histograms without tick labels and for each bin of q^2 a different unknown random number, in the interval (0,1), was used to multiply the yields of signal.

Figure 3.10 shows the invariant meson B mass distributions of the $B^+ \to K^{*+}\mu^-\mu^+$ decay channel for each q^2 bin, Figure 3.11 does it for the low and high q^2 regions and Figure 3.12 shows the correspondent to the $B^+ \to K^{*+}J/\psi$ and $B^+ \to K^{*+}\psi(2S)$ decays channels.

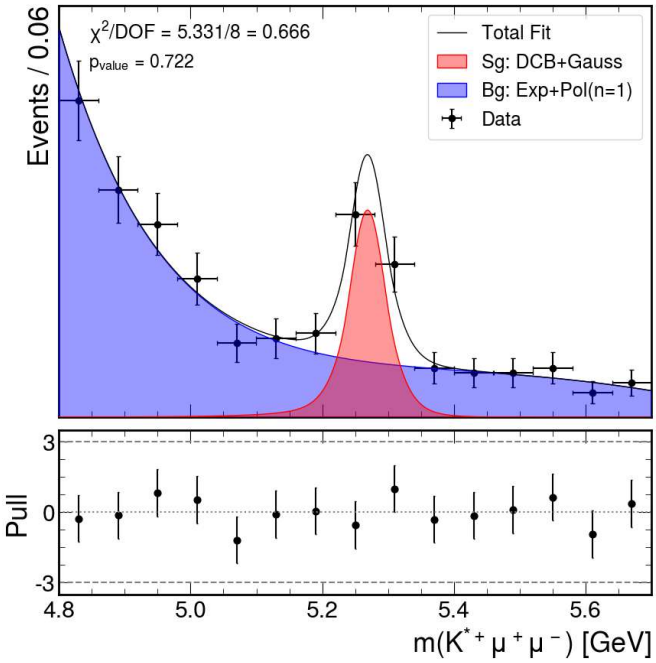


(a) Bin 1

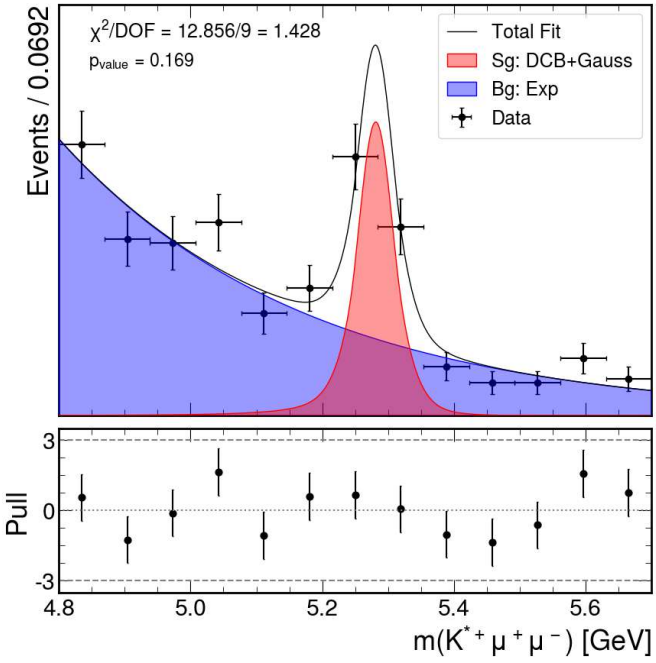


(b) Bin 2

Figure 3.10: Fits of the non-resonant data by bin.

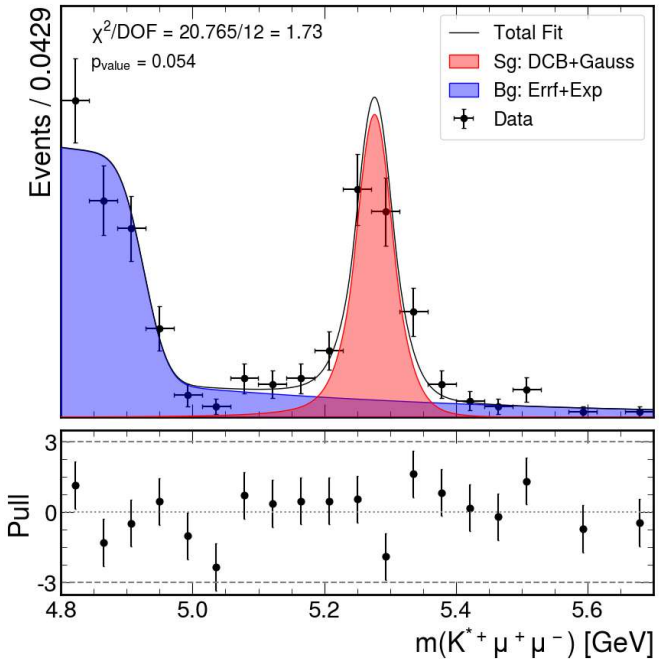


(c) Bin 3

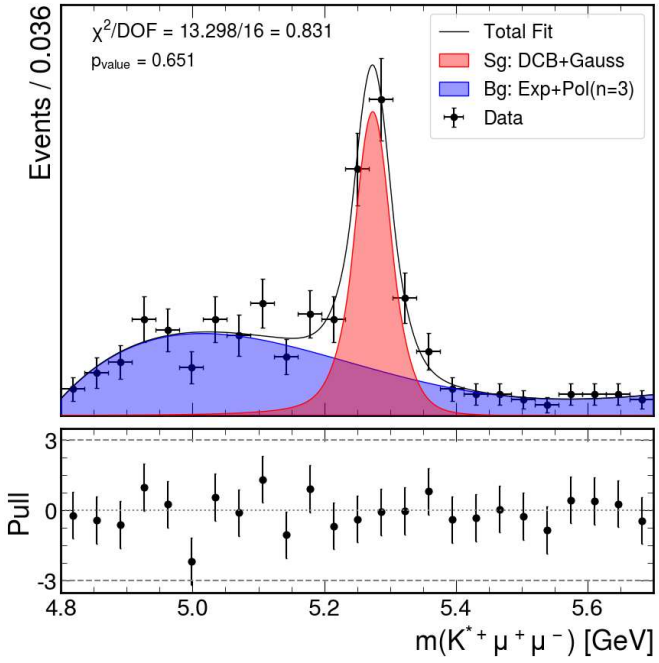


(d) Bin 4

Figure 3.10: Fits of the non-resonant data by bin.

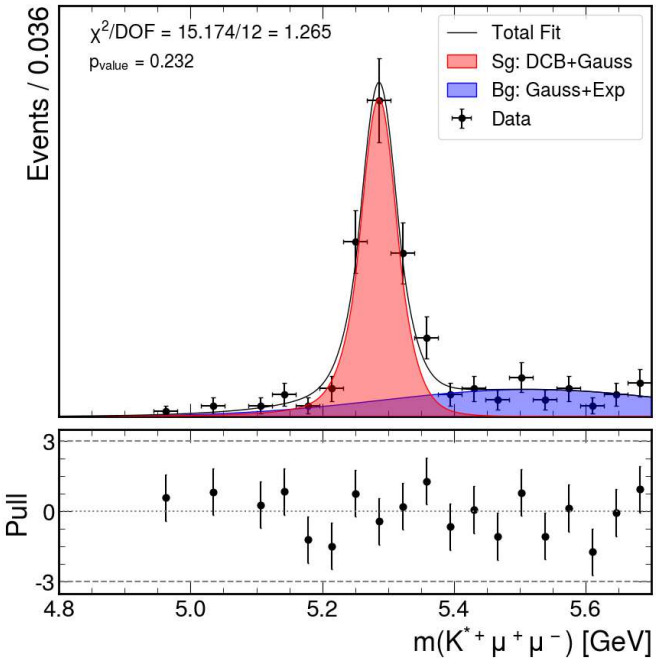


(e) Bin 6



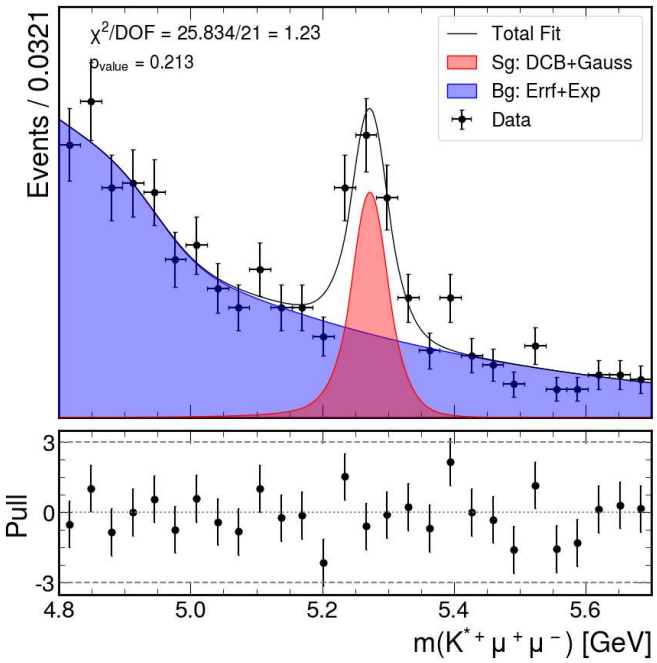
(f) Bin 8

Figure 3.10: Fits of the non-resonant data by bin.

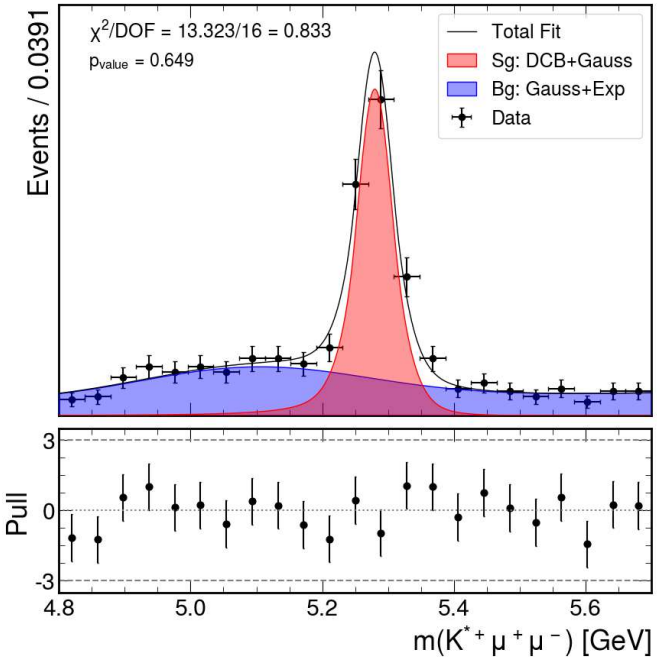


(g) Bin 9

Figure 3.10: Fits of the non-resonant data by bin.

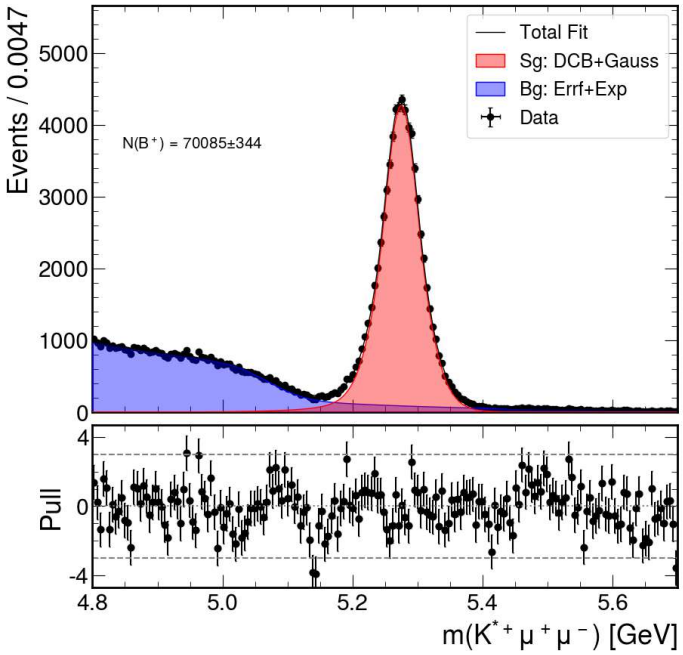


(a) Low q^2

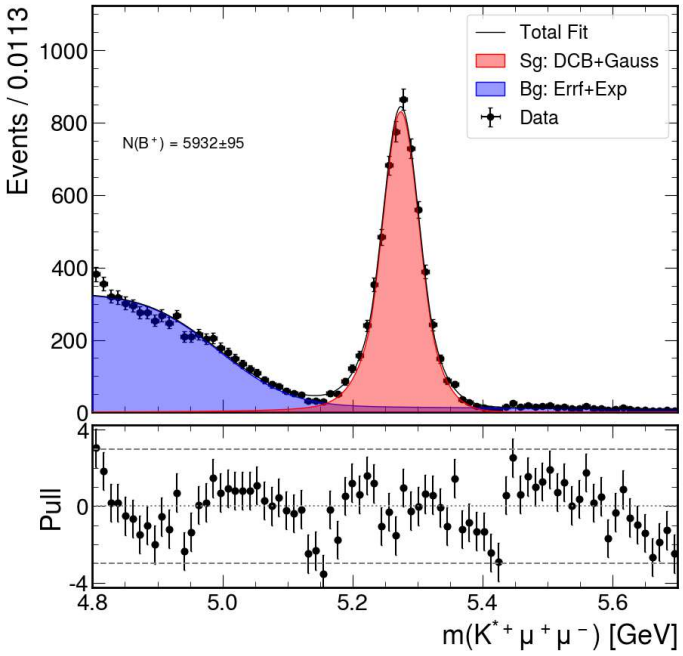


(b) High q^2

Figure 3.11: Fits of the non-resonant data divided in low and high q^2 regions.



(a) J/ψ Resonant Channel



(b) $\psi(2S)$ Resonant Channel

Figure 3.12: Fits of the data of each resonant channel.

Chapter 4

Efficiencies

4.1 Efficiency Determination

As we can see in Equation 1.1, the number of events depends of the efficiency, that in this case it tells us from all the produced events of our decay how many of them get to be counted.

$$\varepsilon = \frac{N_{events}(Reconstructed\ B\ mesons\ after\ the\ full\ selection)}{N_{events}(Produced\ B\ mesons\ in\ the\ collitions)} \tag{4.1}$$

The first thing these events had to cross in order to be possibly counted was to have the minimum requirements that our detector can measure. After that, the events had to make their way through all the filters and selections we discussed in the previous sections.

It is convenient to split the total efficiency in two terms, one of them is called the pre-filter efficiency (ε_1), this measures the efficiency of the generator pre-filter (acceptance), the filters were those showed in Table 2.2 and so ε_1 is defined as:

$$\varepsilon_1 = \frac{N_{events}(B\ mesons\ after\ the\ gen\ filter\ cuts)}{N_{events}(Produced\ B\ mesons\ in\ the\ collitions)} \tag{4.2}$$

The second part of the total efficiency is the reconstruction efficiency (ε_2), this takes into account the triggers, the reconstruction and the selection the events (that have already passed the gen filters) underwent.

$$\varepsilon_2 = \frac{N_{events}(Reconstructed\ B\ mesons\ after\ the\ full\ selection)}{N_{events}(B\ mesons\ after\ the\ gen\ filter\ cuts)} \tag{4.3}$$

This way, the total efficiency can be writen in terms of ε_1 and ε_2 as:

$$\varepsilon = \varepsilon_1 \times \varepsilon_2 \tag{4.4}$$

In the following subsections each component is described with more detail.

4.1.1 Generator-Level Efficiency

The generator-level efficiency provides an estimate of how acceptance affects the counting of b candidates in this measurement. This efficiency is calculated by taking the ratio of the number of events that pass the so-called gen-filter cuts to the total number of events generated. To get these number of events we used the MC only-gen samples that we detailed in section 2, we simply counted the events in the sample and then applied the filters specified in Table 2.2 and counted again the remained events. If we define the bin limits as $[X_1^i, X_2^i]$, were the values of those Xs are given in Table 1.1, the acceptance for the i-th q^2 bin, is defined as follows:

$$\varepsilon_1^i = \frac{N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts})}{N(B^+; X_1^i < q^2_{gen} < X_2^i)} \tag{4.5}$$

The uncertainty of this quantity was computed as:

$$\Delta\varepsilon_1^i = \varepsilon_1^i \sqrt{\left(\frac{\Delta N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts})}{N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts})}\right)^2 + \left(\frac{\Delta N(B^+; X_1^i < q^2_{gen} < X_2^i)}{N(B^+; X_1^i < q^2_{gen} < X_2^i)}\right)^2} \tag{4.6}$$

where $\Delta N = \sqrt{N}$ is the counting error.

4.1.2 Reconstruction Efficiency

The second part of the total efficiency is calculated using MC samples that have the pre-filter cuts already applied. The ratio of the number of fully selected, reconstructed events to the number of generated decays within the appropriate binning is known as the reconstruction efficiency. The reconstruction efficiency for the i-th q^2 bin, is given by:

$$\varepsilon_2^i = \frac{N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts, full selection})}{N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts})} \tag{4.7}$$

The uncertainty was also computed as:

$$\begin{aligned}\Delta\varepsilon_2^i =&\varepsilon_2^i\Bigg[\left(\frac{\Delta N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts, full selection})}{N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts, full selection})}\right)^2 + \\ &+\left(\frac{\Delta N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts})}{N(B^+; X_1^i < q^2_{gen} < X_2^i, \text{gen-filter cuts})}\right)^2\Bigg]^{\frac{1}{2}}\end{aligned} \tag{4.8}$$

4.2 Total Efficiency

The total efficiency was computed using Equation 4.4, while its correspondent uncertainty with:

$$\Delta\varepsilon = \varepsilon\sqrt{\left(\frac{\Delta\varepsilon_1^i}{\varepsilon_1^i}\right)^2 + \left(\frac{\Delta\varepsilon_2^i}{\varepsilon_2^i}\right)^2} \tag{4.9}$$

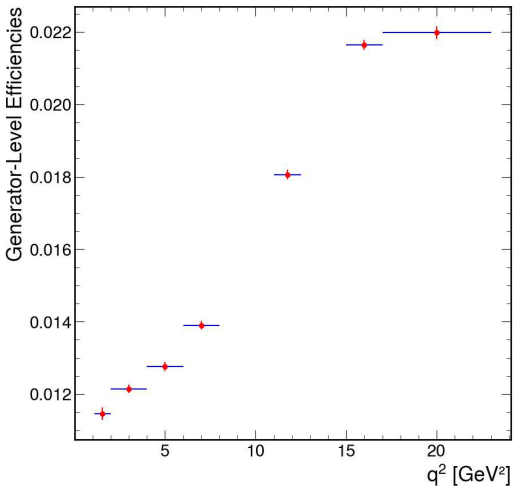
The results for resonant and non-resonant channels are showed in detail in Tables 4.1 and 4.2, we can also graphically see how the non-resonant Prefilter, Reconstruction and Total Efficiencies distributed in Figure 4.1.

q^2 Bin Number	Pre-Filter	Reco	Total
5 (J/ψ)	0.015622 ± 0.000037	0.028317 ± 0.000048	0.000442 ± 0.000001
7 ($\psi(2S)$)	0.018943 ± 0.000041	0.029161 ± 0.000046	0.000552 ± 0.000001

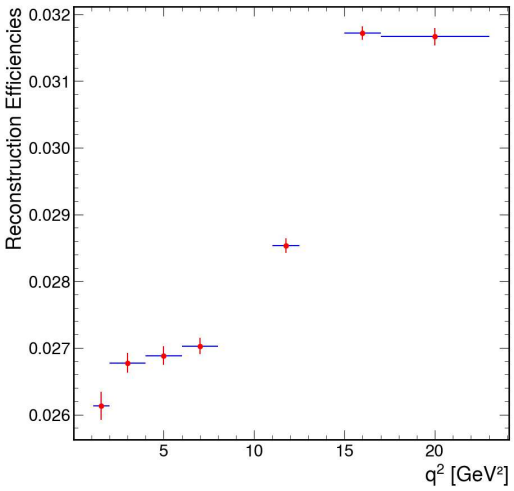
Table 4.1: Prefilter, Reconstruction and Total Efficiencies values by q^2 bin for the resonant channels

q^2 Bin Number	Pre-Filter	Reco	Total
1	0.011457 ± 0.000164	0.026134 ± 0.000210	0.000299 ± 0.000005
2	0.012147 ± 0.000121	0.026772 ± 0.000147	0.000325 ± 0.000004
3	0.012762 ± 0.000117	0.026885 ± 0.000137	0.000343 ± 0.000004
4	0.013901 ± 0.000111	0.027028 ± 0.000119	0.000375 ± 0.000003
6	0.018058 ± 0.000133	0.028534 ± 0.000113	0.000515 ± 0.000004
8	0.021647 ± 0.000140	0.031719 ± 0.000105	0.000686 ± 0.000005
9	0.021980 ± 0.000180	0.031665 ± 0.000132	0.000696 ± 0.000006
Low q^2	0.012273 ± 0.000075	0.026706 ± 0.000090	0.000328 ± 0.000003
High q^2	0.021774 ± 0.000111	0.031698 ± 0.000082	0.000690 ± 0.000004

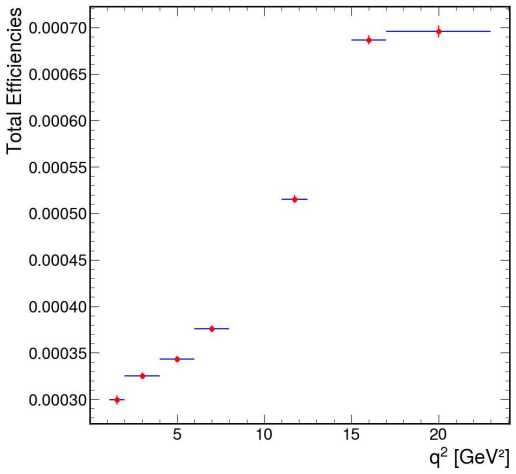
Table 4.2: Prefilter, Reconstruction and Total Efficiencies values by q^2 bin for the non-resonant channels



(a) Generator-Level Efficiencies



(b) Reconstruction Efficiencies



(c) Total Efficiencies

Figure 4.1: Values of the Efficiencies for the different bins of the Non-Resonant Channel

Chapter 5

Results

5.1 Measurement of the Branching Fraction Ratio $\mathbf{BF(B^+ \to K^{*+}\psi(2S))/BF(B^+ \to K^{*+}J/\psi)}$

As an intermediate step, we present a measurement of the ratio of the $B^+ \to K^{*+}\psi(2S)$ to $B^+ \to K^{*+}J/\psi$ branching fractions, we used Equation 1.1 to get the expression:

$$\mathcal{R}_{\psi(2S)} = \frac{BF(B^+ \to K^{*+}\psi(2S))}{BF(B^+ \to K^{*+}J/\psi)} = \frac{N_{\psi(2S)}}{N_{J/\psi}} \frac{\varepsilon_{J/\psi}}{\varepsilon_{\psi(2S)}} \frac{BF(J/\psi \to \mu^+\mu^-)}{BF(\psi(2S) \to \mu^+\mu^-)} \tag{5.1}$$

where $N_{\psi(2S)}/N_{J/\psi}$ is the ratio between the measured yields of $B^+ \to K^{*+}\psi(2S)$ and $B^+ \to K^{*+}J/\psi$ decay channels, $\varepsilon_{J/\psi}/\varepsilon_{\psi(2S)}$ the ratio between the total efficiency of the two channels, and finally $BF(J/\psi \to \mu^+\mu^-) = (5.961 \pm 0.033) \times 10^{-2}$ and $BF(\psi(2S) \to \mu^+\mu^-) = (8.0 \pm 0.6) \times 10^{-3}$ according to the world-average results [12]. Using the values obtained in previous sections, the value of our ratio is:

$$\mathcal{R}_{\psi(2S)} = 0.505 \pm 0.039$$

It is worth mention that the PDG values for the branching fractions are $BF(B^+ \to K^{*+}\psi(2S)) = (6.7 \pm 1.4) \times 10^{-4}$ and $BF(J/\psi \to \mu^+\mu^-) = (1.43 \pm 0.08) \times 10^{-3}$, so the ratio between these is:

$$\mathcal{R}^{PDG}_{\psi(2S)} = 0.469 \pm 0.101$$

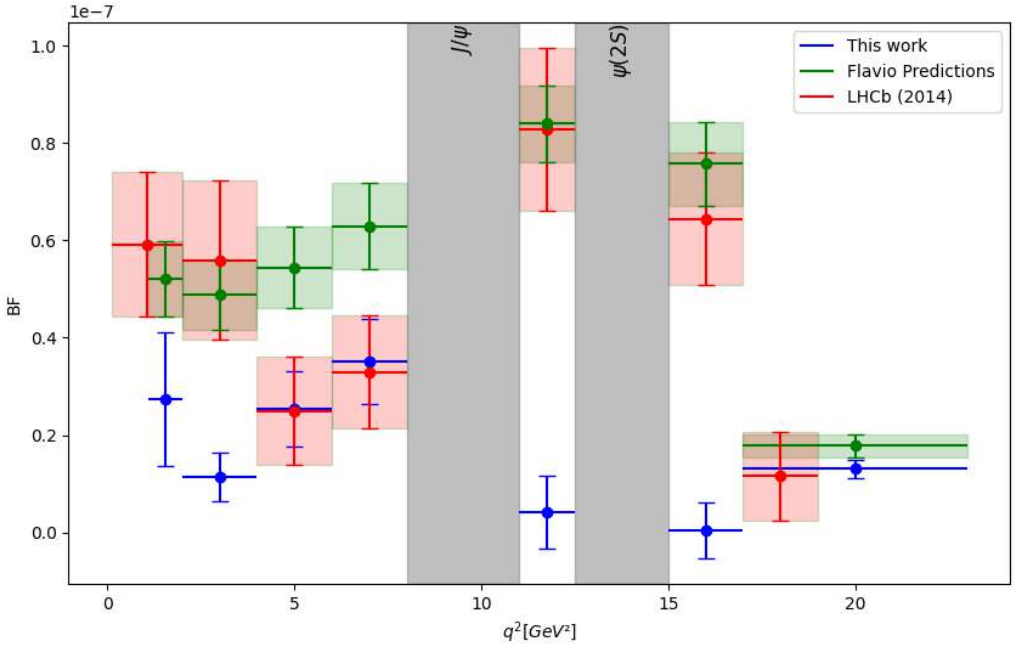
These measurements are consistent within less than a sigma, this is an important cross validation that shows the robustness of the analysis.

5.2 Differential Branching Fraction

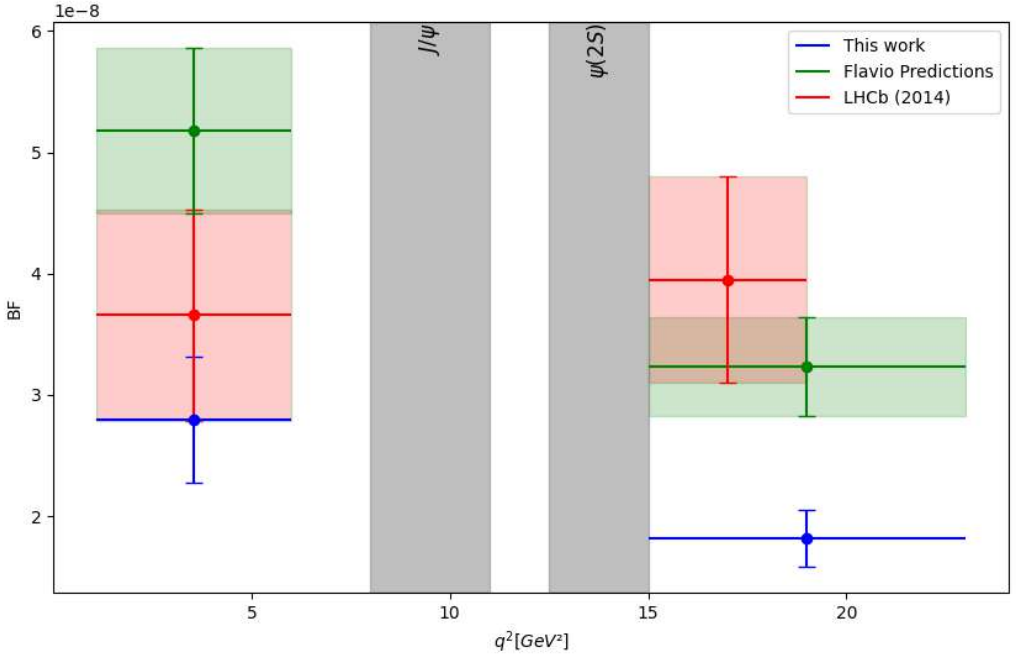
At this point we had everything that was needed to compute the Differential Branching Fractions using the Equation 1.2, it is important to remember that the N_{evts} were multiplied by a random number in the case of the non resonant channel Signal Yields. In Table 5.1 we can see the values we used to compute the branching fractions as well as the values itself, in Figure 5.1 we represent these values and compare them with the values reported by LHCb (Table 1.3) and with some SM predictions we computed using the tool Flavio, which is a Python package for flavour and precision phenomenology in the Standard Model and beyond [29].

q^2 Bin	$N_{\mu\mu}$	$N_{J/\psi}$	$\varepsilon_{\mu\mu}(\times 10^{-4})$	$\varepsilon_{J/\psi}(\times 10^{-4})$	BF
1	14±7	70085±344	2.99±0.05	5.52±0.01	2.743e-08±1.381e-08
2	14±6	70085±344	3.25±0.04	5.52±0.01	1.137e-08±4.915e-09
3	33±10	70085±344	3.43±0.04	5.52±0.01	2.539e-08±7.831e-09
4	50±12	70085±344	3.75±0.03	5.52±0.01	3.513e-08±8.668e-09
6	6±11	70085±344	5.15±0.04	5.52±0.01	4.099e-08±7.518e-09
8	1±15	70085±344	6.86±0.05	5.52±0.01	3.845e-10±5.768e-09
9	103±13	70085±344	6.96±0.06	5.52±0.01	1.302e-08±1.805e-09
Low q^2	85±15	70085±344	3.28±0.03	5.52±0.01	2.795e-08±5.182e-09
High q^2	190±22	70085±344	6.90±0.04	5.52±0.01	1.817e-08±3.343e-09

Table 5.1: Branching Fractions by Bin



(a) Enumerated bins.



(b) Low and High q^2 bins.

Figure 5.1: Comparison of the obtained Blinded BF with the LHCb(2014) measurement and the Flavio prediction.

5.3 Conclusions

The measurements presented in this work will provide more accurate values compared to those conducted by LHCb using data from 2011 and 2012. This improvement is due to the higher statistics available and the fact that the energy of the collisions we used has almost doubled compared to that from 10 years ago.

Even though the results presented here are blinded, the consistency of the ratio between the resonant channels with the PDG values is an indicator of the reliability of the various steps in the analysis.

To provide an even better measurement, we are planning to incorporate data collected in 2023. This should be a straightforward task, as all the techniques and steps have already been developed.
Due to the interest in this decay channel, the next goal is to begin an angular analysis, where previous studies have shown some tensions with the predictions of the Standard Model, making it a promising place to search for new physics.

This work plays a significant role in testing the limits of the SM. By refining the measurement of the Branching Fraction we contribute to either confirm or challenge the existing theoretical models.

Bibliography

[1] Cliff Burgess and Guy Moore. *The Standard Model: A Primer*. Cambridge University Press, 2007.

[2] David Griffiths. *Introduction to elementary particles, 2nd revised edition*. John Wiley & Sons, 2008.

[3] Peter W Higgs. Broken symmetries and the masses of gauge bosons. *Physical review letters*, 13(16):508, 1964.

[4] Atlas Collaboration et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *arXiv preprint arXiv:1207.7214*, 2012.

[5] Cms Collaboration et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *arXiv preprint arXiv:1207.7235*, 2012.

[6] Cern public webpage. 2024. `https://home.cern/science/physics/standard-model`. Accessed: 2024-08-30.

[7] Sheldon L Glashow, Jean Iliopoulos, and Luciano Maiani. Weak interactions with lepton-hadron symmetry. *Physical review D*, 2(7):1285, 1970.

[8] Cern public webpage. 2024. `https://home.cern/science/accelerators`. Accessed: 2024-09-02.

[9] Cern document server. 2022. `https://cds.cern.ch/record/2800984`. Accessed: 2024-08-31.

[10] Cms public webpage. 2024. `https://cms.cern/detector`. Accessed: 2024-09-09.

[11] Cms twiki. 2024. https://twiki.cern.ch/twiki/bin/view/CMSPublic/DPGResultsTRK. Accessed: 2024-09-09.

[12] Particle Data Group, RL Workman, VD Burkert, V Crede, E Klempt, U Thoma, L Tiator, K Agashe, G Aielli, BC Allanach, et al. Review of particle physics. *Progress of theoretical and experimental physics*, 2022(8):083C01, 2022.

[13] Roel Aaij, B Adeva, M Adinolfi, A Affolder, Z Ajaltouni, J Albrecht, F Alessio, M Alexander, S Ali, G Alkhazov, et al. Differential branching fractions and isospin asymmetries of $B \to K(^*)\mu+\mu-$ decays. *Journal of High Energy Physics*, 2014(6):1–22, 2014.

[14] Bernard Aubert, M Bona, Y Karyotakis, JP Lees, V Poireau, E Prencipe, X Prudent, V Tisserand, J Garra Tico, E Grauges, et al. Direct c p, lepton flavor, and isospin asymmetries in the decays $B \to K(^*)l^+l^-$. *Physical review letters*, 102(9):091803, 2009.

[15] J-T Wei and P Chang. Measurement of the differential branching fraction and forward-backword asymmetry for $B \to K(^*)l^+l^-$. *arXiv preprint arXiv:0904.0770*, 2009.

[16] CMS Collaboration. Brilcalc documentation. 2015. CMS internal documentation available at http://cms-service-lumi.web.cern.ch/cms-service-lumi/brilwsdoc.html.

[17] T. Sjöstrand et al. An introduction to pythia 8.2. *Comput. Phys. Commun.*, 191:159, 2015.

[18] D. Lange. The evtgen particle decay simulation package. *Nucl. Instrum. Meth. A*, 462:152, 2001.

[19] T. Przedzinski N. Davidson and Z. Was. Photos interface in C++: Technical and physics documentation. *Comput. Phys. Commun.*, 199:86, 2016.

[20] GEANT4 Collaboration. Geant 4—a simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250, 2003.

[21] L. T. Handoko A. Ali, P. Ball and G. Hiller. Comparative study of the decays $B \to (K, K^*)l^+l^-$ in the standard model and supersymmetric theories. *Phys. Rev. D*, 61(7):074024, Mar 2000.

[22] CMS Collaboration. Performance of the cms muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV. *JINST*, 13:P06015, 2018.

[23] V0producer offline guide. `https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideV0Producer`. Accessed: 2024-07-30.

[24] H. Crotte Ledesma et al. Angular analysis of the decay $B^+ \to K^+\mu^+\mu^-$. *CMS Draft Analysis Note*, (CMS AN-21-020).

[25] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[26] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[27] XGBoost developers. Xgboost documentation. `https://xgboost.readthedocs.io/en/release_1.0.0/index.html`, 2019. Accessed: 2024-08-07.

[28] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR, 2013.

[29] David M Straub. flavio: a python package for flavour and precision phenomenology in the standard model and beyond. *arXiv preprint arXiv:1810.08132*, 2018.