CBM Performance for Λ^0 Hyperon Yield Measurements Using Machine Learning Techniques

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Shahid Khan

aus Swat, Pakistan

Tübingen

2023

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	23.05.2023
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Hans Rudolf Schmidt
2. Berichterstatter:	Prof. Dr. Joseph Jochum

Dedicated to my mother,

who values education more than anything else

Abstract

The phase diagram of the QCD matter in the baryon chemical potential region $500\,\mathrm{MeV} \leq \mu_B \leq 800\,\mathrm{MeV}$ will be studied by the future Compressed Baryonic Matter (CBM) experiment in the beam energy range corresponding to $\sqrt{s_{NN}} = 2.9 - 4.9\,\mathrm{GeV}$. The experiment will be carried out at the Facility for Anti-Proton and Ion Research. A prerequisite for determining the properties of dense baryonic matter is the multi-differential measurement of the yield of (multi-) strange hadrons.

This work evaluates the performance of CBM in measuring the multi-differential (charged tracks multiplicity, transverse momentum, and rapidity) yield of the most abundantly produced Λ^0 baryon for Au-Au collisions at a beam momentum of 12 A GeV/c. The Kalman Filter algorithm is employed to reconstruct the Λ^0 baryon through its weak decay to a proton and π^- topology, which is selected non-linearly using the machine learning algorithm eXtreme Gradient Boosting (XGBoost). The selection is performed multi-differentially in transverse momentum and rapidity for the multiplicity interval $[200, 400]$ of charged tracks to achieve a high signal-to-background ratio.

After the selection, raw-yield extraction is performed multi-differentially through a multi-step fitting routine. The extracted raw-yield is corrected for the efficiency of the reconstruction and selection procedure and the geometrical acceptance of the experiment. The corrected yield is compared to the true yield to validate the reconstruction, selection, yield extraction, and correction procedure. The systematic uncertainties are evaluated by varying the selection parameters and they are typically below 3% but can go up to 6% for high transverse momentum intervals.

Zusammenfassung

Das Phasendiagramm der QCD-Materie im Bereich des chemischen Baryonenpotenzials $500\,\mathrm{MeV} \leq \mu_B \leq 800\,\mathrm{MeV}$ wird mit dem künftigen Experiment für komprimierte baryonische Materie (CBM) im Strahlenergiebereich von $\sqrt{s_{NN}} = 2.9 - 4.9\,\mathrm{GeV}$ untersucht. Das Experiment wird in der Facility for Anti-Proton and Ion Research (FAIR) durchgeführt. Eine Voraussetzung für die Bestimmung der Eigenschaften dichter baryonischer Materie ist die multidifferenzielle Messung der Ausbeute an Hadronen mit einem oder mehreren Strange-Quarks.

In dieser Arbeit wird die Leistungfähigkeit von CBM bei der Messung der multidifferenziellen (Multiplizität der geladenen Spuren, Transversalimpuls und Geschwindigkeit) Ausbeute des am häufigsten produzierten Λ-Baryons für Au-Au-Kollisionen bei einem Strahlimpuls von 12 A GeV/c. Der Kalman-Filter-Algorithmus wird eingesetzt, um das Λ-Baryon aufgrund der Topolgie des schwachens Zerfall in ein Proton und die π^- zu rekonstruieren. Mögliche Λ-Kandidaten werden dem maschinellen Lernalgorithmus eXtreme Gradient Boosting (XGBoost) ausgewählt. Die Auswahl erfolgt multidifferenziell nach transversalem Impuls und Geschwindigkeit für das Multiplizitätsintervall $[200, 400]$ geladener Spuren, um ein hohes Signal-zu-Hintergrund-Verhältnis zu erreichen.

Nach der Auswahl wird die Ausbeute durch eine mehrstufige Fitroutine multidifferenziell extrahiert. Diese Ausbeute wird für die Effizienz des Rekonstruktions und Auswahlverfahrens und die geometrische Akzeptanz des Experiments korrigiert. Die korrigierte Ausbeute wird mit der wahren Ausbeute verglichen, um die Rekonstruktion, die Auswahl, die Extraktion der Ausbeute und das Korrekturverfahren zu validieren. Die systematischen Unsicherheiten werden durch Variation der Auswahlparameter ausgewertet und liegen typischerweise unter 3%, können aber bei hohen Transversalimpulsintervallen bis zu 6% betragen.

Contents

Chapter 1

Introduction

The matter that surrounds us in our daily lives is mostly composed of molecules, which are tiny building blocks made up of atoms. An atom is not the end of the story and it has been found to contain sub-atomic particles such as electrons and nucleons (protons and neutrons). Hitherto, the electrons are considered to be structureless particles but the nucleons are composed of fundamental particles called quarks and gluons. The quarks and gluons interact with each other with a strong force. The set of rules that dictate the behavior of the strong interaction among quarks and gluons is called Quantum-chromo dynamics (QCD). QCD is a quantum field theory and its Lagrangian can be written as:

$$\mathcal{L}_{QCD} = \bar{\psi}_{fl}(i\gamma^{\mu}\mathcal{D}_{\mu} - m_q)\psi_{fl} - \frac{1}{4}G^a_{\mu\nu}G^{\mu\nu}_a \tag{1.1}$$

with ψ_{fl} representing the quark fields and the subscript fl representing their flavors [1]. The γ^{μ} are the Dirac gamma matrices and the co-variant derivative ($\mathcal{D}$) can be expressed as $\mathcal{D}_{\mu} = \partial_{\mu} - \imath g\mathcal{A}^a_{\mu}\lambda_a$. It contains the SU(3), Special Unitary group in 3 dimensions, color symmetric gluon field $\mathcal{A}^a_{\mu}$ with a as a color index and goes from 1 and 8. The Gell-Mann matrices are represented by λ_a and the non-Abelian gauge field strength tensor by $G^a_{\mu\nu}$. While g is related to the strong coupling constant $\alpha_s = g^2/4\pi$, m_q represents the mass of an individual quark.

A property of QCD, asymptotic freedom, reveals that α_s gets weaker at short distances or at large momentum transfer [2, 3]. Another property is that a single quark cannot exist freely. At high energies, due to asymptotic freedom, eq. 1.1 can be solved using perturbative approaches while at lower energies it becomes

non-perturbative and can only be solved by numerical methods such as lattice QCD and effective field theories.

1.1 QCD-Matter Phase Diagram

QCD matter is generally composed of hadrons, quarks, and gluons, which at low temperature and baryon chemical potential μ_B may exist as a pions-dominated hadron gas [4]. The value of μ_B represents the disparity between matter and antimatter, and a zero value implies an ideal balance between the two. At higher temperatures ($T \gtrsim 130\,\text{MeV}$) and small μ_B, the hadron gas can undergo a phase transition to a quark-gluon plasma (QGP), where quarks and colored gluons are deconfined, and the number of effective degrees of freedom increases.

Lattice QCD calculations predict that the phase transition from hadronic matter to QGP is a smooth crossover, i.e., without any abrupt changes when $\mu_B \approx 0$ [5]. However, if $\mu_B/T \gtrsim 2$ then various theoretical approaches predict that the phase transition can become first-order, exhibiting an abrupt change [6]. Unfortunately, lattice QCD calculations fail to predict the value of μ_B and the temperature at which the first-order phase transition begins, known as the critical point [7].

The QCD matter phase diagram maps the different phases of QCD matter as a function of μ_B and temperature as shown in Fig. 1.1. The white-colored curve represents two co-existing phases separated by a first-order phase transition. Along this curve, state variables are discontinuous and become continuous again at the critical point. The discontinuity originates from the fact that the entropy is higher at the QGP side due to the availability of more degrees of freedom. In this diagram, the first-order phase boundary between the hadron gas and QGP ends at a temperature near 150 MeV and μ_B around 400 MeV, which is the critical point. Various theoretical model calculations [6] have placed the critical point at this μ_B. Below this region, the transition from a hadron gas to QGP is represented by a blurry region representing the continuous region, smooth cross-over, of second-order phase change.

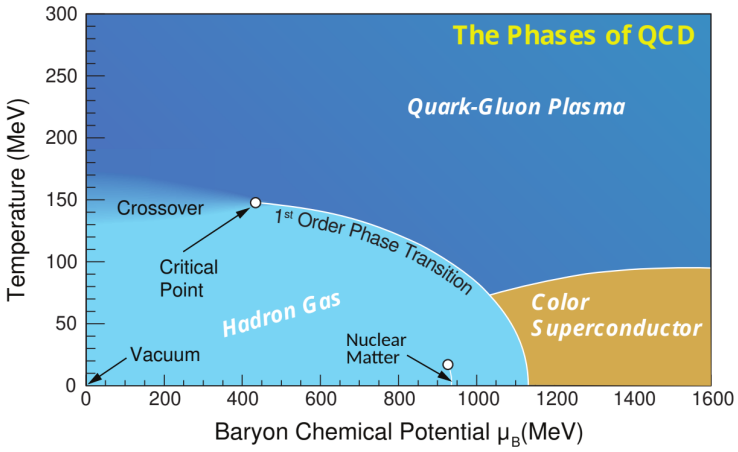


Figure 1.1: The diagram shows different phases of the QCD matter. The image has been taken from [4]. The three phases of the matter, i.e., hadron (light blue) gas, QGP (dark blue), and color superconductivity (brownish yellow) are shown with different colors.

1.2 Heavy-Ions Collisions

The naturally found nuclear matter on earth represents a small part of the QCD matter phase diagram, i.e., at finite μ_B at $T \approx 0$. The extreme environments required to study QCD matter at its various phases are naturally found in neutron star mergers, in the cores of neutron stars, and in the micro-seconds old universe [8]. Those conditions of high energy densities and temperatures can be also made on earth by heavy-ion collisions (HIC) at various energies. Different regions of the phase diagram can be studied by varying the energy of the collision and the size of the colliding nuclei. For example, nuclei collisions at the center of mass energy in the $\sqrt{s_{NN}} \sim$ TeV energies can be used to study the phase diagram near the vanishing μ_B region. Similarly, with heavier nuclei and lower beam energies corresponding to $\sqrt{s_{NN}} \sim 2 - 3\,\text{GeV}$ the high μ_B region can be probed.

When heavy ions are accelerated close to the speed of light, the nuclei get Lorentz contracted and their properties such as the collective dynamics can be explained by a framework called the color-glass-condensate (CGC) [9]. Time dilation makes the color sources (quarks and gluons) static at the time scale of the strong force. The collision of the nuclei leads to the creation of a fireball, which is not in equilibrium. Eventually, partons are created from the fireball, and their collisions with each other result in a locally thermalized phase, i.e., the QGP. After

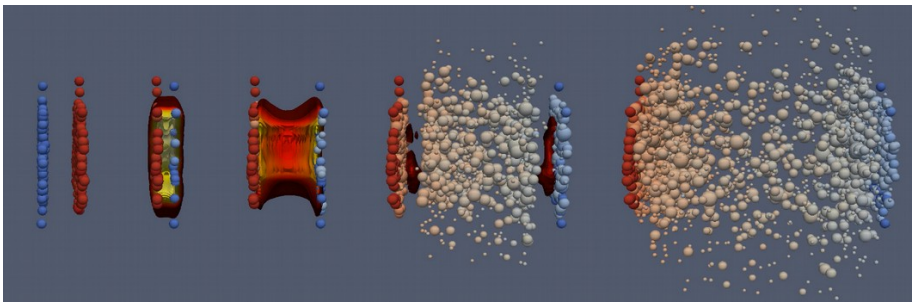


Figure 1.2: The image shows the different stages of the heavy ions collision; the evolution proceeds from left to right. The cartoon has been taken from the MADAI collaboration, Hannah Petersen and Jonah Bernhard [11].

rapid expansion and cooling [10] hadron formation starts and when inelastic collisions vanish their production stops, marking the chemical freeze-out. However, elastic collision among hadrons continues and after some time, this also stops at the kinetic freeze-out. A diagram of the evolution of the collision is shown in Fig. 1.2. On the left side, two nuclei in red and blue color are shown and upon their collision, a pre-equilibrium phase is formed. During equilibration, QGP is formed and is shown by the red phase between the blue and red nucleons. Particle formation is shown in the next two stages of the diagram where elastic and inelastic processes still carry on. This discussion is valid for the $\mu_B \approx 0$ case. At higher μ_B the nuclei are not so much lorentz contracted and the color-glass condensate frame-work may not be applicable.

Various messengers from the different stages of the collision can be used to study its evolution in time. For example, spectra of the emitted particles can reveal information about quantities such as energy density, pressure, and entropy. Similarly, information about the thermodynamics of the system can be extracted from observables such as flow, correlations, fluctuations, particle ratios, etc. Dileptons and photons, i.e., electromagnetic probes, produced at different stages of evolution can escape the medium without interactions and can reveal information about the medium at the time of their production. Observable associated with the QGP phase ($\mu_B \approx 0$ case) is the suppression of the production of J/ψ because cc^- bonding is suppressed due to Debye screening [12]. In the high μ_B region, the extent of equilibration of the fireball can be investigated by measuring excitation functions of (multi)-strange hyperons in A+A collision with

different mass numbers A. This will lead to finding a signal for the onset of deconfinement in QCD matter at high μ_B [13].

To calculate the various particle yields based on theoretical models, collision generators are required. These simulators will give predictions for experiments to improve our understanding of the phases of QCD matter.

1.2.1 Collision Simulators

To describe the results obtained from heavy-ion collisions, various theoretical models have been put forward based on various hypotheses. For example, thermal models [14, 15] hypothesize that global thermodynamic chemical equilibrium is reached. Assuming a thermal source, these models predict the production yields of hadrons at chemical freeze-out. Similarly, hydrodynamic modeling [16] requires a local thermal equilibrium and treats the matter created in the collision collectively like a fluid. This modeling predicts the production yields and spectra of various particle species. The partition function in the case of the thermal model and the equation-of-state (EoS) in the hydrodynamical case drives the physical processes of the collision. However, modern and up-to-date transport models use various assumptions and different models to simulate heavy-ion collisions. Two transport models are used in this work and they are briefly introduced in this section.

The Ultrarelativistic Quantum Molecular Dynamics (UrQMD) [17] is a Monte Carlo event generator that can simulate the collision of a proton with a proton, a proton with a nucleus, and a nucleus with a nucleus at energies ranging from SIS (SchwerIonenSynchrotron) [18] to Large Hadron Collider (LHC)[19] energies, i.e., from a few GeV to TeV. There are 70 different baryons and 39 types of mesons in the model [20]. The cascade mode of the model is established on the propagation of hadrons and re-scattering among hadrons is allowed. This mode offers the solution for the relativistic Boltzmann equation.

The Dubna Cascade Model with the Quark Gluon String Model (DCM-QGSM) and the Statistical Multi-fragmentation Model (SMM) as afterburner [21] is another Monte Carlo event generator. For energies, lower than 1 GeV, the model only considers nucleons, pions, and deltas for the collision dynamics. In compar-

ison, above 10 GeV QGSM is used to describe elementary hadron collisions.

The difference between the two models is that in UrQMD spectators are protons and neutrons, while DCM-QGSM-SMM (DCM) contains spectator fragments and uses coalescence for their formation.

1.2.2 Experimental Facilities to Study the QCD Matter

To study the different phases of the QCD matter, various heavy-ion collisions experiments have been set up in the past at the CERN SPS facility [22], at BNL Relativistic Heavy Ion Collider (RHIC) [23], at the CERN LHC facility, and at the GSI facility [24]. A new phase of matter named the Quark Gluon Plasma (QGP) was discovered at the CERN SPS program and confirmed at the RHIC facility in the 1990s and 2000s; it contains quarks not bound inside nucleons but moving freely inside a plasma of quarks and gluons [25, 26, 27]. The RHIC experiments showed that the plasma is not a weakly interacting gas of its constituents, i.e., the gluons and the quarks [28]. It is like a strongly interacting opaque fluid. The high transverse momentum (p_T) hadron production was studied at the LHC and it was found that the QGP is so dense that high energetic quarks (a few GeV) traversing it lose a large fraction of their energies [29, 30, 31].

Past experiments at RHIC studied the phase diagram from very low (25 MeV) to high (760 MeV) μ_B with beam energies corresponding to $\sqrt{s_{NN}} = 3 - 200$ GeV. But in the high μ_B region corresponding to $\sqrt{s_{NN}} = 3 - 13.7$ GeV, except the 3 GeV, the mid-rapidity region, which is important for physics, was only partially covered by the detector's geometrical acceptance [32]. The high μ_B part at lower energies is challenging and to study the signatures of the phase transition with high detail, high statistics are required. Therefore, new fixed target experiments such as the Compressed Baryonic Matter (CBM) at the Facility for Anti-Proton and Ion Research (FAIR) are being built to solve the issue. To accumulate the required statistics CBM needs to operate at an unprecedented interaction rate of up to 10 MHz.

1.2.3 Strangeness Enhancement as a QGP Observable

Strange quarks do not exist in nuclear matter and are only produced in extreme conditions such as the collision of high-energy heavy ions. A strange quark ($mass \sim 93\,\text{MeV}$) is lighter than heavy quarks, i.e., charm, beauty, and top but is comparatively heavier than light quarks (up and down) [33]. The strangeness quantum number for the strange quark is -1 and is considered to be conserved in strong interaction while it is violated in the weak interaction.

One of the earliest signatures put forward for the formation of QGP was strangeness enhancement [34, 35]. A notable difference between A-A collisions and proton-proton collisions is the chemical equilibration of the production of strange quarks [36, 37]. During the hadronization stage of the QGP, hadrons are formed and the total yield of hadrons containing strange quarks can be measured over a considerable kinematic domain to get insights into the QGP creation and its subsequent evolution [38]. The gluon component of the QGP creates pairs of quarks [39] and their high density at the time of hadronization produces multi-strange hadrons: which are rarely produced in the case of hadron collisions.

Therefore, for the study of the different phases of the QCD matter the study of hadrons containing strange quarks is required. Due to their short lifetimes, particles that contain strange quarks decay near their production point and can be identified and reconstructed by analyzing their daughter particles. The experimental yield for different particle species, especially ones containing short-lived strange quarks, is used to improve models of collision simulations.

1.3 The Facility for Anti-Proton and Ion Research

The GSI Helmholtz Centre for Heavy Ion Research in Darmstadt, Germany has successfully hosted heavy-ion experiments such as High Acceptance Di-Electron Spectrometer (HADES) [40, 41], FOPI [42], and A Large Acceptance DIpole magNet (ALADIN) [43] with its UNILAC (the linear accelerator) and SIS18 accelerator facility. The FAIR facility, next to the GSI facility, will host the heavy-ion synchrotron SIS100 and experiments such as the CBM experiment. Fig. 1.3 shows the GSI facility with blue color and the future FAIR facility with red color.

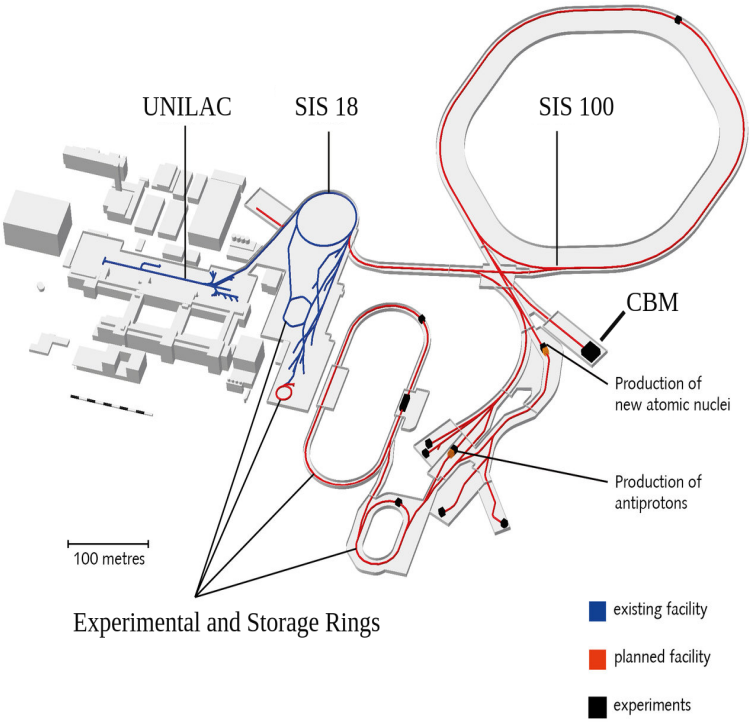


Figure 1.3: The image shows the existing GSI facility (blue color), with blue color, and the future FAIR facility (red), with red color. The CAD drawing has been taken from [46].

The SIS100 accelerator will have super-conducting magnets that will provide magnetic rigidity up to 100 T m [44]. The SIS100, having a radius of around 175 m, will accelerate protons up to 29 GeV beam energy at a rate of 10^9/s [45]. It will also accelerate heavy ions such as Au (other ions such as C and Ca) up to 11 AGeV[1] (14 AGeV). This will enable experiments such as CBM to collide Au nuclei on the Au target to achieve an interaction rate up to 10 MHz at a maximum beam energy corresponding to $\sqrt{s_{NN}} = 4.9$ GeV.

1.4 The CBM Experiment

Microscopic models (transport and hydrodynamic) predict that the density, in the center of an Au-Au collision, is more than 8 times the saturation density ρ_0 at a beam energy of 10 AGeV [47, 48]. Rare probes need to be employed to study the QCD matter at high μ_B, i.e., at conditions with more baryons than anti-baryons. To acquire enough rare probes, one needs to operate at a high beam-target in-

[1] A is for energy per nucleon

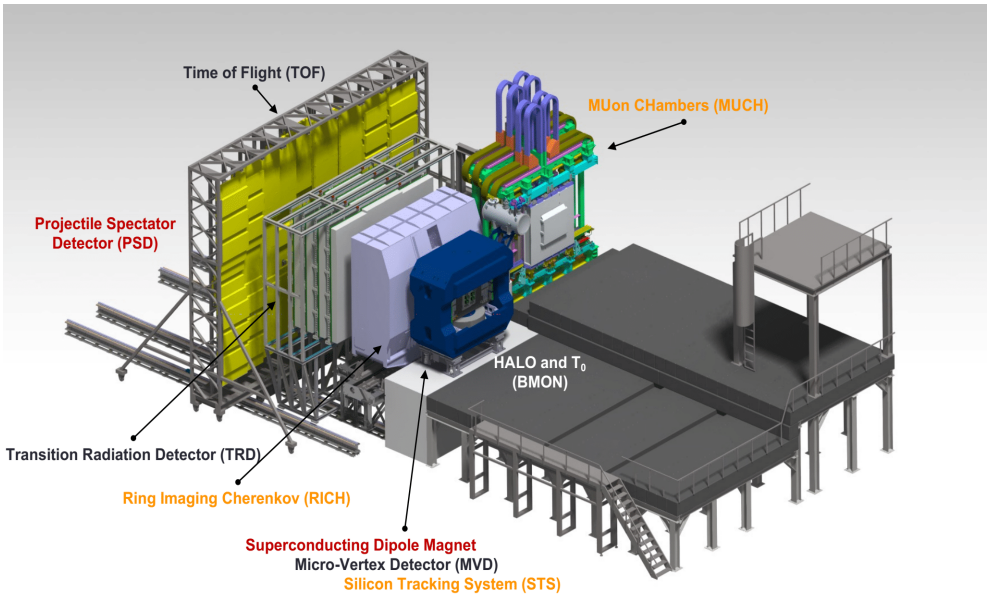


Figure 1.4: The CAD model of the CBM experiment is shown here. The drawing is from the CBM collaboration [49].

teraction rate. A multi-purpose detector with fixed target geometry is ideal for such a scenario. CBM is a fixed target forward geometry experiment designed to operate at high interaction rates, up to 10 MHz [13]. This constrains the detector material budget to be low to not generate secondary particles and the detector to be resilient to radiation.

Due to the operation at a high interaction rate, CBM can detect and measure the yields of hadrons, electrons, and muons at $p - p$ and HIC, with high statistics. The detected charged particles can then be used to reconstruct short-lived mother particles. Due to high statistics, the yields of rare probes will be analyzed multi-differentially. This will require to understand the systematics uncertainties at high precision and reduce them. The experimental setup contains tracking, particle identification, centrality and reaction plane determination systems along with the target. The CAD drawing of the different sub-systems of the CBM detector is shown in Fig. 1.4.

The target for the Au-Au collision will be a segmented one and before it, there will be a diamond beam monitoring (BMON) counter with less than 50 ps time precision. It will give the initial time t_0 of the collision. After the target, the magnet will be placed housing the tracking system, shown by the blue color in Fig. 1.4. It will be followed by different particle identification sub-systems such as the

Ring Imaging Cherenkov detector for electrons identification, shown in Fig. 1.4 by the purple haze color box after the blue magnet. Further electron identification will be achieved by the transition radiation sub-system which lies behind it. The Muon chamber will be used in the alternative setup to measure muons and it is stationed on the right side of the Ring Imaging Cherenkov detector in Fig. 1.4. The time of flight wall is shown by the yellow color and it will identify hadrons. At the very end along the beamline, the geometry determination detector will be placed and it will be followed by a beam dump.

The tracking system will be inside a superconducting dipole magnet of 1 T m field strength with an aperture of $\pm 25^{\circ}$ polar angle [50]. The magnetic field is perpendicular to the beam direction and bends a charged particle passing through the detector based on its momentum-to-charge ratio. The maximum magnet size will be 4.7 m $\times$ 3.73 m and it should have a minimum aperture size of height 1.47 m and width 3.3 m.

1.4.1 Tracking System

The trajectory of a charged particle passing through a magnetic field is curved and the curvature depends on the momentum-to-charge ratio of the particle and also on the strength of the magnetic field. The interaction of a charged particle with a charge-detecting material, in the form of a hit, inside the magnetic field can be used to find the location of the interaction. The hits can be connected to form tracks and if the magnetic field strength is known then momentum can be calculated from the curvature of the track. The tracking system of the CBM collects hits of traversing charged particles in the detector and it is placed inside a magnetic field strength of 1 T m. It consists of two sub-systems, i.e., the main tracking detector in the form of a Silicon Tracking System and a decay vertex reconstruction detector, a Micro-Vertex Detector.

1.4.1.1 The Micro-Vertex Detector

The Micro Vertex Detector (MVD) of the CBM experiment is required to have a spatial resolution of around 5 µm and a low material budget [51]. The position on the z-axis, $5 - 20$ cm after the target in the beam direction, makes it face

more radiation than any other detector sub-system so it needs to be radiation hard. The MVD will enable CBM to reconstruct tracks with momentum as low as 300 MeV/c. It will help in the reduction of combinatorial background and will be able to resolve the vertices of short-lived particles (hyperons and charm hadrons) that decay very close to the primary vertex. Therefore, the MVD will be placed close to the primary vertex. The CAD rendering of the detector is shown in Fig. 1.5a. The MVD will reside inside the target vacuum box, inside the magnet.

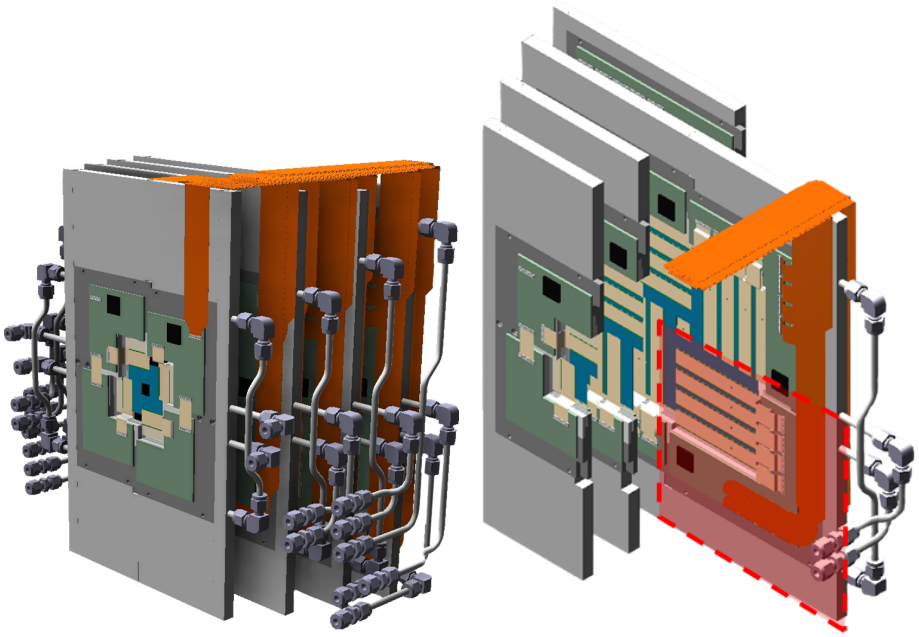
The MVD will have 4 layers of sensors separated by 5 (or 8) cm and the sensors will be CMOS [2] Monolithic Active Pixel Sensors (MAPS) named MIMOSIS. MIMOSIS pixel sensors will be used to make up the MVD detector with each pixel size of 26.9×30.2 µm^2. MIMOSIS sensors were selected for the MVD because of their low material budget, 50 µm thickness, radiation hardness, and 5 µm single-point resolution. A sensor will host a pixel matrix of 1024×504 and will have dimensions of 31×17 mm^2. Each pixel will have an integrated circuit that will amplify, shape, and discriminate the signal. The whole setup will contain up to 300 sensors.

1.4.1.2 The Silicon Tracking System

To reconstruct the tracks of more than 700 charged particles per collision event with momentum resolution of the order of $\Delta p/p^3 < 2\%$, 8 tracking layers of silicon-based detectors are under construction [53, 50]. These 8 layers will be put inside a thermal enclosure in the forward direction of the beam after the target at distances of $30 - 100$ cm and will cover the polar angles between 2.5° and 25°. Minimizing the interaction of charged particles with the sub-system, so that no further charged particles are produced, increases tracking efficiency and improves momentum resolution. Double-sided microstrip sensors (thickness of 320 µm), with a stereo angle of 7.5° will be mounted on carbon fiber support structures. The Front End Electronics will be placed outside the acceptance of the detector along with cooling and mechanical support structures to reduce the material budget. The readout cables, made from multi-layers of polyimide-

[2] Complementary metal–oxide–semiconductor

[3] $\Delta p/p = \frac{p_{reconstructed} - p_{true}}{p_{true}}$



(a) (b)

Figure 1.5: The image shows the drawings of the MVD detector. The left image shows the full MVD setup with its 4 stations, sensors, and cooling pipes (tubes on both sides). The light grey color part consists of the heatsinks and mounting structures. The orange part contains the data and Power Cables. The right side image shows a cross-sectional view with the 3rd station in full showing a module. The image was taken from the MVD section of the CBM website [52].

Aluminium, will transmit analog signals from the sensor to the front-end electronics.

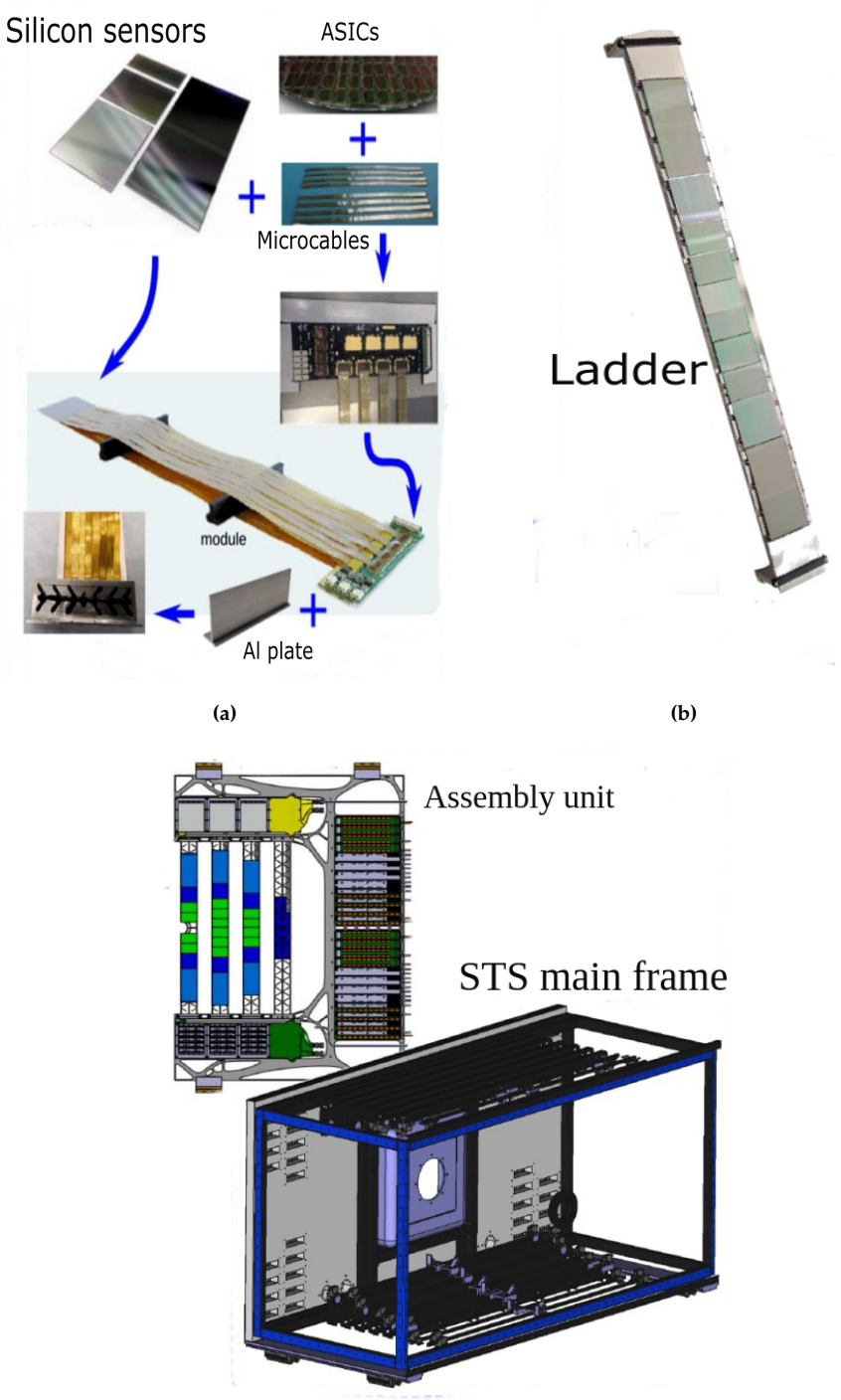
The silicon sensors of the silicon tracking system (STS) are made of n-type float-zone silicon, obtained by vertical zone melting silicon, and implants of p-type material. The thickness of the sensors was chosen as 300 µm to optimize the signal-to-noise ratio and material budget. Thicker sensors produce better signal-to-noise but more multiple scattering as well. The sensors are divided into 1024 strips on each side at a strip pitch of 58 µm. The strip pitch was selected as a trade-off among resolution, the number of readout channels due to the noise of each read-out, and material budget. Smaller polar angles will receive a higher hit density; therefore, the strip length will increase from lower angles to higher angles. The central area of the first two layers of the sensors will constitute sensors with a strip length of 22 mm to get more granularity. The width of the sensors is 62 mm but the height, due to the length of the strips, varies (22, 42, 62 or 124 mm). Each strip has a contact pad for the connection to readout.

The sensors will be connected to the STS-XYTER [4] through microcables to read out, amplify and digitize the signal from the double-sided micro-strip sensors [54, 55]. The sensors (up to 10) and microcables, for readout, are mounted on support structures (called ladders) made of carbon fiber with lengths up to 100 cm, as shown in Fig. 1.6b. This ASIC [5] has 128 channels and to read a single side of a silicon sensor 8 STS-XYTER ASICs are required, which will be put on a single front-end board. There are 106 ladders in total in the STS assembly and 896 STS sensors will be mounted on them. The ladders are then mounted on an Al support frame called C-frame because of its C shape. An assembly unit of STS is shown in Fig. 1.6c consisting of C-frame and electronics boards. Also, Fig. 1.6c shows the main STS frame that will accommodate all the stations.

Simulation studies show that keeping the sensors under $-10\ ^{\circ}C$ will help in preventing the deterioration of the momentum resolution of the detector [56]. To receive equilibrated air flow throughout the detector, the cooling gas will have to flow inside carbon-fiber tubes. Similarly, the front end electronics will also dissipate heat and that can cause the sensors to heat up, so they must also be kept under $-10\ ^{\circ}C$.

[4] Silicon Tracking System-X–Y-Time–Energy Read-out

[5] application-specific integrated circuit



(a) (b)

(c)

Figure 1.6: The image shows the different components of the STS detector setup [57]. (a) The image shows on the top left the double-sided silicon sensors that will be connected through the microcables to the FEBs. (b) Sensors and cables mounted on a ladder. (c) The top image shows an Al C frame to hold the ladders. The FEBs are on the top and bottom of the ladders and the readouts are on the right side. The lower image shows the STS main frame to hold all the layers of the tracking system.

1.4.2 Particle Identification Detectors

Various particle species such as electrons, muons, protons, strange hadrons, and nuclei will be produced by the interaction of the STS-100 beam with the target of the CBM experiment. CBM has various particle identification detectors that can provide particle identification (PID) hypotheses for the measured particles. In this section, the PID sub-systems will be described starting from the one closest to the target along the beam axis.

1.4.2.1 Ring Imaging Cherenkov

Electrons identification is important because the electron-positron pairs in the low-mass ($mass_{e^-e^+} < 1\ \mathrm{GeV/c^2}$) range reveal information about chiral symmetry restoration, the medium mass ($1 < mass_{e^-e^+} < 3\ \mathrm{GeV/c^2}$) range give hints about the fireball temperature and the high mass ($mass_{e^-e^+} > 3\,\mathrm{GeV/c^2}$) is useful for charm quark studies. In the Au-Au collisions of the CBM experiment pions will be produced abundantly and they can be misidentified as electrons. To identify electrons with up to 8A GeV/c and segregate them properly from the pions, the Ring Imaging Cherenkov detector (RICH) [58] is used. The RICH will be placed after the STS and will have a size of $2 \times 4 \times 5$ m [59]. To detect photons, it will have Multi-Anode Photo-Multipliers (MAPMTs) and 80 trapezoidal glass mirror tiles arranged in two half-spheres for focusing the photons. The RICH will use as a radiator CO_2 gas at atmospheric pressure and will have around 28 hits for a single electron cone's ring on the plane containing the MAPMTs. The radius of the cone can be used to filter electrons from other particles.

1.4.2.2 Transition Radiation Detector

The identification of electrons by RICH is restricted to the lower momentum region of $p < 6\,\mathrm{GeV/c}$. Also, the separation of deuterons from ^{4}He is not possible with the CBM primary detector for hadrons identification (the TOF detector), only. The Transition Radiation Detector (TRD) will [60] augment the pion suppression (by 10 to 20 times) in the low-mass, medium-mass, and high-mass regions. It will also help to distinguish nuclei fragments ^{1}H from ^{2}H and ^{4}He. The

TRD will be made from MWPC [6] and will be filled with either Xe/CO_2. The TRD will have 4 layers and will have two different types of modules ($57 \times 57 cm^2$ and $99 \times 99\ cm^2$) and it will be placed after the RICH.

1.4.2.3 The Muon Chamber

For the identification of muons, the Muon CHamber (MuCH) of the CBM experiment will be used [61]. The MuCH will be put behind the STS and will replace the RICH to study di-muons from the decay of low-mass vector mesons and J/Ψ. The MuCH will contain 6 absorbers and tracking stations, to track the particles passing through it. Apart from the first absorber, graphite, and concrete made, the other 5 will be made from Fe with varying thicknesses. For tracking the Gas Electron Multiplier will make up the first two stations because of its resolution and can operate at high rates. The 3rd and 4th stations will be Resistive Plate Chamber based.

1.4.2.4 TOF wall

The Time-Of-Flight (TOF) wall is the main PID sub-detector of the CBM experiment for charged hadrons such as pions, kaons, and protons [62]. It will be constructed using Multi-gap Resistive Plate Chambers and cover an area of $120\,m^2$. The TOF wall will have a time resolution of approximately 80 ps to record the time of charged particles passing through it accurately. Located at a distance of $6 - 10$ m from the target, the TOF wall will face an incident-charged particle flux in the range of $0.1 - 100\ kHz/cm^2$ as a function of angle due to the 10 MHz interaction rate of CBM. Because of the different fluxes, TOF will be modular, and its elements (modules) will be located according to rate requirements. There will be 6 different types of modules and a total of 226 modules will be integrated into the TOF wall.

One can use the time parameter from the time detector t_0 to calculate the mass of a charged particle using the equation:

$$mass^2 = p^2 \left(\frac{t_{TOF}^2}{l^2} - 1 \right) \ (\mathrm{GeV/c^2})^2 \tag{1.2}$$

[6] Multi-Wire Proportional Chambers

with p representing the momentum of the charged track, $t_{TOF} = t_{st} - t_0 - offset$ time between TOF and time detector including offset, l distance, and c the speed of light (constant value). The t_{st} can either come from the TOF wall or another beam monitoring detector, a beam fragmentation detector. Matching a TOF hit to a corresponding track in the tracking system can provide the p of the charged track since p is saved for each track during tracking.

1.4.3 Collision Geometry Determination Detector

The impact parameter between the two colliding nuclei in a heavy-ion collision reveals information about how central the collision was but one cannot directly measure this quantity. Some parts of the colliding nuclei take part in the collision while others do not and are referred to as spectators. To find out the centrality and the orientation of a heavy-ion collision a detector is required to measure the energy and spatial distribution of the spectators. At the current stage, various types of detectors based on other experiments such as the HADES forward wall [63] and the event plane detector of the STAR experiment [64] are under consideration for such a sub-system. As of March 2023, the underlying technology for the forward detector is still under investigation.

1.4.4 Data Acquisition System

There are no straightforward observables that can be used to apply a hardware trigger on the streaming readout system, therefore, the CBM experiment, due to its high interaction rate, will produce data at a rate of around 1 TB/s [65]. A system containing 200 FPGA-based Common Readout Interface boards will preprocess the data before sending it to a computing farm via optical fibers. Events are reconstructed and filtered by the First-level event selector at the computing farm. The selection of interesting events is performed from the physics point of view, and the data of the selected events will be saved on tape.

1.4.5 CBM Reconstruction Chain

A charged particle traversing the tracking detectors leaves hits in the detectors. Three hits are then combined to form triplets which are then converted to track candidates and selection criteria are applied to select tracks. Tracks are fitted by the Kalman filter method that returns various track parameters. Tracks sharing origin in time and space are grouped to form a single event.

When a charged particle traverses the bulk region of the silicon sensor of the STS, electron-hole pairs are created and they drift in the presence of an applied electric field toward their respective electrodes. Current sensing amplifiers amplify the current and the response is digitized by a digitizer. Adjacent hit strips by a charged track with a joint time stamp are grouped in a cluster. The center of gravity (COG) equation ($X_{COG} = \sum_{cluster} S_i x_i / \sum_{cluster} S_i$) gives the cluster position when the position of the ith strip, x_i, and the signal amplitude, S_i are known. If the two clusters are associated with the p and n sides of a single sensor and they are inside a pre-defined time window then the cluster is referred to as a single hit. In the case of MVD, the discriminator on the pixel detects a hit, and then this information is forwarded to the readout.

A cellular automaton based track finder algorithm is used to reconstruct tracks in the CBM experiment [66]. Cellular automaton was chosen as the optimum candidate, in terms of speed and efficiency, over other algorithms such as conformal mapping, hough transformation, and track following. The algorithm takes in hits, containing position and time coordinates, in the input from the tracking system and reconstructs tracks. The algorithm has 3 main steps:

1. Three hits on consecutive stations are combined into tracklets (triplets)
2. Tracklets containing two common hits are merged to form candidates for tracks
3. Selection criteria are applied to the candidates to filter reconstructed tracks.

The above steps are repeated three times to reduce combinatorics and increase efficiency. In the first attempt easier tracks (high-momentum primary tracks) are reconstructed, and the associated hits are eliminated in the other two iterations. In the second iteration, low-momentum primary tracks are reconstructed

and the related hits are discarded in the next iteration. Finally, all other tracks are searched. In step (1), only those hits are combined that coincide with each other in time. Step (1) is the most computationally expensive and creates huge combinatorics but step (2) and step (3) reduces that consecutively. Tracks candidates' construction begins at the last layer and goes toward the target. The triplet structure was selected because momentum can be calculated from three hits.

The Kalman Filter (KF) method is used to estimate track parameters [67]. A state vector is used to parameterize the track and gets updated by a measurement of a hit. The Kalman Gain, which depends on the uncertainties of the measurement and track estimate, controls the updating of the state vector. The χ^2 matrix encapsulates the difference between the measurement and the track estimate. The quality of the Kalman fit is checked using the distribution of the χ^2. The saved track parameters include initial and final coordinates, slopes, time, and q/p (q is the electromagnetic charge sign). Similarly, the position of the PV is found by the KF method where extrapolated to the PV state vectors of tracks are used as measurements.

Stable charged particles such as protons and electrons or particles with the $c\tau$ (c is the speed of light and τ is the mean half-life) larger than the length of the tracking detector are reconstructed directly by tracking. However, short-lived particles (e.g. $\boldsymbol{K_s^0}$ and $\boldsymbol{\Lambda^0}$) with $c\tau$ smaller than the length of the tracking detector need to be reconstructed by using their daughter tracks. If a short-lived particle decays and the hits of the charged daughter tracks are available then the point of the decay of the mother particle, the secondary vertex (SV), is reconstructed using the KF method in the KFParticle package [67].

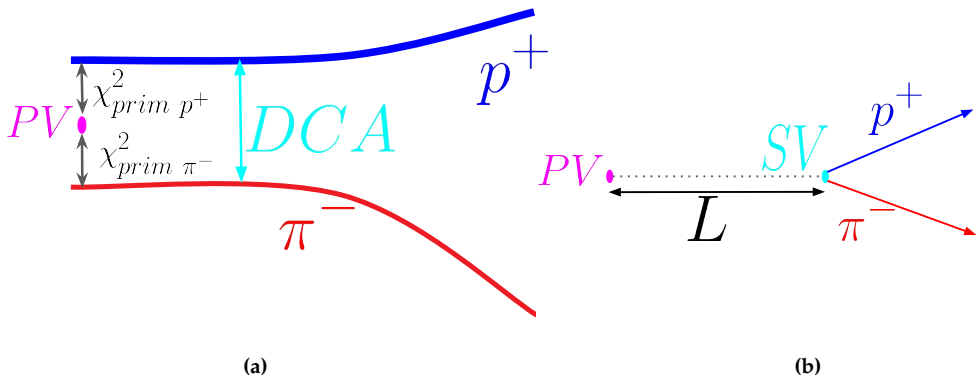
The event builder algorithm groups the reconstructed tracks into different events based on the time, which is obtained as a fit parameter during track fitting using the Kalman filter fit. Tracks are propagated to the PV and the time parameter value in the KF along with its error is used for the assignment of a track to a particular event. Initially, a track with the least χ^2 and smallest time error σ_t is used as a seed for the event. Other tracks are added to the event if they coincide within $3\sigma_t$ of the time of the seed of the event. After the event builder, the tracks of an individual event are saved separately.

1.4.6 Reconstruction of Λ^0 Hyperon

For the reconstruction of short-lived particles through a specific decay channel, all the candidate daughters are combined. The reconstructed short-lived candidates are mostly combinatorial pairs when the particles to be reconstructed are rare particles. Therefore, certain variables need to be reconstructed with the help of which true short-lived particles (signal) can be distinguished from the combinatorial combinations (background). This work focuses on the reconstruction of Λ^0 (referred to as Λ from now onwards) hyperon and its selection criteria optimization so the variables which have separation power for Λ will be discussed here. The decay with the largest branching ratio (63.9%) for Λ is $\Lambda^0 \rightarrow p^+\pi^-$ with p^+ representing a proton and π^- representing the negatively charged pion [68]. To reconstruct Λ through the mentioned decay channel, all positive tracks (hypothesized as protons) will be combined with all negative tracks (hypothesized as pions).

Au-Au collisions are simulated with collision simulators, at $p_{beam} = 12$ A GeV/c, such as DCM and UrQMD and the produced particles are transported through the CBM setup (APR20 version [69]) in the Geant4 [70] engine. The tracks are found by the cellular automaton package and the KFParticle package (PFSimple [71]) is used to reconstruct Λ candidates from all the negatively and positively charged tracks. The analysis has been performed event by event and not time based. In the future, it should be performed on time based simulations but the over all procedure of this analysis will not change. The combinatorics contain more combinatorial background than the signal (MC true Λ). From now onwards, when the data generated by these generators will be mentioned it will mean that the CBM reconstruction chain has been implemented and Λ candidates have been reconstructed. If the discussion will be about simulated Λs by these generators, without the CBM reconstruction chain, then the word simulated Λs will be used.

The distance of the closest approach between the PV and the daughter (d) of a short-lived particle when extrapolated to the PV can be used as a criterion to separate the signal from the background. This means that a daughter track that originates closer to the PV is most likely not produced by hyperon decay.



(a) (b)

Figure 1.7: The variables associated with a Λ decay to p^+ (blue line) and π^- ((red)) are illustrated. The variables $\chi^2_{prim\ p^+}$, $\chi^2_{prim\ \pi^-}$, and DCA associated are illustrated in 1.7a. The separation between the PV (magenta circle) and SV (cyan circle) L is illustrated in the drawing in 1.7b.

The extrapolation of d to the PV means that the uncertainties will also increase because there is no tracking layer very close to the PV. The distance needs to be divided by the errors to take the uncertainties into account. The distance of a daughter to the PV divided by the error is represented by $\chi^2_{prim\ d}$ in this work. It is defined as:

$$\chi^2_{prim\ d} = \Delta r^T_{d-PV}(C_d + C_{PV})^{-1}\Delta r_{d-PV} \tag{1.3}$$

with Δr_{d-PV} representing the distance of the closest approach between the daughter d and the PV [72]. The covariance matrix associated with the track of the daughter d is C_d, and the covariance matrix of the state of the PV is C_{PV}. Fig. 1.7a illustrates the Λ decay in terms of its daughters, i.e., p^+ (blue line) and π^- (red line), and the magenta circle represents the PV. For the daughter p^+, $\chi^2_{prim\ d}$ is illustrated as $\chi^2_{prim\ p^+}$ and for the π^- it is shown as $\chi^2_{prim\ \pi^-}$. For the $\Lambda \rightarrow p^+ + \pi^-$ decay the π^- (p^+) from 17000 Au-Au events generated by UrQMD, variable $\chi^2_{prim\ \pi^-}$ ($\chi^2_{prim\ p^+}$) is shown in Fig.1.8d (1.8e), with red color showing the background and black color showing the signal distribution.

The point of closest approach between the daughters is beneficial for the initial approximation of the SV and also for the segregation of the signal from the background. The shortest point of approach can be used to calculate the distance of the closest approach (DCA) and those tracks with longer distances can be regarded as combinatorial backgrounds. The DCA is illustrated in Fig. 1.7a with the cyan color. Similarly, those tracks which have small values (a few cm)

of DCA can be regarded as good candidates for the signal. The DCA, in units of cm, distribution is shown in Fig. 1.8b for signal and background. Finding the SV requires the extrapolation of the daughter tracks. This increases the errors of the parameters of the tracks and therefore DCA can be normalized to its errors in the form of χ^2_{geo} variable. It can be defined as:

$$\chi^2_{geo} = \Delta r^T_{d1-d2}(C_{d1} + C_{d2})^{-1}\Delta r_{d1-d2} \tag{1.4}$$

with Δr_{d1-d2} representing the distance between the two tracks, and C_{d1} and C_{d2} the error matrices of the daughter tracks. A smaller portion of the χ^2_{geo} distribution is shown in Fig. 1.8a for better visualization of the two distributions.

The distance between the PV and the SV can also be used to separate the signal from the background. This distance is illustrated as a drawing in Fig. 1.7b as a double-sided arrow and indicated by L. The selection criterion based on such a variable works because the smaller the distance between the PV and the SV the smaller the chance that the daughter particles are originating from a decay. Therefore, a variable $L/\Delta L$, i.e., the distance between PV and SV normalized on the errors is made for the isolation of the signal from the background. The $L/\Delta L$ for the Λ decay is shown in Fig. 1.8c in a smaller range.

For the identification of charged particles, PID detectors can be used alongside the tracking detectors. In the case of Λ, the charged decay daughters are π^- and p^+ and for their identification, the $mass^2$ information from the TOF wall can be used. The distribution of $mass^2$ in units of $(\mathrm{GeV/c^2})^2$ for p^+ (π^-) is shown for signal and background in Fig. 1.8g (1.8f). The variables used in this study are summarized in Table 1.1. Also, the other detectors of CBM such as the TRD and RICH can be useful to separate the electrons from the π^- but they are not used in this study. In the future, the information from them can be also incorporated into the reconstruction and selection of short-lived particles.

Variable	Description
$\chi^2_{prim\ \pi^-}$	Squared distance normalized to error between π^- and PV
$\chi^2_{prim\ p^+}$	Squared distance normalized to error between p^+ and PV
DCA	Distance of closest approach between p^+ and π^-
χ^2_{geo}	Squared DCA normalized to its error
$L/\Delta L$	Normalized to its error distance between PV and SV
$mass^2_{\pi^-}$	$mass^2$ of the π^- from TOF
$mass^2_{p^+}$	$mass^2$ of the p^+ from TOF

Table 1.1: The variables associated with the Λ decay and its daughters, i.e., p^+ and π^-.

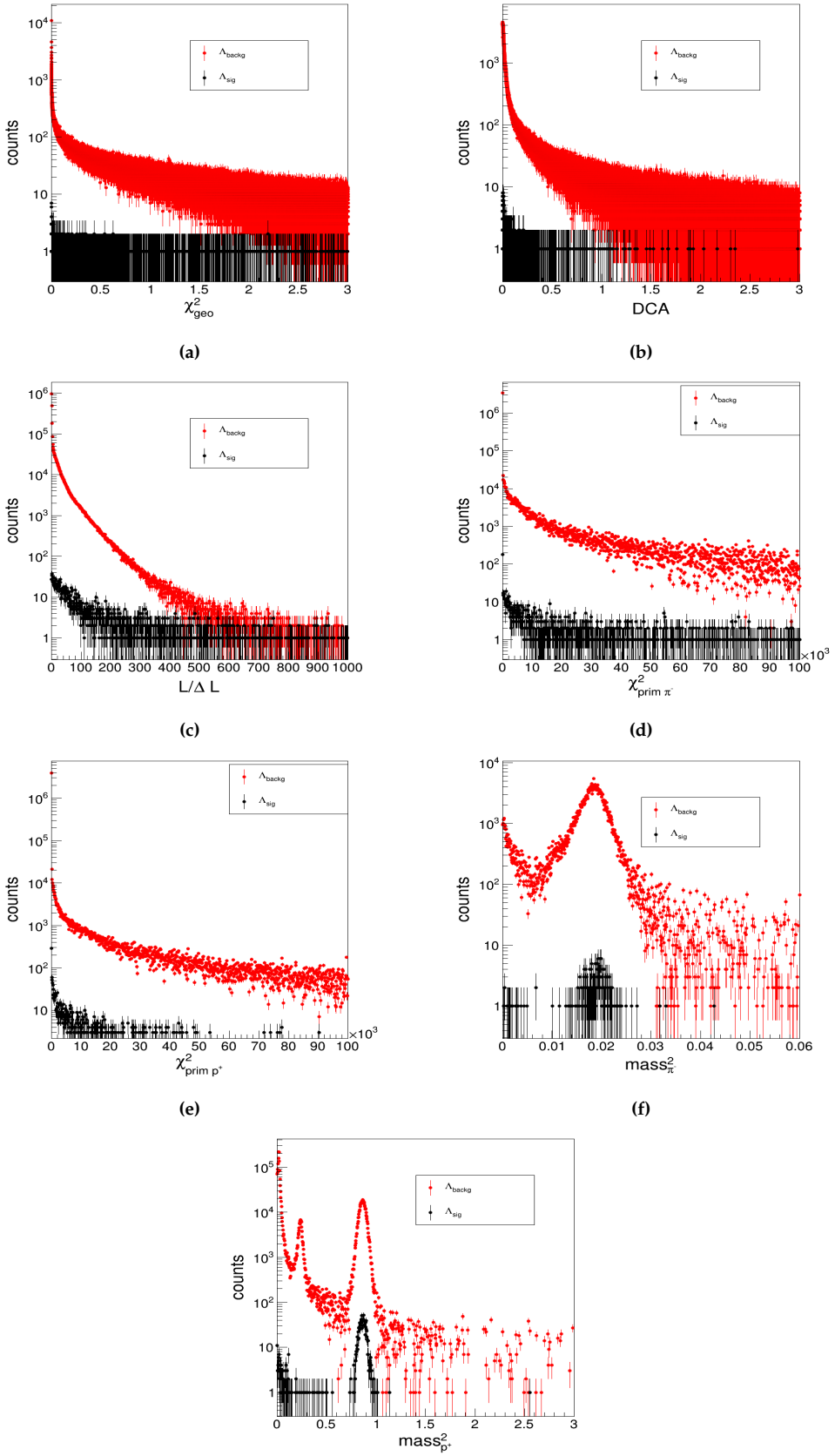


Figure 1.8: The variables associated with the Λ reconstruction are shown here in the log scale on the y-axis. The Λ candidates were reconstructed from the UrQMD simulated Λs.

Chapter 2

Machine Learning Theory

New accelerator facilities are aiming to provide high beam luminosities, driving research to upgrade detector technologies [73]. The increase in the beam intensities has allowed HIC experiments to operate at increasing interaction rates, i.e., up to a few MHz [74]. Simulations of HICs through collision generators predict experimental results such as particle yields with high precision, and the interaction of the produced particles with detector material is also well simulated within Geant4 [70]. For the processing of this data, algorithms are required that can learn from the simulation and experimental data and analyze the real experimental data. Machine Learning (ML) algorithms have been used in HIC experiments for various purposes such as tracking, identifying particles, and selection criteria optimization of reconstructed particles [75, 76, 77].

This chapter defines ML algorithms and describes the eXtreme Gradient Boosting (XGBoost) algorithm. It also highlights the optimization of hyperparameters (HPs) of ML models using different techniques. Various tools to quantify the performance of different ML models are also discussed briefly. Finally, the interpretation of ML models is described.

2.1 Introduction to Machine Learning

ML is a division of computer science where algorithms are designed to predict an outcome (y') by learning from the data. The algorithm may be used to find patterns or learn correlations in the data. The algorithm learns from the variables or

features ($x = x_1, x_2, \cdots, x_n$) of the data. After learning, the algorithm can predict an outcome when provided with the same features of the data not yet seen by the algorithm during the training stage. Mathematically, the ML model is a mapping from the features to the prediction, i.e., $f(x) = y'$. Sometimes in the training of the algorithm, the target or output (y) is also given as an input, the goal is to learn the correlations between the variables and the target, this type of learning is called supervised learning [78]. In another case, unsupervised learning, the algorithm is given only the features, and the algorithm clusters subsets of data based on the values of the features. In both cases, after training the ML model, it is deployed on the unseen or test data, and the performance of the algorithm is evaluated on this unseen data.

In the case of supervised learning a cost function $c(y, y')$ is used to train the algorithm and also to evaluate its performance on the test data. During the learning stage, the algorithm learns from the data by minimizing the cost function. A well-trained algorithm has a lower c value on the test data. If the dependent variable, y, is a continuous value then the mapping $f(x)$ is called a regression, and often the cost function $c(y, y') = (y - y')^2$ is used. If the response y is a category or a number then the prediction is known as classification and the most used cost function is the negative log-likelihood defined as

$$c(y, p) = -[y \times \log(p) + (1 - y) \times \log(1 - p)]. \tag{2.1}$$

Here p is the predicted probability or output, and the minus sign ensures that the minimization of $c(y, p)$ is required to make the difference between y and p small, without the minus maximization would have been required. If the number of categories is two then the classification is called a binary classification and this thesis focuses on binary classification. For simplification, the two classes will be 0 (noise/background) and 1 (signal), and the goal will be to segregate these two classes on the basis of certain input features.

To perform the task of a binary classification various ML algorithms are available and they are optimized for speed and performance. For example, the ML algorithm decision tree (sec. 2.2) is a simple and fast algorithm but with limited power of prediction on complex data sets. Complex means that the data is large and has many variables and each variable has a low separation power of

distinguishing the different classes. While complex algorithms such as XGBoost (sec. 2.3.1) are useful for large and complex data sets they take often longer to be trained and deployed.

2.2 Decision Trees

A Decision Tree (DT) is a sequential ML model that is used in supervised learning for classification and regression. In each step, an if-else condition is applied to the data to partition it into different groups. The point where the condition is applied is called a node while the data passing the condition is stored in one category, i.e., a so-called leaf, while the data not passing the condition is stored in another leaf. The selection of the condition at each node is made through a process that aims to maximize the purity of the data in the leaf nodes relative to the data in the parent node. Because of finding the quickest way to optimize the selection, DTs are called greedy algorithms. The goal is to have a leaf with one class only if no pruning criteria, e.g. a limit on the depth of the tree, are set. Pruning prevents the DT from over-fitting on the training data and therefore makes it generalized. DTs are preferred over other complicated black-box models because they can be easily interpreted [79].

One of the selection criteria for deciding to split the data is maximizing the information gain

$$I = E_{\text{node}} - < E_{\text{leaves}} > . \tag{2.2}$$

The entropy (E), measures the impurity of a leaf/node, for each class i is defined as $E = \sum_{i=1}^{n} (-p_i log_2 p_i)$ with p as the probability of each class in a node/leaf. The $< E_{\text{leaves}} >$ represents the average entropy of the leaves. The probability p for a class i is calculated by taking the ratio of the samples of this class to the total number of samples. Every variable is scanned, and the variable which gives the maximum information gain at a particular value is selected. At the next node, again the same selection is applied and sometimes one variable is used multiple times. This makes the selection non-linear.

Table 2.1 shows some made-up data for two classes, i.e., 1 and 0 as specified by the target variable. Figure 2.2 shows a made-up structure of a DT. Tree depth

is the number of decision nodes in a tree, for example, the tree of Fig.2.2 has a depth of 2. The goal of the decision tree is to classify the two classes based on the variables provided. The entropy at node 0 can be calculated as the sum of the entropy of each class $E = -(1/2)(-1) + -(1/2)(-1) = 1$, where the probability of each class is $1/2$. Leaf 1 gets the same target value samples so its entropy is 0, since $log_2 1 = 0$ and the entropy of the other leaf (labeled as node 1) is $E = (-1/4)(-2) + (-3/4)(-0.415) = 0.811$. An information gain of $I = 1 - (0 + 0.81)/2 = 0.594$ has been achieved through the first selection. This information gain is more than the gain which can be achieved if Var2 is used instead of Var1, i.e., $I = 1 - (0 + ((-2/5) * (-1.322)) + ((-3/5)(-0.737)) = 0.51$. Similarly, at node 1 a selection criterion on Var2 splits the data to increase the information gain. The response of the tree to predict the target in terms of variables Var1 and Var2 can be written as

$$f(Var) = \sum_{j=1}^{3} \mathcal{O}_j I\{(Var1, Var2) \in R_j\} \tag{2.3}$$

with R as a leaf and $\mathcal{O}_j$ as the response, output value of the individual leaf, of the DT model, [78]. Thus for an element in a leaf $x \in R_j$ the tree response can be summarized as a constant value, i.e., $f(x) = \mathcal{O}_j$. In the example (Tab. 2.1), sample 1 in leaf 1 has a prediction value of 1.

S.No.	Target	Var1	Var2
1	1	10	30
2	0	20	30
3	1	15	30
4	0	30	20
5	1	30	10
6	0	30	30

Table 2.1: A table of pseudo-data to show how a DT will classify the two classes of the target variable based on the variables Var1 and Var2. S.No. is the sample number.

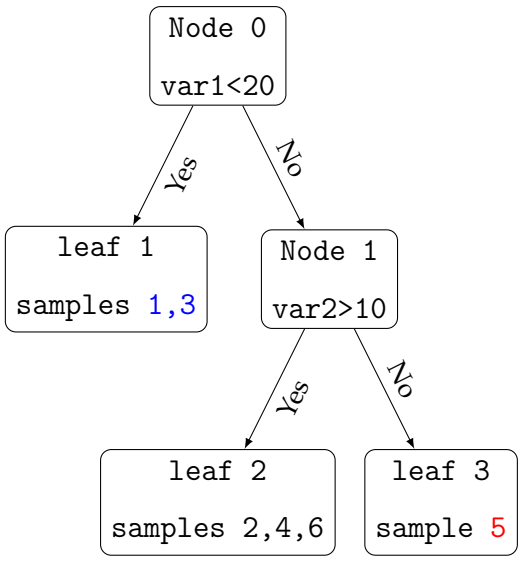


Table 2.2: A DT to classify the classes in the target according to Var1 and Var2.

2.3 The Gradient Boosting Algorithm

DT-based results are easily interpretable but the algorithm, with a limited tree depth, is a weak learner (models performing a little better than guessing). Therefore, the DT algorithm only works well for less complex and smaller datasets. One way to combine weak learners, such as DTs, to get a strong learner, with good accuracy, is gradient boosting [80]. The DTs are combined in an iterative way to get a final classifier that takes into account all the previous results of all the trees. Suppose, there is a dataset $\mathcal{D} = (x_i, y_i)_{i=1}^{n}$, with $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ representing the independent variables and y the target variable of the n samples in the data. A cost function $c(y_i, F(x_i))$ can be used to quantify the prediction performance of ML model $F(x)$ on some data. In the case of classification, the differentiable cost function often used is the negative log-likelihood (eq. 2.1) of the observed data (y_i) given the ML prediction $F(x_i)$, i.e.,

$$c(y_i, F(x_i)) = -[y_i \times \log(F(x_i)) + (1 - y_i) \times \log(1 - F(x_i))]. \tag{2.4}$$

The main gradient boosting algorithm proceeds in the following steps [78]:

1. Initialise the ML model with a constant value

$$F_0(x) = \arg\min \sum_{i=1}^{n} c(y_i, F(x_i)) \tag{2.5}$$

2. Fit m-th DT to the data where $m \in \{1, 2, 3, \cdots, M\}$:

   (a) Compute the pseudo-residuals (r_{im}) for m$-$th tree for all n samples at $F(x) = F_{m-1}(x)$, i.e.,

$$r_{im} = -\Big[\frac{\partial c(y_i, F(x_i))}{\partial F(x_i)}\Big] \tag{2.6}$$

   (b) By fitting a regression tree to the r_{im} values, leaves R_{jm} are created. For each leaf j in tree m, i.e., $j = 1, \cdots, J_m$, the output value is determined by

$$\mathcal{O}_{jm} = \arg\min \sum_{x_i \epsilon R_{ij}} c(y_i, F_{m-1}(x_i) + \mathcal{O}) \tag{2.7}$$

   often $\mathcal{O}_{jm}$ is not easy to solve and Taylor expansion is used

$$c(y_i, F_{m-1}(x_i) + \mathcal{O}) \approx (y_i, F_{m-1}(x_i)) + grad\ c(y_i, F_{m-1}(x_i))\mathcal{O} - \frac{1}{2} hess\ c(y_i, F_{m-1}(x_i))\mathcal{O}^2 \tag{2.8}$$

(c) Update the previous prediction

$$F_m(x) = F_{m-1}(x) + \eta \sum_{j=1}^{J_m} \mathcal{O}_{jm} I(x \epsilon R_{jm}) \tag{2.9}$$

3. The final classifier is the combination of all trees in $F_M(x)$.

eq. 2.5 is initialized for binary classification with $log(p/(1-p))$ of the different classes present in the data with p as the probability of one of the classes in the data. With the selected cost function eq. 2.4, the pseudo-residuals in eq. 2.6, will become $r_{im} = y_i - p$, i.e., the difference between the actual class label and the probability for individual entry. The algorithm is named after the step in eq. 2.6 as Gradient-Boosting. The grad (related to gradient) in eq. 2.8 is the first order derivative ($\frac{d}{dF(F_{m-1})}$) and the hess (related to Hessian) is the 2nd order one ($\frac{d^2}{dF(F_{m-1})^2}$). The η in eq. 2.9 is a regularization term and it scales the contribution of the tree, $\sum_{j=1}^{J_m} \mathcal{O}_{jm} I(x \epsilon R_{jm})$, by a factor $0 < \eta < 1$. Empirical evidence [80] suggests that smaller values of η result in better test data performance but require more trees (M) with lower values of depth; therefore, F(x) becomes computationally expensive. Subsampling is another regularization parameter that reduces computing usage but increases the performance of the model on the test data. If a subsampling of 0.8 is selected this will lead to the selection of 80% of the total training data and the next tree will be trained on it. This will reduce the computing power by 20% and will also make the model less dependent on the training data.

For example, the initial prediction for the data in table 2.1 will be $F_0(x) = log(p/(1-p)) = log(0.5/0.5) = 0$. Therefore, all candidates will be treated as background and pseudo-residuals in eq. 2.6 will be calculated and will be $r = (-1, 0, -1, 0, -1, 0)$. A tree is used to predict the pseudo-residuals, instead of the target, using the variables Var1 and Var2. To prevent the tree from over-fitting onto the data $\eta = 0.2$ can be used and the final classifier in eq. 2.9 p will be $F_1(x) = 0 + 0.2 \times \sum_{j=1}^{J_1} \mathcal{O}_{j1} I(x \epsilon R_{j1})$. To improve the result, another tree could be added and this time the prediction of $F_1(x)$ will be the initial prediction and the final classifier will be $F_2(x) = F_0(x) + F_1(x) + 0.2 \times \sum_{j=1}^{J_2} \mathcal{O}_{j2} I(x \epsilon R_{j2})$.

2.3.1 XGBoost

To create a library that uses the gradient boosting algorithm efficiently and effectively Tianqi Chen and Carlos Guestrin introduced Extreme Gradient Boosting (XGBoost). XGBoost is better than other boosting libraries in terms of speed and accuracy. It offers parallel processing and can handle missing values. It has won several ML competitions and is an open-source project [81].

2.3.1.1 The XGBoost Algorithm

Since XGBoost is a gradient boosting algorithm, most of the theory of sec. 2.3 applies here. However, some changes were made to the original algorithm to modify it for speed and accuracy. An initial prediction, $F_0 = 0$, is used for all candidates of the target variable, and residuals r_{im} (eq. 2.6) are calculated. DTs are added iteratively to predict the residuals using variables of the data, excluding the target variable, and the final classifier is a combination of all the previous ones. For a classification problem, the cost function in eq. 2.4 is used but an additional term is added for regularization, i.e., $\gamma R + 0.5\lambda\mathcal{O}^2$. R represents the total number of leaves and γ is a pruning term, i.e., it controls the depth of a decision tree. The regularization term $0.5\lambda\mathcal{O}^2$ takes into account the output value of a leaf $\mathcal{O}$ and then shrinks it by a factor λ. The output values ($\mathcal{O}_{jm}$) of the leaves (R_{jm}) for the m-th tree are again calculated using the Taylor expansion mentioned in eq. 2.8. If I_j is the instance set of leaf j then $\mathcal{O}_{jm}$ in terms of the new cost-function become

$$\mathcal{O}_{jm} = \frac{\sum_{i\in I_j} grad_i}{\sum_{i\in I_j} hess_i + \lambda}. \tag{2.10}$$

However, instead of using the impurity measure mentioned in sec. 2.2 as the splitting criteria for node into leaves, XGBoost uses a score given as

$$score = -\frac{1}{2}\sum_{j=1}^{R} \frac{(\sum_{i\in I_j} grad_i^2)}{\sum_{i\in I_j} hess_i + \lambda} + \gamma R. \tag{2.11}$$

The gain achieved by two leaves, I_1 and I_2, after a split is given by

$$Gain = \frac{1}{2}\left[\frac{\sum_{i\in I_1} grad_i^2}{\sum_{i\in I_1} hess_i + \lambda} + \frac{\sum_{i\in I_2} grad_i^2}{\sum_{i\in I_2} hess_i + \lambda} - \frac{\sum_{i\in I_{node}} grad_i^2}{\sum_{i\in I_{node}} hess_i + \lambda}\right] - \gamma. \tag{2.12}$$

A node is split into leaves if eq. 2.12 is greater than zero. So γ and score control the splitting of the node to leaves and therefore γ regularizes the depth of the tree.

Finding the best split exactly using eq. 2.12 is very costly for bigger data sets and therefore an approximate method is also available. In this method (hist tree), the data in a variable is divided into percentiles and then the splitting of a node to leaves is checked on these percentiles. Whichever percentile corresponds to the biggest gain will be selected to split the data. If a feature has missing values then during the data splitting from node to leaves, the missing data entries are added to different percentiles and the one resulting in the highest gain is selected.

2.3.1.2 Hyperparameters of XGBoost

Apart from the HPs discussed in section 2.3, i.e., tree depth (*max_depth*), number of trees (*n_estimator* or n_{est} for short),η, and subsampling, XGBoost offer a wide variety of HPs.

- λ: regularization parameter controlling output values of leaves. It is also known as L2 regularization and a higher value will mean a more conservative model.
- γ- pruning parameter that controls tree depth
- α: L1 regularization that control leaf outputs, i.e., eq. 2.10. Increasing it makes the model more conservative.
- subsample: By defining this fraction the algorithm selects a subset of the training data based on this fraction. Subsample 0.8 means that XGBoost would randomly sample 80% of the training data before growing trees to minimize overfitting.
- scale position weight (scale P-W): This controls the weights of different classes if the data is unbalanced: different classes do not equally populate the data.

Here L1 regularization originates from Lasso regression where the absolute value of output is regularized by a regularization parameter λ. While in L2 regulariza-

tion, i.e., Ridge regression the square of the output is regularized by the λ. There are also other HPs but since they are not used in this work, they will not be discussed here.

2.3.2 Treelite

The Python implementation of XGBoost is slow when it comes to predicting the outcome for a very large data set e.g. data of Terabytes in size. In high-energy physics, selection criteria are generally optimized on a local computer (host machine) on a smaller data set. The optimized selection criteria are then utilized on larger data on a supercomputer cluster (target machine) with 1000s of Central Processing Units (CPU).

The treelite [82] can convert the python-based XGBoost model into a C++ library and therefore offer accelerated performance in terms of speed. It was made with the intention to install treelite on the local machine and then take the trained ML model to the host machine without installing any software. The library converts the decision rules of XGBoost into if-else conditions. For example, a tree node division into leaves is converted into a single if-else rule and if the leaf further divides then other if-else conditions are added into the parent if or else condition.

2.4 Hyperparameter Optimization

ML algorithms are often complex mapping functions from observables to an output distribution. To fit the mapping functions such that they fit the train data and also have useful predictions on unseen data, the number of the free parameters of the mapping function needs to be found. The mapping is then fitted on the training data by finding the values of the fit parameters. The resulting mapping is then tested on some unseen part of the data (test data). Some parameters improve the performance of the mapping on the available data while others control over-fitting on it so that the prediction power on unseen data is comparable to the one on training data. Generally, a cost function, such as $c(x) = \textit{true label} - \textit{ML prediction}$, i.e., the difference between the true target

label value and the ML output value, is used to quantify the performance of the ML model. The free parameters of the mapping are often so many that one needs an automatized tool to find them.

If there is an ML model and it has N HPs to be optimized then the total space of HPs is:

$$\mathbf{X} = \chi_1 \times \chi_2 \times \cdots \chi_N. \tag{2.13}$$

The individual domain of the i-th HP is χ_i and a vector from this space would be $x \in \mathbf{X}$. When an ML model such as a classifier is applied to certain data it gives predictions for the target label. Algorithmically, one is trying to minimize or maximize a cost function $c(x)$ such as:

$$x^* = \arg\min_{x \in \mathbf{X}} c(x). \tag{2.14}$$

Here the cost function measures the performance of the ML model on the test data and the x^* is the best HPs vector. The goal of an HP optimization algorithm should be to find x^* efficiently in terms of computation power.

For example, in the case of XGBoost, these parameters can be the number of trees (n_{est}), max_depth, the η, etc. If an HP, like the number of trees, is selected and the domain is selected to be between 1 and 1000, then $\chi_{n_{est}} = \{1, 2, \cdots, 1000\}$. Similarly, if an HP such as the depth of each tree is selected to have values between 1 and 20, then $\chi_{max\_depth} = \{1, 2, \cdots, 20\}$. The total space would be $\mathbf{X}_{XGBoost} = \chi_{n_{est}} \times \chi_{max\_depth}$. Examples of $x \in \mathbf{X}_{XGBoost}$ will be $x_1 = (1, 1)$, $x_2 = (1000, 1)$, and $x_3 = (1000, 20)$. In total $1000 \times 20 = 20,000$ different x will be in $\mathbf{X}_{XGBoost}$.

2.4.1 Sequential Model-based Global Optimization

Brute force searches such as grid search, where HP points are placed on a lattice, are computationally expensive because it searches the provided space extensively [83]. As an alternative, Bayesian optimization [84] can be used to find the HP more quickly. In sequential model-based (SMB) bayesian optimization a subsample of $\mathbf{X}$ is selected with a sampling strategy such as random, quasi-random, and Latin hypercube sampling. On a sub sample of $\mathbf{X}$, a set $\mathcal{E} = \{(x_1, c(x_1), \cdots, (x_i, c(x_i))\}$ is created by evaluating the expensive cost function, $c(x)$ at each vector $x \in \chi$. A surrogate model $\mathcal{S}$ (generally probabilistic regression) is fitted

to $\mathcal{E}$ and a probability distribution $p(c(x)|x, \mathcal{E})$ is created. The probability distribution should contain the uncertainty of the surrogate model for mimicking the cost function. New locations within $\mathbf{X}$ are iteratively selected by optimizing a selection function $\mathcal{A}$ which uses the surrogate model instead of the expensive objective $c(x)$ for the HP optimization.

2.4.1.1 Tree-structured Parzen Estimators

Tree-structured Parzen Estimators (TPE) is an SMB optimization algorithm. Instead of predictive distribution over the cost function, $p(c(x)|x, \mathcal{E})$, it models $p(x|c(x), \mathcal{E})$ and generates two density functions, i.e., h(x) and j(x) [85]. Both h(x) and j(x) model the domain variables when the cost function is below and above a specified quantile $c(x)^*$ (usually set to 15%):

$$p(x|c(x), \mathcal{E}) = \begin{cases} h(x) & \text{if } c(x) < c(x)^* \\ j(x) & \text{if } c(x) \geq c(x)^* \end{cases} \tag{2.15}$$

The ratio $h(x)/j(x)$ is related to the selection function $\mathcal{A}$ and is used to predict new HP. A tree of Parzen estimators for conditional HP is used by TPE and has been shown to perform well on different datasets [85, 86]. It is easier to understand, computationally less expensive than conventional methods and offers the option of parallelization. The disadvantage of TPE is that it does not model interactions between HP.

2.4.1.2 Evolutionary Strategy

The finding of HPs is an optimization problem, and since Evolutionary Strategy (ES) is a stochastic search algorithm, therefore, can also be used to find HPs. In an ES, a subset of $\mathbf{X}$ is taken as a population and then one $x_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$ is selected randomly as a parent. An offspring x_i' is generated from the parent by adding an independent mutation (δ_i), sampled from a normal distribution to one of its N variables e.g., $x_i' = (x_{i1} + \delta_i z, x_{i2}, \cdots, x_{iN})$. Here z is a random vector coming from a normal distribution. The mutation part is the stochastic part of the algorithm. A selection procedure then filters the worst samples from the population and therefore one is left with a number of x as the best HPs [87]. The

disadvantage of these algorithms is that they are very slow and therefore computationally expensive.

A popular ES algorithm is the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [88]. In this method, the offspring is created not from a single parent but from a weighted mean of the population where the best candidates get a higher weight. The offspring are then mutated by adding terms to variables from a normal distribution with zero mean and variance that comes from a covariance matrix. This covariance matrix is updated at each iteration and therefore the new population generated is based on it.

2.4.2 Optuna

Optuna [89] is an HP optimization library that offers efficient sampling and pruning algorithms. The library provides different sampling methods such as grid search, random search, relational, e.g., CMA-ES, and independent, e.g., Tree Parzen Estimators (TPE). The relational method uses the correlations among the parameters while the independent method samples each parameter independently. The library optimizes the HP by using a loss function, which takes in HP and gives back a validation score. In every trial, an improved validation score is searched. Optuna can generate HPs through the "suggest API" within limits provided by the user. The library makes the search more efficient by applying pruning methods through its "should prune API." Through pruning, it terminates a trial in which a pre-defined condition is not met.

2.4.3 Cross Validation

For a simpler data set, with fewer variables having great separation power between classes, one often splits the data into two parts, i.e., train and test data. The ML algorithm is trained on the train part and is evaluated on the test part. However, if the data sample is complex, more variables with less separation power and one needs the HPs of the ML model to be tuned separately, then one can divide the data into three equal parts. One part of the data will be used for finding HPs, one for training, and one for testing the ML model. Often generation

and handling of such big data are expensive. There is a better strategy of k-fold cross-validation where one divides the data into the conventional train and test data and then further divides the train data into k equal subsets [78]. Then one uses all k-1 subsets, selected randomly out of k subsets, to train the ML model on a particular vector of HPs and uses the kth subset to evaluate the model performance. The same process is performed k times and the average performance of the model is calculated. In the search to find a new and better HPs vector than the previous one, another HP vector goes through k-fold cross-validation. This process is repeated several times and the best HPs are found.

After finding the best HPs the complete training data is used to train the ML model. Now to check the performance of the ML model on data that it has not seen during the HPs optimization stage, the model is deployed on the test data. The difference between the test data target and the ML model output is used to quantify the performance of the ML model.

2.5 Model Performance Evaluation

To quantify the performance of the ML model on the train-test data sets a criterion is required. This criterion should reveal the variance and bias of the ML model on the train-test data sets. Often a confusion matrix (CM) is used to quantify the performance of an ML classifier on different data but a selection needs to be applied to the ML output for that. A CM [90] for a binary classifier contains 2 rows and 2 columns as shown in Fig. 2.1 left. The top left column shows the number of true class 1 candidates correctly classified as class 1 candidates by the classifier i.e true positives (TP). The top right plot shows the number of true class 1 candidates misidentified as class 0 candidates, i.e., false negatives (FN). The bottom right shows the class 0 candidates correctly classified, i.e. true negatives (TN), while the bottom left shows the misclassified background, i.e., false positives (FP). To check the performance of the ML model on many selections, creating and analyzing CMs is time-consuming.

The CMs for many selections on the ML output can be quantified by making a plot of true positive rate (TPR) vs false positive rate (FPR) in a Receiver Operating

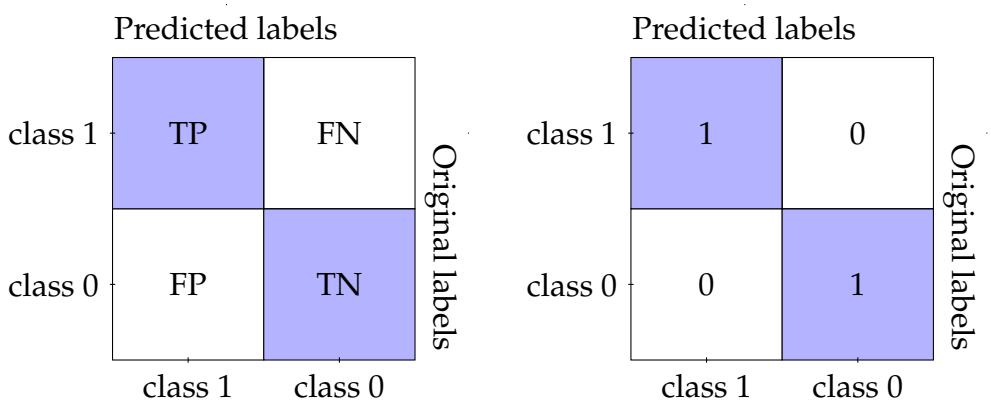


Figure 2.1: The left drawing shows a confusion matrix for a binary classifier. The y-axis labels are the true labels of the data, and the x-axis labels are the classifier-predicted labels. The right side drawing shows a CM normalized to 1 for an ideal classifier.

characteristic Curve (ROC). Where TPR and FPR are defined as:

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{TN}{TN + FP}. \tag{2.16}$$

In terms of two classes in the binary classification, the TPR is the ratio of true class 1 candidates classified correctly to the total true candidates of class 1. Similarly, FPR is the ratio of the correctly classified class 0 candidates to the total class 0 candidates. The Area Under the ROC Curve (AUC) of the ROC plot can be used as a criterion for distinguishing different models. Fig. 2.2 shows the performance of an ML model on two data. Generally, a higher value of ROC-AUC indicates better classification performance for an ML model, but this is not always the case. For example, a model with more complexity can give a better ROC-AUC on training data (Fig. 2.2 left, blue dash-dot line) but not better results on the test data ((Fig. 2.2 right). Therefore, a good classifier will have high ROC-AUC on both train and test data. An ideal ML model will have ROC-AUC equal to one while random guessing will have a 50% chance of success if the two classes are equally likely to appear in the data.

2.6 Model Interpretability

A supervised ML model is a complex mapping from input variables (the independent ones) to a target variable (the dependent one) with many hyper-parameters and is not easily interpretable. The functionality of ML models is not easily un-

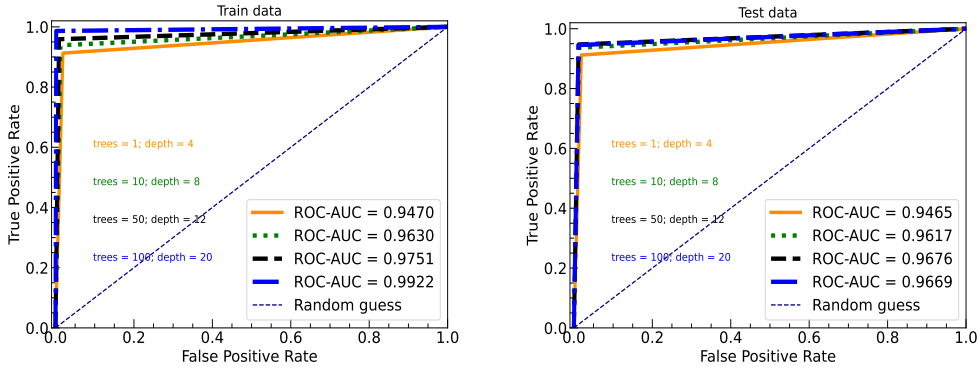


Figure 2.2: The left side images shows the ROC plot for the train data set. The right side plot shows the ROC plot for the test data. All the models trained had a fixed learning rate of 0.1.

derstandable by mere visual inspection. Approximation of an ML model by a simpler model is required to interpret the results of the ML model intuitively. Models which approximate the ML model locally explain individual predictions of the ML model and then selecting multiple such predictions reveals the inner workings of the ML model. For example, the simpler explanation model can be a linear regression model or a decision tree.

The Shapley Additive exPlanation (SHAP) library approximates the original ML model, $f(x)$, locally (on a single input x) by an explanatory model, $g(x')$, [91]. It is not made for some specific ML model and can be applied to any ML model. A mapping function $x = h_x(x')$ connects the original inputs x to simplified inputs x'. Individual feature $i \in N$ (e.g. the DCA between two tracks), is given a SHAP score ϕ_i and the ML model is approximated in additive feature attribution way as:

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{N} \phi_i x'_i \tag{2.17}$$

with $x' \in \{0,1\}^N$. If features of the output $f(x)$ were unknown, then the model would predict the base value ϕ_0. Each feature contribution can be written as:

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(N-z'-1)!}{N!}[f_x(z') - f_x(z' \setminus i)]. \tag{2.18}$$

Here $z' \subseteq x'$ shows all z' vectors that have non-zero entries as a subset of the non-zero entries of x'. The number of non-zero values in z' is represented by $|z'|$. In eq. 2.18, the $f_x(z') = f(h_x(z'))$ shows the presence of the feature in the model prediction. The $z' \setminus i$ in $f_x(z' \setminus i)$ symbolize the configuration $z'_i = 0$ and shows

the absence of the feature in the model prediction. This means that each feature contribution should be calculated as the difference between its presence $(f_x(z'))$ and absence $(f_x(z' \setminus i))$ in the model prediction. Missing features should have no influence, i.e., $x'_i = 0 \Rightarrow \phi_i = 0$. In Kernel SHAP, a minimization algorithm fits the local approximation $g(z')$ to the ML model f using the squared loss function L for finding the ϕ over a set of samples in the simplified input space weighted by the local kernel $\pi_{z'}$, i.e.,

$$\arg\min_{g \in G} L(f, g, \pi_{x'}). \tag{2.19}$$

The explanation model g for sample x minimizes the loss L. L quantifies how close the explanation model is to the output of the ML model, i.e.,

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} \left[f(h_x^{-1}(z')) - g(z') \right]^2 \pi_{x'}(z'). \tag{2.20}$$

G encompasses all similar simple explanatory models, for example, all possible linear regression models[92]. Kernel SHAP has a unique weighted local kernel, which can be mathematically expressed as:

$$\pi_{x'}(z') = \frac{N - 1}{(N \; choose \; |z'|) \, |z'| (N - |z'|)}. \tag{2.21}$$

Chapter 3

Performance for Multi-differential Yield Measurement

The hot and dense matter created during heavy-ion collisions can be studied in a fixed target experiment by measuring the multi-differential yields of hyperons. To calculate the yield of a hyperon, the tracks of its daughter particles are reconstructed and then combined to form candidates for the hyperon. This leads to a huge combinatoric and a selection process is required for its removal. Detectors are made for general purposes in fixed target experiments; they are not perfectly optimized for a particular decay scheme in terms of placing tracking detectors downward the target. The purpose is to reconstruct most particles in order to get a complete picture of the collision. Often the decay vertex of a short-lived particle cannot be clearly resolved from the primary vertex when the detector geometry is not optimized for this type of decay. Therefore, the application of selection criteria will reduce the combinatorial background, but the particle yield also gets diminished. To not diminish the yield of a rare decay, mild selection criteria are applied and some background is left under the signal peak of the invariant (inv.) mass distribution of the decay. A fitting routine is used to estimate the yield and the remaining combinatorial background. The particle yield needs to be corrected for efficiency and geometrical acceptance. The systematic uncertainties are evaluated by varying the selection criteria.

This chapter focuses on the selection criteria optimization for Λ hyperon through the ML algorithm of XGBoost. The performance of the ML model will be evalu-

ated using ROC-AUC curves and the selection criteria visualization will be performed with SHAP. A comparison is also made to manually optimized selection criteria.

The next chapter (ch. 4) focuses on the multi-differential yield extraction procedure through a fitting routine. The yield is efficiency corrected and the systematics are evaluated from the response of the corrected yield to the variation in selection criteria. The corrected yield and the simulated yields are compared and a summary is also be presented.

3.1 The Selection Criteria Optimization of Λ

Λ hyperons are crucial to study the deconfined matter created during heavy-ion collisions because they contain a strange quark. Since they are the most abundantly produced strange baryons at FAIR energies they are useful for assessing the systematic uncertainty of a selection procedure. For the same number of simulated collision events, a rarer decay than Λ has lower statistics and the systematic uncertainty calculation gets less precise. Λs are neutral particles that decay via weak interaction and their tracks are not registered in the tracking detectors. Also, their $c\tau$ is 7.89 cm [93], so their decay vertex is on average shifted from the primary vertex and lies between the first layer of the micro-vertex detector and the target. A Λ can decay through multiple decay channels but the decay channel with the highest branching ratio (63.9%) is $\Lambda \rightarrow p^+ + \pi^-$, so this channel will be studied in this work. There are Λs that are produced in the collision of heavy nuclei and they are referred to as primary Λs in this work. There are also Λs which are produced in the decay of other particles such as Ξ^0 and Ξ^- [93]; also inelastic processes such as the interaction of charged particles with the detector material can produce Λs called secondary Λs. In this work, the contribution of the secondary Λs to the yield will not be discussed. In appendix A.1, the reason for the exclusion of secondaries is discussed.

Data, products of Au-Au collisions at $p_{beam} = 12\ A$ GeV/c, from two different collision simulators, i.e., UrQMD and DCM are passed through the CBM Geant4 setup. KFParticle package [67, 71] is used to reconstruct Λ candidates.

The combinatorics contain more combinatorial background than signal and the signal and the background produced in such collisions depend on the centrality class, p_T, and rapidity (y_{Lab}). The values of the various variables, associated with the daughters of Λ, vary in different intervals because of the response of the CBM detector and the tracking algorithms. selection criteria need to be applied to the variables to segregate signal from the combinatorial background in various centrality, p_T, and y_{Lab} intervals. Simulations of the variables of the signal are easier to produce than those associated with random combinatorial backgrounds. The reason is that the random background can come from various sources, such as protons and pions produced due to the interaction of the collision-produced particles with a detector component that was not taken into account during simulation. This can be overcome by using the combinatorial background from real data in a phase space of inv. mass distribution where Λs are not expected, i.e., $inv.\ mass < 1.1$ and $inv.\ mass > 1.13$. Also, this optimization needs to be performed for different energies of the colliding nuclei.

The optimization of selection criteria in a multi-dimensional space of variables manually is a laborious job. It gets more demanding if one wants to optimize the selection criteria non-linearly for each energy of colliding nuclei for different centrality, p_T, and y_{Lab} intervals. ML algorithms can optimize selection criteria non-linearly and multi-dimensionally in an automatized way.

3.2 Data Preparation for ML

KFParticle reconstructed Λ candidates generated by the DCM model are treated as a simulation while those generated by UrQMD are considered real experimental data. When the FAIR facility will start operating, the experimental CBM data will replace the 2nd model data but the procedure will remain the same. This type of analysis is partially data-driven because the easier to simulate data comes from simulation and the less predictable part comes from data .

In an average UrQMD (DCM) Au-Au collision, the ratio of π^- and p^+ pairs produced in a Λ decay to combinatorial background is around $3.8 \times 10^{-7}(3 \times 10^{-7})$. Therefore, to enhance the number of signal candidates, 5×10^6 events of

the DCM model are taken and only primary MC true Λ candidates are selected for the training of the algorithm. The signal of the DCM data is shown in Fig. 3.1 and it is distributed non-uniformly in different $p_T - y_{Lab}$ bins. To make the ML model treat various $p_T - y_{Lab}$ regions differently, one needs to train one ML model for each region. Similarly, the production of signal candidates depends on the impact parameter of the collision. A central collision produces more Λ hyperons than a peripheral one. A central collision also produces other charged particles in higher abundance than a peripheral which can lead to a decrease in the efficiency of the tracking system. This efficiency will be discussed in sec. 4.2. In this section, the analysis of the multiplicity interval of charged tracks $[200, 400]$, p_T $[0, 0.6]$ GeV/c, and rapidity $[0, 1.6]$ is presented as an example. The analysis of the other $p_T - y_{Lab}$ intervals of this multiplicity interval are also performed and some of its figures are added in the appendix A.

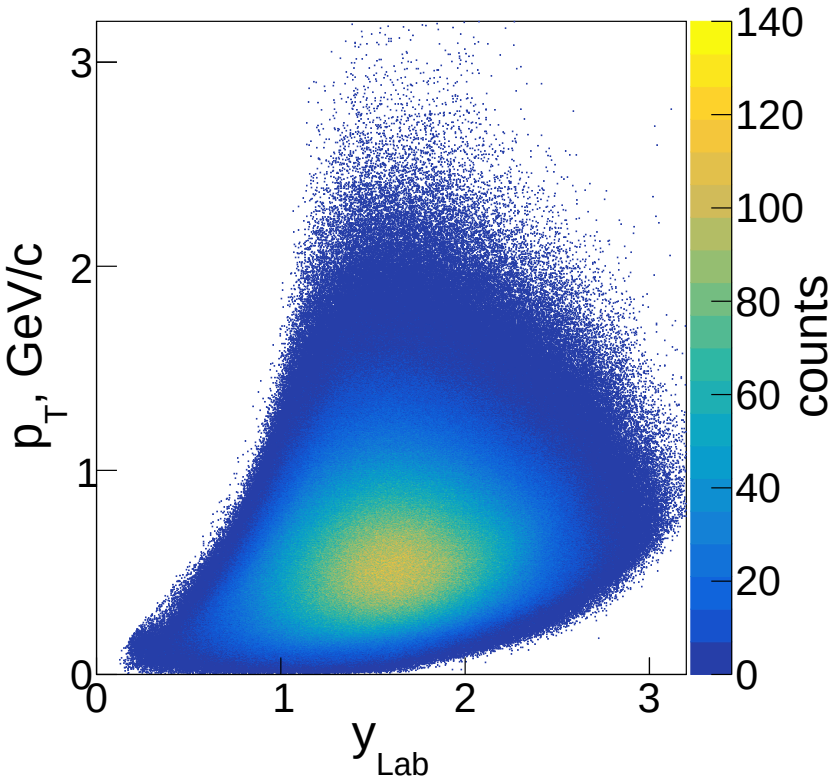
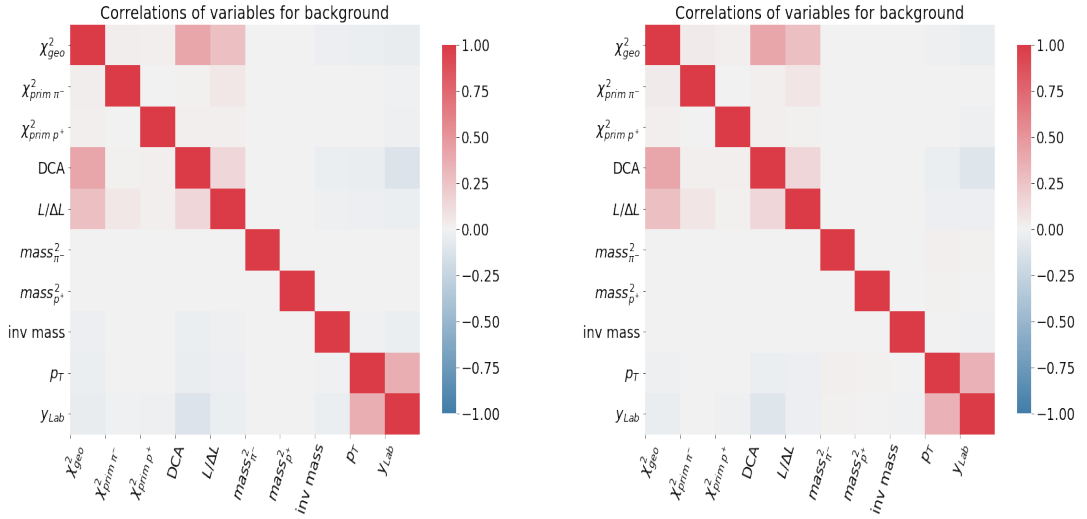


Figure 3.1: The DCM-generated MC Λ distribution after reconstruction for 5×10^6 Au-Au events, produced at $p_{beam} = 12$ A GeV/c.

The variables (discussed in sec. 1.4.6) that are used to separate the signal from

the background are: χ^2_{geo}, $\chi^2_{prim\ \pi^-}$, $\chi^2_{prim\ p^+}$, DCA, $L/\Delta L$, $mass^2_{pi^-}$ and the $mass^2_{p^+}$. The distribution of MC true Λ and its background were plotted for each variable in Fig. 1.8. The Pearson correlation coefficient $\rho_{x,y} = \frac{Covariance_{x,y}}{\sigma_x,\sigma_y}$ between every two variables for the MC true background set is plotted in Fig. 3.2. The standard deviation of each variable is represented by its σ. The correlation of the variables for the background outside (inside) the inv. mass peak of the Λ, $mass < 1.1\,\mathrm{GeV/c^2}$ and $mass > 1.13\,\mathrm{GeV/c^2}$ ($1.1\,\mathrm{GeV/c^2} < mass < 1.13\,\mathrm{GeV/c^2}$), is shown in Fig. 3.2b (3.2a). Both images show that there are no strong correlations between the various variables and the inv. mass, p_T, and y_{Lab} distributions of the background. Also, the correlations are the same for both types (under the peak and in side bands) of the background, which means that the background is a mere combination of pairs and it is independent of inv. mass.



(a) (b)

Figure 3.2: The plot shows the correlations among various variables for the background data, i.e., KFParticles reconstructed Λ candidates by combining random pairs of positively and negatively charged tracks generated by UrQMD Au-Au collisions at $p_{beam} = 12\ A$ GeV/c. The left (right) plot is for the background lying outside (inside) the Λ peak on the inv mass distribution.

The MC true signal is selected from the DCM-QGSM-SMM model data in the 5σ region around the Λ peak. The background distribution is selected in the sidebands of the Λ peak from the UrQMD model data. These two data sets are combined in such a way that 3 times more background than signal is taken and Fig. 3.3 shows the inv. mass distribution of this data set. This data is divided into

equal-size train and test data. The training data is used for optimizing the hyper-parameters of the XGBoost algorithm, discussed in section 3.3, and training the model. The performance of the model will be evaluated on the test data.

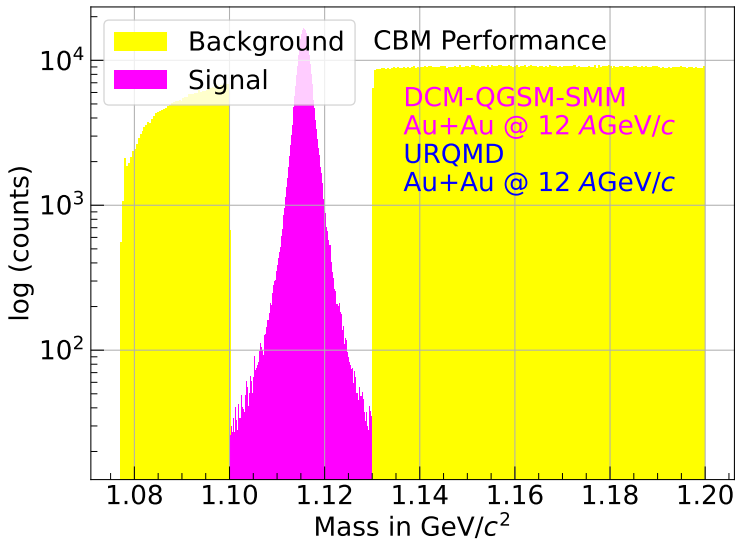


Figure 3.3: The graph shows the true MC Λ candidates, magenta, selected from the DCM model in the 5σ region around the Λ peak along with combinatorial background selected (yellow) from the UrQMD model in the sidebands.

3.3 XGBoost Hyper-parameters Tuning

The Optuna library is used for finding the hyper-parameters (sec. 2.3.1.2) of the XGBoost algorithm. It offers many sampling methods but two of them, TPE (sec. 2.4.1.1) and CMA-ES (sec.2.4.1.2), are used in this study. Each sampling method was iterated three times. Each time the number of trials was 5, the range (minimum or min and maximum or max) of various hyperparameters is given in Table 3.1. In each trial, 3-fold cross-validation (sec. 2.4.3) is performed and the model performance is judged on the validation part (or the testing part of the train data). The number of subsample was fixed at 0.8, the area under the curve

HPs	min	max	TPE1	TPE2	TPE3	CEs1	CEs2	CEs3
n_est	100	500	300	160	240	300	250	300
α	2	30	26	11	25	16	5	16
scale P-W	1	10	2	4	6	5	2	6
γ	0	1	0.58	0.02	0.44	0.49	0.70	0.61
η	0.01	1	0.06	0.08	0.12	0.10	0.10	0.09
max depth	0	10	6	8	6	5	4	5
AUC (10^{-2})			99.47	99.47	99.47	99.47	99.47	99.47

Table 3.1: The table shows the hyper-parameters (HPs), the minimum and maximum range of a hyper-parameter, and the best value returned by TPE and CmaEs (CEs) for each hyperparameter. The TPE3 values were used for the selection of Λs of the interval: mulitplicity$= [200, 400]$, $p_T = [0, 0.6]$ GeV/c, and rapidity $= [0, 1.6]$

.

(AUC) was used as the evaluation metric, and the hist tree method was used for speed. The model, trained on the best hyper-parameters of an individual trial, is evaluated on the test part of the 3-fold, and its evaluation AUC is plotted for each sampler in Fig. 3.4. The higher the AUC the better the model. The number, e.g., TPE1 shows the first attempt of using the TPE sampler for 5 trials and the best with an individual attempt shows that during a single trial, multiple hyper-parameters can be found but the best is saved. It can be concluded that, if performed for 5 trials, the two sampling algorithms, i.e., TPE and CmaEs perform similarly in finding the best hyper-parameters as all the curves converge

after the 3rd trial. This means that both methods perform similarly over 5 trials and both methods are suitable for this work. However, TPE is faster, therefore it was selected as a default method.

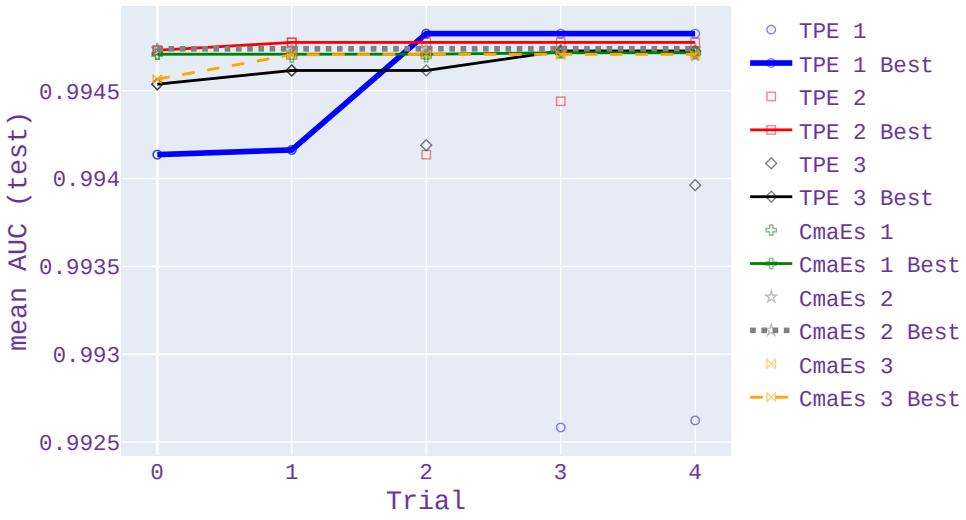
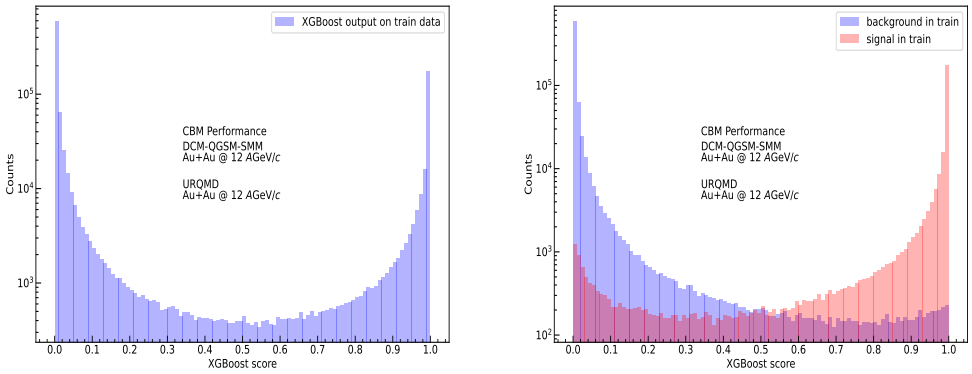


Figure 3.4: The plot shows the average AUC score, on the validation data, of the best hyper-parameters found in a search using TPE and CMA-ES in each trial. Each point on the graph shows the validation score of the ML model based on the hyper-parameters found by the hyper-parameter search algorithm.

3.4 Bias and Variance Check

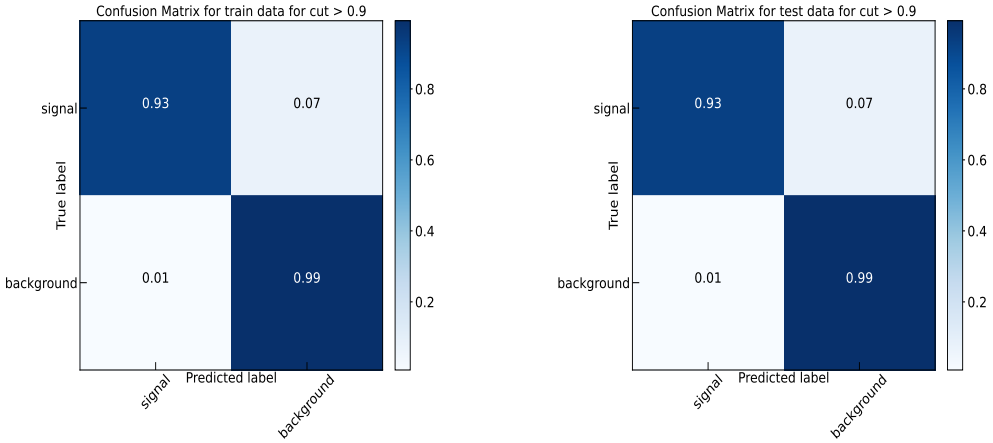
After finding the best hyper-parameters for the training data, the model is deployed first on the train and then on the test data set, and it returns a distribution of probabilities for each set. The probability for each Λ candidate to be a signal will be referred to as XGBoost score. Fig. 3.5a shows the overall output of the ML model on the train data while Fig. 3.5b plot shows the true signal and background distributions in the XGBoost score. According to Fig.3.5 the model puts signal like Λ candidates near 1 and background like near 0. This means that if a selection threshold, greater than zero, is applied on the XGBoost score to select Λ candidates then some true signal candidates will be also lost. To understand the behavior of selection criteria on the XGBoost score in terms of the true signal loss, a confusion matrix needs to be plotted. Fig.3.6a shows the confusion matrix for train data, normalized to 1, for a particular selection, i.e., 0.9 on the XGBoost score. All candidates with an XGBoost score greater than 0.9 are selected. The confusion matrix shows that at this selection on the XGBoost score, 93% of the MC true Λs are correctly identified while $< 1\%$ background has also passed the selection. Also, 7% of true Λs were lost by this selection but it also removed 99% of the total background. Similarly, Fig. 3.7a shows the XGBoost score for the test data and Fig. 3.7b shows the MC signal and background distributions in the score. Fig. 3.6b right shows the confusion matrix for the test data where the selection is applied on the 0.9 threshold.

Fig. 3.5b and Fig. 3.7b can be combined in one image to look at the performance of the model on the training and testing data, along with MC information, as shown in Fig.3.8 and it shows that the model output for the two data sets looks similar. Also, the two confusion matrices (Fig.3.6a, Fig.3.6b right) reveal that the selection of a particular XGBoost score results in similar ML model performance on both train and test data sets. To ensure that the ML model is behaving similarly on the train and test data one needs to calculate the confusion matrix for each selection on the XGBoost score. The ROC curve (Fig.3.9) is a plot of True Positive Rate (TPR) vs False Positive Rate (FPR) for all the thresholds that one wants to apply on the XGBoost score. The area under the ROC curve can be used as a criterion to judge the performance of an ML model. Fig.3.9 also confirms that



(a) **(b)**

Figure 3.5: The left side image shows the output of the XGBoost model on the train data set. The right side image shows the distribution of MC true Λ candidates and MC background in the predictions

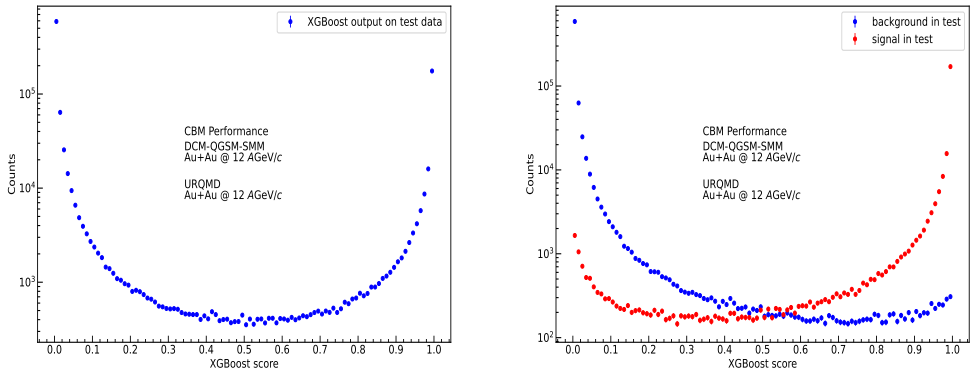


(a) **(b)**

Figure 3.6: The left plot shows the confusion matrix for train data while the right shows for test data. The two confusion matrices differ from each other in the 3rd digit after the decimal point.

the model is working similarly on the two data sets because the AUC is differing only by 0.1%.

Since the model was trained on enhanced data, it is tested on full UrQMD generated events at $p_{beam} = 12\ A$ GeV/c passed through the CBM Geant4 setup. The inv. mass distribution of the Λ candidates before (blue) and after (red) the application of ML selection criteria is shown in Fig. 3.10. The combinatorics is so huge before the application of the selection criteria that the Λ peak is not visible to the naked eye. After the deployment of the ML model, the background is reduced and the peak is clearly visible. This visual test suggests that the model



(a) **(b)**

Figure 3.7: The left image shows the output of the XGBoost model on the test data set. The right image shows the distribution of MC true Λ candidates and MC background in the predictions

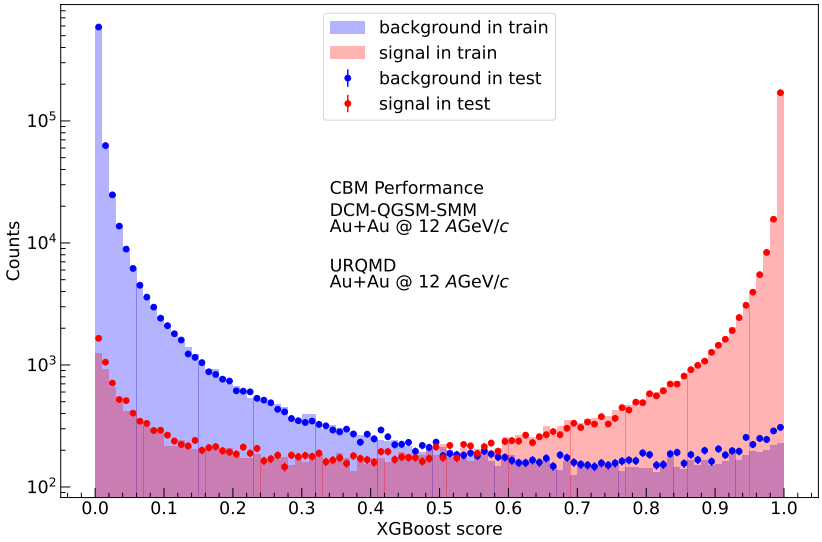


Figure 3.8: The graph shows the XGBoost score for both the train and test data set. This graph is the combination of Fig 3.5 and 3.7. The performance of the ML model on the train (filled histogram) and test (circles) is almost the same.

performs similarly on enhanced data and realistic simulated data.

Selection criteria can reduce combinatorial background, but it can also reject signal. To quantify the loss of signal candidates, an ML efficiency, denoted by ε_{ML}, can be introduced. This represents the ratio of the selected signal (Λ_{slc}) to the reconstructed signal (Λ_{recons}). A $p_T - y_{Lab}$ distribution of ε_{ML} for the threshold of 0.9 on the XGBoost score is shown in Figure 3.11, where it can be observed that the efficiency reaches up to 0.9.

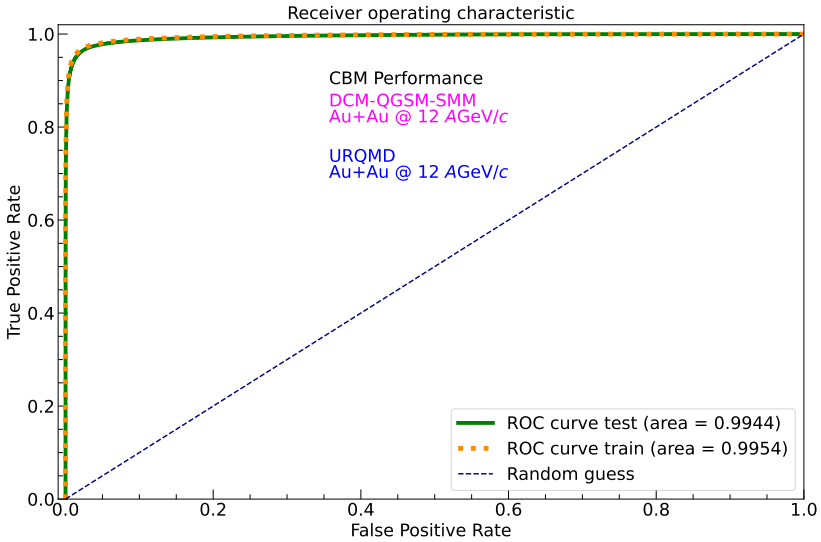


Figure 3.9: The graph shows the ROC curves for the train (dotted orange line) and test data (green line). The AUC of the ROC curve is shown in the legend.

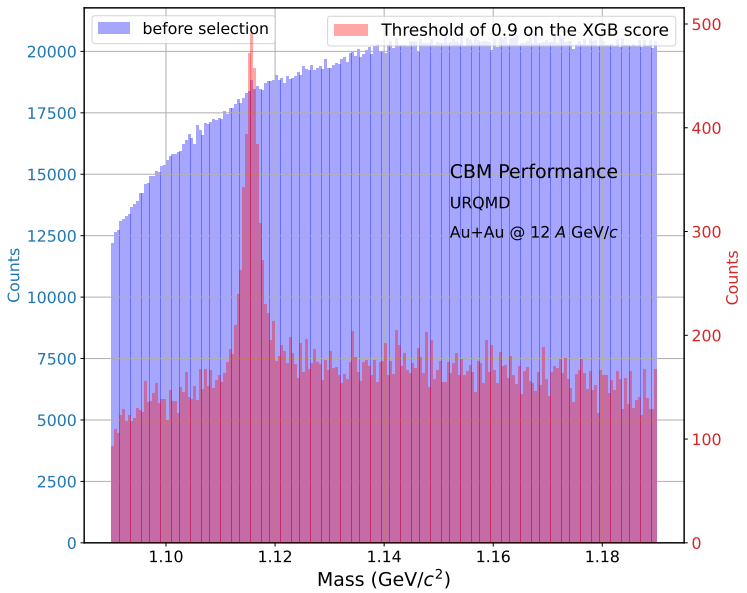


Figure 3.10: The inv. mass distribution of the UrQMD model data before (blue) and after (red) the application of ML-based selection criteria. The counts for the data before the application of ML are on the left side y-axis and for the data after ML model deployment on the right side.

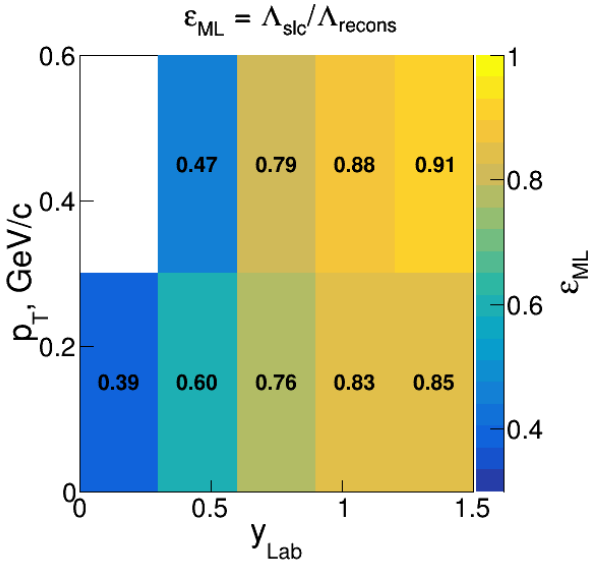


Figure 3.11: The ε_{ML} is shown here for the threshold of 0.9 on the XGBoost score on the UrQMD data.

3.5 Comparison between Manually & ML Optimized Selection Criteria

In this section, the ML-based selection criteria optimization is compared to manually optimized selection criteria. The manual selection criteria for Λ hyperon for CBM were based on signal-to-background ratio maximization and have been discussed here [72]. The optimum values of the hypercube found were: $\chi^2_{prim} > 18.4$, $\chi^2_{geo} < 3$, $DCA < 1$, and $L/\Delta L > 5$. On top of these, one selects π^- and p^+ which is performed by selection on the $mass^2$ information that is obtained from the TOF wall detector. The selection criteria applied to the $mass^2$ variable for the π^- (p^+) are shown in Fig. 3.12 left (right) as red lines. The procedure behind this particle identification method has been discussed in this [94] work.

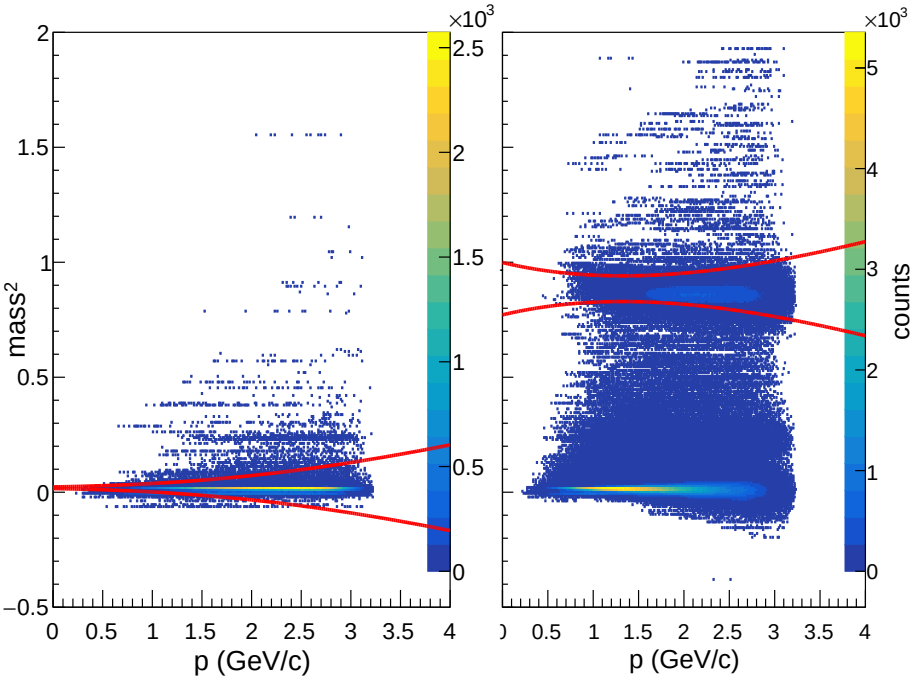


Figure 3.12: The graph shows the momentum (p) on the x-axis and $mass^2$ in $(\mathrm{GeV/c^2})^2$ on the y-axis, for UrQMD data. The PID selection criteria (red) applied to the $mass^2$ variable associated with all negatively (left) and positively (right) charged tracks are shown by the red lines.

The manually optimized selection criteria, MSC, (blue), and the ML optimized (red) are applied to 2×10^6 UrQMD generated events, transported through the Geant4 CBM setup are shown in Fig. 3.13. In the lower part of Fig. 3.13, the ratio

of the two Λ distributions is plotted. The errors are propagated using:

$$\sigma_{ratio} = \frac{\Lambda_{ML}}{\Lambda_{MSC}} \sqrt{(\frac{\sqrt{\Lambda_{ML}}}{\Lambda_{ML}})^2 + (\frac{\sqrt{\Lambda_{MSC}}}{\Lambda_{MSC}})^2} \tag{3.1}$$

with Λ_{ML} representing the bin count of the ML selected Λ distribution and Λ_{MSC} the manually optimized one. The ML-optimized selection criteria are non-linear in a multi-dimensional space and achieve a much better signal-to-background ratio while maintaining a higher efficiency. The efficiency of primary and secondary Λ are shown as text in Fig. 3.13, although the study of secondary Λ is not the focus of this study it reflects the fact that any type of true Λ is not thrown away from the distribution by ML. Similar plots have been generated for other intervals of $p_T - y_{Lab}$ and they are presented in the appendix sec. A.4.

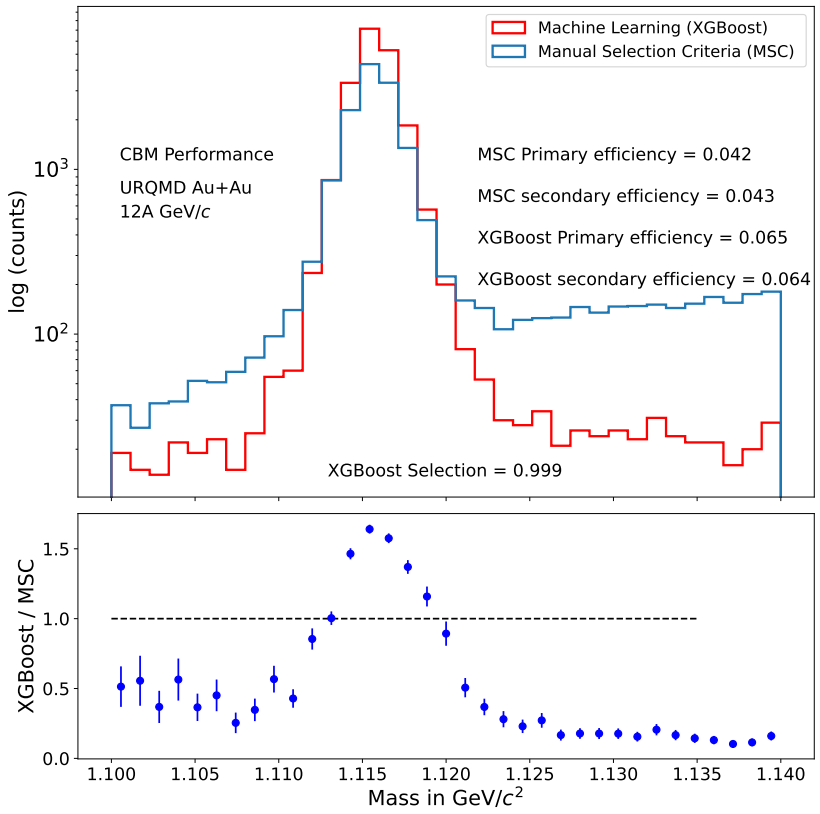


Figure 3.13: The top plot shows the Λ candidates selected by ML-optimized selection criteria (red) and manually optimized selection criteria (blue). The bottom plot shows the ratio of the two Λ distributions.

In the mass window of $1.1 - 1.13\,\mathrm{GeV/c^2}$, the signal and background candidates were counted in data after the application of ML and manual selection criteria, separately. The signal-to-background ratio for the ML selection criteria

was 43.3, while for the manually optimized hypercube was 5.9. Therefore, at the same efficiency ML selection criteria have a more than seven times better signal-to-background ratio than the hypercube.

The distributions of the variables of the Λ candidates that the ML optimized selection criteria have filtered are plotted in Fig. 3.14, and the black (red) filled circles show the MC signal (background) distribution. In contrast to Fig. 1.8, the signal distribution is in abundance. Also, the distributions of the Λ variables after the application of MSC are plotted with blue (magenta) unfilled squares showing the signal (background) distributions in Fig. 3.14. These distributions show that the background is still dominant in some places and can be suppressed by applying stricter selection. The χ^2_{geo} distribution in Fig. 3.14a goes up to the value of 8 while the DCA in Fig. 3.14b goes up to 0.02. On the other hand, the MSC applied χ^2_{geo} distribution goes up to 3 and DCA up to 1. The $L/\Delta L$ in Fig. 3.14c starts from the value 90 and the $\chi^2_{prim\ \pi^-}$ ($\chi^2_{prim\ p^+}$) in Fig. 3.14d (Fig. 3.14e) starts from 20×10^3 (1500). Sometimes the TOF hit is incorrectly assigned to a particle track and the ML algorithm, in this case, selects the Λ candidates based on other variables than the mass of p^+ and π^- as shown by Fig. 3.14f and Fig. 3.14g. The two images show that certain true p^+ are mismatched to tracks with $mass^2$ greater than 2 GeV/c^2. Similarly some true π^- are assigned an incorrect $mass^2$ of electrons.

In the case of ML selection criteria, certain variables start (end) from some low (high) value meaning that the lower (higher) values are rejected. This is not the true picture because the selection criteria of ML are non-linear and some intermediate values of some variables are also eliminated. This means that multiple variables are used non-linearly to apply a certain selection. The selection applied to the XGBoost score, in the comparison case, is to maintain similar efficiency to that of the manual selection criteria. The distributions may change on a different threshold on the XGBoost score and this makes ML-based selection easily adjustable according to one's need. If one wants high efficiency with a low signal-to-background ratio then one applies a low threshold on the XGBoost score and vice versa. The manual-based hypercube is more rigid and to increase (decrease) efficiency one will have to change the hypercube. Also, manual-based selection

criteria depend on the $mass^2$ information but some π^- never reach the TOF detector and they are automatically rejected in the first stage of the selection. The matching algorithm that connects a TOF hit to a track from the tracking system is not ideal and therefore some tracks are mismatched to an incorrect $mass^2$ value. Manually optimized selection criteria are more vulnerable to this mismatch than ML selection criteria.

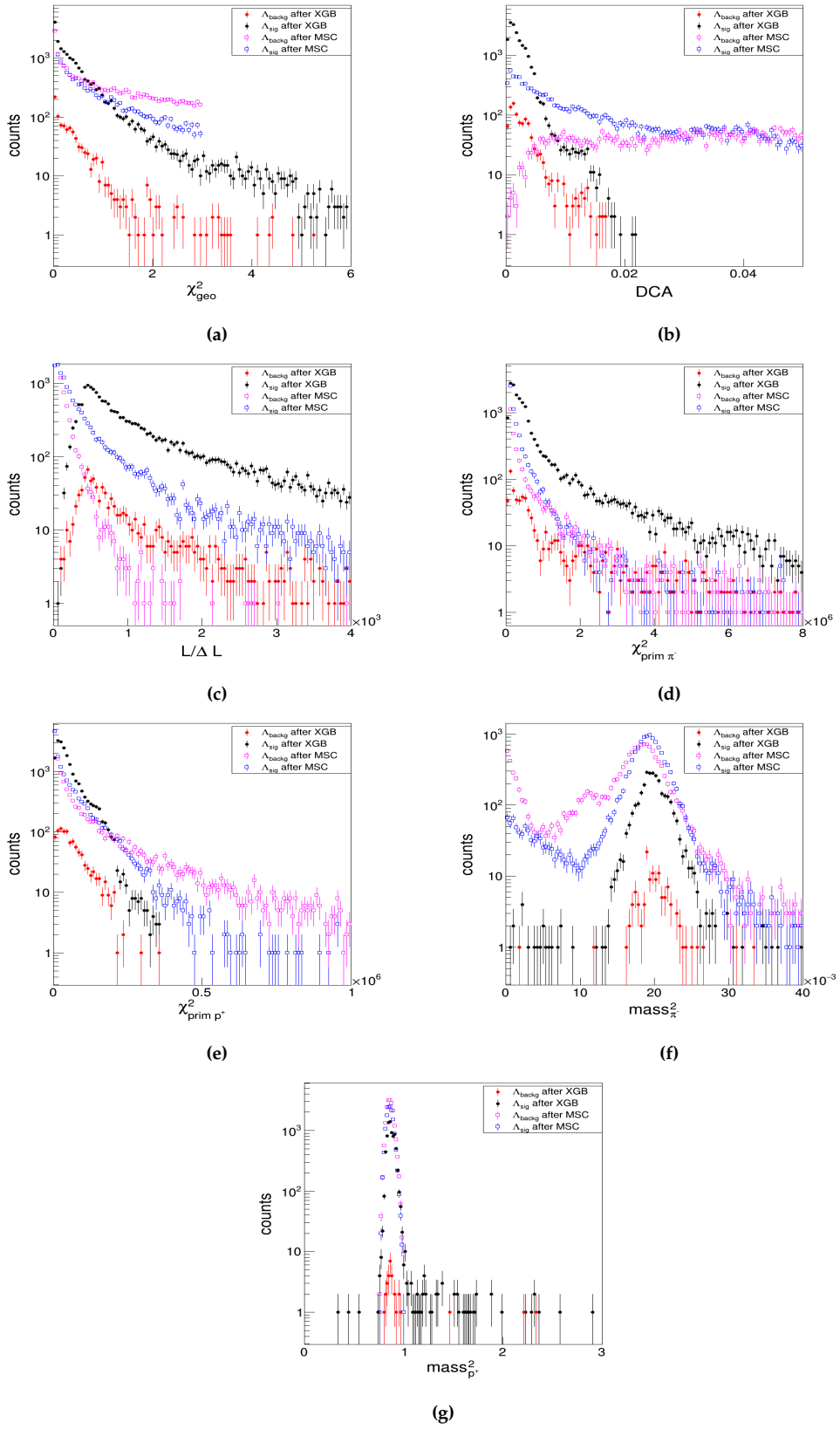


Figure 3.14: The graphs show the distributions of variables associated with Λ after the application of ML-optimized and manually-optimized selection criteria, for UrQMD-generated data. The signal is represented by black full circles (blue open squares) and the background with red full circles (magenta open squares) for the ML-optimized (manually optimized) selection criteria applied.

3.6 Visualization of the ML Model and Ranking of the Variables

ML models such as XGBoost contain a large number of hyperparameters and that makes them not easily explainable. Since the global explanation of the behavior of these models is complicated, SHAP (Sec. 2.6) can be used to approximate the behavior locally. Fig. 3.15 shows the SHAP values of the variables associated with the ML model for one Λ candidate. SHAP approximates the ML model performance on this single sample. The variable which has the highest SHAP score is put on the top while the variable with the lowest SHAP score is on the bottom. The decision to assign the XGBoost score to a candidate can be considered as the sum of its SHAP scores. For example, in Fig. 3.15 the sum of all SHAP scores is less than zero and therefore this candidate is assigned a low XGBoost score (0.011).

For a number of samples, the SHAP model approximates the ML model generally. The higher the number of samples the better the SHAP model fits the real ML model. For 3×10^4 samples from the training data, the SHAP plot is shown in Fig. 3.16. It shows that the model is using primarily the distances of the positive and negative tracks to the PV, i.e., $\chi^2_{prim\ p^+}$ and $\chi^2_{prim\ \pi^-}$ as the top classification variables with the biggest SHAP scores. The SHAP scores for the values of variable $\chi^2_{prim\ p^+}$ go up to 4 while the SHAP score for the values of variable $\chi^2_{prim\ \pi^-}$ end up below 3. This means that $\chi^2_{prim\ p^+}$ is much more useful for signal identification. Also, it shows that the higher values (red color dots) of these two variables are more useful in the classification of signal (positive SHAP value score). The higher the distance to the primary vertex of the daughter tracks the more likely the model treats the candidates as the signal. This is further cemented by Fig.3.17a where the SHAP values for the variable $\chi^2_{prim\ p^+}$ are plotted as a function of the variable values. The SHAP score for the candidates with $\chi^2_{prim\ p^+} < 20$ are mostly negative. These candidates are more likely to originate from the primary vertex and not from a Λ decay. The plot for the $\chi^2_{prim\ \pi^-}$ vs its SHAP values is plotted in Fig.3.17b and the higher the value of the variable the higher the SHAP score. However, the SHAP score does not go as high as it does for the $\chi^2_{prim\ p^+}$

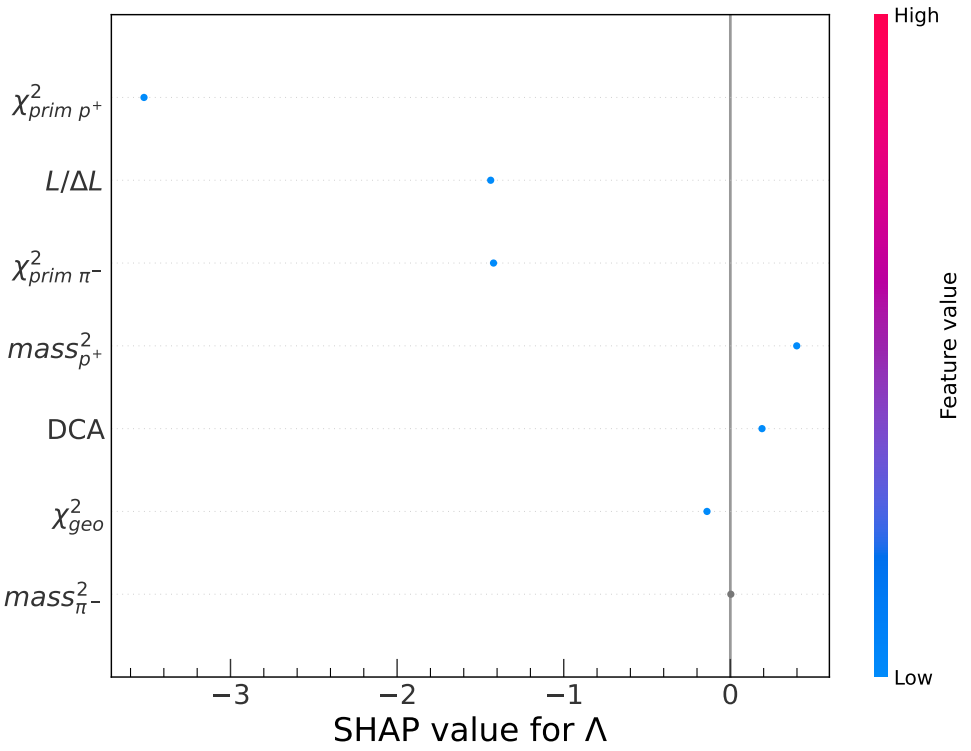


Figure 3.15: The graph shows the SHAP values associated with each feature for a single Λ candidate on the x-axis and the variables on the y-axis. The color bar shows the values of the variables with red meaning higher values and blue meaning lower values. The grey color shows missing values. The single entry was taken from the train data. At least two candidates are required to compare different feature values, that's why no red color is visible here.

variable.

The model finds the distance, normalized to its error, between primary vertex and secondary vertex ($L/\Delta L$) the 3rd useful, shown by Fig. 3.16. Fig. 3.17e shows that the lower values (less than 10) of this variable are given more negative SHAP scores. The SHAP values mostly go above zero around 10 and then go below zero after 20. The higher values of the mass of the positive track ($mass^2_{p^+}$) are also correlated with the signal because they have been given high SHAP values in Fig. 3.16. There are also some grey dots here which are tracks where the information of the $mass^2$ variable from the TOF detector is not available. These missing values are sometimes also given SHAP scores as the algorithm can take them into account, as discussed in sec. 2.3.1.1. The SHAP values for the variable $mass^2_{p^+}$ are shown in Fig. 3.17d and they show that a high SHAP score is given to values near the mass of p^+, $mass_{p^+} = 0.938\,\mathrm{GeV/c^2}$ [33]. Higher values of distance (DCA and

χ^2_{geo}) between the two tracks are given lower SHAP scores, i.e., these candidates are treated as background. The SHAP values for DCA plotted against DCA value in Fig. 3.17c shows that the closer the two tracks come toward each other the higher the SHAP score and vice versa. The mass of the negatively charged track ($mass^2_{pi^-}$) is the least useful for the segregation of signal from the background.

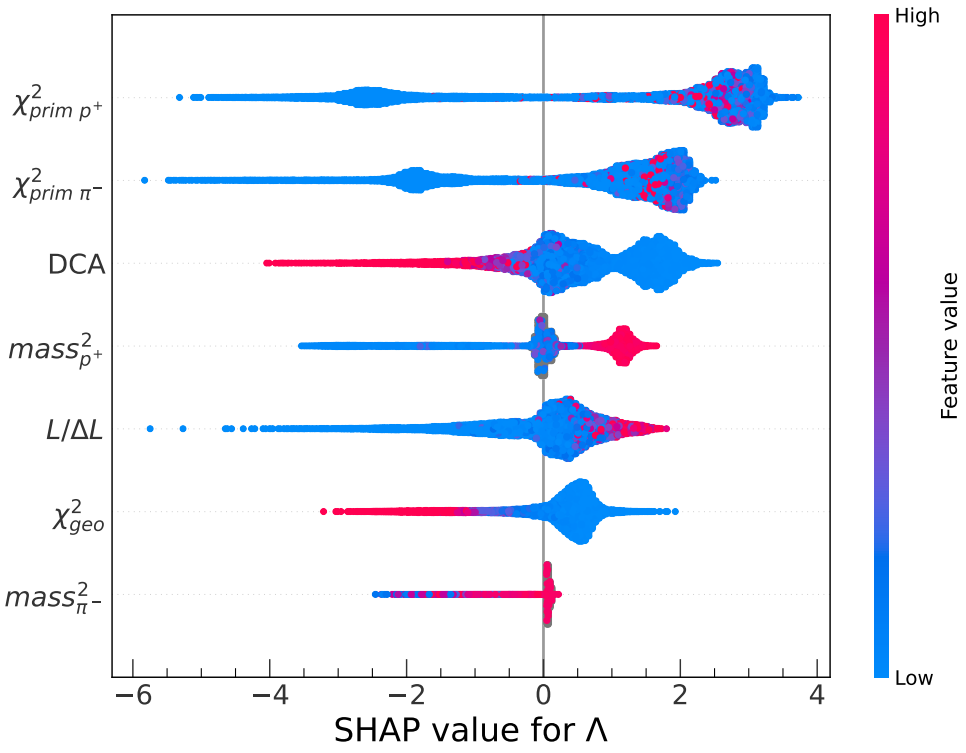


Figure 3.16: The graph shows the features on which the model was trained on the y-axis and the SHAP score on the x-axis.

The SHAP analysis provides insights into how the machine learning (ML) model is making decisions based on the underlying decay topology of the variables. As discussed in section 1.4.6, the distance of the daughter track from the primary vertex is a crucial variable for identifying whether it originates from the primary vertex or from the decay of a particle. Consistently with the decay topology, the ML algorithm assigns higher rankings to daughter tracks that are farther from the PV. Additionally, the proximity of two tracks to each other is a key feature for identifying tracks that arise from decay. Once again, the ML model reflects the decay topology by assigning higher scores to lower values of the DCA between the tracks. The farther away the secondary vertex from the primary vertex the more likely the tracks are from decay and the SHAP analysis also reveals

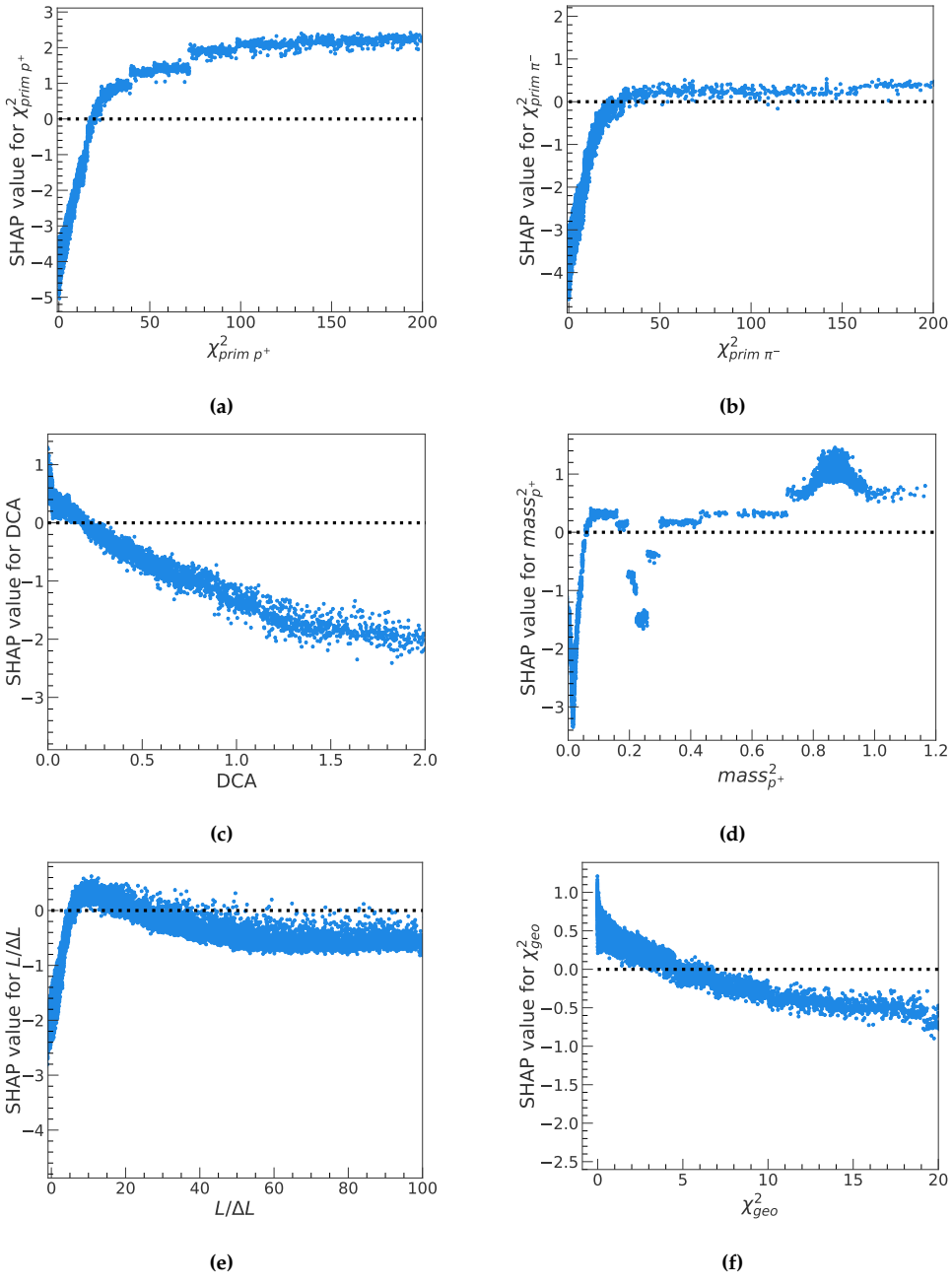


Figure 3.17: SHAP score for different variables is shown against the values of the variables.

the same.

Further SHAP analysis is described in the appendix section A.2. That analysis includes SHAP plots for the signal-only data with a high XGBoost score, i.e., greater than 0.99. The SHAP distributions show that the ML model performs the same on signal-only data as it does on the total data. Each candidate's individual value of a variable gets a score and the final decision is made on the final sum of all the scores from all the variables.

3.7 Conclusions on Selection Criteria Optimization

Selection optimization was performed using the ML algorithm XGBoost multi-differential in multiplicity, transverse momentum and rapidity. The selection criteria reduces the background but also rejects signal and the ML efficiency term quantifies the rejection. The ML efficiency value depends on the threshold on the XGBoost score and this will be discussed in chapter 4, but at a threshold of 0.9 it can go up to 0.9. The SHAP explanation revealed that the ML model makes decisions based on the underlying decay topology.

The ML selection criteria because of its non-linear nature in a multi-dimensional space show better suppression of background than the manual hypercube selection at a better efficiency. This resulted in a seven times better signal-to-background ratio in the ML case. ML selection criteria are less vulnerable to mismatches in the TOF matching algorithm than the manually optimized selection criteria. Also in the case where there is no TOF information available for the π^- and p^+ the ML selection criteria could still make a decision while the manual selection criteria could not.

Chapter 4

Yield Extraction and Systematic Uncertainties

After the application of ML-optimized selection criteria, the raw yield of the Λ hyperon is extracted differentially in p_T and y_{LAB} in a multi-step inv. mass fitting routine. The signal distribution is approximated using the Double-Sided Crystal Ball (DSCB) function [95], while background estimation is performed using a fourth-order polynomial ($pol4$). The yield is obtained by integrating the DSCB function within a few σ regions around the Λ inv. mass peak. Counting the total MC true signal is considered as the yield of the DCM model data and the ratio of this yield to the simulated yield defines the total efficiency. The total efficiency is then used to correct the yield obtained from UrQMD data, through the fitting routine.

4.1 Raw-Yield Extraction Procedure

After the application of the XGBoost model on the data and the selection of a certain XGB score as a signal discrimination threshold, some combinatorial background is still left, as shown by the tails of the inv. mass in Fig. 3.13. The separation of this combinatorial background from the signal is performed using a multi-step fitting routine on the invariant inv. mass distribution of the Λ candidates. The functions used to fit the signal and background distribution were selected because they resulted in a lower $\chi^2_{red} = \chi^2/NDF$. Here χ^2 is the weighted

sum of squared residuals and the number of degrees of freedom (NDF) is the difference between the number of observations and the number of fitted parameters. However, a comparison between the DSCB and a Double Gaussian (DG) is also performed here.

The DSCB function is a combination of two Crystal Balls and is defined as:

$$f(m; A_0, \mu, \sigma, a_L, a_R, n_L, n_R) = A_0 \times \begin{cases} A_L(B_L - G)^{-n_L} & \text{if } G < -a_L \\ e^{-(G)^2/2} & \text{if } a_R > G \geq -a_L \\ A_R(B_R + G)^{-n_2} & \text{if } G \geq a_R \end{cases} \quad (4.1)$$

with $A_{L/R} = (n_{L/R}/a_{L/R})^{n_{L/R}} \times e^{-a_{L/R}^2/2}$, $B_{L/R} = \frac{n_{L/R}}{a_{L/R}} - a_{L/R}$ and $G = \frac{m-\mu}{\sigma}$. The fit parameter μ is the meanwhile σ is the standard deviation of the Gaussian. The A_0 is a normalization coefficient. The parameters of the power tail $(A_{L/R}(B_{L/R} - G)^{-n_{L/R}})$ are $a_{L/R}$ and $n_{L/R}$. The $a_{L/R}$ is the parameter that decides the switching to the Gaussian from the power tail on the left/right.

The 4th-order polynomial is defined as:

$$pol4 = p_0 + \sum_{i=1}^{4} p_i \frac{C^i}{i!} \quad (4.2)$$

with $C = x - mass_\Lambda$, where $mass_\Lambda = 1.115\,\mathrm{GeV/c^2}$ [96], and the p_is are the fitting parameters.

The DG consists of two Gaussians centered around the same mean μ and is defined as:

$$DG(m, A, \mu, \sigma_1, B, \sigma_2) = A \left[\frac{(1-B)}{\sqrt{2\pi}/\sigma_1} e^{\frac{-(m-\mu)^2}{2\sigma_1^2}} + \frac{(B)}{\sqrt{2\pi}/\sigma_2} e^{\frac{-(m-\mu)^2}{2\sigma_2^2}} \right] \quad (4.3)$$

where σ_1 and σ_2 are the standard deviations of the two Gaussians. Parameters A and B are coefficients.

To ensure the stability of the fitting routine, a multi-step procedure is employed. The yield is extracted multi-differentially by dividing the data into 10 intervals of p_T, with a size of $0.3\,\mathrm{GeV/c}$ for each interval. Each p_T interval is further divided into 10 intervals of y_{Lab}, with a width of 0.3. This section focuses on the $p_T = [0.3, 0.6]$ GeV/c and $y_{Lab} = [1.2, 1.5]$ interval. The same procedure is applied to all other p_T and y_{Lab} intervals, although the results are not discussed

here. The inv. mass fits of the other intervals are plotted in the Appendix section A.4.

In the first step of the fitting routine, MC signal-only data is selected from the DCM-QGSM-SMM data with an XGBoost threshold of 0.53. This data is used to create a histogram of the inv. mass, with 500 bins in the range $1.08 - 1.2\,\mathrm{GeV/c^2}$. The DSCB parameters $(A_0, \mu, \sigma, a_L, a_R, n_L, n_R)$ are initialized with values given in appendix sec. A.4, and their ranges are bounded. The fitting range is set as 4.5 standard deviations of the histogram data around the mean value of the histogram. The histogram and the fitted DSCB function are displayed in Fig. (4.1). The yield (Λ_{yield}) is computed by integrating the fit function over an $\pm 11\sigma$ region centered around the μ of the Gaussian parameters of the fit function. The $\pm 11\sigma$ region covers the peak region of the inv. mass. However, due to issues in the reconstruction, a very small number of true Λs can end up far away from the main peak, and they are excluded from fitting. To estimate the uncertainty of the fit integral ($\sigma_{\Lambda_{yield}}$), it is assumed that the relative error is the same as for the total integral. The value of a fit parameter, such as μ, and the uncertainty in the value of this fit parameter, such as σ_μ, are used to calculate the ratio $\frac{\sigma_\mu}{\mu_{value}}$ which is multiplied by the integral of the fit function. The value of $\Lambda_{yield} \times \frac{\sigma_\mu}{\mu_{value}}$ estimates the uncertainty of the fit integral. The number of true Λs are counted as well and the sum is called MC true yield. The obtained yield agrees with the MC true yield within the uncertainty of the fit integral.

Fitting the DG function to the signal-only distribution was also implemented in the same way as the DSCB, and the yield was calculated as the integral of the DG function in the 3σ (larger σ of the two σs) region around μ. Although the integral of the fit function agrees with the total number of MC true Λ candidates, the fit does not describe the tails of the distribution very well, as shown in Fig. 4.2. Also, a visual inspection of the ratio plot reveals that the fit function does not describe the peak region as well as the DSCB function does. Additionally, the χ^2_{red} value (9.2) was larger than that of DSCB (2.6).

In the second step of the fitting routine, the inv. mass distribution of the UrQMD data is utilized. The invariant mass window $(\mu - 8\sigma) - (\mu + 8\sigma)$ is excluded, using the σ and μ fit parameters from the first step. Consequently, the

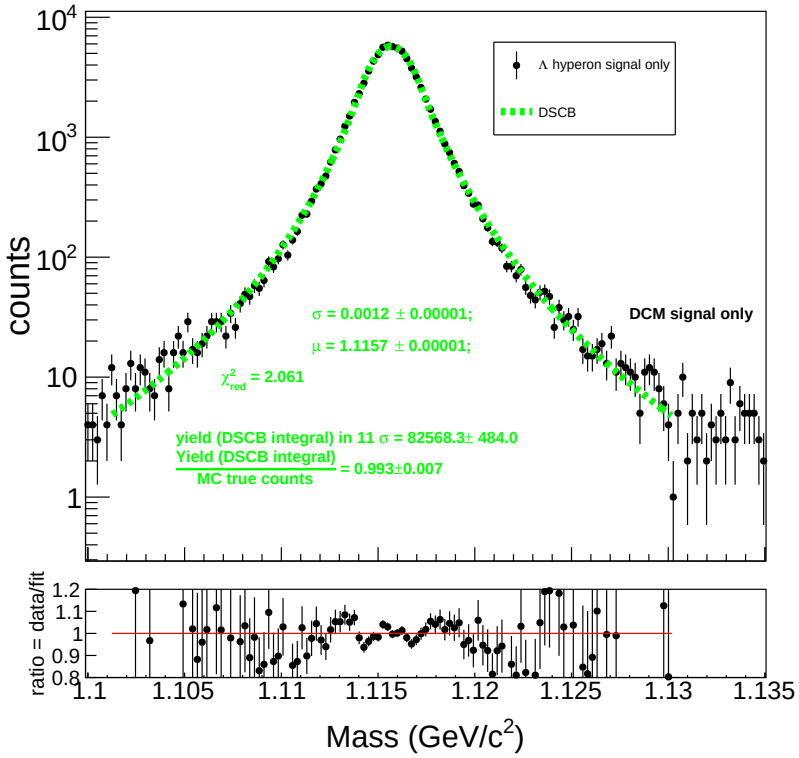


Figure 4.1: The inv. mass histogram, in black circular markers, of the DCM-QGSM-SMM data along with the DSCB fit, green dotted curve, is plotted for $p_T(\mathrm{GeV/c}) - y_{LAB} = [0.3, 0.6] - [1.2, 1.5]$ is shown here. XGBoost probability selection of 0.53 is applied. The bottom plot shows the ratio plot (black circles) of the data and the fit. The red line is the $ratio = 1$.

invariant mass range lying beyond the red perpendicular lines shown in Fig.4.3 is fitted with a $pol4$ function. The fitted function is extended to the inv. mass peak region and is shown by the magenta-dotted curve. As actual accelerator facility-produced data will replace the UrQMD data in the future, this process will continue to work because the background data will be chosen outside the Λ peak region.

The third step of the fitting routine involves fitting the total invariant mass distribution with the $DSCB + pol4$ function. The initialization of the parameters uses the fit parameters obtained from the last two steps. The red dotted curve in Fig. 4.4 represents the resulting total fit. To calculate the raw yield, the signal-only function, which is the DSCB (green dotted curve), is integrated inside the 11σ region (green perpendicular dotted lines) around μ. The raw yield agrees

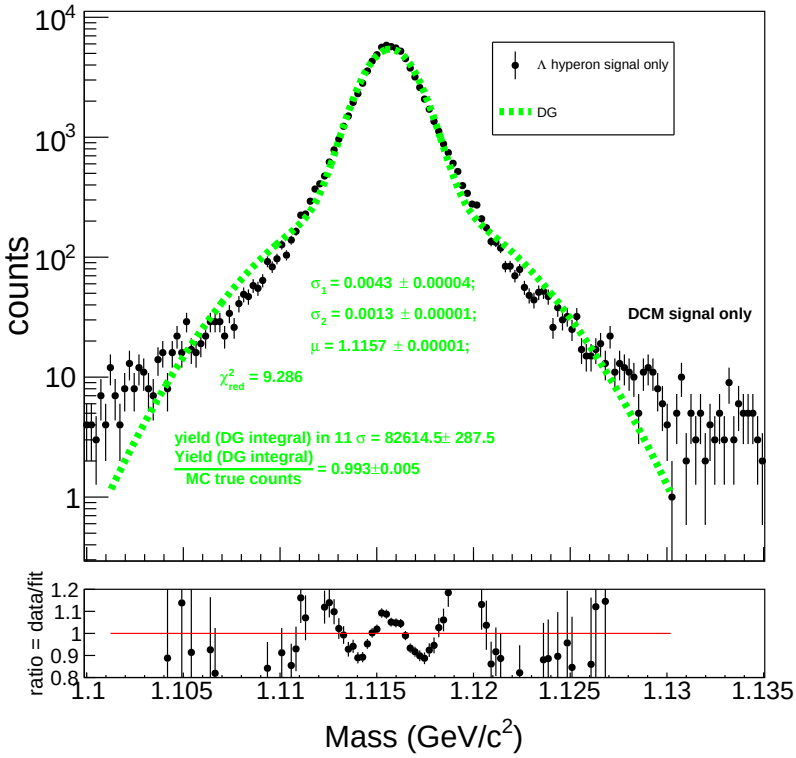


Figure 4.2: The MC true signal only inv. mass histogram of the DCM model-generated data (black circles) is fitted by a DG fit function (green dotted curve).

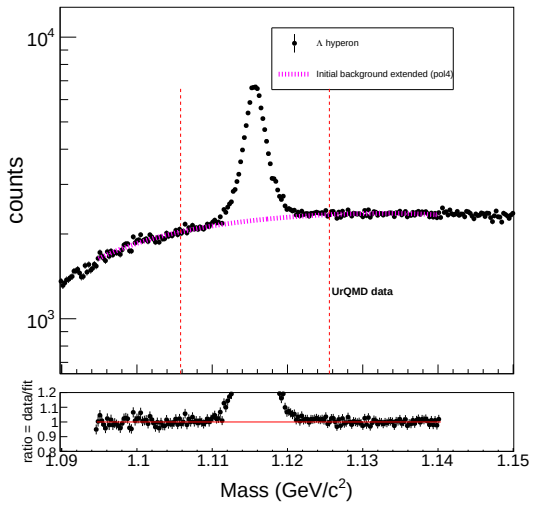


Figure 4.3: The inv. mass of the UrQMD model data (black circles) is fitted in the sidebands of the Λ peak (the region outside the red perpendicular lines) with a *pol*4 (dotted magenta curve).

with the MC true signal (found by counting all MC true) within the uncertainty.

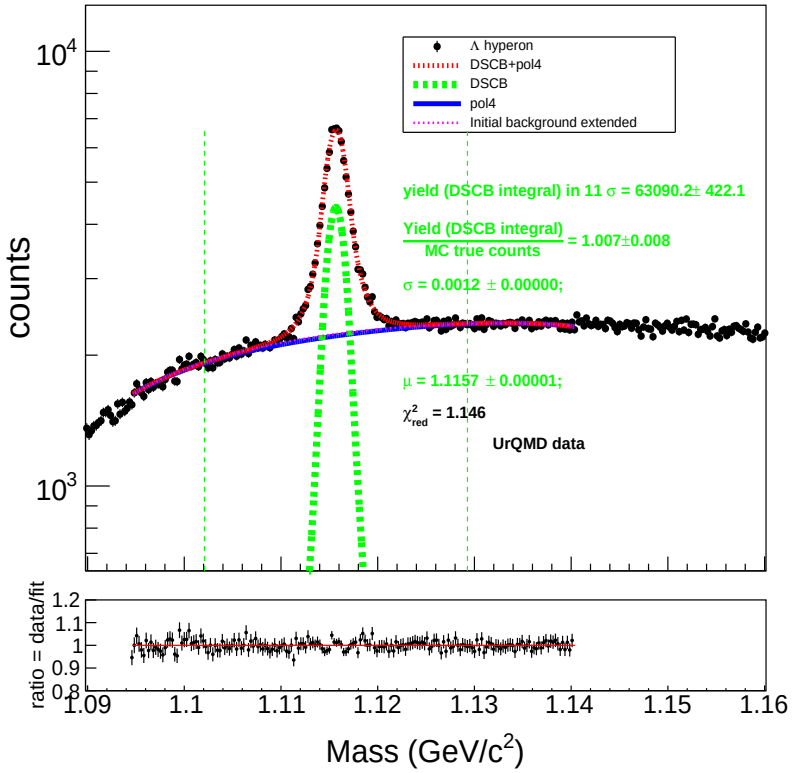


Figure 4.4: The inv. mass of the Λ hyperon of UrQMD data (black circles) is fitted with $DSCB + pol4$ (red dotted curve). The $DSCB$ only part (green dotted curve) of the total fit function approximates the signal part while the $pol4$ (blue curve) approximates the background-only distribution. The perpendicular green dotted lines on both sides of the Λ peak show the inv. mass range where the signal function is integrated for yield calculation.

4.2 Efficiency Correction of Raw-Yield

Fixed target experiments, such as CBM, adopt a forward spectrometer geometry with an angle coverage of 2.5°-25° to cover the mid-rapidity region. Nonetheless, not all Λ candidates produced in heavy-ion collisions are detected by this detector geometry. The detector's inefficiency to detect all Λ candidates is quantified by the geometrical acceptance (Acc) of the experiment. Furthermore, even those Λ candidates that pass through the detector geometry are not always successfully reconstructed due to the limited efficiency of the candidate reconstruction chain. About 63.9% [96] of the total Λ candidates decay through the $\Lambda \rightarrow p^+ + \pi^-$ decay channel. The efficiency for reconstruction and branching ratio are combined into the term ϵ_{comb}. Therefore, if N is the number of simulated Λs (Λ_{sim}), only $Acc \times \epsilon_{comb} \times N$ will be reconstructed by the CBM experiment.

The $p_T - y_{Lab}$ distribution of simulated Λ by DCM for 2×10^6 Au-Au events is illustrated in Fig. 4.5a, corresponding to charged tracks multiplicity interval $= [200, 400]$ per collision. The simulated signal shows $p_T - y_{Lab}$ dependence and this is due to the simulation model with its modeling physics. Fig. 4.5b presents the reconstructed true Λs by the CBM reconstruction chain. To calculate the $Acc \times \epsilon_{comb}$, the number of reconstructed Λ candidates is divided by the number of simulated ones. The $p_T - y_{Lab}$ distribution of $Acc \times \epsilon_{comb}$ for Λ candidates for the multiplicity interval $[200, 400]$ is shown in Fig. 4.5c, while Fig. 4.5f shows the zoomed-in $p_T = [0, 0.6]$ GeV/c and $y_{Lab} = [0, 1.5]$ interval. The reconstruction efficiency for Λ is non-uniform in different $p_T - y_{Lab}$ intervals and it can go up to 0.52. Similarly, the $Acc \times \epsilon_{comb}$ was calculated for the multiplicity interval $[0 - 200]$ (shown in Fig. 4.5d) and it can go up to 0.54. The efficiency for multiplicity interval $[0 - 200]$ is divided by the efficiency of multiplicity interval $[200, 400]$, and the result is shown in Fig. 4.5e. The ratio of the efficiencies of two multiplicity intervals in all intervals of p_T and y_{Lab} is not one, indicating that the $Acc \times \epsilon_{comb}$ is non-uniform across different multiplicity, p_T and y_{Lab} intervals. This non-uniformity is due to the detector's response.

Performing selection and yield extraction using all the data within a single interval is unlikely to result in a successful correction procedure. This is because the simulated signal depends on both p_T and y_{Lab}, and the reconstruction effi-

ciency is non-uniform across different intervals of multiplicity, p_T and y_{Lab}. To correct for the non-uniformity of the signal in different intervals of p_T, y_{Lab} and multiplicity, it is important to perform the optimization of selection criteria, yield extraction, and yield correction multi-differentially.

ML-based selection criteria reject combinatorial background but may also eliminate signal, leading to a certain efficiency (ϵ_{ML}) for each selection based on the XGB score. There is an inverse relationship between efficiency and the use of an XGB score as selection criteria for candidates and it will be discussed in sec. 4.4.1. Fig.4.6a shows the ML-only efficiency at the XGB score of 0.53, reaching up to 96% for the multiplicity $[200, 400]$ interval. Since the selected Λ candidates (Λ_{slc}) are a subset of the reconstructed Λ candidates (Λ_{recons}), the uncertainty in the ML efficiency is calculated using the error propagation of binomial statistics. The uncertainty [97] is defined as:

$$\sigma_{\epsilon_{ML}} = \left((1 - \frac{2\Lambda_{slc}}{\Lambda_{recons}})\sigma^2_{\Lambda_{slc}} + \frac{\Lambda^2_{slc}}{\Lambda^2_{recons}}\sigma^2_{\Lambda_{recons}} \right) / \Lambda^2_{recons} \tag{4.4}$$

where $\sigma_{\Lambda_{slc}}$ ($\sigma_{\Lambda_{recons}}$) is the square root of Λ_{slc} (Λ_{recons}). ML efficiency can be combined with the $\epsilon_{comb} \times Acc$ to obtain a total efficiency, i.e.,

$$\epsilon_{total} = \epsilon_{ML} \times \epsilon_{comb} \times Acc. \tag{4.5}$$

Fig.4.6b displays the values of ϵ_{total}, which reach up to 43%. The associated uncertainty ($\sigma_{\epsilon_{total}}$) is calculated using equation 4.4, but replacing Λ_{recons} with simulated Λ candidates (Λ_{sim}). To correct for the lost signal candidates, the yield obtained in section 4.1 requires correction by the factor ϵ_{total}, which depends on the selection threshold applied to the XGB score.

The DCM collision simulator data is treated as a simulation, and efficiency is calculated using this model. Conversely, UrQMD data is treated as experimental data, and its yield is corrected on the efficiency obtained on the DCM data, after the yield extraction procedure. The corrected yield Λ_{corr} is obtained by:

$$\Lambda_{corr} = \frac{\Lambda_{yield}}{\epsilon_{total}} \tag{4.6}$$

The uncertainty of the corrected yield is calculated by considering that the ϵ_{total} of the UrQMD is correlated to the ϵ_{total} of the DCM model. This correlation is due

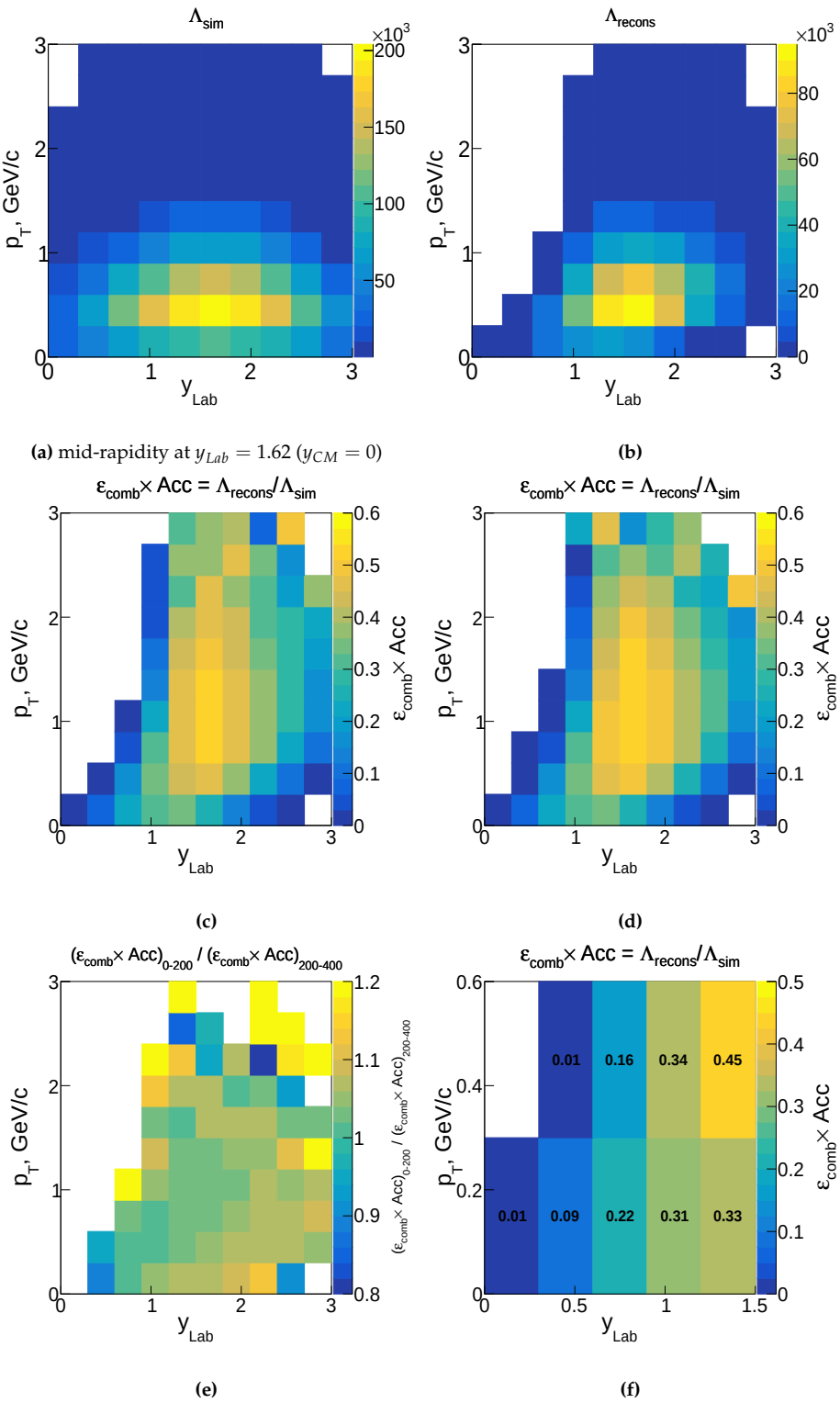
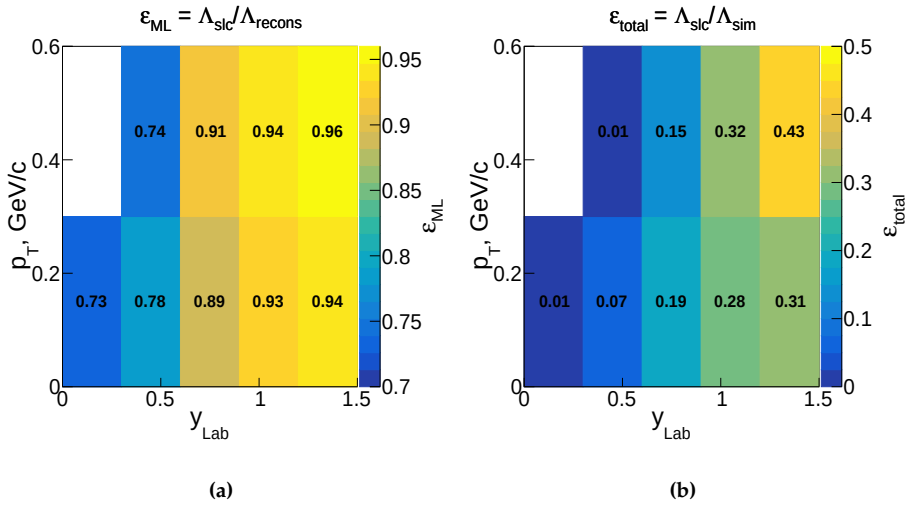


Figure 4.5: Fig. 4.5a shows the production of Λ candidates by the DCM generator for 2×10^6 Au-Au events; the reconstructed ones by the CBM reconstruction chain are shown in 4.5b, in the multiplicity interval $= [200, 400]$. The $Acc \times \epsilon_{comb}$ efficiency is plotted in 4.5c and for a smaller $p_T - y_{Lab}$ interval it is plotted in 4.5f. The $Acc \times \epsilon_{comb}$ for the the multiplicity interval $= [0 - 200]$ are shown in 4.5d and the ratio of 4.5d to 4.5c is shown in 4.5e. In 4.5e the $(\varepsilon_{comb} \times Acc)_{0-200}$ ($(\varepsilon_{comb} \times Acc)_{0-200}$) is $Acc \times \epsilon_{comb}$ for multiplicity $[0 - 200]$ ($[200 - 400]$) interval.



(a) **(b)**

Figure 4.6: The ML-only efficiency is shown in Fig. 4.6a and the total efficiency, i.e., $\epsilon_{ML} \times \epsilon_{comb} \times Acc = \epsilon_{total}$ is shown in Fig. 4.6b.

to the fact that ϵ_{total} is independent of the collision generator. This means that the ML-based selection and CBM reconstruction mechanism do not depend on the choice of collision generator. This discussion is followed in the next section (sec. 4.3) in detail. The formula for the uncertainty calculation of the corrected yield ($\sigma_{corr\ yield}$) is given by

$$\sigma_{corr\ yield} = \frac{\Lambda_{yield}}{\epsilon_{total}} \sqrt{(\frac{\sigma_{\Lambda_{yield}}}{\Lambda_{yield}})^2 + (\frac{\sigma_{\epsilon_{total}}}{\epsilon_{total}})^2 - 2\frac{\sigma_{\epsilon_{total}}\sigma_{\Lambda_{yield}}}{\Lambda_{yield}\epsilon_{total}}} \tag{4.7}$$

here σ_{yield} is the error on the fit integral of the DSCB function, and $\sigma_{\epsilon_{total}}$ is the uncertainty in the ϵ_{total}.

4.3 Efficiencies Comparison

The $\epsilon_{comb} \times Acc$ should be independent of the collision generator. The Λ $\epsilon_{comb} \times Acc$ is plotted in Fig. 4.7c, for the UrQMD data. The ratio of the Λ $\epsilon_{comb} \times Acc$ for DCM (Fig. 4.5f) and UrQMD is shown in Fig. 4.7a (4.7b) for $p_T = [0, 0.3]$ ($p_T = [0.3, 0.6]$) GeV/c. The ratio plots show that the two match within the statistical uncertainties. This shows that the $\epsilon_{comb} \times Acc$ for Λ of the CBM experiment is independent of the collision generator.

Similarly, the ML-based selection criteria application is also independent of the collision generator. In the same manner as ε_{total} for DCM (Fig. 4.6b), the ε_{total} for the UrQMD data is shown in Fig. 4.8c for the threshold of 0.53 on the XGB score. The ratio of the ε_{total} for DCM and UrQMD is shown in Fig. 4.8a (4.8b)

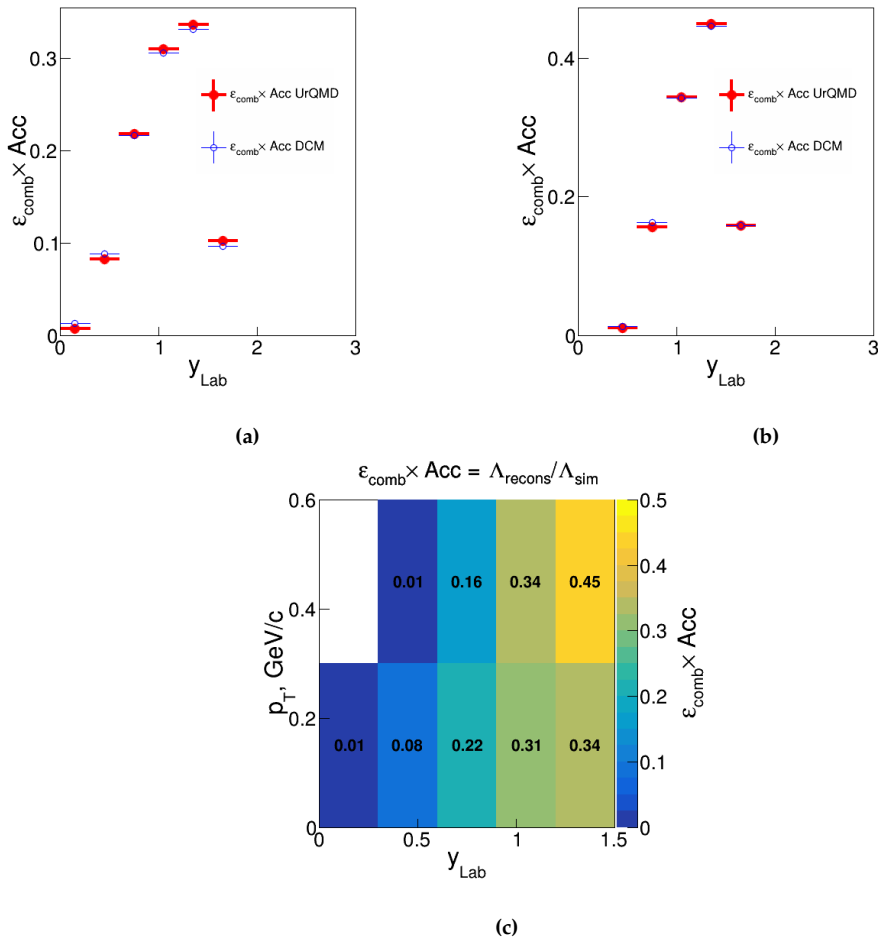


Figure 4.7: Fig. 4.7c shows the $\varepsilon_{comb} \times Acc$ for true Λ generated by the UrQMD. The ratio of the $\varepsilon_{comb} \times Acc$ for true Λ for two collision generators is shown in Fig. 4.7a (4.7b) for $p_T = [0, 0.3]$ ($p_T = [0.3, 0.6]$) GeV/c .

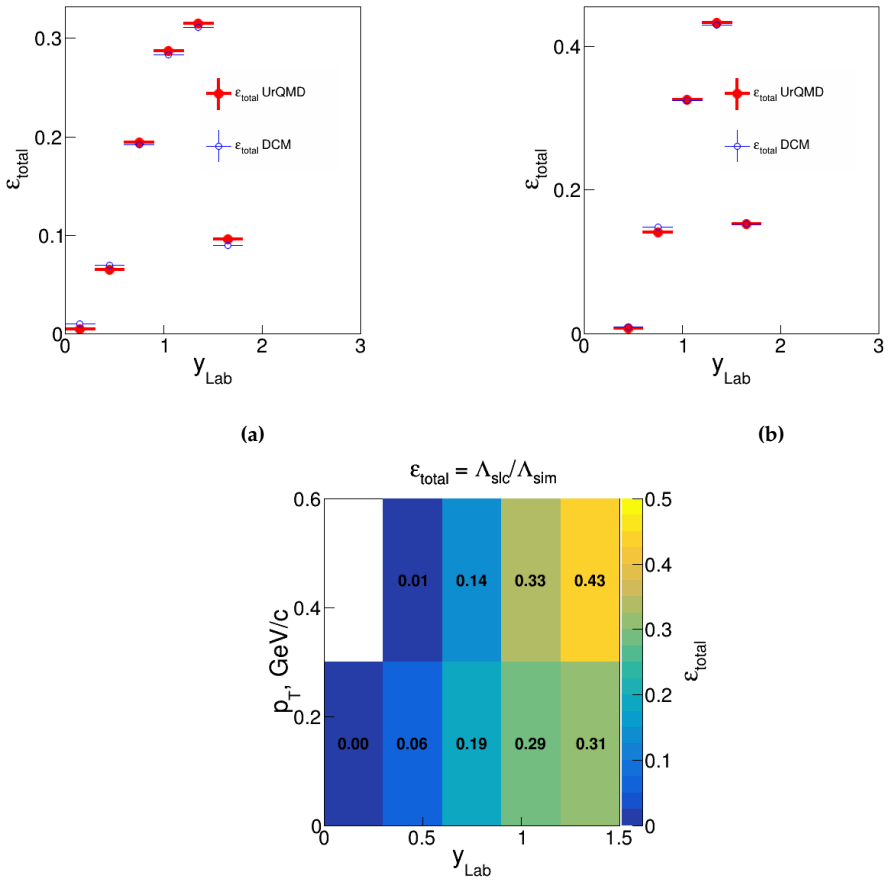
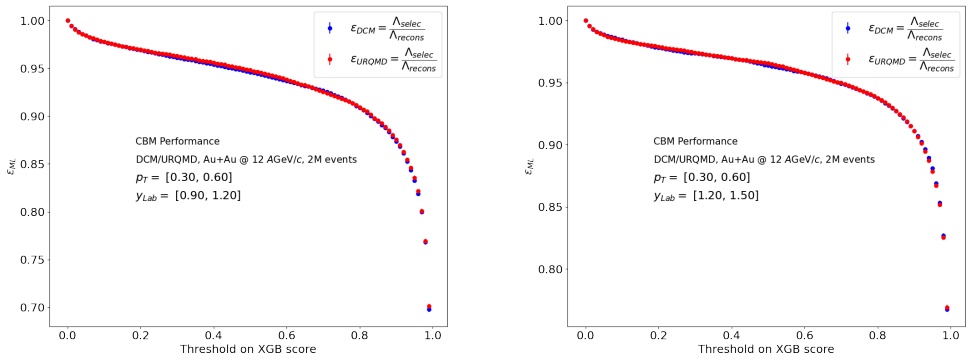


Figure 4.8: Fig. 4.8c shows the ε_{total} for true Λ generated by the UrQMD. The ratio of the ε_{total} for true Λ for two collision generators is shown in Fig. 4.8a (4.8b) for $p_T = [0, 0.3] ((p_T = [0.3, 0.6])$ GeV/c.

for $p_T = [0, 0.3]$ $(p_T = [0.3, 0.6])$ GeV/c. The ratio plots show that the two match within the statistical uncertainties. This shows that the ε_{total} of the selection, and reconstruction is independent of the collision generator.

On top of that, the ML-based selection efficiency will not change if one changes the threshold on the XGB score for two different collision generators. The $p_T = [0.3, 0.6]$ GeV and $y_{Lab} = [1.2, 1.5] ([0.9, 1.2])$ interval has been taken and the ε_{ML} efficiency for UrQMD (red) and DCM (blue) is plotted as a function of XGB score in Fig. 4.9b (4.9a). The ε_{ML} of the two collision generators vary in the same way with the variation of the selection on the XGB score. The two efficiencies overlap within the uncertainties showing that the ML-based selection is independent of the collision generator.



(a) **(b)**

Figure 4.9: The variation of ε_{ML} for UrQMD(red circles) and DCM(blue circles) is shown as a function of XGB score

The independence of the ε_{total} on a collision generator is crucial for the future data taking of the experiment.

4.4 Systematic Uncertainties Evaluation

To calculate the systematic uncertainty on the ML selection, the variation of the selection on the XGBoost (XGB) score is needed. The variation of selection on the XGB score will change the raw yield and therefore the corrected yield. The systematic uncertainty will be the standard deviation of the distribution of the corrected yield for different thresholds on the XGB score. The systematic uncertainty of the fitting procedure (sec. 4.1) is calculated as the difference between the MC yield obtained by counting and the one achieved through the fitting routine. The total systematic error is the sum of the two in quadrature.

4.4.1 Variation of Corrected Yield with XGB Selection

Increasing the threshold of selection on the XGB score reduces combinatorial background but also decreases the number of true signal candidates, referred to as ML efficiency in Fig. 4.10. This is because the separation between the true Λ and the combinatorial background is not perfect. In Fig. 4.10, the efficiency decreases slowly when the threshold on the XGB score is incremented between 0.08 and 0.8 with a linear kind of response ($a + bx$). For higher thresholds, the efficiency decreases exponentially ($ce^{-x/d}$). Therefore, the threshold on the XGB score will be varied between 0.4 and 0.8 to avoid the region of the dramatic fall in efficiency.

The efficiency of machine learning (ML) varies with the threshold on the XGB score, which affects the raw yield achieved through the fitting routine. Fig.4.11 shows the variation of the raw yield with the variation on the threshold on XGB score for the $p_T = [0.3, 0.6]$ GeV/c and $y_{Lab} = [1.2, 1.5]$, interval. The value of the raw yield comes from the integral of the signal function (Sec.4.1) and the error bars are the associated errors on the integral. Fig.4.12 shows the significance ($\frac{\Lambda_{yield}}{\sqrt{\Lambda_{yield}+background}}$) as a function of the XGB score variation. A 3rd-order polynomial ($f_3 = a_0 + a_1 x + a_2 x^2 + a_3 x^3$) has been fit to the significance, a_i represents the i-th fit parameter. The uncertainties of the fit function [98] with covariance

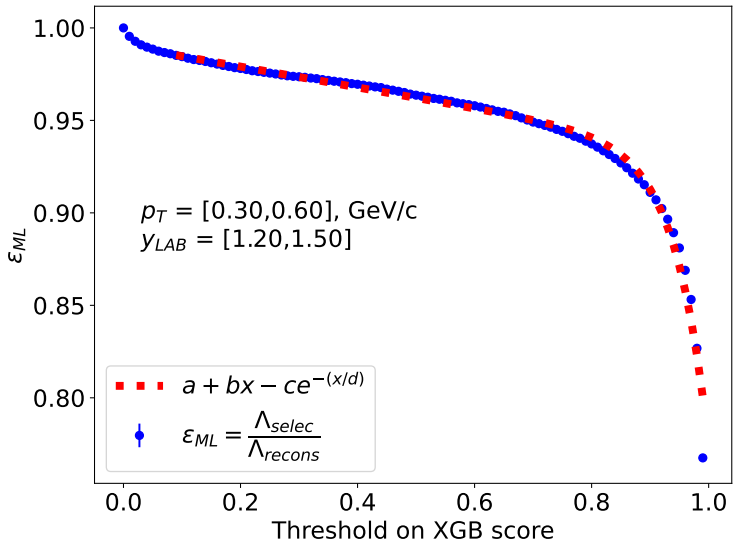


Figure 4.10: The graph shows the ML efficiency variation on changing the selection on XGB probability for two different p_T bins. The efficiency errors are calculated using Binomial statistics (eq. 4.4).

matrix cov are calculated for 1000 uniform steps (x_0) between 0.4 and 0.8 as:

$$\Delta f_3^2\big|_{x=x_0} = \sum_{i=1}^{4}\sum_{j=1}^{4}\left(\frac{\partial f_3}{\partial a_i}\right)\Big|_{x=x_0}\left(\frac{\partial f_3}{\partial a_j}\right)\Big|_{x=x_0} cov(a_i, a_j). \tag{4.8}$$

The maximum of f_3 is found at the XGB score of 0.53, therefore, the default threshold on the XGB score is set to 0.53 ($\varepsilon_{ML\ def} = 0.961$), and the selection is varied between 0.4 ($\varepsilon_{ML\ low} = 0.969 = \varepsilon_{ML\ def} + 0.7\%\varepsilon_{ML\ def}$) and 0.8 ($\varepsilon_{ML\ high} = 0.937 = \varepsilon_{ML\ def} - 2.6\%\varepsilon_{ML\ def}$) with a step size of 0.002. The asymmetry $(0.7, 2.6\%)$ in the efficiency variation cannot be adjusted because on the left side, the efficiency reaches 1 and on the right side it goes up to 0.8, as shown in Fig. 4.10.

The raw yield is efficiency corrected, eq. 4.6, and the corrected yield as a function of the XGB score threshold is shown in Fig. 4.13a with red circles. The standard deviation of this data is considered as the systematic uncertainty of the selection procedure and it is less than 0.2% for the $p_T = [0.3, 0.6]$, $y_{Lab} = [1.2, 1.5]$ and multiplicity $= [200, 400]$ interval. The bars on the red circles are the associated statistical uncertainties calculated from the fit procedure and the efficiency correction procedure. The uncertainties are estimated using eq. 4.7. The blue circle and the bars represent the corrected yield at the default XGB threshold. Similarly, Fig. 4.13b shows the corrected yield at different XGB scores but the yield was calculated through MC counting.

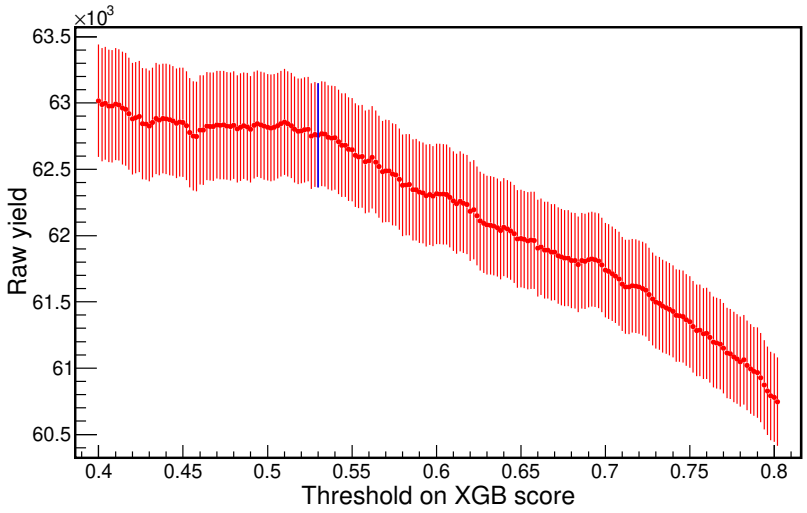


Figure 4.11: The variation of the raw yield as a function of selection on the XGB score. The blue dot with bars shows the raw yield at the default threshold of 0.53 on the XGB score with its errors.

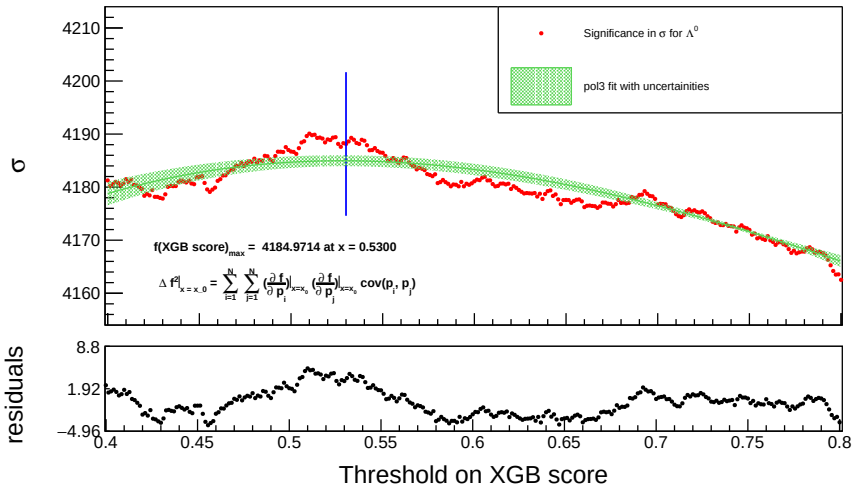
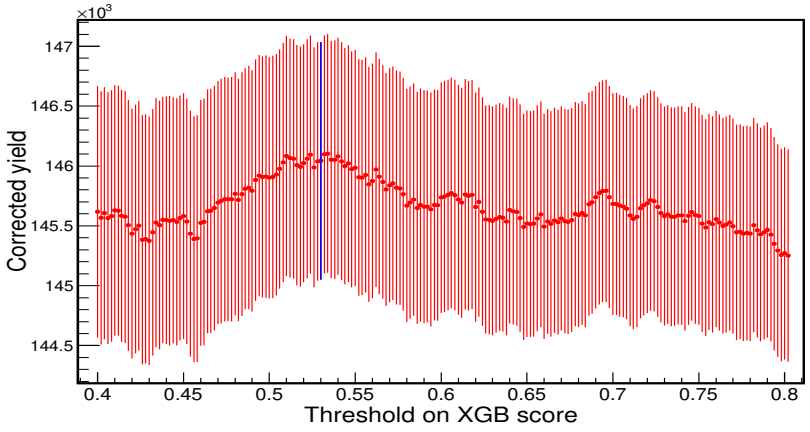
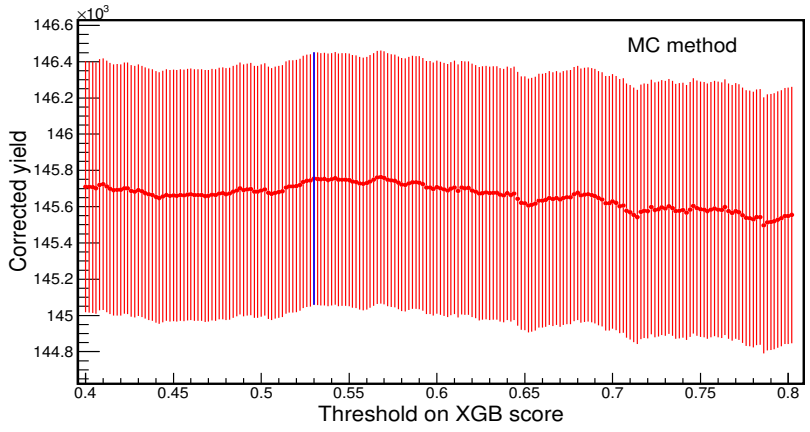


Figure 4.12: The variation of significance (red circles) as a function of the threshold applied on the XGB score for $p_T = [0.3, 0.6]$, GeV/c and $y_{LAB} = [1.2, 1.5]$. A 3rd-order polynomial has been fit to the significance plot and is shown by the green curve and its uncertainties. The lower plot shows the residuals ($data - f_3$).

The black open circles in Fig. 4.14, represent the corrected yield obtained at various XGB scores and it is the projection of Fig. 4.13a. The magenta circle is the corrected yield at the default XGB threshold and its statistical uncertainty is shown by the dotted magenta perpendicular lines. The simulated yield is shown by the cyan triangle and its statistical uncertainty is shown by the cyan lines. The corrected yield at the default XGB threshold matches the simulated yield



(a)



(b)

Figure 4.13: (4.13a) The corrected yield as a function of the selection on the XGB score. The yield is obtained through the integration of the signal fit function. (4.13b) The graph shows the corrected yield, the yield is obtained through MC counting, as a function of the threshold on the XGB score.

(difference $< 0.8\%$) within the statistical uncertainties. The green open squares represent the y-axis projections of Fig.4.13b. The red square is the corrected yield, obtained through MC counting, at the default threshold on the XGB threshold. The difference between the magenta circle and the red square gives the systematic uncertainty estimate for the fitting routine (σ_{fit}) and it is less than 0.2% for the $p_T = [0.3, 0.6]$, $y_{Lab} = [1.2, 1.5]$ and multiplicity $= [200, 400]$ interval. The standard deviation of the black open circle estimates the systematic uncertainty (σ_{XGB}) of the corrected yield as a variation of the XGB threshold. The total systematic uncertainty is calculated as:

$$\sigma = \sqrt{\left(\frac{\sigma_{fit}}{\Lambda_{corr\ yield\ def}}\right)^2 + \left(\frac{\sigma_{XGB}}{\Lambda_{corr\ yield\ def}}\right)^2} \tag{4.9}$$

with $\Lambda_{corr\ yield\ def}$ representing the corrected yield at the default threshold on the XGB score. The total systematic is less than 0.3% for this particular interval.

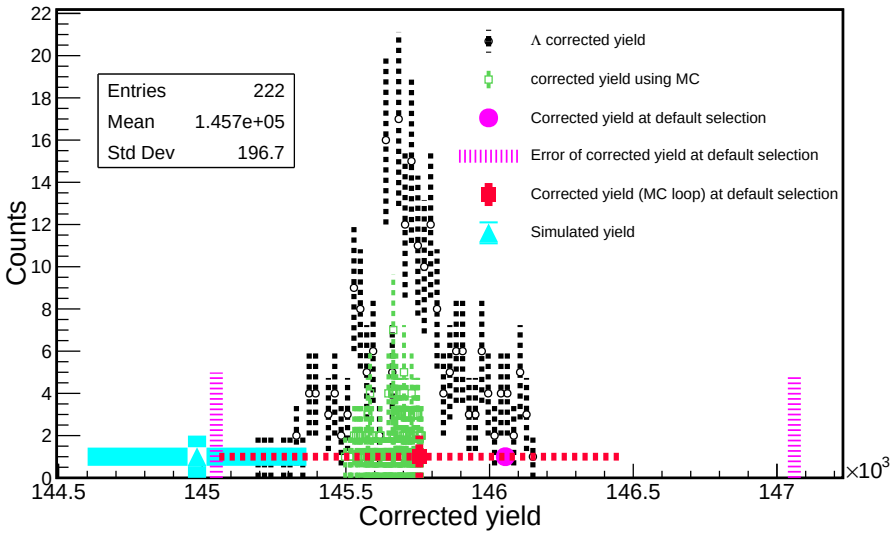


Figure 4.14: The corrected yield (black circles) for different XGB scores. The magenta circle shows the corrected yield at the default selection while the perpendicular magenta lines represent its uncertainty. The cyan triangle shows the simulated yield. The green histogram is the corrected yield obtained through MC counting of the signal. The red square is the corrected yield at the default XGB score and the yield is obtained through MC counting.

The efficiency calculation on a single generator (model A) and then correcting the yield of the same model (model A) on its efficiency is also checked. The ML model is applied to Λ candidates generated by 1M UrQMD events and the yield

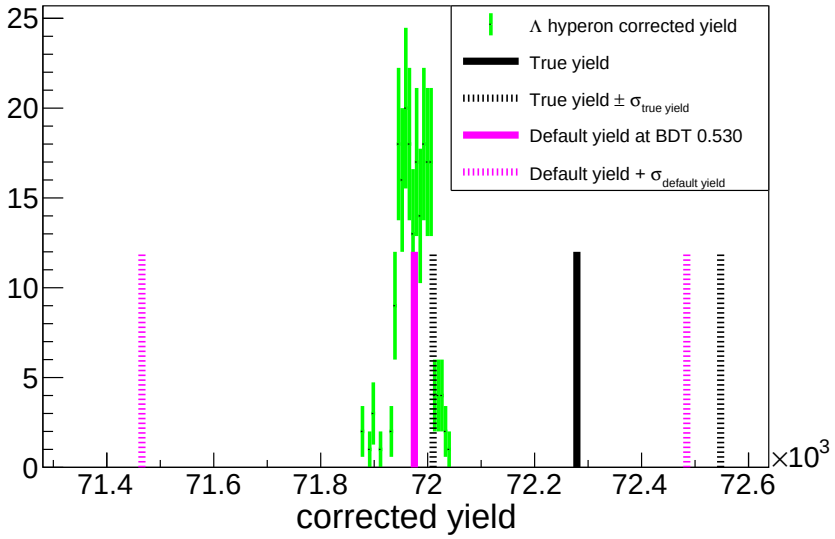


Figure 4.15: The corrected yield for various XGB scores. The yield is obtained from 1M events while the efficiency is calculated on another 1M events, generated with the same collision generator.

is calculated by summing all the MC Λs, for one selection on the XGB score. The efficiency is calculated on another 1M data of UrQMD on that same XGB score. The Λ yield from the first 1M is corrected by the efficiency obtained from the 2nd 1M events. The process is repeated for multiple selections on XGB score as was performed previously and the efficiency corrected yield is shown in Fig. 4.15 The difference between the corrected yield at the default XGB score of 0.53 (magenta line) and the true simulated yield (black line) is less than 0.5% and both match within the statistical errors.

The selection criteria optimization is repeated for the intervals: $p_T = [0.6, 3] - y_{Lab} = [0, 1.6]$, $p_T = [0, 0.6] - y_{Lab} = [1.6, 3]$, and $p_T = [0.6, 3] - y_{Lab} = [1.6, 3]$. The p_T is in the units of GeV/c. The multiplicity interval was the same as in the previous analysis, i.e., multiplicity of charged tracks $= [200, 400]$. The multi-differential $p_T - y_{Lab}$ yield extraction and the efficiency correction procedure were performed for the above three intervals. Plots of the invariant mass distribution along with the fitting curves have been plotted in the appendix section A.4. The plots to extract the systematic uncertainty for a few intervals are shown in sec. A.5. The performance in the very low populated p_T and low y_{Lab} regions has not been shown in this work because of the availability of simulated data. The

CBM common production only has 2×10^6 UrQMD events available and this is not enough to populate these bins with high statistics.

4.5 Transverse Momentum-Rapidity Spectra

The Λ hyperons generated by UrQMD and DCM were transported through the CBM Geant4 setup and reconstructed using the CBM reconstruction chain, for the multiplicity of charged tracks $= [200, 400]$. The reconstructed candidates of both models were subjected to ML-based SC. The DCM-generated data was used to obtain the efficiency, which was used to correct the UrQMD raw yield. The reconstructed and corrected spectra (red circles) and the MC spectra (black squares) of the UrQMD data have been plotted in Fig. 4.16. The function:

$$Fit\ func(A, p_T) = \frac{A p_T}{T^2 + m_\Lambda T} exp\left(-\frac{\sqrt{p_T^2 + m_\Lambda^2} - m_\Lambda}{T}\right) \tag{4.10}$$

is fitted to the reconstructed spectra with A as and T as free parameters and $mass_\Lambda = 1.115\,683\,\mathrm{GeV/c^2}$. This function comes from a thermal ansatz and the derivation is discussed in the appendix of the work in [99]. The obtained values of T (A) for the $[0.9, 1.2]$,$[1.2 - 1.5]$, $[1.5 - 1.8]$, and $[1.8, 2.1]$ y_{Lab} intervals are 0.205 ± 0.015 ($1.17 \times 10^5 \pm 8 \times 10^3$), 0.2145 ± 0.016 ($1.4 \times 10^5 \pm 10^4$), 0.219 ± 0.016 ($1.47 \times 10^5 \pm 1.1 \times 10^4$), and 0.211 ± 0.016 ($1.38 \times 10^5 \pm 10^4$). These intervals of rapidity contain the mid-rapidity region, i.e., $y/2 = 0.25(\log(\frac{E+p_{beam}}{E-p_{beam}})) = 1.62$ where beam momentum is $p_{beam} = 12A\mathrm{GeV/c}$, energy is $E = \sqrt{mass_{p^+}^2 + p_{beam}^2}\mathrm{GeV}$, and $mass_{p^+} = 0.938\,\mathrm{GeV/c^2}$. The spectra show that CBM has p_T and rapidity coverage over midrapidity.

The statistical uncertainties for the corrected (simulated) spectra are shown by the red lines (black dotted lines) while the systematic uncertainties are shown with the blue shaded area, in Fig. 4.16. For better visualization of the uncertainties, one p_T interval is zoomed in. The systematic uncertainity is less than 6%, 2%, 3%, and 3% for the y_{Lab} intervals $[0.9, 1.2]$, $[1.2, 1.5]$, $[1.5, 1.8]$, and $[1.8, 2.1]$, respectively. The lower statistics in the high p_T intervals might be contributing to the high systematics and can be improved with higher statistics. The corrected spectra match the simulated spectra within the statistical and systematic uncertainties.

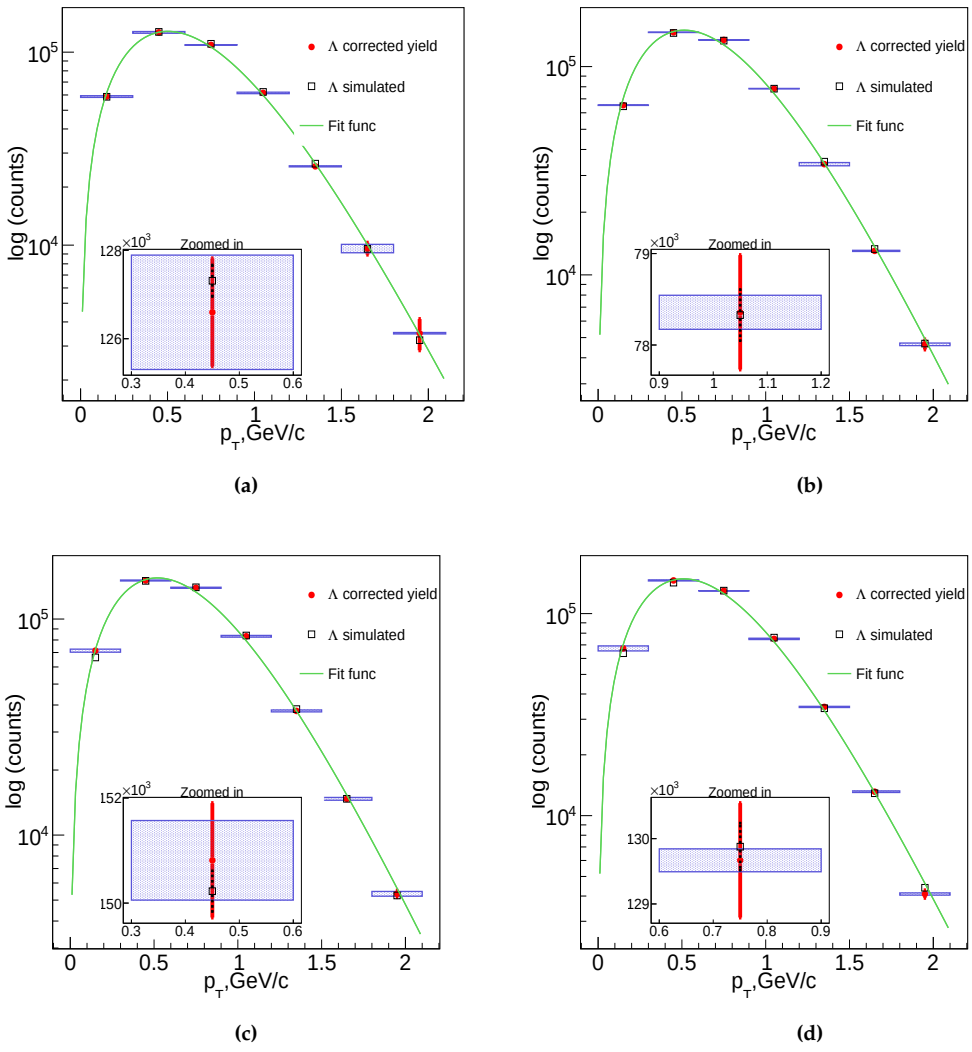


Figure 4.16: The graphs 4.16a, 4.16b, 4.16c, and 4.16d show the corrected p_T spectra (red circles) for the y_{Lab} intervals of $[0.9, 1.2]$,$[1.2 - 1.5]$, $[1.5 - 1.8]$, and $[1.8, 2.1]$. The MC true simulated spectra, before reconstruction and selection, are shown with black unfilled rectangles. The systematic uncertainties are shown with a filled blue area.

Chapter 5

Summary and Outlook

The multi-differential yield of (multi-)strange hyperons can reveal information about the matter created during heavy-ions collision. The rare production of hyperons requires optimized selection criteria to suppress the combinatorial background. Since Λ is the most abundantly produced hyperon at FAIR energies, it is useful for the systematic uncertainty evaluation of the selection process. Achieving a large signal-to-background ratio with high efficiency requires optimizing selection criteria multi-differentially in centrality, transverse momentum, and rapidity for each collision energy. Manual optimization can be performed by adjusting the selection criteria to maximize the signal-to-background ratio. However, performing this optimization non-linearly in a multi-dimensional space across different intervals becomes a laborious task. ML algorithms can optimize selection criteria non-linearly in an automatized way for every interval. The optimization of selection criteria for the Λ hyperon has been investigated using ML algorithms such as tree-based eXtreme Gradient Boosting (XGBoost). The ML-based selection criteria delivered at least seven times better signal-to-background ratio at a higher efficiency than the manually optimized selection criteria. For some $p_T - y_{Lab}$ interval (sec. A.3), the signal-to-background ratio was more than forty times better. The CBM experiment is currently not yet operational, so the goal of this work was to develop methods in preparation for the future.

Two collision generators were employed, DCM-QGSM-SMM for simulation and UrQMD for data. This approach is useful to check for biases in the procedure. In the future, the UrQMD data will be replaced by data from the experiment

but the developed framework will remain the same. Monte Carlo (MC) signal was selected from the Λ peak region of the simulation, while three times more background candidates were selected from data outside the Λ peak region. To develop and assess a machine learning (ML) model, training and testing datasets were created. The model was trained on the training dataset and evaluated on the testing dataset to ensure that it did not over-fit. After the model demonstrated good performance on the testing dataset, it was applied to analyze 2×10^6 events from both simulation and data.

After applying ML-based selection criteria to simulation and data events, the raw yield of the hyperon was extracted using a multi-differential fitting routine. To ensure stability, the routine was performed in multiple steps. The signal is fitted using a double-sided crystal ball function (DSCB), followed by a polynomial fit for the background. Finally, the entire invariant mass distribution of the data is fit with a sum of DSCB and a polynomial, while initializing the parameters from the last two steps. The yield was obtained as the integral of the DSCB in the ± 11 sigma region around the peak. To correct the yield for efficiency and acceptance, an efficiency $\times Acc$ factor obtained from the simulation was applied. The systematic uncertainties for the selection procedure and the fitting routine were separately calculated and the two were added in quadrature. The total systematic uncertainty is generally below 3% but in the high p_T intervals it can go up to 6%.

It is worth mentioning that the removal of secondary Λ contamination was not performed.

To optimize the selection criteria, this study focused on one multiplicity interval of charged particles, specifically $[200, 400]$, for beam momentum of 12 A GeV/c of the CBM experiment. The interval was divided into four $p_T - y_{Lab}$ intervals and separate optimizations were conducted for each interval. This study did not explore other multiplicity intervals or energies, leaving these as areas for future investigation. Also, the fitting procedure needs to be applied to the other multiplicities of the data but its performance has been tested on high and low p_T intervals and it seems pretty robust.

The inclusion of other variables from other sub-systems of the CBM detector will be tried in the future to better select the p^+ and π^-. The information from

the other detectors for example RICH and TRD can help to suppress the contamination from electrons for the π^-. Also, an improvement in the TOF matching algorithm can help reduce the background further.

Appendix A

To not break the flow of the discussion of the analysis some of the figures and discussions have been avoided in the last two chapters and they are provided here. SHAP plots of signal-only data with a high threshold applied to the XGB score are discussed here. Also a discussion about the removal of secondary Λ from this analysis is provided. The details of the fitting routine and some plots for other $p_T - y_{Lab}$ intervals were not discussed before and they are shown here. The systematic uncertainty calculation plots of the remaining intervals and their spectra is also added here.

A.1 Primary and secondary Λs Separation

There are Λs that are produced directly in the collision and therefore are called primary Λs. These Λs tell us about the nature of the deconfined matter created by heavy-ions collisions. However, these collisions also produce other particles which decay into Λs. Similarly, there are Λs that are produced due to the interaction of particles with the detector material. All other types of Λ except primaries are called secondaries. In order to study the matter created in the collision, the two Λ types need to be separated from each other. In this section, it is illustrated that the full separation is not possible but the full description is not put forward.

To separate primary Λs from secondaries and from the background, a multi-class classifier was applied to both train and test data. The number of classes available is three and the algorithm returns a probability distribution for each class. Most of the variables available associated with the Λ decay topology were used to separate the different classes. Fig. A.1 shows the performance of the model for the primary Λs on the train-test data. The background (blue) distribution peak is away from the overlapping peaks of the primary (red) and secondary (green) distributions. The model can separate the background from the two distributions of primaries and secondaries, but it cannot separate the secondaries from the primaries.

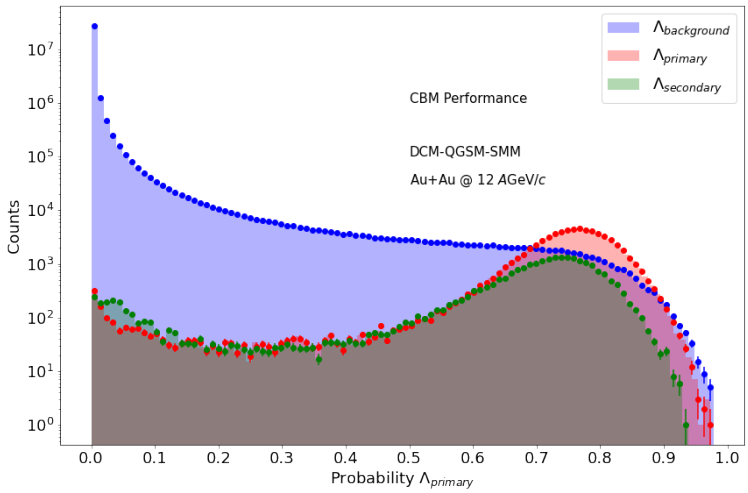


Figure A.1: The distribution of primary and secondary Λ along with the background distribution for the train and test data. The shaded area histograms are for the train data and the circles represent the test data.

A.2 SHAP Explanation

This section is a continuation of the discussion in section 3.6. The SHAP score for signal-only training data with an XGB score above 0.99 is plotted in Fig. A.2 to see the performance of the model. These signals show that the most useful variables for the ML classifier are the distances of the p^+ and π^- tracks to the primary vertex. But the $\chi^2_{prim\ p^+}$ values go up to a SHAP score of nearly 4 while the $\chi^2_{prim\ \pi^-}$ SHAP score ends below 3. Fig. A.3a shows that for signal with XGB score above 0.99 higher SHAP score is given to higher values of the $\chi^2_{prim\ p^+}$ variable. Also, a higher SHAP score is given to higher values for the $\chi^2_{prim\ \pi^-}$ variable, shown in Fig. A.3b. The distance between the two daughter tracks is found to be the 3rd best variable according to the SHAP score. Fig. A.3d shows that the lower values of the DCA are given a higher SHAP score and since χ^2_{geo} is the same variable but normalized so the same behavior is observed for it, as shown in Fig. A.3e. But in terms of ranking, it is the 5th best variable according to the SHAP score.

The mass of the p^+ is the 4th useful variable according to SHAP calculation and Fig. A.3f shows the SHAP values plotted for this variable. Fig. A.3f reveals that though these are all true p^+ they have been matched with incorrect hits in the TOF wall. Most of these protons have a mass near the mass of the proton, $m_{p^+} = 0.938\,27\,\mathrm{GeV/c^2}$ [33] and the algorithm has given those protons all positive SHAP scores. Improving this matching from the TOF team will help also this analysis, as the p^+ assigned with lower mass values have been given negative SHAP scores by the ML model. The 6th best variable found by the model is normalized to its error distance between the PV and SV. Fig. A.3c shows the SHAP values for $L/\Delta L$ and they peak between 5 and 15.

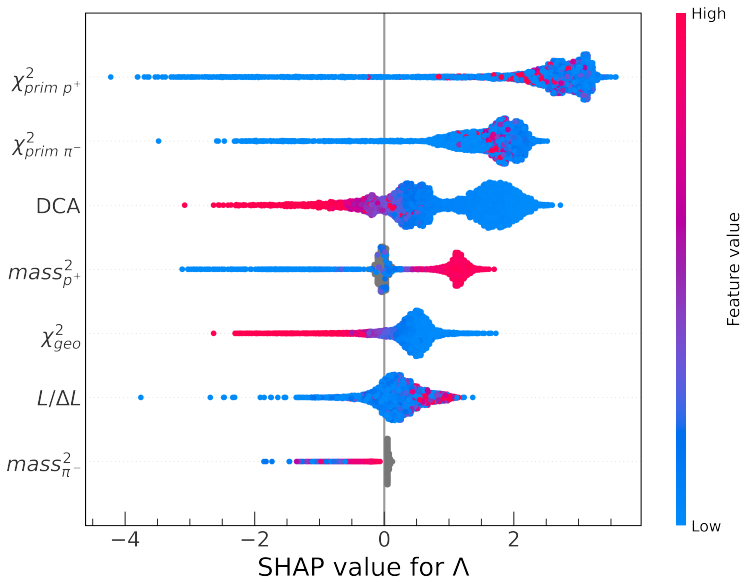


Figure A.2: SHAP for signal only with XGB score above 0.99

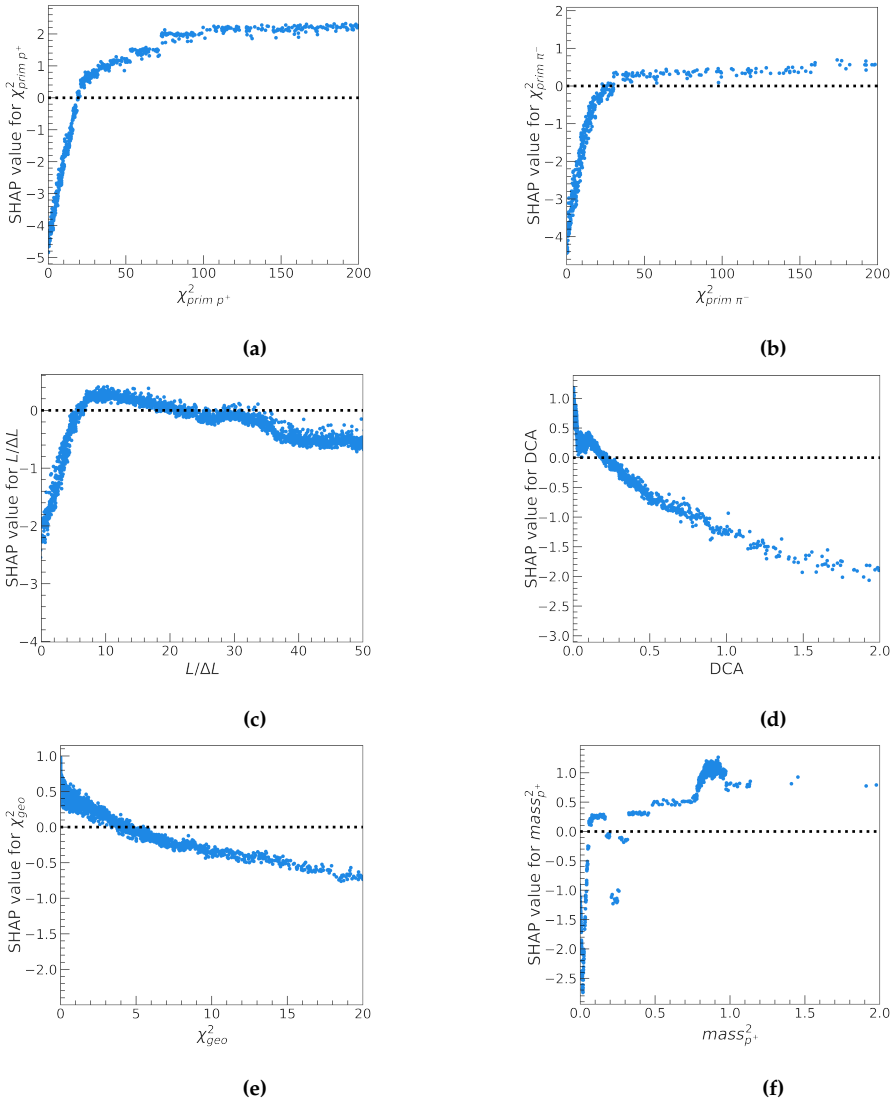


Figure A.3: The SHAP score is plotted on the y-axis for the values of the variables used for Λ segregation from the background.

A.3 Comparison of ML to Manual SC

The comparison between the hypercube selection criteria optimized manually and selection criteria optimized through XGBoost had been discussed in sec. 3.5. The other intervals of $p_T - y_{Lab}$ are compared in Fig. A.4. Thresholds on the XGB scores that have similar efficiency to that of manually optimized hypercube selection criteria are applied. ML-optimized selection criteria show slightly better efficiency and very good background suppression than the manually optimized SC.

In the mass window of $1.1 - 1.13\,\mathrm{GeV/c^2}$, the true signal and background candidates were counted for both the data, i.e., after the application of ML and manual SC. The signal-to-background ratio for the ML selection criteria method were 366, 179, and 373 while for the manually optimized hypercube method, they were 15, 4.3, and 13 for the intervals $p_T(\mathrm{GeV}/c) - y_{Lab} = [0.6, 3] - [0, 1.6], [0, 0.6] - [1.6, 3]$, and $[0.6, 3] - [1.6, 3]$, respectively. This means that the signal-to-background ratio in the case of ML selection criteria is at least 24 times more than in the manually optimized selection criteria case.

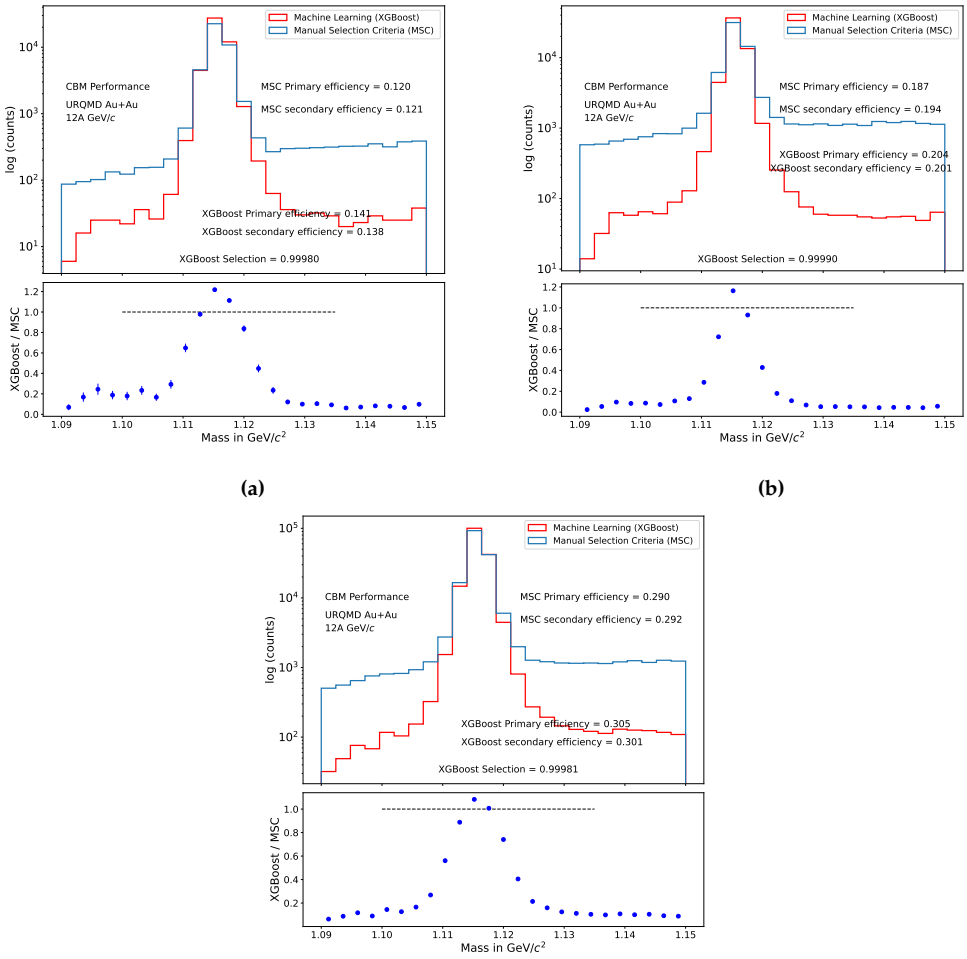
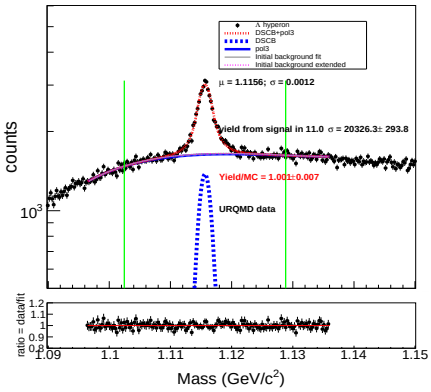


Figure A.4: The Figs. A.4a, Fig. A.4b, Fig. A.4c show the comparison of manually and ML optimized selection criteria for the intervals $p_T = [0.6, 3] - y_{Lab} = [0, 1.6]$, $p_T = [0, 0.6] - y_{Lab} = [1.6, 3]$, and $p_T = [0.6, 3] - y_{Lab} = [1.6, 3]$. The p_T is in GeV/c.

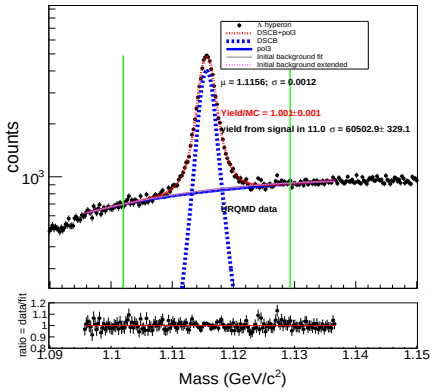
A.4 Fitting Routine

The invariant mass histograms of the various intervals shown in sec. 4.5 are plotted here. The multi-step fitting routine mentioned in section 4.1 is utilized here.

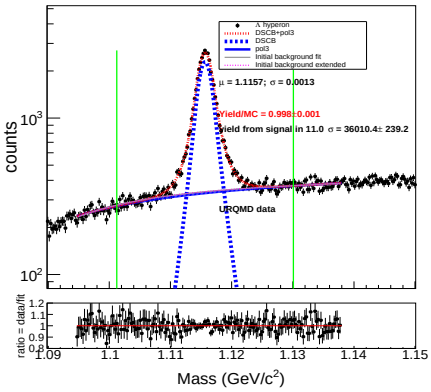
The first step of the fitting routine starts with the fitting of the signal only part of the simulation with a DSCB function. The parameters are initialized (bounded) as $\mu = 1.11567(1.113, 1.119)$, $\sigma = 0.0012$ $(0.0012 \times (1-4), 0.0012 \times (1+0.0012))$, $a_{L/R} = 1(0, 10)$, and $n_{L/R} = 1(0, 100)$.



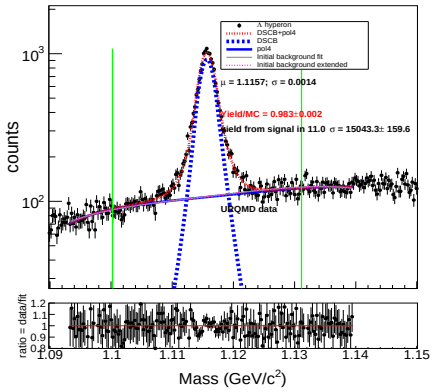
(a)



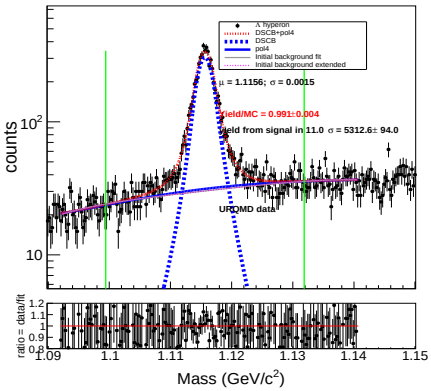
(b)



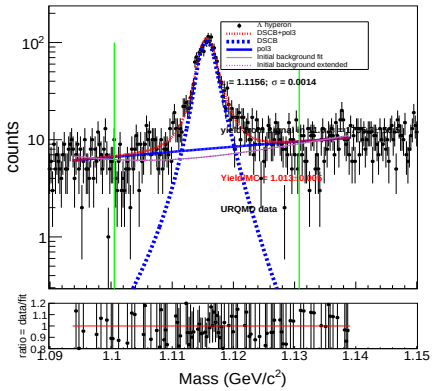
(c)



(d)

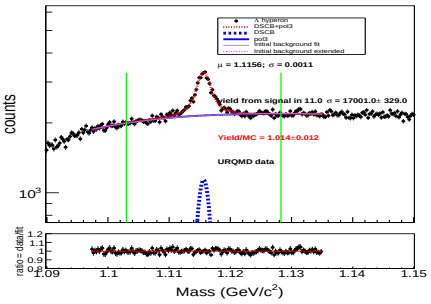


(e)

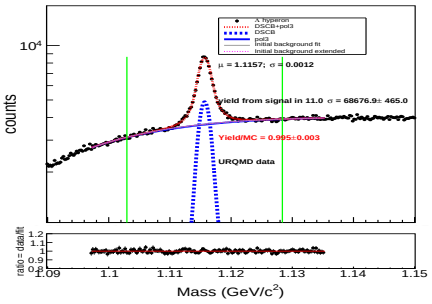


(f)

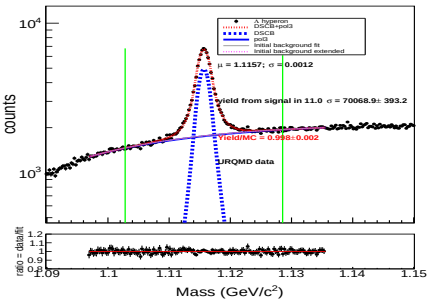
Figure A.5: The Figures A.5a, A.5b, A.5c, A.5d, A.5e, and A.5f show the invariant mass histogram of the $y_{Lab} = [1.2, 1.5]$ interval and different intervals of p_T i.e., $[0, 0.3]$, $[0.6, 0.9]$, $[0.9, 1.2]$, $[1.2, 1.5]$, $[1.5, 1.8]$, and $[1.8, 2.1]$. The p_T is in the units of GeV/c.



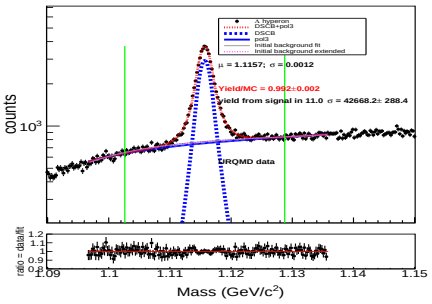
(a)



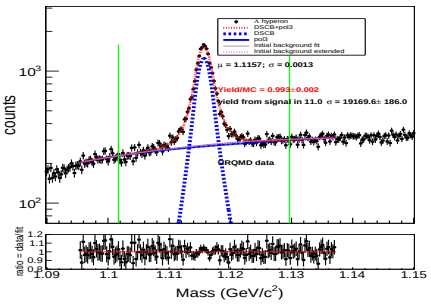
(b)



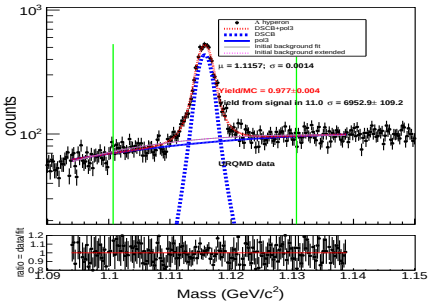
(c)



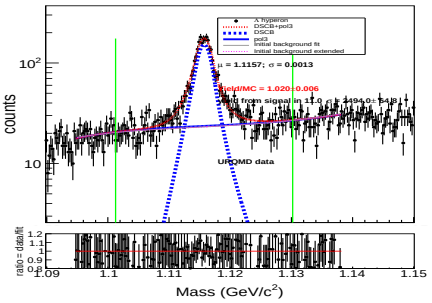
(d)



(e)



(f)

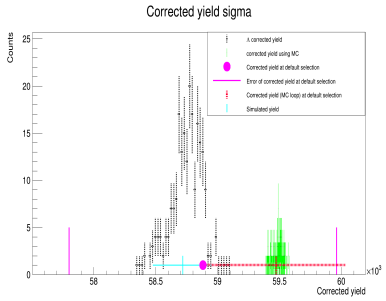


(g)

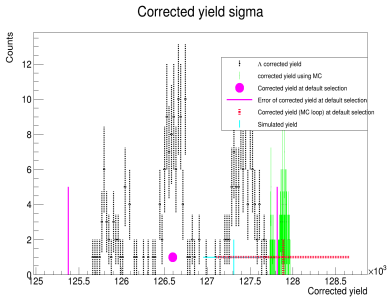
Figure A.6: The Figures A.6a, A.6b, A.6c, A.6d, A.6e, A.6f, and A.6g show the invariant mass histogram for the $y_{Lab} = [1.5, 1.8]$ interval and p_T intervals of $[0, 0.3]$, $[0.3, 0.6]$, $[0.6, 0.9]$, $[0.9, 1.2]$, $[1.2, 1.5]$, $[1.5, 1.8]$, $[1.8, 2.1]$. The p_T is in GeV/c units.

A.5 Additional Plots for Systematic Uncertainity

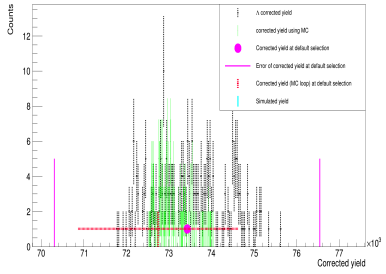
The procedure discussed in sec. 4.4 has been used here to calculate the systematic uncertainties for a few transverse momentum and rapidity intervals. The two sources of the systematic uncertainity evaluated here are the selection procedure and the fitting routine.



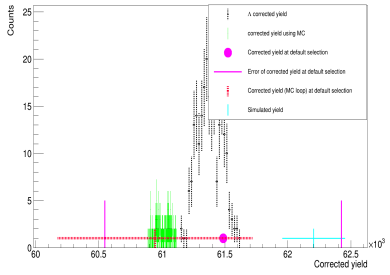
(a)



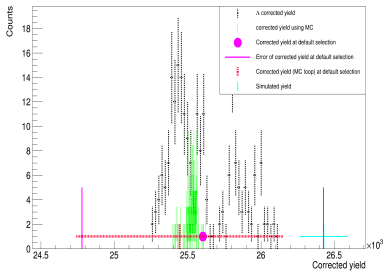
(b)



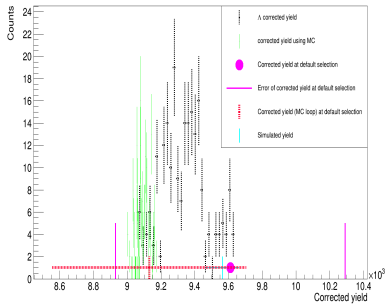
(c)



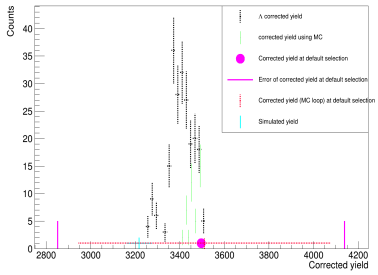
(d)



(e)



(f)



(g)

Figure A.7: The Figures A.7a, A.7b, A.7d, A.7e, A.7f, and A.7g show the invariant mass histogram for the $y_{Lab} = [0.9, 1.2]$ interval and p_T intervals of $[0, 0.3]$, $[0.3, 0.6]$, $[0.6, 0.9]$, $[0.9, 1.2]$, $[1.2, 1.5]$, $[1.5, 1.8]$, $[1.8, 2.1]$. The p_T is in GeV/c units.

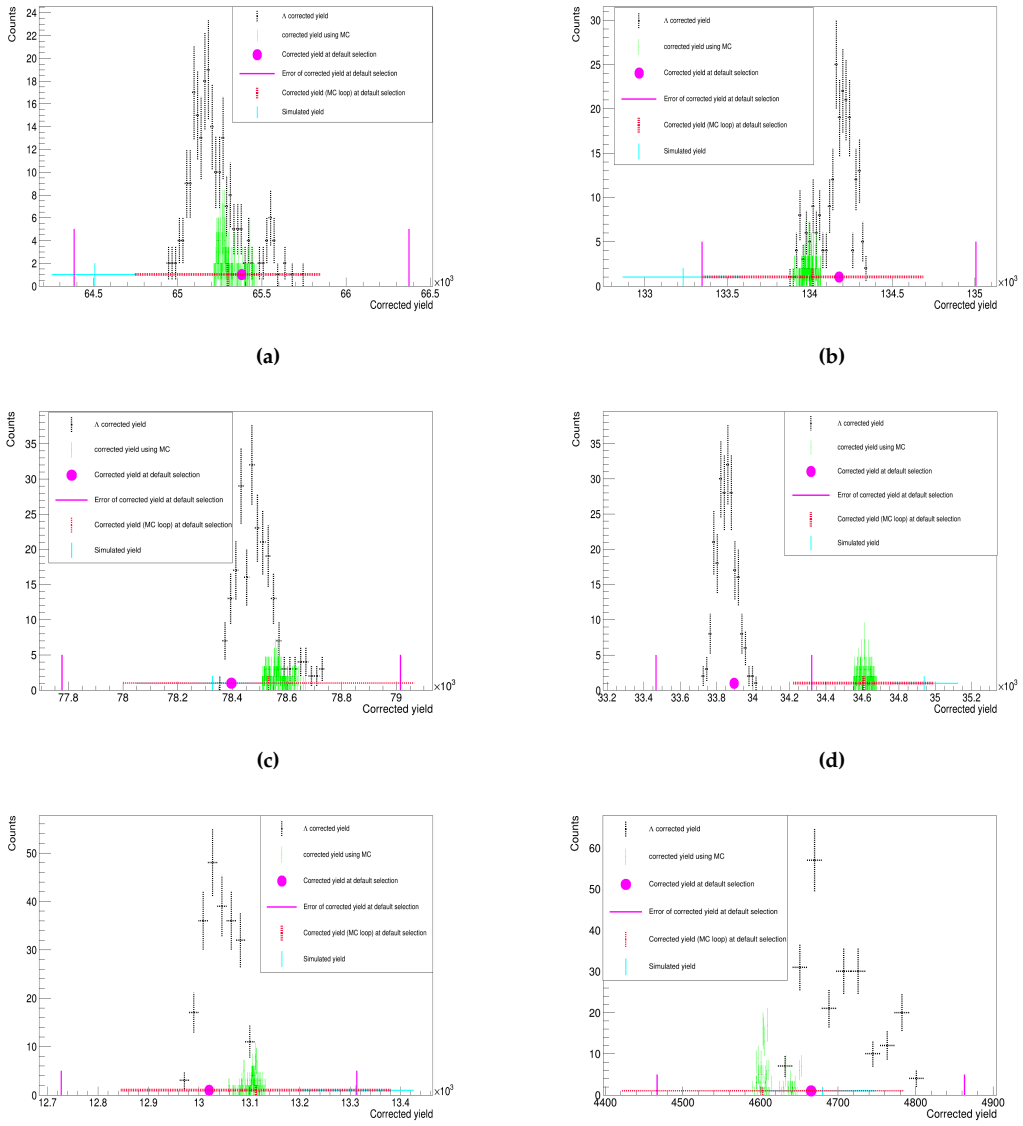


Figure A.8: The Figures A.8a, A.8b, A.7c ,A.8c, A.8d, A.8e, and A.8f show the invariant mass histogram for the $y_{Lab} = [1.2, 1.5]$ interval and p_T intervals of $[0, 0.3]$, $[0.6, 0.9]$, $[0.9, 1.2]$, $[1.2, 1.5]$, $[1.5, 1.8]$, and $[1.8, 2.1]$. The p_T is in GeV/c units.

Contributions to CBM

In this work, if a plot is borrowed from someone else's work, then the credit has been given to the creator of the image in the caption. The results obtained in chapters 3, 4, and appendix A are part of this work and have not been borrowed from any other work. The details about the packages used in this work are mentioned below.

The Au-Au collisions simulation through the UrQMD and DCM collisions generator and then transporting it through the CBM setup inside Geant4 was made available by the CBM collaboration's common production team. The author of this work is not involved in those productions. To reconstruct Λ candidates from the data the PFSimple package [71] was used and again the author is not involved in its development. The machine learning implementation for the CBM experiment code was developed by the author. This means the hyperparameters search, training, and testing of the ML algorithm and the explanation through the SHAP library. The treelite conversion code was written in collaboration with Viktor Klochkov.

For the comparison of the ML and the manually optimized SC, the plots were generated by the author. However, the PID selection criteria optimization, for the π^- and p^+, and the manual selection criteria optimization, for Λ, are part of this work only. Only once the manually optimized selection criteria are used in sec. 3.5 for the comparison with ML-based SC.

For the yield extraction (sec: 4.1), the ALICE HF inv. mass fitting routine class [100] was used. This class was not in use by the CBM collaboration so the necessary parts of the code were applied to the CBM data by the author. The author modified the class by introducing the DSCB function. The class was used to fit the data by taking the advantage of the available signal and background

approximation functions. The visualization plots of the inv. mass and the fits are created for this work only and are not a duplication of anyone else's work. For the efficiency calculation (sec:4.2) the AnalysistreeQA package was used to extract the simulated yield and the author has not developed the package in any way. After that, the efficiency calculation and correction were done by the author.

The code for the systematic uncertainties is the author's contribution and so is the code for Λ spectra and their fitting.

Acknowledgments

This work would not have been possible without the constant support of my supervisor Prof. Dr. Hans Rudolf Schmidt and mentors Dr. Ilya Selyuzhenkov and Dr. Andrea Dubla. I will always be indebted to them for providing me with this opportunity to work with them and learn from them.

My colleagues-cum-friends Dr. Viktor Klochkov and Oleksii Lubynets have played pivotal roles in helping me finish this work. They helped me use the packages that they have developed for the CBM experiment. They have also been kind enough to look at my C++ code and find out the bugs in them. No words can help me express how grateful I am to have them as friends.

I am very thankful to all my friends and family members for their continuous support and help. My friend Khushi cooked for me when I was busy writing my master's thesis and she has done the same when I was writing my Ph.D. thesis. She has been very kind and generous in providing any help that I ever wanted. My friends Shayan, Ali, Bilal, Chiragh, Shantanu, and Sibgha have also cheered me up when I needed it, apart from making spicy cuisine for me. Other friends like Taner, Stephen, Cathia, and Davide have assured me that I am competent enough to finish this work.

My colleagues from the Physikalisches Institut: Dr. Christian Strandhagen Kshitij A. have taken out time of their busy schedules to read my thesis and have offered very insightful comments. Iarsolav has been crucial for me to understand the detectors of the STS and Elena has helped me learn the basics of ROOT. Torsten, Lukas, and others have offered help when I needed it.

References

[1] Michael E Peskin and Daniel V Schroeder. *An introduction to quantum field theory Addison*. 1995.

[2] G Zweig. "CERN preprint TH-401 (1964); H. Fritzsch and M. Gell-Mann, eConf C720906V2 (1972) 135; DJ Gross and F. Wilczek". In: *Phys. Rev. Lett* 30 (1973), p. 1343.

[3] H David Politzer. "Reliable perturbative results for strong interactions?" In: *Physical Review Letters* 30.26 (1973), p. 1346.

[4] Adam Bzdak et al. "Mapping the phases of quantum chromodynamics with beam energy scan". In: *Physics Reports* 853 (2020), pp. 1–87.

[5] Yasumichi Aoki et al. "The order of the quantum chromodynamics transition predicted by the standard model of particle physics". In: *Nature* 443.7112 (2006), pp. 675–678.

[6] Fei Gao and Jan M Pawlowski. "Chiral phase structure and critical end point in QCD". In: *Physics Letters B* 820 (2021), p. 136584.

[7] Philippe de Forcrand. "Simulating QCD at finite density, PoS". In: *LAT2009* 10 (2009), p. 41.

[8] John Ellis. "From little bangs to the big bang". In: *Journal of Physics: Conference Series*. Vol. 50. 1. IOP Publishing. 2006, p. 8.

[9] Francois Gelis et al. "The color glass condensate". In: *Annual Review of Nuclear and Particle Science* 60 (2010), pp. 463–489.

[10] Tapan K Nayak. "Heavy ions: results from the Large Hadron Collider". In: *Pramana* 79.4 (2012), pp. 719–735.

[11] *MADAI collaboration*. URL: https://madai.phy.duke.edu/indexc151.html?page_id=452.

[12] Helmut Satz. "The Quark-Gluon Plasma". In: *arXiv preprint arXiv:1101.3937* (2011).

[13] T Ablyazimov et al. "Challenges in QCD matter physics: The scientific programme of the Compressed Baryonic Matter experiment at FAIR". In: *The European Physical Journal A* 53.3 (2017), pp. 1–14.

[14] Jean Cleymans and Krzysztof Redlich. "Chemical and thermal freeze-out parameters from 1 A to 2 0 0 A GeV". In: *Physical Review C* 60.5 (1999), p. 054908.

[15] Anton Andronic et al. "Decoding the phase structure of QCD via particle production at high energy". In: *Nature* 561.7723 (2018), pp. 321–330.

[16] Josef Sollfrank et al. "Hydrodynamical description of 200A GeV/c S+ Au collisions: hadron and electromagnetic spectra". In: *Physical Review C* 55.1 (1997), p. 392.

[17] Steffen A Bass et al. "Microscopic models for ultrarelativistic heavy ion collisions". In: *Progress in Particle and Nuclear Physics* 41 (1998), pp. 255–369.

[18] *SIS18*. URL: https://www.gsi.de/en/work/project_management_fair/sis100sis18_sis/heavy_ion_synchrotron_sis18.

[19] Lyndon Evans and Philip Bryant. "LHC machine". In: *Journal of instrumentation* 3.08 (2008), S08001.

[20] Tom Reichert et al. "Comparison of heavy ion transport simulations: Ag+ Ag collisions at E lab= 1.58 A GeV". In: *Journal of Physics G: Nuclear and Particle Physics* 49.5 (2022), p. 055108.

[21] AS Botvina et al. "Production of spectator hypermatter in relativistic heavy-ion collisions". In: *Physical Review C* 84.6 (2011), p. 064904.

[22] *CERN SPS*. URL: https://home.cern/science/accelerators/super-proton-synchrotron.

[23] M Harrison, T Ludlam, and S Ozaki. "RHIC project overview". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 499.2-3 (2003), pp. 235–244.

[24] *GSI*. URL: `https://www.gsi.de/`.

[25] Ulrich Heinz and Maurice Jacob. "Evidence for a new state of matter: An assessment of the results from the CERN lead beam programme". In: *arXiv preprint nucl-th/0002042* (2000).

[26] John Adams et al. "Experimental and theoretical challenges in the search for the quark–gluon plasma: The STAR Collaboration's critical assessment of the evidence from RHIC collisions". In: *Nuclear Physics A* 757.1-2 (2005), pp. 102–183.

[27] K Adcox et al. "Formation of dense partonic matter in relativistic nucleus–nucleus collisions at RHIC: experimental evaluation by the PHENIX collaboration". In: *Nuclear Physics A* 757.1-2 (2005), pp. 184–283.

[28] Miklos Gyulassy and Larry McLerran. "New forms of QCD matter discovered at RHIC". In: *Nuclear Physics A* 750.1 (2005), pp. 30–63.

[29] Berndt Müller, Jürgen Schukraft, and Bolesław Wysłouch. "First results from Pb+ Pb collisions at the LHC". In: *Annual Review of Nuclear and Particle Science* 62 (2012), pp. 361–386.

[30] J Schukraft. "Heavy ion physics at the Large Hadron Collider: what is new? What is next?" In: *Physica Scripta* 2013.T158 (2013), p. 014003.

[31] Peter Braun-Munzinger et al. "Properties of hot and dense matter from relativistic heavy ion collisions". In: *Physics Reports* 621 (2016), pp. 76–126.

[32] D Almaalol et al. "QCD Phase Structure and Interactions at High Baryon Density: Completion of BES Physics Program with CBM at FAIR". In: *arXiv preprint arXiv:2209.05009* (2022).

[33] Masaharu Tanabashi et al. "Review of Particle Physics: particle data groups". In: *Physical Review D* 98.3 (2018), pp. 1–1898.

[34] Johann Rafelski and Rolf Hagedorn. *From hadron gas to quark matter, 2*. Tech. rep. 1980.

[35] Johann Rafelski. “Extreme states of nuclear matter-1980: From:“Workshop on Future Relativistic Heavy Ion Experiments” held 7-10 October 1980 at: GSI, Darmstadt, Germany”. In: *The European Physical Journal A* 51.9 (2015), p. 115.

[36] Peter Braun-Munzinger, Krzysztof Redlich, and Johanna Stachel. “Particle production in heavy ion collisions”. In: *Quark–Gluon Plasma 3*. World Scientific, 2004, pp. 491–599.

[37] Johann Rafelski. “Strangeness enhancement: challenges and successes”. In: *The European Physical Journal Special Topics* 155.1 (2008), pp. 139–166.

[38] Peter Koch, Berndt Müller, and Johann Rafelski. “From strangeness enhancement to quark–gluon plasma discovery”. In: *International Journal of Modern Physics A* 32.31 (2017), p. 1730024.

[39] Johann Rafelski and Berndt Müller. “Strangeness production in the Quark-Gluon plasma”. In: *Physical Review Letters* 56.21 (1986), p. 2334.

[40] G Agakichiev et al. “HADES collaboration”. In: *Nuclear Physics, Section A* 774 (2006), pp. 940–941.

[41] *HADES at GSI*. URL: `https://hades.gsi.de/?q=node/2`.

[42] *FOPI*. URL: `https://www-fopi.gsi.de/`.

[43] W Henning. “Physics with SIS/ESR at GSI”. In: *Nuclear Physics A* 538 (1992), pp. 637–648.

[44] Norbert Herrmann. “Status and Perspectives of the CBM experiment at FAIR”. In: *EPJ Web of Conferences*. Vol. 259. EDP Sciences. 2022.

[45] M Durante et al. “All the fun of the FAIR: fundamental physics at the facility for antiproton and ion research”. In: *Physica Scripta* 94.3 (2019), p. 033001.

[46] *FAIR facility*. URL: `https://www.gsi.de/en/researchaccelerators/fair`.

[47] I. C. Arsene et al. “Dynamical phase trajectories for relativistic nuclear collisions”. In: *Phys. Rev. C* 75 (3 Mar. 2007), p. 034902.

[48] Bengt Friman et al. *The CBM physics book: Compressed baryonic matter in laboratory experiments*. Vol. 814. Springer, 2011.

[49] *CBM collaboration*. URL: https://www.cbm.gsi.de/.

[50] HR Schmidt, CBM Collaboration, et al. "The silicon tracking system of the CBM experiment at FAIR". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 936 (2019), pp. 630–633.

[51] J. Stroth and M. Deveaux. *Technical Design Report for the CBM: Micro Vertex Detector (MVD)*. Tech. rep. 1. 2022, 157 p.

[52] *CBMMVD*. URL: https://www.cbm.gsi.de/detectors/mvd.

[53] Johann Heuser et al., eds. *[GSI Report 2013-4] Technical Design Report for the CBM Silicon Tracking System (STS)*. Darmstadt: GSI, 2013, 167 p.

[54] M Dogan et al. "Quality assurance test of the STS-XYTERv2 ASIC for the silicon tracking system of the CBM experiment". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 976 (2020), p. 164278.

[55] K Kasinski et al. "Characterization of the STS/MUCH-XYTER2, a 128-channel time and amplitude measurement IC for gas and silicon microstrip sensors". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 908 (2018), pp. 225–235.

[56] K Agarwal et al. "Progress towards the development of cooling demonstrator for the STS detector of the CBM experiment at FAIR". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 936 (2019), pp. 691–692.

[57] M Teklishyn. "Development of the space-time self-triggering Silicon Tracking System for the CBM Experiment at FAIR". In: (2022).

[58] *Technical Design Report for the CBM Ring Imaging Cherenkov Detector*. Tech. rep. 2013, 215 p.

[59] J Adamczewski-Musch et al. “Status of the CBM and HADES RICH projects at FAIR”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 952 (2020), p. 161970.

[60] *The Transition Radiation Detector of the CBM Experiment at FAIR : Technical Design Report for the CBM Transition Radiation Detector (TRD)*. Tech. rep. FAIR Technical Design Report. Darmstadt, 2018, 165 p. DOI: 10.15120/GSI-2018-01091.

[61] Subhasis Chattopadhyay et al., eds. *Technical Design Report for the CBM : Muon Chambers (MuCh)*. Darmstadt: GSI, 2015, 190 S.

[62] Norbert Herrmann, ed. *Technical Design Report for the CBM Time-of-Flight System (TOF)*. Darmstadt: GSI, 2014, 182 S.

[63] OV Andreeva et al. “Forward scintillation hodoscope for nuclear fragment detection at the high acceptance dielectron spectrometer (HADES) setup”. In: *Instruments and Experimental Techniques* 57 (2014), pp. 103–119.

[64] Joseph Adams et al. “The STAR event plane detector”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 968 (2020), p. 163970.

[65] Xin Gao et al. “Throttling strategies and optimization of the trigger-less streaming DAQ system in the CBM experiment”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 978 (2020), p. 164442.

[66] Valentina Akishina and Ivan Kisel. “Time-based Cellular Automaton track finder for the CBM experiment”. In: *Journal of Physics: Conference Series*. Vol. 599. 1. IOP Publishing. 2015, p. 012024.

[67] Sergey Gorbunov. “On-line reconstruction algorithms for the CBM and ALICE experiments”. PhD thesis. Frankfurt am Main, Johann Wolfgang Goethe-Univ., Diss., 2013, 2013.

[68] PA Zyla et al. “79. Baryon Decay Parameters”. In: *Prog. Theor. Exp. Phys* 2020 (2020).

[69] *CBMAPR20*. URL: https://redmine.cbm.gsi.de/versions/14.

[70] Sea Agostinelli et al. "GEANT4—a simulation toolkit". In: *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303.

[71] Oleksii Lubynets, Ilya Selyuzhenkov, and Viktor Klochkov. "CBM Performance for Λ Hyperon Directed Flow Measurements in Au+ Au Collisions at 12 A GeV/c". In: *Particles* 4.2 (2021), pp. 288–295.

[72] Maksym Zyzak, Ivan Kisel, and Peter Senger. *Online selection of short-lived particles on many-core computer architectures in the CBM experiment at FAIR*. Tech. rep. Collaboration FAIR: CBM, 2016.

[73] Philip Allport. "Applications of silicon strip and pixel-based particle tracking detectors". In: *Nature Reviews Physics* 1.9 (2019), pp. 567–576.

[74] Tetyana Galatyuk. "Future facilities for high μ_B physics". In: *Nuclear Physics A* 982 (2019), pp. 163–169.

[75] Serguei Chatrchyan et al. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 30–61.

[76] *TRACKML*. URL: https://sites.google.com/site/trackmlparticle/home?authuser=0.

[77] Denis Derkach et al. "Machine-Learning-based global particle-identification algorithms at the LHCb experiment". In: *Journal of Physics: Conference Series*. Vol. 1085. 4. IOP Publishing. 2018, p. 042038.

[78] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[79] Sotiris B Kotsiantis. "Decision trees: a recent overview". In: *Artificial Intelligence Review* 39.4 (2013), pp. 261–283.

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

[81] 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*. 2016, pp. 785–794.

[82] Hyunsu Cho and Mu Li. "Treelite: toolbox for decision tree deployment". In: *Proc. Conf. Syst. Mach. Learn.(SysML)*. 2018.

[83] Matthias Feurer and Frank Hutter. "Hyperparameter Optimization". In: *Automated Machine Learning: Methods, Systems, Challenges*. Ed. by Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. Cham: Springer International Publishing, 2019, pp. 3–33. ISBN: 978-3-030-05318-5. DOI: 10.1007/978-3-030-05318-5_1. URL: https://doi.org/10.1007/978-3-030-05318-5_1.

[84] Ian Dewancker, Michael McCourt, and Scott Clark. "Bayesian optimization for machine learning: A practical guidebook". In: *arXiv:1612.04858* (2016).

[85] James Bergstra et al. "Algorithms for hyper-parameter optimization". In: *Advances in neural information processing systems* 24 (2011).

[86] 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*. PMLR. 2013, pp. 115–123.

[87] Hans-Georg Beyer and Hans-Paul Schwefel. "Evolution strategies a comprehensive introduction". In: *Natural computing* 1.1 (2002), pp. 3–52.

[88] Nikolaus Hansen and Andreas Ostermeier. "Completely derandomized self-adaptation in evolution strategies". In: *Evolutionary computation* 9.2 (2001), pp. 159–195.

[89] Takuya Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631.

[90] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[91] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[92] Christoph Molnar. "A guide for making black box models explainable". In: *URL: https://christophm. github. io/interpretable-ml-book* (2018).

[93] P.A. Zyla et al. "Review of Particle Physics". In: *PTEP* 2020.8 (2020), p. 083C01. DOI: 10.1093/ptep/ptaa104.

[94] Viktor Klochkov. "Anisotropic flow measurements at FAIR and SPS energies". PhD thesis. Frankfurt U., 2019.

[95] M Oreglia et al. "Study of the reaction $\psi^{'} \rightarrow \gamma\ \gamma$ J ψ". In: *Physical Review D* 25.9 (1982), p. 2259.

[96] Particle Data Group et al. "Review of particle physics". In: *Progress of Theoretical and Experimental Physics* 2020.8 (2020), p. 083C01.

[97] *ROOT TH1 Divide for binomial*. URL: https://root.cern.ch/doc/master/TH1_8cxx_source.html#l02950.

[98] Luca Lista. *Statistical methods for data analysis in particle physics*. Vol. 941. Springer, 2017.

[99] Andrzej Wilczek. "Lambda production in p+ p interactions at SPS energies". In: (2015).

[100] *ALIHFINVMASS*. URL: https://github.com/alisw/AliPhysics/blob/master/PWGHF/vertexingHF/AliHFInvMassFitter.h.

List of Figures

List of Tables