



UNIVERSIDADE ESTADUAL DE CAMPINAS
Instituto de Física Gleb Wataghin

GABRIEL REIS GARCIA

MACHINE LEARNING METHODS FOR STRANGENESS
RECONSTRUCTION IN ALICE

MÉTODOS DE MACHINE LEARNING PARA RECONSTRUÇÃO DE
ESTRANHEZA NO ALICE

CAMPINAS
2023

GABRIEL REIS GARCIA

MACHINE LEARNING METHODS FOR STRANGENESS
RECONSTRUCTION IN ALICE

MÉTODOS DE MACHINE LEARNING PARA RECONSTRUÇÃO DE
ESTRANHEZA NO ALICE

Tese apresentada ao Instituto de Física Gleb Wataghin da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Ciências na área de Física Aplicada.

Thesis presented to the Gleb Wataghin Institute of Physics of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor of science in the area of applied physics.

Orientador: Prof. Dr. Jun Takahashi

ESTE TRABALHO CORRESPONDE À VERSÃO FINAL DA
TESE DEFENDIDA PELO ALUNO GABRIEL REIS GARCIA,
E ORIENTADA PELO PROF. DR. JUN TAKAHASHI.

CAMPINAS
2023

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Física Gleb Wataghin
Lucimeire de Oliveira Silva da Rocha - CRB 8/9174

G165m Garcia, Gabriel Reis, 1995-
Machine learning methods for strangeness reconstruction in ALICE /
Gabriel Reis Garcia. – Campinas, SP : [s.n.], 2023.

Orientador: Jun Takahashi.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Física
Gleb Wataghin.

1. Aprendizado de máquina. 2. Partículas estranhas (Física nuclear). 3.
Grande Colisor de Hádrons (França e Suíça). 4. Experimento ALICE. 5. Plasma
de quarks e glúons. I. Takahashi, Jun, 1971-. II. Universidade Estadual de
Campinas. Instituto de Física Gleb Wataghin. III. Título.

Informações Complementares

Título em outro idioma: Métodos de machine learning para reconstrução de estranheza no
ALICE

Palavras-chave em inglês:

Machine learning

Strange particles

Large Hadron Collider (France and Switzerland)

ALICE experiment

Quark-gluon plasma

Área de concentração: Física Aplicada

Titulação: Doutor em Ciências

Banca examinadora:

Jun Takahashi [Orientador]

Pedro Cunha de Holanda

Ettore Segreto

Andre Sznajder

Tiago Fiorini da Silva

Data de defesa: 24-11-2023

Programa de Pós-Graduação: Física

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-5298-2881>

- Currículo Lattes do autor: <http://lattes.cnpq.br/3174816999793596>



INSTITUTO DE FÍSICA
GLEB WATAGHIN

MEMBROS DA COMISSÃO EXAMINADORA DA TESE DE DOUTORADO DO ALUNO GABRIEL REIS GARCIA - RA 146156, APRESENTADA E APROVADA AO INSTITUTO DE FÍSICA GLEB WATAGHIN, DA UNIVERSIDADE ESTADUAL DE CAMPINAS, EM 24/11/2023.

COMISSÃO JULGADORA:

- Prof. Dr. Jun Takahashi – Presidente e Orientador (IFGW/UNICAMP)
- Prof. Dr. Pedro Cunha de Holanda (IFGW/UNICAMP)
- Prof. Dr. Ettore Segreto (IFGW/UNICAMP)
- Prof. Dr. Andre Sznajder (Instituto de Física da Universidade do Estado do Rio de Janeiro)
- Prof. Dr. Tiago Fiorini da Silva (Instituto de Física da USP)

OBS.: Ata da defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

CAMPINAS

2023

"...It ain't about how hard you hit.
it's about how hard you can get hit
and keep moving forward..."

Agradecimentos

Primeiramente, gostaria de agradecer ao meu orientador, Prof. Dr. Jun Takahashi, por ter tornado este trabalho possível com sua orientação, discussões, incentivo, confiança e presença em todas as etapas do desenvolvimento deste projeto. Agradeço também aos demais membros, atuais e anteriores, do grupo HadrEx: Prof. Dr. David Dobrigkeit Chinellato, Gianni Shigeru Setoue Liveraro, Vitor Hugo Ribeiro, Gabriel Rabelo Soares, Cícero Muncinelli, João Paulo Pichetti, Prof. Dr. André Veiga Giannini, Prof. Dr. Tiago José Nunes da Silva e Prof. Dr. Donato Giorgio Torrieri, pelo apoio e cooperação que foram fundamentais para o desenvolvimento deste trabalho.

Agradeço a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo financiamento deste trabalho Proc. 2019/03743-0 e pela oportunidade de realizar um projeto de Bolsa Estágio de Pesquisa no Exterior (BEPE) Proc. 2022/01562-1 durante o doutorado, que foi crucial para a conclusão do trabalho. O financiamento ocorreu dentro do projeto temático "Física Hadrônica em Colisões Nucleares de Altas Energias" Proc. 2017/05685-2. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Processo nº 88887.336577/2019-00.

Agradeço aos demais professores: Prof. Dr. Marcus Aloizio Martinez de Aguiar, Prof. Dr. Ricardo Luís Doretto, Prof. Dr. Fernando José Von Zuben, Prof. Dr. Marcelo Knobel, Profa. Flávia Sobreira, Prof. Dr. André Monteiro Paschoal, Prof. Dr. Ernesto Kemp, Prof. Dr. Antônio Riul Júnior, Prof. Dr. Pedro Cunha de Holanda, Prof. Dr. Ettore Segreto, Prof. Dr. Andre Sznajder e Prof. Dr. Tiago Fiorini da Silva, pelas discussões e sugestões que me permitiram aprimorar esta tese, desenvolver o projeto e pelas disciplinas lecionadas que me proporcionaram atingir o conhecimento necessário para produzir este trabalho.

Agradeço a minha família e amigos: Heliana Luisa Guardiano Reis, Reinaldo Garcia, Rafael Reis Garcia, Danilo Rodrigues de Assis Elias, Marcos Luiz Ferreira Gomes, Aderlon Uchechukwu Agoha, Flávia do Carmo Silva, Amanda Vettorazzo Halsman, Caique Conde Rodrigues, João Pedro Oliveira Pompiani dos Santos, William Freitas, Valéria Mattiello, Mateus Corradini Lopes, Maria Clara Giassetti Medeiros Corradini Lopes, Viviane Therezinha de Faria Fonseca, Eliane Valente e aos demais que eu possa ter esquecido de mencionar ao escrever, pelo incondicional suporte, carinho e companhia que me motivaram e sustentaram ao longo desta jornada.

Por fim, gostaria de agradecer a comunidade do Instituto de Física Gleb Wataghin como um todo: alunos, funcionários e professores, por fazerem deste lugar uma segunda casa para mim. Por terem me proporcionado uma experiência fantástica na universidade que vai muito além do escopo deste trabalho e da produção intelectual.

A todos vocês o meu mais profundo e sincero, muito obrigado.

Resumo

O objetivo do projeto de doutorado foi explorar as aplicações de técnicas de *Machine Learning* em análise de dados e desenvolver uma nova metodologia para abordar um problema físico específico: a reconstrução de bárions multi-estranhos. Este trabalho foi desenvolvido em colaboração com o Experimento ALICE (*A Large Ion Collider Experiment*). O ALICE tem como objetivo estudar matéria fortemente interagente em densidades de energia tão altas que um estado da matéria chamado Plasma de Quarks e Glúons (QGP) se forma. Isso é alcançado usando um conjunto de detectores projetados e otimizados para medir colisões ultra-relativísticas de íons pesados no LHC (*Large Hadron Collider*) localizado no CERN. O aumento da estranheza é um dos possíveis observáveis para investigar o QGP, portanto, precisamos de medições precisas da produção de partículas estranhas. Detectar essas partículas envolve a reconstrução de seus vértices de decaimento e a determinação da massa invariante de várias combinações de trajetórias. No entanto, a presença de inúmeras combinações de trajetórias que não estão associadas ao processo de decaimento introduz um fundo significativo nos espectros de massa invariante. Para lidar com esse desafio, cortes de seleção com base em variáveis topológicas são comumente usados para suprimir o fundo combinatório. Uma vez que esses parâmetros topológicos frequentemente exibem correlações, há um potencial considerável para melhorias por meio do uso de técnicas de análise multivariacional, como o *Machine Learning*. Neste trabalho, a implementação de métodos de *Machine Learning* resultou em maior eficiência e significância, aumentando assim a precisão dos espectros de momento transversal das partículas estudadas. O *Machine Learning* também permitiu o acesso a um ponto experimental sem precedentes na região de baixo momento transversal. A extensão do limite inferior melhorou o procedimento de extrapolação, resultando em uma maior precisão na produção global de partículas.

Palavras-chave: Aprendizado de máquina, Partículas estranhas (Física nuclear), Grande Colisor de Hádrons (França e Suíça), Experimento ALICE, Plasma de quarks e glúons

Abstract

The objective of the PhD project was to explore the applications of Machine Learning techniques in data analysis and develop a novel methodology to address a specific Physics problem: The reconstruction of multi-strange baryons. This work was developed within the A Large Ion Collider Experiment (ALICE) collaboration. ALICE aims to study strongly interacting matter in such high energy densities that a state of matter called Quark-Gluon Plasma (QGP) forms. It accomplishes this using a set of detectors designed and optimized to measure ultra-relativistic collisions of heavy nuclei at the Large Hadron Collider (LHC) located at CERN. Strangeness enhancement is one possible observable to probe the QGP, therefore we need precise measurement for the production of strange particles. Detecting these particles involves reconstructing their secondary decay vertices and determining the invariant mass of various track combinations. However, the presence of numerous combinations tracks that are not associated with the decay process introduces a significant background in the invariant mass spectra. To address this challenge, selection cuts based on topological variables are commonly employed to suppress the combinatorial background. Given that these topological parameters often exhibit correlations, there is a considerable potential for improvement through the use of multivariate analysis techniques, such as Machine Learning. In this work the implementation of Machine Learning methods has resulted in enhanced efficiency and significance, therefore increasing the precision of the transverse momentum spectra of the studied particles. Machine Learning also allowed the access to an unprecedented experimental point in the low transverse momentum region. The extension of the inferior limit improved the extrapolation procedure, leading to improved precision of the overall particle production yield.

Keywords: Machine Learning, Strangeness (Nuclear physics), Large Hadron Collider (France and Switzerland), ALICE experiment, Quark-Gluon Plasma

List of Figures

1.1	Flowchart illustrating the difference in paradigm between conventional computing and Machine Learning.	19
1.2	Schematic of a Neural Network MLP.	22
1.3	Schematic of a BDT.	23
2.1	The Standard Model of Particle Physics. Figure from [12].	25
2.2	Depiction of color confinement resulting in hadronization.	26
2.3	Summary of measurements of α_s as a function of the respective energy scale Q . Figure from [14].	27
2.4	Phase diagram (temperature, net baryon density) of QCD matter, ranging from regular nuclear matter to Quark-Gluon Plasma. Figure from [20]	28
2.5	Multistrange (anti)baryons as signature of QGP, Figure from [21]	29
2.6	p_T -integrated yield ratios of strange and multi-strange hadrons to $(\pi^+ + \pi^-)$ as a function of $\langle dN_{ch}/d\eta \rangle$ measured in the rapidity interval $ y < 0.5$. The empty and dark-shaded boxes show the total systematic uncertainty and the contribution uncorrelated across multiplicity bins, respectively. The values are compared to calculations from MC models [22, 23, 24], and to results obtained in Pb-Pb and p-Pb collisions at the LHC [25, 26, 27]. Figure from [28].	30
3.1	Coordinate system of ALICE.	32
3.2	ALICE's Inner Tracking System (ITS). Figure from [30]	33
3.3	ALICE's Time Projection Chamber (TPC). Figure from [31]	34
3.4	TPC dE/dx as a function of momentum with superimposed Bethe-Bloch lines for various particle species for RUN2 Pb-Pb low interaction rate runs. Figure from [32]	34
3.5	Distribution of the sum of the amplitudes in the V0 detectors for Pb-Pb at $\sqrt{s_{NN}}=5.02$ TeV. The threshold boundaries that define each centrality class are represented by the vertical lines. The inset shows a zoom in the most peripheral region. The Glauber Negative Binomial Distribution (NBD-Glauber) fit is shown in red. Figure from [33]	35
4.1	Cascade decay topology of Ξ^-	37
4.2	Topological variables of the cascade decay. Ξ^- is given as example.	38
4.3	Parametrization of cosine of pointing angle cuts as a function of p_T . Cascade $\cos(PA)$ is shown in red and V0 $\cos(PA)$ is shown in blue.	39

4.4	Prevision distribution of the Ξ^- Neural Network for the background data (blue) and signal data (orange) in the validation set.	41
4.5	Neural network output distribution for signal (in red) and background (in gray) in the test set for $0.8 \leq p_T < 1.0 \text{ GeV}/c$	42
4.6	Classification threshold analysis for $0.8 \leq p_T < 1.0 \text{ GeV}/c$. Signal and background counts from the standard analysis are shown in red. Signal and background counts from the ML analysis are shown in blue. Full lines indicates background, dashed lines indicates signal. Signal intersection at threshold=0.86 and background intersection at threshold=0.66. Inner graphic shows significance as a function of classification threshold for both methods and the intersection at threshold=0.28.	43
4.7	Classification thresholds of the intersections of signal counts (in black), background counts (in red) and significance (in blue) as a function of p_T . The dashed line indicates the average classification threshold for background intersections.	44
4.8	Efficiency of reconstructing Ξ^- using an artificial neural network (in blue) in comparison with the standard analysis of topological cuts (in red). Open points indicate the efficiencies for ± 0.1 of the classification threshold choice. Bellow efficiencies are the ratio from the standard method. The average gain is 22%.	45
4.9	Significance of reconstructing Ξ^- using an artificial neural network (in blue) in comparison with the standard analysis of topological cuts (in red). Open points indicate the significance for ± 0.1 of the classification threshold choice. Bellow significance are the ratio from the standard method. The average gain is 19% and 37% in the lowest p_T bin.	45
4.10	Invariant mass spectrum of Ξ^- for $1.2 \leq p_T < 1.4 \text{ GeV}/c$ in real data reconstructed by the neural network analysis (in black) in comparison with the spectrum reconstructed by the standard analysis (in red). Dashed curves indicate the fits performed by signal extraction. Ratio of the extracted signals show a 22% improvement due to the use of neural network.	47
4.11	Invariant mass spectrum of Ξ^- for $0.6 \leq p_T < 0.8 \text{ GeV}/c$ in real data reconstructed by the neural network analysis. (in black) in comparison with the spectrum reconstructed by the standard analysis (in red). Dashed curves indicate the fits performed by signal extraction. Ratio of the extracted signals show a 22% improvement due to the use of neural network.	48
5.1	Ratio between the statistical uncertainty of the Ξ^- corrected p_T spectra obtained with the neural network analysis by the one obtained with the standard analysis in the 0-10% centrality class.	50
5.2	Ξ^- in $0.8 < p_T < 1.0 \text{ GeV}/c$ candidate selection systematic uncertainty of the NN (blue) and standard analysis (red). The blue markers are the NN extracted signal in each threshold interval. Blue line is the NN average extracted signal. Blue dashed lines are the estimate systematic for NN. Red line is the extracted signal from the standard analysis. Red dashed lines are the standard analysis systematic.	52

5.3	Ω^+ in $2.8 < p_T < 3.2$ GeV/c candidate selection systematic uncertainty of the BDT (black) and standard analysis (red). The black markers are the BDT extracted signal in each threshold interval. Black line is the BDT average extracted signal. Black dashed lines are the estimate systematic for BDT. Red line is the extracted signal from the standard analysis. Red dashed lines are the standard analysis systematic.	52
5.4	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 0-10% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	54
5.5	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 0-10% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	54
5.6	High-Low fit extrapolation to low p_T of $\Xi^- + \Xi^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the High-Low fits used to estimate the systematic uncertainty. Neural network analysis is shown in blue and the standard analysis in red.	55
5.7	High-Low fit extrapolation to low p_T of $\Xi^- + \Xi^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the High-Low fits used to estimate the systematic uncertainty. BDT analysis is shown in black and the standard analysis in red.	56
5.8	High-Low fit extrapolation to low p_T of $\Omega^- + \Omega^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the High-Low fits used to estimate the systematic uncertainty. Neural network analysis is shown in blue and the standard analysis in red.	56
5.9	High-Low fit extrapolation to low p_T of $\Omega^- + \Omega^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the High-Low fits used to estimate the systematic uncertainty. BDT analysis is shown in black and the standard analysis in red.	57
5.10	Secondary function fit extrapolation to low p_T of $\Xi^- + \Xi^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the m_T exponential fits used to estimate the systematic uncertainty. Neural network analysis is shown in blue and the standard analysis in red.	58
5.11	Secondary function fit extrapolation to low p_T of $\Xi^- + \Xi^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the m_T exponential fits used to estimate the systematic uncertainty. BDT analysis is shown in black and the standard analysis in red.	58

5.12	Secondary function fit extrapolation to low p_T of $\Omega^- + \Omega^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra. Dashed line for Neural Network is the m_T exponential fit and dashed line for the standard analysis is the Hagedorn fit used to estimate the systematic uncertainty. Neural network analysis is shown in blue and the standard analysis in red.	59
5.13	Secondary function fit extrapolation to low p_T of $\Omega^- + \Omega^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra. Dashed line for BDT is the m_T exponential and dashed line for the standard analysis is the Hagedorn fit used to estimate the systematic uncertainty. BDT analysis is shown in black and the standard analysis in red.	59
6.1	Transverse momentum spectra of $\Xi^- + \Xi^+$ (left) and $\Omega^- + \Omega^+$ (right) in Pb-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV for all centrality classes. Obtained with the standard analysis (top), a Neural Network analysis (center) and a Boosted Decision Tree analysis (bottom). Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra.	62
6.2	$\Xi^- + \Xi^+$ production yield as a function of centrality class. The standard analysis is shown in red, Neural Network analysis in blue and BDT analysis in black. Lower plot presents the ratio of uncertainties from Machine Learning based methods to the standard analysis.	64
6.3	$\Omega^- + \Omega^+$ production yield as a function of centrality class. The standard analysis is shown in red, Neural Network analysis in blue and BDT analysis in black. Lower plot presents the ratio of uncertainties from Machine Learning based methods to the standard analysis.	64
6.4	Ratio of multi-strange baryons to pions as a function of charged particle multiplicity compared with previous measurements already presented in figure 2.6.	65
A.1	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 10-20% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	72
A.2	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 20-30% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	73
A.3	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 30-40% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	73

A.4	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 40-50% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	74
A.5	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 50-60% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	74
A.6	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 60-70% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	75
A.7	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 70-80% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	75
A.8	Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 80-90% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	76
A.9	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 10-20% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	76
A.10	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 20-30% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	77
A.11	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 30-40% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.	77

A.12	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 40-50% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis. . . .	78
A.13	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 50-60% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis. . . .	78
A.14	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 60-70% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis. . . .	79
A.15	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 70-80% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis. . . .	79
A.16	Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 80-90% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis. . . .	80

List of Tables

3.1	Charged particle multiplicity in each centrality class	35
4.1	Decay channels of strange baryons	37
4.2	Selection criteria	38
4.3	Topological cuts from standard analysis.	39
5.1	Systematic Uncertainties for $\Xi^-(\Xi^+)$ in percentage (%)	51
5.2	Systematic Uncertainties for $\Omega^-(\Omega^+)$ in percentage (%)	51
5.3	Systematic Uncertainties for $\Xi^-(\Xi^+)$ in percentage (%)	53
5.4	Systematic Uncertainties for $\Omega^-(\Omega^+)$ in percentage (%)	53
5.5	Standard analysis relative uncertainties of production yield of $\Xi^- + \Xi^+$ ($\Omega^- + \Omega^+$) .	60
5.6	Neural network analysis relative uncertainties of production yield of $\Xi^- + \Xi^+$ ($\Omega^- + \Omega^+$)	60
5.7	BDT analysis relative uncertainties of production yield of $\Xi^- + \Xi^+$ ($\Omega^- + \Omega^+$) . . .	60
6.1	Standard analysis: Particle production yield	63
6.2	Neural Network: Particle production yield	63
6.3	Boosted Decision Tree: Particle production yield	63

Contents

1	A brief introduction to Machine Learning	18
1.1	Definition of Machine Learning	18
1.2	Linear regression	20
1.3	Classification problem	21
1.3.1	Artificial Neural Networks	22
1.3.2	Boosted Decision Trees	23
2	Heavy-ion physics	24
2.1	Elements of the standard model for heavy-ion physics	24
2.1.1	Strong interaction	25
2.2	Quark-Gluon Plasma	27
3	A Large Ion Collider Experiment	31
3.1	Coordinate system and kinematic variables	31
3.2	Measurements for strangeness reconstruction	33
4	Data analysis methodology	36
4.1	ALICE data flow	36
4.2	Strangeness Reconstruction	36
4.3	Machine Learning analysis	40
4.3.1	Training results in the test set	41
4.4	Analysis in the complete data set	46
5	Computation of Uncertainties	50
5.1	Statistical uncertainties	50
5.2	Systematical uncertainties	51
5.2.1	Candidate selection	51
5.2.2	Signal extraction	53
5.2.3	Other sources of systematic uncertainties	53
5.2.4	Extrapolation to low transverse momentum	55
6	Results and discussion	61
6.1	Transverse momentum spectra	61
6.2	Particle production yield	61

6.3 Strangeness Enhancement	65
7 Conclusion	66
Bibliography	68
A Comparison of transverse momentum spectra	72
B Codes developed for the ALICE framework	81

Chapter 1

A brief introduction to Machine Learning

Since it is a thesis in physics this chapter is meant to introduce core concepts and terminology of the field of Machine Learning that were used in this work. It starts by presenting two definitions of Machine Learning and a brief discussion on the fact that even though the term “Machine Learning” is showing up very often in almost every field of knowledge these days, it is just a problem-solving paradigm that is already used in physics, quite often, perhaps without even knowing that it satisfies the definitions. It is also introduced a class of problems that can be solved with Machine Learning, called classification problem, and the two types of Machine Learning algorithms used in the developed methodology: Artificial Neural Network and Boosted Decision Tree.

1.1 Definition of Machine Learning

Definition 1.1.1. *“Machine Learning is to give a computer the ability to perform a task without being explicitly programmed to do it.”*

The “Learning” part of the concept is due to the fact that usually the computer’s ability to perform a task, in this way, is obtained through an iterative process called training and that resembles the idea of the human ability to learn by conditioning. Figure 1.1 illustrates the difference in paradigm between conventional computing and Machine Learning [1].

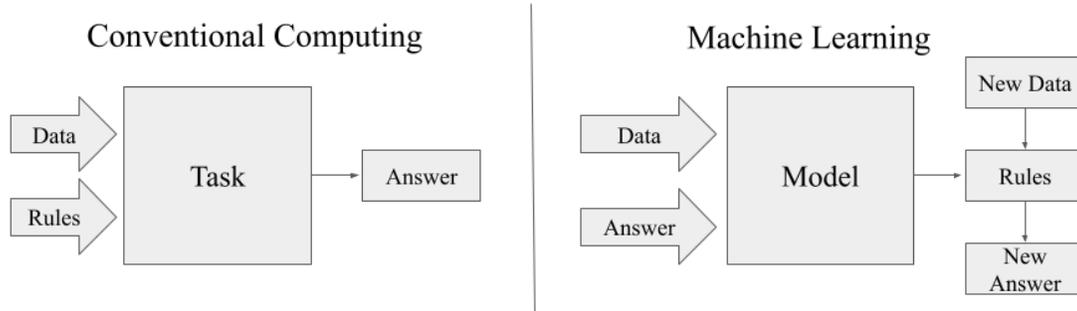


Figure 1.1: Flowchart illustrating the difference in paradigm between conventional computing and Machine Learning.

In conventional computing the rules are the necessary code that one writes to execute a task and get an answer. In that scenario the computer was explicitly programmed to do so, therefore it does not satisfy the definition of Machine Learning. The data is a possible input of the code in conventional computing.

In Machine Learning the data is a necessary input to a model that can learn rules and with those rules execute a task. This is what the phrase ‘...without being explicitly programmed...’ means in the definition. The answer is a possible input to the model, when it is present we call it “Supervised Learning” and when it is not we call it “Unsupervised Learning”.

Examples of tasks that can be performed using supervised learning are classification and regression. Examples of tasks that can be performed using unsupervised learning are pattern recognition, clustering and data generation.

A more formal definition of Machine Learning can be enunciated as [2]

Definition 1.1.2. “Given a task T with an experience E , over a data set, and a performance metric for the task $P(T)$, an algorithm is Machine Learning if $P(T)$ increases with E .”

$$\Delta P(T) > 0 \Leftrightarrow \Delta E > 0. \quad (1.1.1)$$

One could reasonably argue that equation 1.1.1 is also a good definition of the word "learn", and given the context of computing and algorithms, the expression "Machine Learning" becomes not just a fancy name for a field of study but also a literal semantic statement.

1.2 Linear regression

Linear regression is often used as an introductory example in Machine Learning [3]. The goal of linear regression is to find the line that best describes a data set. Technically, it represents a line when the data is in a two-dimensional Cartesian plane (\mathbb{R}^2). In a more general case of \mathbb{R}^n , with an euclidean metric, the linear regression finds the best hyper-plane that describes the data set.

Consider a data set with size S in which each data can be written as a vector $\vec{X}_i = (x_{i1}, \dots, x_{in})$, where each entry is a real number, called feature, and Y_i is a known answer for each \vec{X}_i with $1 \leq i \leq S$. The linear regression problem use the hypothesis that Y is a linear function of \vec{X} . Let

$$P_i = w_0 + \vec{W} \cdot \vec{X}_i, \quad (1.2.1)$$

be a linear prediction of each entry in the data set. To solve the linear regression problem is to determine the parameters w_0 and $\vec{W} = (w_1, \dots, w_n)$, called weights, to best describe the data set, that is make P_i as close as possible of Y_i for every data.

It is common in the literature to redefine $\vec{X} = (1, x_{i1}, \dots, x_{in})$ and $\vec{W} = (w_0, w_1, \dots, w_n)$ to simplify expression for P , that becomes just

$$P_i = \vec{W} \cdot \vec{X}_i, \quad (1.2.2)$$

incorporating the w_0 weight, also called bias term, in the dot product.

The solution to this problem involves defining a cost function, the classic choice is the mean squared error between Y_i and P_i , given by

$$J \equiv \frac{1}{S} \sum_{i=1}^S (Y_i - P_i)^2 = \frac{1}{S} \sum_{i=1}^S (Y_i - \vec{W} \cdot \vec{X}_i)^2. \quad (1.2.3)$$

And then minimizing J with respect to \vec{W} . Since we are dealing with a linear hypothesis, there is an analytical solution for the optimization, that is going to be presented for the sake of completeness.

It is convenient to define a data matrix $\Phi \in \mathbb{R}^{S \times N}$, where $N = n + 1$. Such matrix has in each of its lines one of the S data vectors \vec{X}_i with n features plus the unit in the redefinition for equation 1.2.2. Then we can say that for the entire data set

$$\vec{P} = \Phi \vec{W}, \quad (1.2.4)$$

where $\vec{P} \in \mathbb{R}^S$ is the vector with all predictions that we want to approach $\vec{Y} \in \mathbb{R}^S$, the vector with all answers. Let $\vec{E} = \vec{Y} - \vec{P}$ be the error vector. With that, equation 1.2.3 becomes

$$J = \frac{1}{S} |\vec{E}|^2, \quad (1.2.5)$$

therefore

$$\min_{\vec{W}} J = \min_{\vec{W}} |\vec{E}|^2. \quad (1.2.6)$$

To optimize the the error with respect to the weights we do

$$\frac{d}{d\vec{W}}|\vec{E}|^2 = \frac{d}{d\vec{W}}|\vec{Y} - \Phi\vec{W}|^2 = 0 \quad (1.2.7)$$

and find that

$$\vec{W} = (\Phi\Phi^T)^{-1}\Phi^T\vec{Y}. \quad (1.2.8)$$

Here we conclude that the solution of the linear regression requires the inverse matrix of $\Phi\Phi^T$ to exist.

Note that the task of finding the best hyper-plane to describe the data reduces to an optimization problem. And that is the core of the Machine Learning paradigm. For example one can write a code that performs a gradient descent to minimize the cost-function, therefore finding an optimal \vec{W} and consequently the best hyper-plane. This logic satisfies definition 1.1.1 of Machine Learning. Even though the discussion started with a linear hypothesis the paradigm does not change if it was any other type of function. That means, every time we execute a fitting procedure in physics, we are using Machine Learning.

1.3 Classification problem

The reconstruction of strange particles is a far more complicated problem to be addressed and solved with a linear hypothesis. Nevertheless the problem we want to solve falls into the category of classification problem, that is one possible application of supervised learning techniques. Just like in the linear regression, the data set is composed of feature vectors, however the answer associated to each data is a discrete information, called class.

When there are only two possible classes we call it a binary-classification problem. The model to solve such a problem receives features as input and its output is actually not discrete, it is constructed to be interpreted as the probability P of the input belonging to one of the classes, with $C = 1$ being a possible known answer. The probability of the input belonging to the other class, that has $C = 0$, is simply $1 - P$. A Machine Learning algorithm to solve this type of problem can be structured as follows:

1. Parameterize P .
2. Select a data sample containing features and their respective known classes C , called training set.
3. Define a cost function $J(P, C)$.
4. Iterate over the training set to minimize J .

In this work two different types of Machine Learning algorithms were used, that are essentially, two different ways of parameterize P .

1.3.1 Artificial Neural Networks

The neuron is the basic unit of a neural network, such a structure can receive any number of inputs and has one number as its output. For the purpose of this work it is enough to define the perceptron [4]. Which is a type of neuron that performs the scalar product of the input with its own parameters, called synaptic weights and then passes this number through a, usually, non-linear function, called activation function.

Neural networks consist of interconnected neurons, forming complex structures like a Multi-Layer Perceptron (MLP), a particular type of architecture where all the neurons in a layer are fully connected with the ones of the next. Figure 1.2 depicts the described structure, with three input neurons, one inner layer with four neurons and one output neuron.

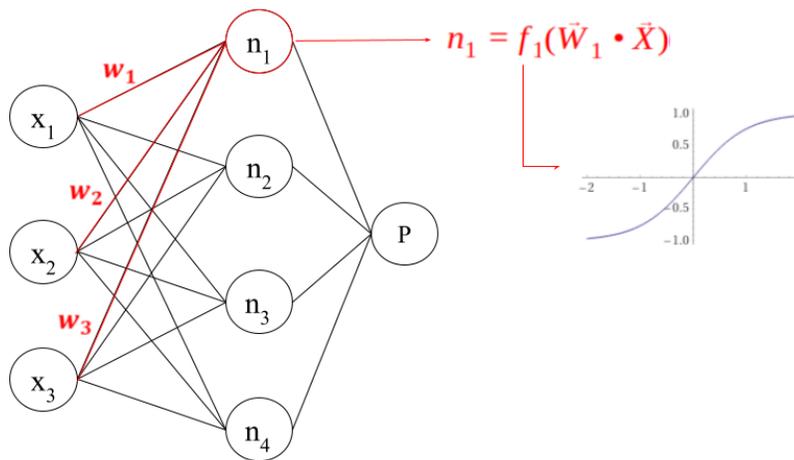


Figure 1.2: Schematic of a Neural Network MLP.

$P(w)$ is determined by a sequence of scalar products between neuron inputs and synaptic weights. In the example of figure 1.2 we have for each of the four inner neurons

$$n_i = f_i \left(\sum_{j=1}^3 w_{ij} x_j \right), \quad (1.3.1)$$

therefore the output is given by

$$P = f_P \left(\sum_{i=1}^4 w_{Pi} n_i \right) = f_P \left(\sum_{i=1}^4 w_{Pi} f_i \left(\sum_{j=1}^3 w_{ij} x_j \right) \right) \quad (1.3.2)$$

and the more inner layers we have, more complicated it gets.

The optimization of neural networks is typically numerical and often involves methods like gradient descent and back-propagation to iteratively adjust weights and minimize the cost function. This approach allows us to address classification problems that go beyond linear relationships and enables the development of more intricate models for the reconstruction of strange particles.

1.3.2 Boosted Decision Trees

Boosted Decision Trees are a bit different, since they don't have parameters like the synaptic weights. They only have what we call hyper-parameters. For the neural networks, hyper-parameters are the fixed choices of the model, such as the number of layers, and the numbers of neurons in each layer. Boosted Decision Trees have fixed hyper-parameters, like the number of estimators and their depth. However the training process of such algorithm minimizes J by modifying the contents inside the hyper-parameters.

Decision trees are simple models, they work by recursively splitting the data set into subsets based on the features, eventually forming a tree-like structure with decision rules at each node. Boosting involves training a series of decision trees sequentially, where each tree corrects the errors made by the previous ones. It is an ensemble learning method that combines the predictions of multiple decision trees to create a robust and accurate model.

Each tree is, essentially, a sequences of "yes or no, questions" about the features, to define a score. The depth gives how many questions are asked by each estimator, and the number of estimators are how many sequences of questions are performed. In the end the scores are combined as the model's prediction. Figure 1.3 illustrates the idea. In the depicted example there are n estimators of depth 2.

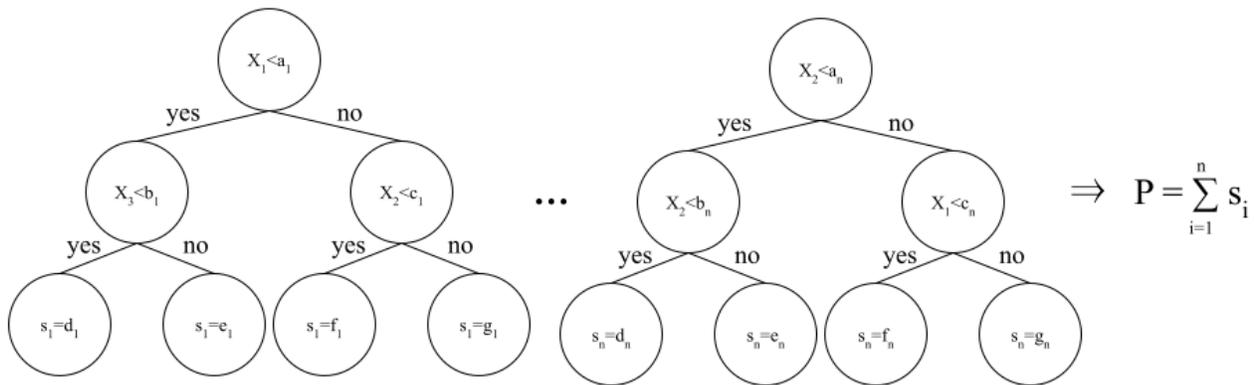


Figure 1.3: Schematic of a BDT.

Chapter 2

Heavy-ion physics

In this chapter an introduction to the some aspects of the standard model of particle physics is presented emphasizing the strong interaction between quarks and gluons and giving a highlight to the strange quark. The discussion then moves towards the main object of study of the field of heavy-ion physics, a state of matter called quark-gluon plasma (QGP), and then is introduced the observable used to probe such state regarding the highlighted strange quark that motivates this work of improving the measurement of strange particles.

2.1 Elements of the standard model for heavy-ion physics

Since 1978 the standard model of particles physics is one of the most well succeeded scientific theories in human history, in the sense of prediction capabilities and precision [5]. Almost 20% of every Nobel prize in physics ever awarded was somehow related with this model that describes the fundamental structures of matter and its interactions.

The particles of the standard model, presented in figure 2.1, consists of fermions and bosons and for each of them there is also a respective anti-particle that has the same mass as the corresponding particle but carries the opposite electric charge and quantum numbers. The fermions are particles of spin $\frac{1}{2}$ and the building blocks of matter. They are divided into three generations of quark and leptons. For the quarks each generation has two types, that we call flavors, a positive ($+\frac{2}{3}e$) and negative ($-\frac{1}{3}e$) electrically charged, and the flavors gets heavier, and therefore unstable, with the generations. Only the up and down quarks (1st generation) are stable. The strange quark is the lightest ($m_s=96 \text{ MeV}/c^2$) unstable flavor and the top quark is the heaviest ($m_t=173 \text{ GeV}/c^2$) weighting about the same as an entire caffeine molecule.

For the leptons each generation has one electrically charged ($-1e$) flavor, that like the quarks gets heavier and unstable across generations, and one respective flavor of neutrino that are electrically neutral. Neutrinos behave differently with respect of mass and stability, since they can oscillate they must have at least a tiny mass but it's not yet well understood why or how they acquire mass [6, 7, 8].

The bosons are particles of integer spin and the mediators of the interactions. Three fundamental interactions are described by the standard model. The gluon mediates the strong interaction,

the photon mediates the electromagnetic interaction and the W and Z bosons are the mediators of the weak interaction. All of them have spin 1 and because of that are also called vector bosons. And finally the last discovered piece of the model is the Higgs boson with spin 0 and because of that also called scalar boson. The interaction with such particle results in the generation of mass of every elementary particle through the Higgs mechanism of spontaneous symmetry breaking [9]. I'm not going to delve into the mathematical aspect of the theory of the standard model in this work, however its worth mentioning that it resides in the regime of Quantum Field Theory [10], that symmetry is one very fundamental aspect of it, and also bounded by theorem [11] with one very fundamental aspect of nature that are conservation laws.

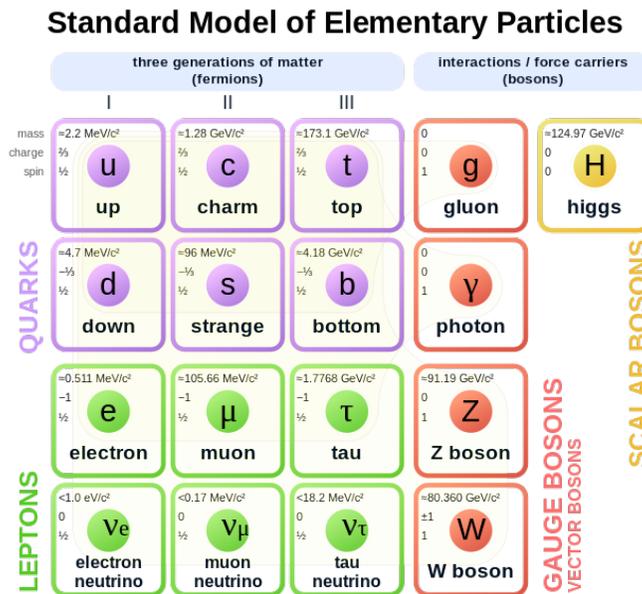


Figure 2.1: The Standard Model of Particle Physics. Figure from [12].

2.1.1 Strong interaction

In this work the focus is set into the strong interaction that happens only between quarks and gluons. Like the well known electromagnetism that requires a particle to have an electric charge to interact, the strong interaction have its own charge that appears in three types and three anti-types. The total charge of a system is neutral if there are present one charge of each type, one charge of each anti-type or a charge of type and its respective anti-type, this logical way of summing zero is called algebra and it behaves like this for the strong interaction because of the symmetry group of the theory $SU(3)$ [13] that implies in the conservation law of such charges. We are already familiar with a concept that satisfies the described algebra: The RGB color system where $R+G+B = \text{white } (0)$. And that is why the charge of the strong interaction is called color charge, and the theory that describes it is called Quantum Chromodynamics (QCD).

QCD has two fundamental properties to be discussed: color confinement and asymptotic freedom. However before entering the discussion I believe to be important to briefly define some terminology:

- "parton": is either a quark or gluon.
- "hadron": any particle formed by quarks or anti-quarks.
- "meson": a particle formed by a quark and an anti-quark.
- "baryon": a particle formed by three quarks.
- "anti-baryon": a particle formed by three anti-quarks.

Color confinement

Note that the definitions of meson, baryons and anti-baryons calls back to the given algebra of a neutral color system. It is the fundamental property, called color confinement, that essentially states that every measurable particle must be color neutral, therefore we can only measure bound states of quarks and gluons and never the elementary partons alone.

If one tries to pull apart, let's say, a pair of quark and anti-quark the attraction between them will increase with the distance and the potential energy will become high enough to generate the masses of a new pair of quarks and anti-quarks. This idea is illustrated in figure 2.2 and defines the concept of hadronization.

There is a particular interest in hadrons containing strange quarks, more specifically for this work, the goal was to improve the detection of multi-strange baryons, that possesses more than one strange quark. They are the $\Xi^- (ssd)$ and $\Omega^- (sss)$ and their respective anti-particles $\Xi^+ (\bar{s}\bar{s}\bar{d})$ and $\Omega^+ (\bar{s}\bar{s}\bar{s})$.

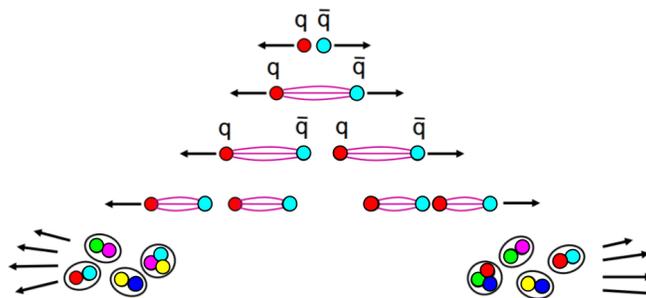


Figure 2.2: Depiction of color confinement resulting in hadronization.

Asymptotic freedom

In the perturbative framework of QCD, predictions for observables are expressed in terms of the running coupling $\alpha_s(\mu_R^2)$, with μ_R being an renormalization scale. In a given QCD process the running coupling indicates the effective strength of the strong interaction when the scale μ_R approaches the momentum transfer Q of the process.[14] The second fundamental property of QCD, arises when calculating rate of the running α_s . For this discussion is sufficient to say that

$$\frac{d\alpha_s}{d\mu_R^2} < 0, \quad (2.1.1)$$

which means that the running coupling of QCD is monotonic decreasing, as shown in figure 2.3 . Therefore the strength of the strong interaction becomes weak for processes with high Q .

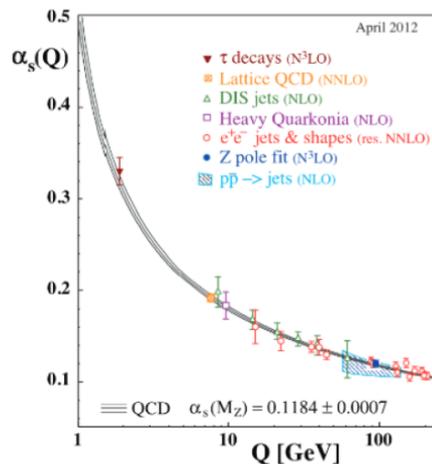


Figure 2.3: Summary of measurements of α_s as a function of the respective energy scale Q . Figure from [14].

2.2 Quark-Gluon Plasma

With everything discussed up to this point, it is possible now to introduce the main object of study of the field of heavy-ion physics: The state of matter called quark-gluon plasma (QGP), that forms under extreme conditions of pressure or temperature. For example in a heavy nuclei collision in the Large Hadron Collider (LHC) at CERN, the European Organization for Nuclear Research located in Geneva in the France–Switzerland border, or the Relativistic Heavy-Ion Collider (RHIC) at BNL, the Brookhaven National Laboratory located in Upton, Long Island in the United States.

In the QGP the partons are locally deconfined, which means that their dynamics within the plasma is not limited by the color confinement property. It may seem like a contradiction, however the QGP as a whole must be color neutral and the state itself is not directly measurable. However throughout the years several observables were proposed as QGP signatures and measured in heavy-ion collisions [15, 16], for example anisotropic flow, jet quenching, number of constituent quark

scaling and the one signature that is the focus of this work: Strangeness enhancement. Therefore the detection of strange particles is one way of probing such state of matter.

Results obtained with lattice QCD [17] calculations estimates a phase transition to the QGP, as illustrated in Figure 2.4 . In the regime of a heavy-ion collision the transition to QGP should happen at temperatures around 200 MeV. Which is in the same order of magnitude of twice the mass of the strange quark, therefore the QGP can thermally produce strange quarks through a process called gluon fusion [18, 19].

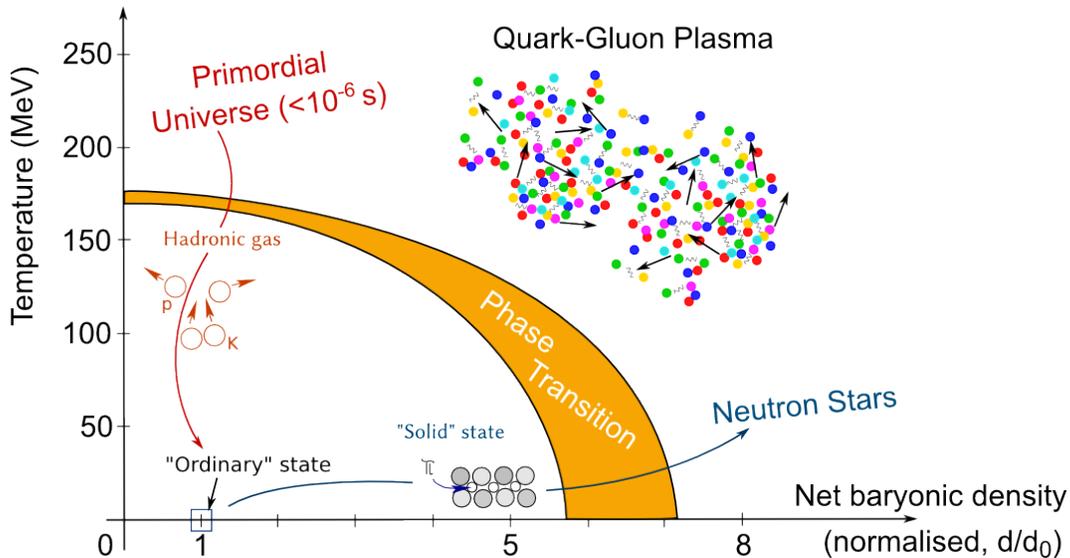


Figure 2.4: Phase diagram (temperature, net baryon density) of QCD matter, ranging from regular nuclear matter to Quark-Gluon Plasma. Figure from [20]

The thermal production of strange quarks inside a deconfined medium favors their combination into multi-strange baryons, as illustrated in figure 2.5. The ratio of measured multi-strange baryons to pions (the most abundantly produced hadron) as a functions of charged particle multiplicity defines the observable known as strangeness enhancement. Charged particle multiplicity is denoted by $\langle dN_{ch}/d\eta \rangle_{|\eta|<0.5}$, that is the average counts of charged particles produced inside a pseudo-rapidity window. That number is associated with the number of participant nucleons in a heavy-ion collision, that gives the expected collective effect to allow a phase transition.

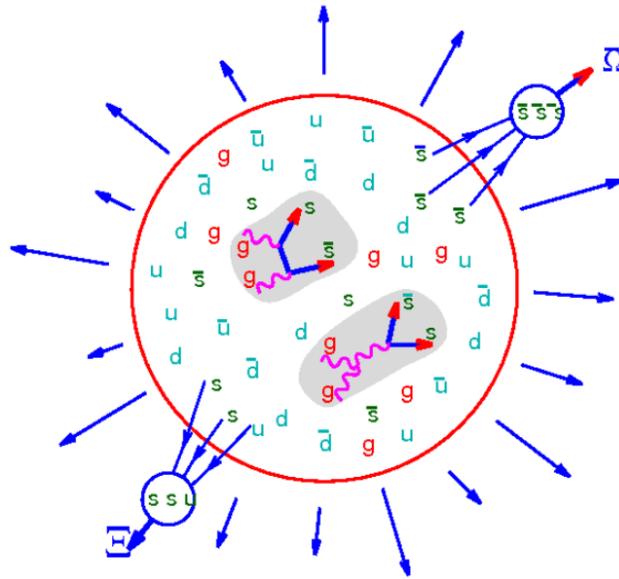


Figure 2.5: Multistrange (anti)baryons as signature of QGP, Figure from [21]

Figure 2.6 shows the strangeness enhancement observable, measured in several collisions systems, however ALICE still haven't published it in Pb-Pb at $\sqrt{s_{NN}}=5.02$ TeV, which is the collision system reported in this thesis. An interesting aspect of figure 2.6 is that it also shows an enhancement with respect to the multiplicity of proton-proton collision where QGP is not expected to happen.

That is the context and background that motivates this work of exploring new tools for the analysis of experimental data in the ALICE collaboration. Seeking to improve the methodology of strangeness reconstruction, but first we need to introduce the ALICE experiment and briefly explain how it acquires the data to be analyzed.

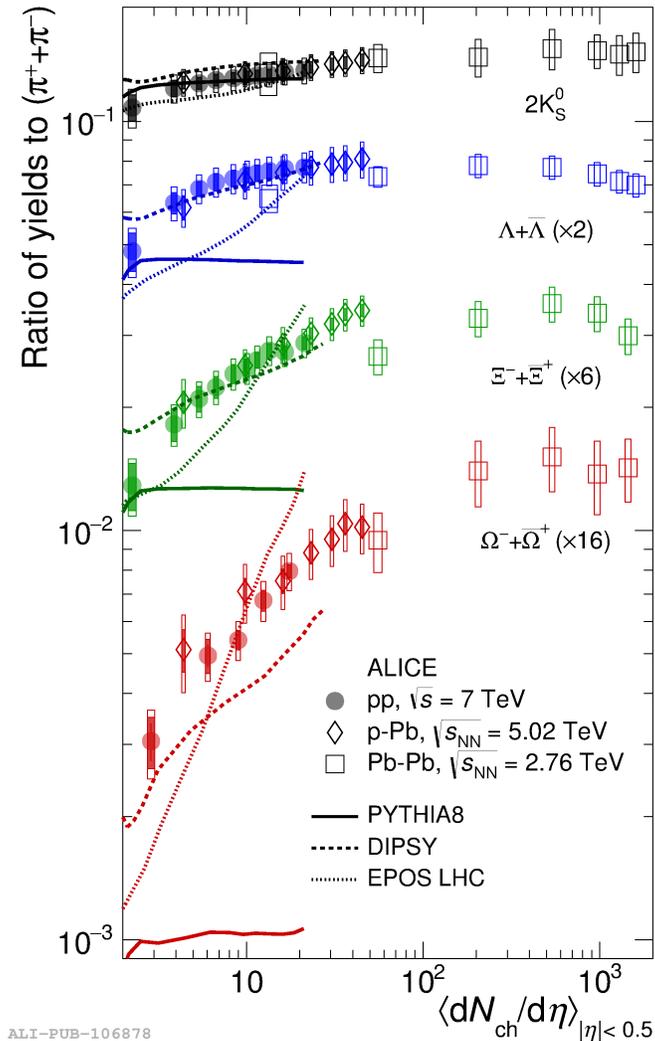


Figure 2.6: p_T -integrated yield ratios of strange and multi-strange hadrons to $(\pi^+ + \pi^-)$ as a function of $\langle dN_{ch}/d\eta \rangle$ measured in the rapidity interval $|y| < 0.5$. The empty and dark-shaded boxes show the total systematic uncertainty and the contribution uncorrelated across multiplicity bins, respectively. The values are compared to calculations from MC models [22, 23, 24], and to results obtained in Pb-Pb and p-Pb collisions at the LHC [25, 26, 27]. Figure from [28].

Chapter 3

A Large Ion Collider Experiment

This short chapter presents a description of the A Large Ion Collider Experiment (ALICE) and the introduction of some key aspects and variables in experimental heavy-ion collisions necessary for the following chapter about the methodology for the strangeness reconstruction.

ALICE is one of the four major experiments located around the LHC, the other three are ATLAS, CMS and LHCb. ALICE is a complex of detector designed for measuring the products of ultra-relativistic heavy-ion collisions.

3.1 Coordinate system and kinematic variables

ALICE uses right-handed orthogonal Cartesian coordinate system, with the origin at the beams interaction point. \hat{x} is perpendicular to the beam direction, aligned with the local horizontal and pointing to the center of the LHC. \hat{y} is perpendicular to \hat{x} and the beam direction and pointing upward. \hat{z} is parallel to the beam direction and points accordingly to the right-hand rule. ALICE's muon arm sits at negative \hat{z} . The azimuthal angle ϕ goes counter-clockwise from \hat{x} to \hat{y} , looking from positive \hat{z} . The polar angle θ goes from \hat{z} to the xy plane, also called transverse plane. The described coordinate system is depicted in figure 3.1 .

For the analysis it is convenient to look at the coordinate system as a transverse component and a longitudinal one. The most important transverse kinematic variable is the transverse momentum (p_T) defined by

$$p_T^2 = p_x^2 + p_y^2. \quad (3.1.1)$$

Since \hat{z} is the direction of the beam (longitudinal) every collision, also called event, has initial $p_T \approx 0$. Therefore the transverse plane is associated with particles produced by the collision.

The two main longitudinal variables are the rapidity (y)¹ and pseudo-rapidity (η) defined as follows

$$y = \frac{1}{2} \ln \left(\frac{E + p_z c}{E - p_z c} \right), \quad (3.1.2)$$

¹Rapidity is also represented by "y" and should not be confused with the \hat{y} direction of the coordinate system. In the definition of transverse momentum p_y is the momentum in the \hat{y} direction.

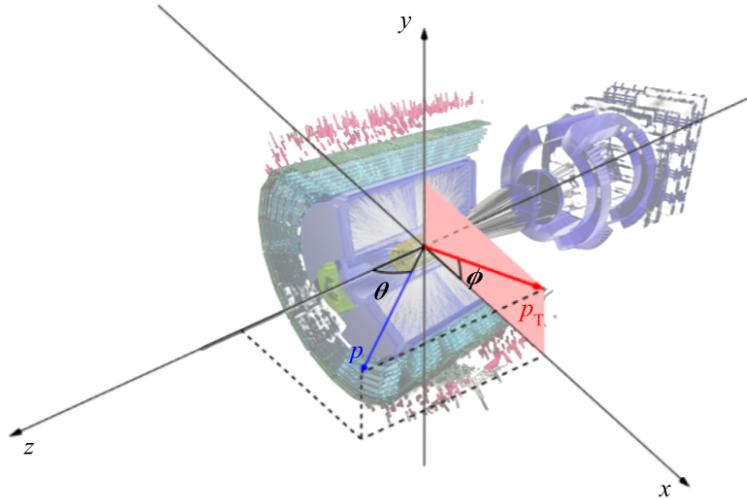


Figure 3.1: Coordinate system of ALICE.

$$\eta = \frac{1}{2} \ln \left(\frac{|\vec{p}| + p_z}{|\vec{p}| - p_z} \right) = - \ln \left[\tan \left(\frac{\theta}{2} \right) \right]. \quad (3.1.3)$$

It is worth noting that η is Lorentz invariant since it can be reduced to a function of θ only. And also, since

$$E^2 = (mc^2)^2 + (|\vec{p}|c)^2, \quad (3.1.4)$$

in the limit of high momentum, with respect to the laboratory frame, rapidity and pseudo rapidity are the same.

$$(|\vec{p}|c) \gg mc^2 \Rightarrow E \approx |\vec{p}|c \Rightarrow y \approx \eta. \quad (3.1.5)$$

With the introduced variables it is now possible to define one core observable of this work, the transverse momentum distribution, also called p_T spectrum, that can be denoted as

$$f(p_T) = \frac{1}{N_{ev}} \frac{d^2 N}{dp_T dy}. \quad (3.1.6)$$

It was chosen to already present it normalized by the number of events (N_{ev}). Such observable, essentially, tells us how many particles, per interval of p_T and in a given rapidity window (usually $|y| < 0.5$), are produced in a heavy-ion collision. It is worth noting that the transverse momentum spectra does not denote a continuous derivative in the sense of calculus, however each experimental point that builds up the distribution is given by the counts of particle in each p_T interval divided by Δp_T and all that is performed inside a rapidity window, given by the detector acceptance, and also divided by the window size Δy . Therefore, it represents a numeric derivative.

3.2 Measurements for strangeness reconstruction

A detailed description of the ALICE apparatus can be found in reference [29]. In this work the focus is going to be the simple description of some detectors and the set of measurements necessary to perform the strangeness reconstruction analysis and build the p_T spectra of the multi-strange baryons.

Central detectors

The two main central detectors in ALICE are the Inner Tracking System (ITS), which is a silicon detector, illustrated in figure 3.2. And the Time Projection Chamber (TPC), which is a gas detector, illustrated in figure 3.3. Together they are responsible for determine the primary vertex (PV), the point in space in which the initial Pb-Pb collision happened. They are also responsible for determining the trajectory of the particles produced in the events.

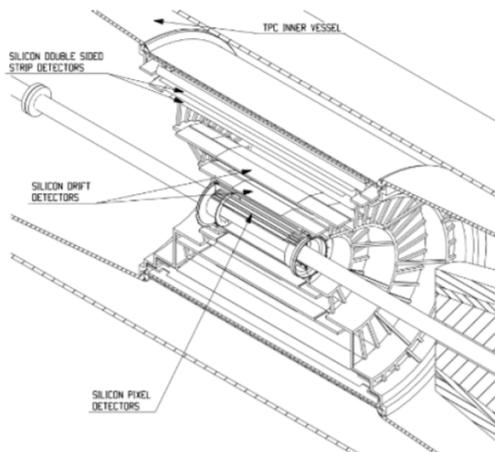


Figure 3.2: ALICE's Inner Tracking System (ITS). Figure from [30]

Another important feature of the TPC is to provide particle identification (PID) information based on the energy loss of a charged particle crossing it. It is necessary for the strangeness reconstruction analysis the identification of protons, kaons and pions.

Figure 3.4 presents how the the TPC performs PID. The x axis is the momentum p normalized by charge z , in units of the elementary electron charge e . The y axis is the energy loss inside the detector. The curves for each particle were obtained with superimposed Bethe-Bloch parameterization and are well discriminated [32].

Forward detectors

The main forward detectors are the V0A and V0C, which are scintillator detectors. They are responsible for the centrality estimation of Pb-Pb collisions. That is, essentially a proxy of the impact parameter.

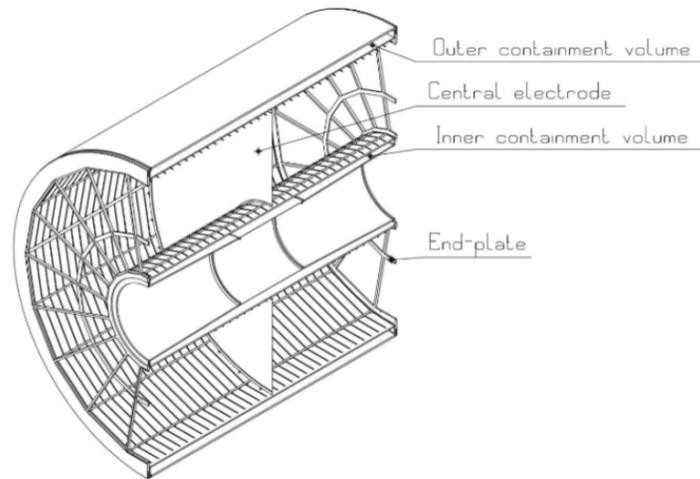
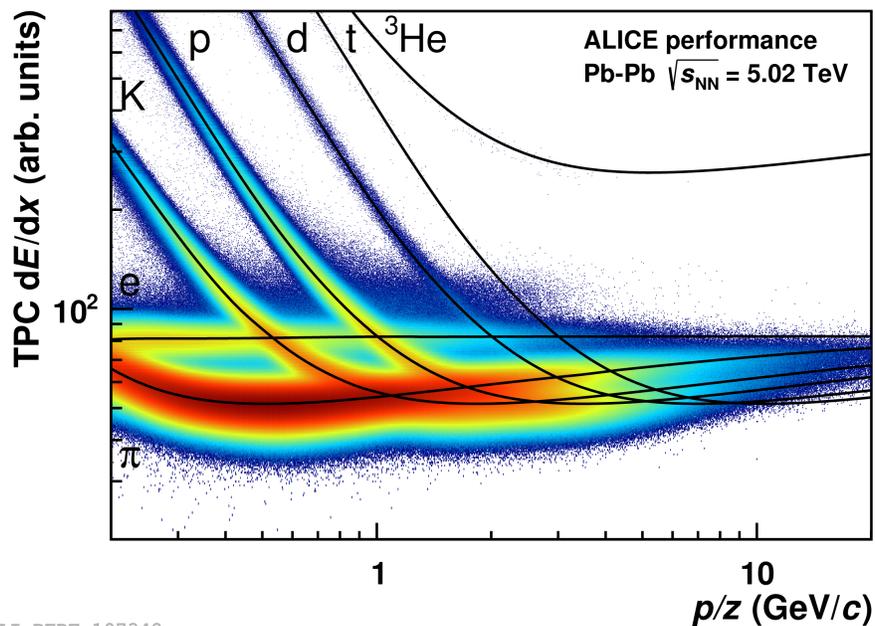


Figure 3.3: ALICE's Time Projection Chamber (TPC). Figure from [31]



ALI-PERF-107348

Figure 3.4: TPC dE/dx as a function of momentum with superimposed Bethe-Bloch lines for various particle species for RUN2 Pb-Pb low interaction rate runs. Figure from [32]

The centrality estimation is performed by splitting the events, in a given set, in classes based on the signal measured by the scintillator. The higher the amplitude, the more central the collision was and then the event tends to be placed in a lower centrality class. 0-10% is the lowest centrality class used in this work, therefore it contain the subset of most central events. 80-90% was the higher centrality class used in this work, therefore it contain the subset of least central events, or most peripheral ones. Figure 3.5 presents the centrality estimation procedure [33].

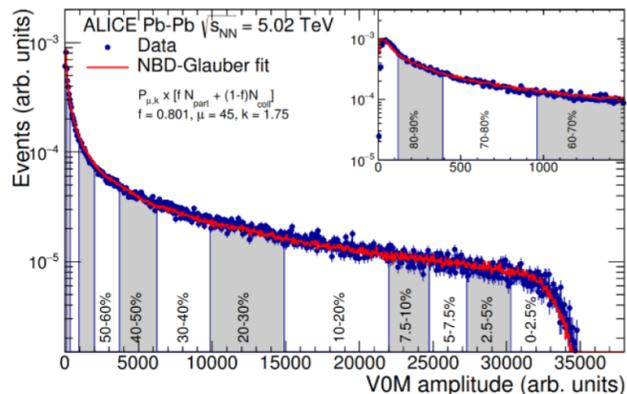


Figure 3.5: Distribution of the sum of the amplitudes in the V0 detectors for Pb-Pb at $\sqrt{s_{NN}}=5.02$ TeV. The threshold boundaries that define each centrality class are represented by the vertical lines. The inset shows a zoom in the most peripheral region. The Glauber Negative Binomial Distribution (NBD-Glauber) fit is shown in red. Figure from [33]

The centrality classes can also be mapped into charged particle multiplicity [34, 35, 36], as presented in table 3.1.

Table 3.1: Charged particle multiplicity in each centrality class

Centrality class	$\langle dN_{ch}/d\eta \rangle_{ \eta <0.5}$
0-10%	1764 ± 49.75
10-20%	1180 ± 31
20-30%	786 ± 31
30-40%	512 ± 15
40-50%	318 ± 12
50-60%	183 ± 8
60-70%	96.3 ± 5.8
70-80%	44.9 ± 3.4
80-90%	17.52 ± 1.9

With all that in hand, we use the PID information from TPC, the PV information and tracking from ITS and TPC to calculate geometrical information about the identified particles that have fundamental importance in the strangeness reconstruction procedure. Since we count the multi-strange baryons by looking into possible decay products, using the geometry of the tracks to distinguish non-correlated tracks to actual decay products.

The analysis is done separately in every centrality class that we have access to, thanks to the V0 detectors. By obtaining the corrected p_T spectra in every centrality class, we can extract to total production yield of multi-strange baryons as a function of $\langle dN_{ch}/d\eta \rangle_{|\eta|<0.5}$, using table 3.1, and by dividing it by the production yield of pions [37] we can obtain the strangeness enhancement observable.

Chapter 4

Data analysis methodology

This chapter starts with a brief introduction of the data processing from the raw signal of the detectors to the actual data set in which the analysis was performed in this work. Then it covers the concepts of strangeness reconstruction, as well as the steps necessary to obtain the results of multi-strange baryon production from ALICE data.

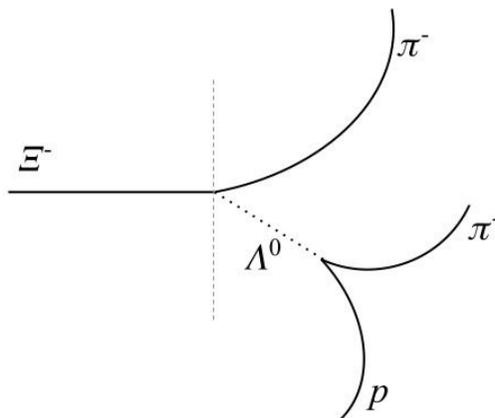
4.1 ALICE data flow

ALICE has a complex online-offline framework for extract, process and storage data [38]. The read out of raw data from the detectors is performed by the Data Acquisition system (DAQ) and gathering the information of the other four online systems: Central Trigger Processor (CTP), Detector Control System (DCS), Experiment Control System (ECS) and High-Level Trigger (HLT), the framework called Shuttle, process the data and store it in the ROOT format.

ROOT [39] is a C++ framework developed at CERN to, essentially, store and analyze data in an efficient way. The data stored by Shuttle is then curated by the ALICE Data Preparation Group (DPG) and becomes available for analyzers, usually in the format of ESD (Event Summary Data), that contains the information about the primary vertex, tracks and multiplicity, that we actually need to perform the strangeness reconstruction analysis.

4.2 Strangeness Reconstruction

Strangeness reconstruction is the data analysis procedure used to count the production of strange particles in high energy collisions. In this work the particles of interest were the multi-strange baryons (Ξ^- , Ξ^+ , Ω^- and Ω^+). These particles are unstable and decay inside the ALICE detector in the presence of a magnetic field. The particles produced in these decays are called daughters and their tracks are measured by the ITS and TPC. The geometry of the decay is called cascade topology because it consists of a decay into a charged meson and neutral baryon that also decays into a V0 topology, that is a charged meson and a charged baryon. Figure 4.1 illustrates the process for for Ξ^- .

Figure 4.1: Cascade decay topology of Ξ^- .

The decay channels for each particle are listed below with the respective branching ratios (BR) in table 4.1. In the cascade decay topology 10 variables can be defined as presented in figure 4.2 and they are used in the reconstruction analysis. Five of these variables are associated only with the V0 decay and together with the other five they compose the set called topological variables.

Table 4.1: Decay channels of strange baryons

Cascade	BR	V0	BR
$\Xi^- (dss) \rightarrow \pi^- (\bar{u}d) + \Lambda^0 (uds)$	100%	$\Lambda^0 (uds) \rightarrow p (uud) + \pi^- (\bar{u}d)$	64%
$\Xi^+ (d\bar{s}\bar{s}) \rightarrow \pi^+ (u\bar{d}) + \bar{\Lambda}^0 (\bar{u}\bar{d}\bar{s})$	100%	$\bar{\Lambda}^0 (\bar{u}\bar{d}\bar{s}) \rightarrow \bar{p} (\bar{u}\bar{u}\bar{d}) + \pi^+ (u\bar{d})$	64%
$\Omega^- (sss) \rightarrow K^- (\bar{u}s) + \Lambda^0 (uds)$	68%		
$\Omega^+ (\bar{s}\bar{s}\bar{s}) \rightarrow K^+ (u\bar{s}) + \bar{\Lambda}^0 (\bar{u}\bar{d}\bar{s})$	68%		

To reconstruct these particles we select various combinations of three identified tracks, called candidates. For example, for the Ξ^- analysis each candidate is a set of $[\pi_b^-, \pi_v^-, p_v]$, that can correspond to a secondary vertex. Where π_b^- denotes a pion that, in this candidate, is considered to be the bachelor track and π_v^- and p_v , in this candidate, are considered to be from the V0 decay. Then we determine the invariant mass of each candidate and populate a histogram with such values, often called invariant mass spectra. Then count the strangeness production by extracting the signal of the invariant mass spectra.

However, the presence of numerous candidates that are not associated with the decay process introduces a significant background in the invariant mass spectra. To address this challenge, selection cuts based on topological variables are commonly employed to suppress the combinatorial background. This procedure is going to be called standard analysis in this work. As these topological parameters often exhibit correlations, there exists a considerable potential for improvement through the use of Machine Learning (ML) analysis techniques solving a binary classification problem: "Is this candidate signal (an actual multi-strange baryon decay) or is it combinatorial background?".

the listed cut in the cosine of this angle [40]. All other selection criteria are detector related and are applied to assure good track quality.

The standard analysis of topological cuts consists in populate the invariant mass spectra only with candidates with topological variables within the values in table 4.3 .

Table 4.3: Topological cuts from standard analysis.

Topological variable	Cut value for $\Xi(\Omega)$
DCA(V0 tracks)	< 1.0 cm
DCA(V0 negative track - PV)	> 0.2 cm
DCA(V0 positive track - PV)	> 0.2 cm
V0 transverse decay radius	> 3.0 cm
V0 $\cos(PA)$	$> 0.95^\dagger$
DCA(bachelor - V0)	$< 1.0(0.6)$ cm
DCA(bachelor - PV)	> 0.1 cm
Cascade transverse decay radius	$> 1.2(1.0)$ cm
Cascade $\cos(PA)$	$> 0.95^\dagger$
DCA(V0 - PV)	> 0.1 cm
V0 Invariant Mass Window	± 0.005 GeV/c ²

\dagger : These are baseline values for cosine of pointing angle selections that can be optimized with a p_T -dependent parametrization presented in figure 4.3 .

Note that there is an additional variable in the table 4.3 , V0 Invariant Mass Window, that is actually not topological. However it is an applied cut, given that the cascade decays produces a Λ^0 particle, it is expected that the invariant mass of the V0 tracks are close to m_{Λ^0} .

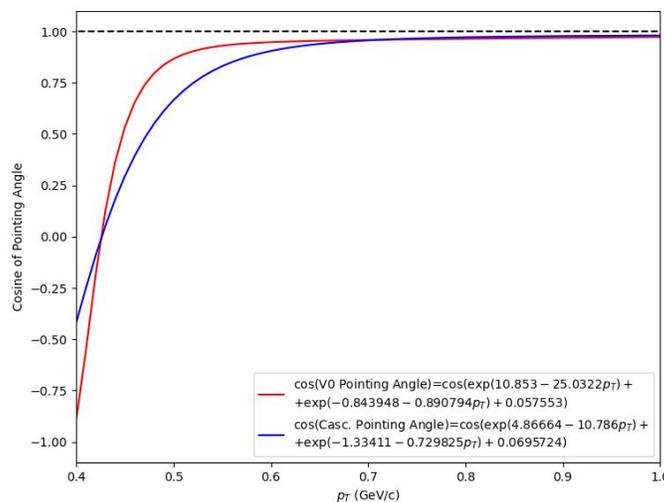


Figure 4.3: Parametrization of cosine of pointing angle cuts as a function of p_T . Cascade $\cos(PA)$ is shown in red and V0 $\cos(PA)$ is shown in blue.

4.3 Machine Learning analysis

To perform the Machine Learning analysis a training, validation and test set were built from Monte Carlo (MC) simulated data. Eighth models were trained in total. One for each particle, four neural networks and four boosted decision trees. The models for Ξ^- and Ξ^+ trained with around 200,000 candidates each. Evenly divided in signal and background respecting the transverse momentum and centrality distributions of the MC data set. For Ω^- and Ω^+ it were around 50,000 candidates each also evenly divided in signal and background respecting the transverse momentum and centrality distributions of the MC data set. Each validation set had a quarter of the training set size also respecting the mentioned distributions.

The neural networks were built with the TensorFlow library [41] with the architecture of one input layer with 11 neurons, four inner layers with 256, 64, 16 and 4 neurons respectively and one output layer with 1 neuron. The activation function of all neurons was the sigmoid function

$$f(x) = (1 + e^{-x})^{-1}. \quad (4.3.1)$$

The cost function was the binary cross entropy, and the optimizer was ADAM with $\epsilon = 10^{-8}$ and learning rate $lr = 10^{-4}$ [42].

The boosted decision trees were built with the XGBoost library [43] with the architecture of 100 estimators of depth 2. The cost function, called 'objective' in XGBoost, was the binary logistic function.

All the models receives as input the 11 variables from table 4.3 . That means the task performed by the models is to calculate the probability that the tracks, from a given candidate, were originated in a cascade decay. Based on topological information.

The validation sets were used during the training process to avoid over fitting. Every model were trained until convergence. Figure 4.4 shows the distributions of the output P from the Ξ^- Neural Network, in the respective validation set. Since we have access to the known answers in the validation set as well it is possible to see how the model splits the classes.

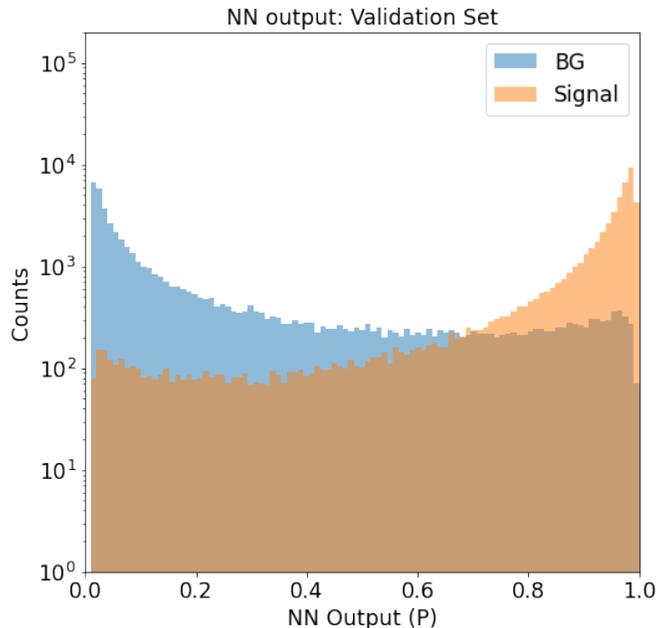


Figure 4.4: Prevision distribution of the Ξ^- Neural Network for the background data (blue) and signal data (orange) in the validation set.

4.3.1 Training results in the test set

After training, the models were used to predict the test set. Figure 4.5 presents an output in the region $0.8 < p_T < 1.0$ GeV/c. Low transverse momentum regions are more challenging due to the presence of higher background counts and a special attention was given to it in the analysis.

The standard analysis were also applied to the same data set for comparison. The ML classification has a freedom of choice, which is the minimum value of P to classify a given candidate as a the particle of interest, this value is called classification threshold. So only candidates with P greater than the threshold populates the invariant mass spectra. Traditional analysis of classification typically uses $P > 0.5$, but in this work we delve into the implications of changing such parameter.

Figure 4.6 shows how the counts of signal and background change with the classification threshold and compare it to the signal and background counts obtained with the standard analysis. The background counts falls faster then the signal ones which is expected given the output distributions. What is interesting is what is happening in the intersections with the standard analysis.

When both methods yields the same amount of background, at classification threshold 0.66, the signal counts obtained with machine learning is higher than the standard one. When both methods yields the same amount of signal, at classification threshold 0.86, the background counts obtained with machine learning is higher than the standard one.

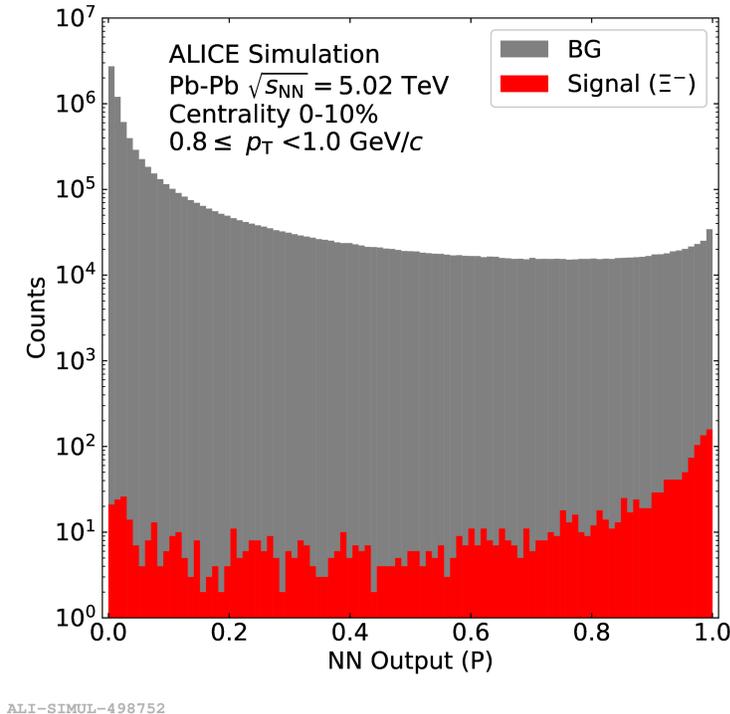


Figure 4.5: Neural network output distribution for signal (in red) and background (in gray) in the test set for $0.8 \leq p_T < 1.0 \text{ GeV}/c$.

One important variable in this type of analysis is the significance

$$Significance \equiv \frac{Signal}{\sqrt{Signal + 2 \times Background}}, \quad (4.3.2)$$

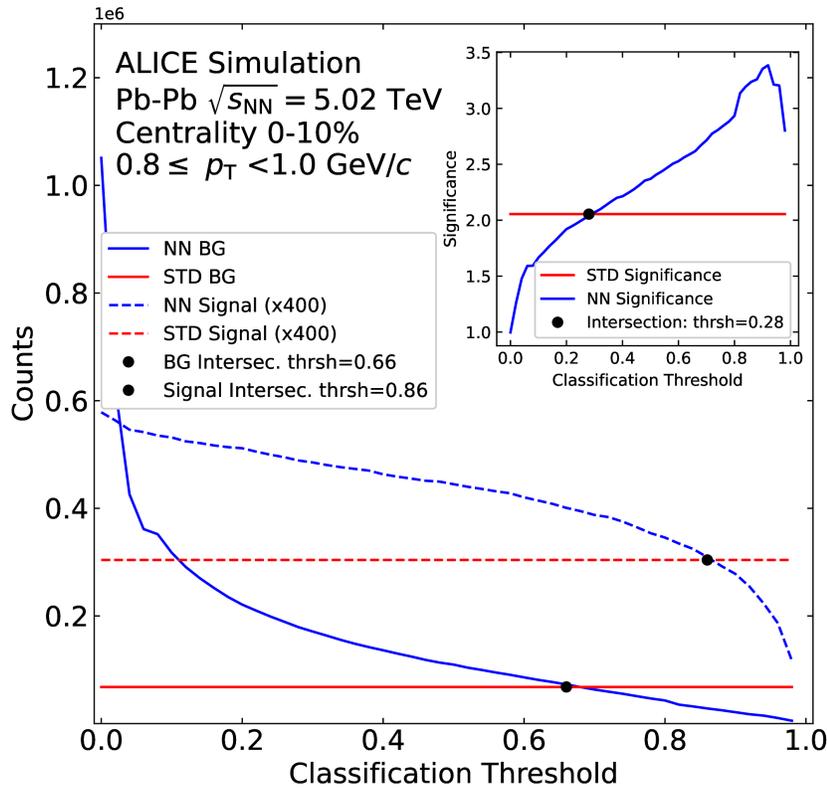
that is shown in the inset plot in figure 4.6 and from the intersection with the standard value, at classification threshold 0.28, until the classification threshold limit 1.0, the machine learning presents improvement.

Another important variable is the efficiency, which is simply the amount of signal classified correctly divided by the total counts of signal in the data set. Therefore the signal counts intersection is the same as the efficiency intersection, since both analysis were performed in the same data set.

Since efficiency decreases with threshold, significance increases and the significance intersection occurs before the efficiency intersection, every choice of threshold between them results in a simultaneous improvement of the machine learning in both variables with respect to the standard analysis.

Figure 4.7 shows the intersection points of signal, background and significance across transverse momentum. In each p_T interval any choice of threshold between the blue and black points provides simultaneous improvement.

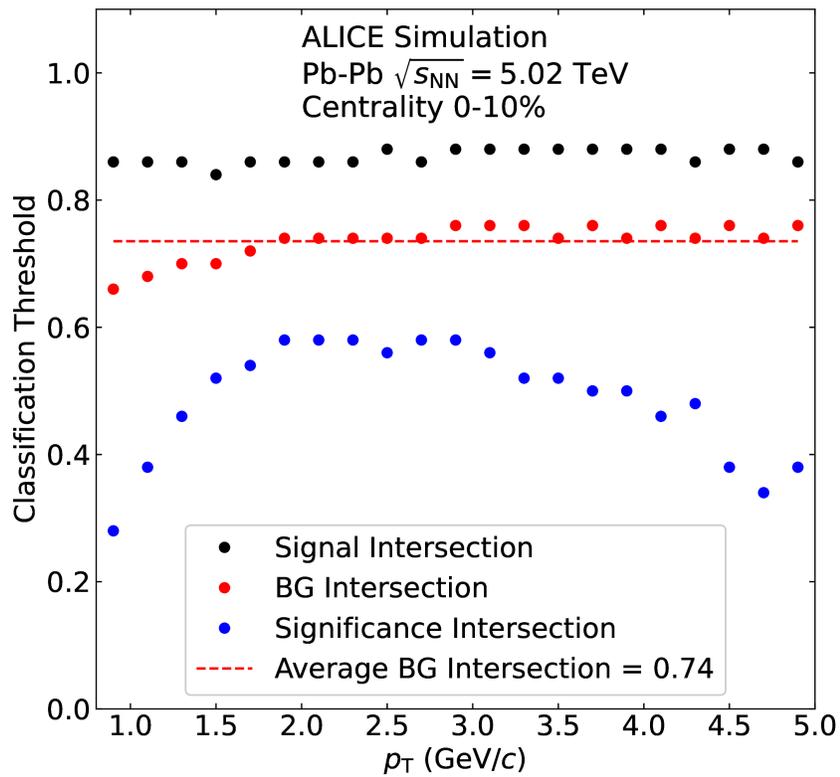
The average background intersection threshold, of 0.74, was chosen to compare the efficiency



ALI-SIMUL-498756

Figure 4.6: Classification threshold analysis for $0.8 \leq p_T < 1.0$ GeV/c. Signal and background counts from the standard analysis are shown in red. Signal and background counts from the ML analysis are shown in blue. Full lines indicates background, dashed lines indicates signal. Signal intersection at threshold=0.86 and background intersection at threshold=0.66. Inner graphic shows significance as a function of classification threshold for both methods and the intersection at threshold=0.28.

and significance of the methods as a function of p_T as reported in figures 4.8 and 4.9. That also shows ± 0.1 variations in the chosen threshold. For efficiency the overall gain is 22% and for significance 19%, being 37% in the lowest p_T bin.



ALI-SIMUL-498760

Figure 4.7: Classification thresholds of the intersections of signal counts (in black), background counts (in red) and significance (in blue) as a function of p_T . The dashed line indicates the average classification threshold for background intersections.

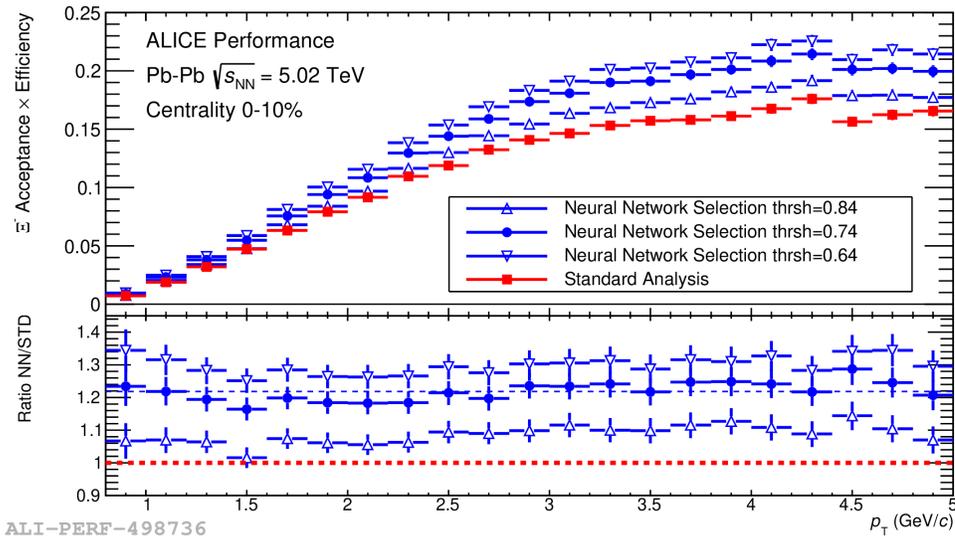


Figure 4.8: Efficiency of reconstructing Ξ^- using an artificial neural network (in blue) in comparison with the standard analysis of topological cuts (in red). Open points indicate the efficiencies for ± 0.1 of the classification threshold choice. Below efficiencies are the ratio from the standard method. The average gain is 22%.

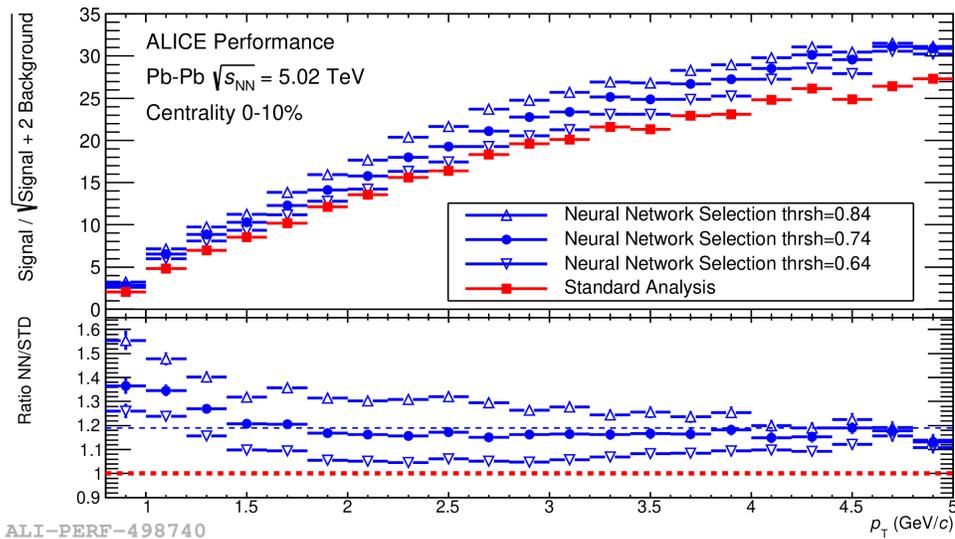


Figure 4.9: Significance of reconstructing Ξ^- using an artificial neural network (in blue) in comparison with the standard analysis of topological cuts (in red). Open points indicate the significance for ± 0.1 of the classification threshold choice. Below significance are the ratio from the standard method. The average gain is 19% and 37% in the lowest p_T bin.

4.4 Analysis in the complete data set

With the trained machine learning models the next step of the methodology consists in classify the complete data set of Pb-Pb collision, at center of mass energy $\sqrt{s_{NN}} = 5.02$ TeV, measured with ALICE in 2015. It consists of 42 million events, that can not be simply downloaded and analyzed.

ALICE has its own framework to process and analyze this data, called *AliPhysics* that runs in the CERN computing grid. However it did not have any support for machine learning techniques. One challenging part of the job was to write the necessary code that could operate exactly like the trained machine learning models and incorporate it into the *AliPhysics* framework. The framework is C++ based and three classes were developed to allow the usage of machine learning in strangeness reconstruction: *AliMachineLearning*, *AliBDT* and *AliNeuralNetwork*. The codes are now part of the official repository of the ALICE collaboration and are presented here in Appendix B.

The classes are rather simple, however they operate exactly as needed. An intermediary step is needed since the models were trained in python. One must first save the structure of the trained model from python and then place it in the appropriate data structure using the ROOT framework to be read and used by the classes.

The classes are then used by the official strangeness analysis task that gives as output a set of three dimensional histograms containing counts in bins of invariant mass, p_T and centrality. For each particle, there is one histogram filled with counts from the standard analysis and 50 histograms filled with counts from the neural network analysis and 50 histograms filled with counts from the boosted decision tree analysis. Each of those 50 histograms are filled with different values of threshold, varying from 0 to 0.98, with steps of 0.02.

Nine centrality classes were studied. For each of them there are at least 22 (6) p_T bins for Ξ^\pm (Ω^\pm) that requires a signal extraction from the invariant mass histogram. Figure 4.10 gives an example of invariant mass histogram comparing the one obtained with the neural network with the standard one.

Signal Extraction

The signal extraction is performed in two steps, in the first one a fit of a Gaussian plus a first order polynomial is performed in the region $(m-0.01, m+0.01)$, where m denotes the PDG mass of the particle [14]

- $m = 1.32171$ GeV/ c^2 for Ξ^\pm
- $m = 1.67245$ GeV/ c^2 for Ω^\pm

The second step uses the fitted Gaussian's mean (μ) and standard deviation (σ) to adjust a second order polynomial in the region $(\mu - 6\sigma, \mu - 3\sigma) + (\mu + 3\sigma, \mu + 6\sigma)$, that is ignoring the peak region. That is consider the background function (f_{BG}). The extracted signal is then

$$S_{ext} \equiv \int_{\mu-3\sigma}^{\mu+3\sigma} H dx + \int_{\mu-3\sigma}^{\mu+3\sigma} f_{BG} dx, \quad (4.4.1)$$

where x simply denotes the horizontal component of the invariant mass spectra, and H denotes the entire invariant mass histogram, therefore it's integral is simply the sum of the counts within the peak region divided by the invariant mass bin size (Δx).

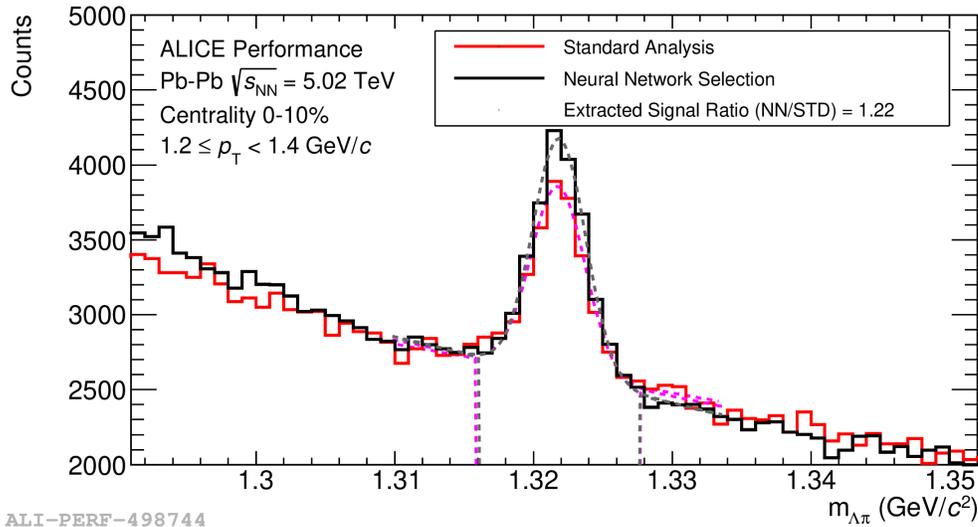


Figure 4.10: Invariant mass spectrum of Ξ^- for $1.2 \leq p_T < 1.4$ GeV/c in real data reconstructed by the neural network analysis (in black) in comparison with the spectrum reconstructed by the standard analysis (in red). Dashed curves indicate the fits performed by signal extraction. Ratio of the extracted signals show a 22% improvement due to the use of neural network.

Signal Extraction fine tuning

The lowest p_T bin in the standard analysis is $0.8 \leq p_T < 1.0$ GeV/c for the Ξ^\pm and $1.2 \leq p_T < 1.6$ GeV/c for the Ω^\pm . However we believed that it would be possible to extend the limit in the machine learning analysis given the improvements in efficiency and significance. And turns out, it is possible, however a fine tuning in the signal extraction procedure was needed to effectively count the Ξ^\pm in $0.6 \leq p_T < 0.8$ GeV/c and the Ω^\pm in $0.8 \leq p_T < 1.2$ GeV/c.

The fine tuning procedure consists in defining a function that is the sum of a Gaussian plus a second order polynomial. The μ and σ are fixed from the first fit. The entire histogram is adjusted without ignoring the peak region, and with a larger fit range ($\mu - 12\sigma, \mu + 12\sigma$). The new f_{BG} is building using the found parameters for the second order polynomial. Figure 4.11 illustrates the procedure and compares it with the previous described one.

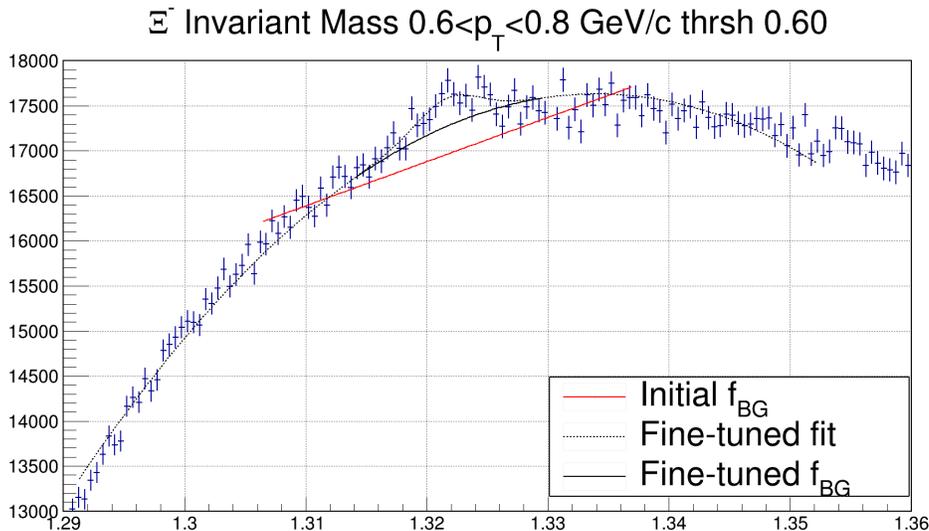


Figure 4.11: Invariant mass spectrum of Ξ^- for $0.6 \leq p_T < 0.8$ GeV/c in real data reconstructed by the neural network analysis. (in black) in comparison with the spectrum reconstructed by the standard analysis (in red). Dashed curves indicate the fits performed by signal extraction. Ratio of the extracted signals show a 22% improvement due to the use of neural network.

The extension of the inferior p_T limit was possible for every centrality class in the Ξ^\pm analysis and in centrality classes from 0-40% in the Ω^\pm analysis.

Corrected p_T spectra

The signal extraction provides particles counts per p_T bin separated into centrality classes. To compute the transverse momentum distribution of the particles it is also necessary to apply the efficiency correction factor. Bringing back the expression from previous chapter

$$f(p_T) = \frac{1}{N_{ev}} \frac{d^2 N}{dp_T dy} \quad (4.4.2)$$

is the value of the corrected p_T spectra at each p_T bin.

$$N = \frac{S_{ext}}{e} \quad (4.4.3)$$

is given by the extracted signal S_{ext} corrected by the efficiency e .

To compute the efficiency we apply the methodology used to build the invariant mass spectra to a so called associated Monte Carlo data set, built to reproduce well enough the observables of the data set that are being analyzed. In the associated Monte Carlo we simply count how many particles of interest are kept by the methodology, and the ratio with the total number of particles generated in the Monte Carlo gives the efficiency.

Then in each point of the corrected p_T spectra N is divided by the size of the respective p_T bin Δp_T and also by $\Delta y = 1$, since the acceptance rapidity region is $|y| < 0.5$.

Extrapolation to low p_T

To account for the total particle production yield we need to estimate how many particles are produced between $p_T = 0$ and the lowest p_T bin of the spectra. This procedure is called extrapolation to low p_T and was performed with a blast-wave fit of the spectra [44]. Which is a phenomenological description for the p_T spectra produced in heavy-ion collisions depending on collective variables, given by

$$\frac{1}{p_T} \frac{dN}{dp_T} \propto \int_0^R r dr m_T I_0 \left(\frac{p_T \sinh \rho}{T_{kin}} \right) K_1 \left(\frac{m_T \cosh \rho}{T_{kin}} \right), \quad (4.4.4)$$

where m_T is the transverse mass, I_0 and K_1 are modified Bessel functions, R is the radius of the medium,

$$\rho = \tanh^{-1} \beta_T \quad (4.4.5)$$

is the velocity profile, where

$$\beta_T = \left(\frac{r}{R} \right)^n \beta_s \quad (4.4.6)$$

is the transverse velocity distribution, parameterized by the surface velocity β_s .

After this fit we can integrate the obtained function from $p_T = 0$ to the lowest p_T bin of the spectra, to count the production in the non-measured region. Associated with each point of the spectra and to the extrapolation there are statistical uncertainties and a series of systematic uncertainties that must be calculated. This particular discussion is the purpose of the next chapter.

Chapter 5

Computation of Uncertainties

It was shown that the ML based methodologies simultaneously improved the efficiency and significance in the reconstruction of multi-strange baryons. As well as enabled the access to an unprecedented experimental data point in the low transverse momentum region of the corrected p_T spectra. In this chapter it is discussed the implications of such improvements by evaluating and comparing the statistical uncertainties, and the sources of systematical uncertainties of the corrected p_T spectra that are affected by the change in the analysis methodology.

5.1 Statistical uncertainties

The statistical uncertainties are originated in the invariant mass spectra and therefore propagated through the procedure of signal extraction described in the previous chapter. The improvements due to ML analysis reflects in the statistical uncertainty of the corrected spectra as shown in figure 5.1 that presents the ratio between statistical uncertainties of the Neural Network analysis by the standard analysis in the 0-10% centrality class for Ξ^- .

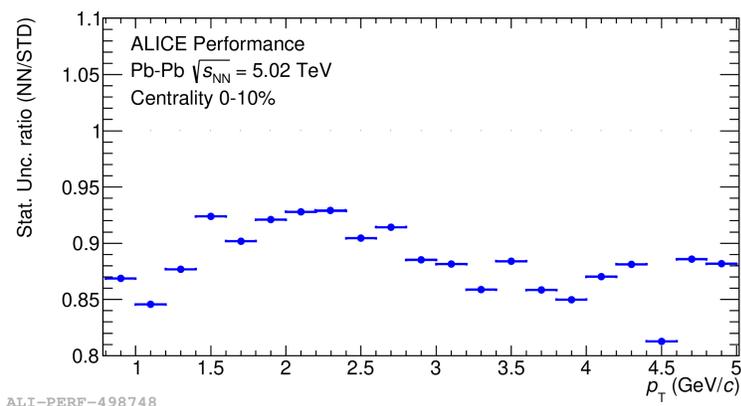


Figure 5.1: Ratio between the statistical uncertainty of the Ξ^- corrected p_T spectra obtained with the neural network analysis by the one obtained with the standard analysis in the 0-10% centrality class.

5.2 Systematical uncertainties

There are several sources of systematic uncertainties that can be accounted for. They were studied in detail in a previous work [40]. Here are presented the two main sources of systematic uncertainties of the p_T spectra that are affected by the change in methodology and the two sources of systematic uncertainties of the extrapolation to low p_T procedure that are the dominant terms in the total uncertainty of the production yield.

5.2.1 Candidate selection

Essentially the ML analysis substitutes the topological cuts from the the standard analysis by a cut in the probability output of the ML, called selection threshold. What is typically done to evaluate the systematic of such a selection is vary each topological cut individually by an arbitrary amount, usually 10%, and take the difference in extracted signal of each variation and sum them in quadrature. To reproduce this idea in the ML analysis it was taken in consideration the invariant mass spectra in a stable region in threshold selection, from 0.4 to 0.8, with 21 steps of 0.02 in threshold. The signal of each of them were extracted and the average signal calculated. The systematic uncertainty was assumed to be the maximum difference between the average and a signal in the threshold region.

Tables 5.1 and 5.2 presents three p_T bins that were used to compare the systematic of the standard analysis with the ML ones. To illustrate the method described for computing the systematic of ML figure 5.2 shows the the uncertainty for Ξ^- in $0.8 < p_T < 1.0$ GeV/c of the Neural Network and 5.3 shows the the uncertainty for Ω^+ in $2.8 < p_T < 3.2$ GeV/c of the BDT, both compared to the standard analysis.

Table 5.1: Systematic Uncertainties for $\Xi^-(\Xi^+)$ in percentage (%)

	$0.8 < p_T < 1.0$ GeV/c	$2.4 < p_T < 2.6$ GeV/c	$4.8 < p_T < 5.0$ GeV/c
NN	7.7 (7.3)	1.5 (1.0)	2.5 (3.0)
BDT	9.1 (4.4)	1.3 (1.4)	1.3 (1.8)
STD	13.4 (10.5)	1.8 (1.7)	2.4 (2.5)

Table 5.2: Systematic Uncertainties for $\Omega^-(\Omega^+)$ in percentage (%)

	$1.6 < p_T < 2.0$ GeV/c	$2.8 < p_T < 3.2$ GeV/c	$3.6 < p_T < 4.0$ GeV/c
NN	1.9 (3.8)	8.6 (7.5)	5.0 (8.0)
BDT	4.7 (10.0)	7.5 (3.9)	4.5 (9.7)
STD	11.4 (11.1)	8.5 (9.2)	7.7 (16.6)

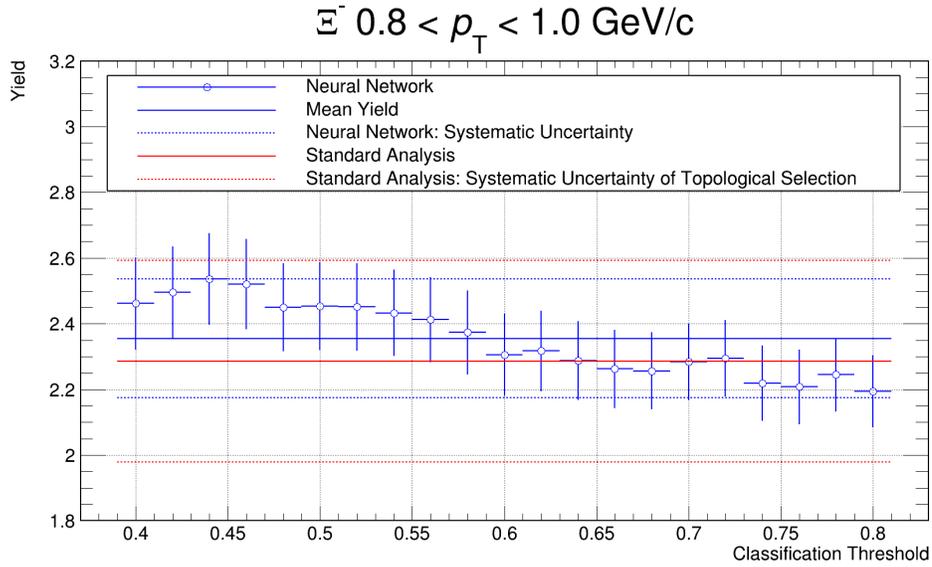


Figure 5.2: Ξ^- in $0.8 < p_T < 1.0 \text{ GeV}/c$ candidate selection systematic uncertainty of the NN (blue) and standard analysis (red). The blue markers are the NN extracted signal in each threshold interval. Blue line is the NN average extracted signal. Blue dashed lines are the estimate systematic for NN. Red line is the extracted signal from the standard analysis. Red dashed lines are the standard analysis systematic.

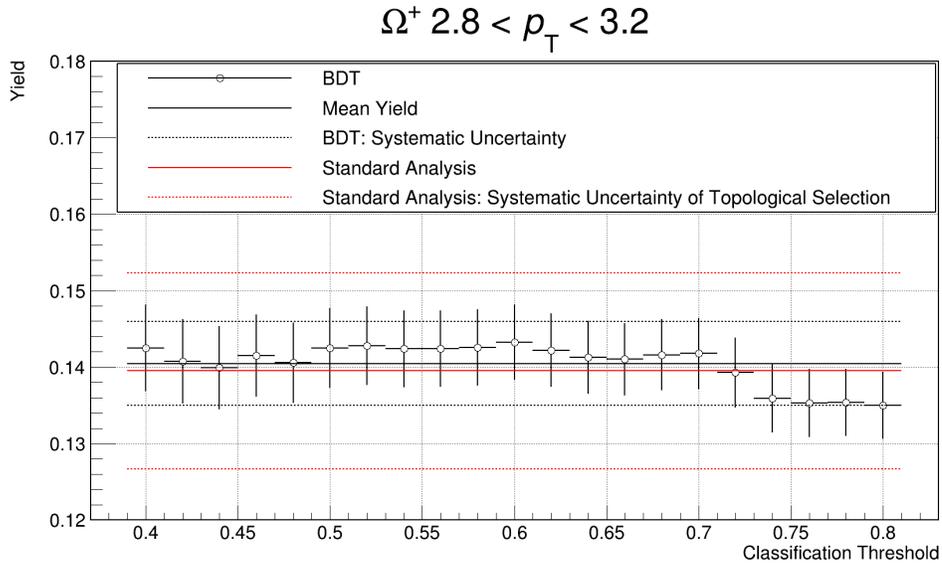


Figure 5.3: Ω^+ in $2.8 < p_T < 3.2 \text{ GeV}/c$ candidate selection systematic uncertainty of the BDT (black) and standard analysis (red). The black markers are the BDT extracted signal in each threshold interval. Black line is the BDT average extracted signal. Black dashed lines are the estimate systematic for BDT. Red line is the extracted signal from the standard analysis. Red dashed lines are the standard analysis systematic.

5.2.2 Signal extraction

The last source of systematical uncertainty calculated for the corrected p_T spectra was associated with the signal extraction procedure. First the peak region was redefined from $(\mu - 3\sigma, \mu + 3\sigma)$ to $(\mu - 2.75\sigma, \mu + 2.75\sigma)$ and $(\mu - 3.25\sigma, \mu + 3.25\sigma)$ and the redefinition that produced the higher difference in extracted signal was chosen. Then the function used to fit the background region was redefined from a second order polynomial to a first order and a third order and again the redefinition that produced the higher difference in extracted signal was chosen. The two contributions were then summed in quadrature to estimate the signal extraction systematic. For the fit of the third order polynomial the background region was chosen to be $(\mu - 7\sigma, \mu - 3\sigma)$ and $(\mu + 3\sigma, \mu + 12\sigma)$ to maintaining the exact same procedure in the reference analysis for comparison purposes [40].

5.2.3 Other sources of systematic uncertainties

There were also taken into account other sources of systematic uncertainties imported from previous analysis, presented in tables 5.3 and 5.4, that were assumed to don't change significantly by the change in the methodology.

Table 5.3: Systematic Uncertainties for $\Xi^- (\Xi^+)$ in percentage (%)

p_T (GeV/c)	0.8 to 1.0	2.4 to 2.6	4.8 to 5.0
Material budget	4.2 (5.3)	1.5 (1.7)	0.0 (0.0)
Transport code	0.0 (1.0)	0.0 (1.0)	0.0 (1.0)
Track selection	3.5 (2.4)	1.0 (1.5)	1.9 (1.9)
Particle Identification	0.0 (1.4)	0.9 (1.0)	1.0 (0.9)
Proper Lifetime	2.3 (1.7)	0.6 (0.5)	1.6 (1.4)
Event centrality selection	1.6 (1.7)	0.6 (0.7)	0.8 (0.8)

Table 5.4: Systematic Uncertainties for $\Omega^- (\Omega^+)$ in percentage (%)

p_T (GeV/c)	1.6 to 2.0	2.8 to 3.2	3.6 to 4.0
Material budget	2.2 (2.6)	1.6 (1.6)	0.0 (0.0)
Transport code	0.0 (1.0)	0.0 (1.0)	0.0 (1.0)
Track selection	0.9 (1.4)	1.3 (0.6)	1.2 (0.8)
Particle Identification	2.4 (4.6)	2.0 (2.3)	5.6 (4.8)
Proper Lifetime	2.1 (2.0)	0.7 (0.8)	0.4 (0.7)
Competing decay rejection	2.9 (2.2)	4.8 (5.1)	2.3 (1.8)
Event centrality selection	1.0 (1.0)	0.7 (0.8)	0.4 (0.7)

All the sources of systematic uncertainties of the corrected p_T spectra were summed in quadrature. Figures 5.4 and 5.5 present the corrected p_T spectra of $\Xi^- + \Xi^+$ and $\Omega^- + \Omega^+$ with all the described uncertainties comparing each ML model with the standard analysis in the 0-10% centrality class. The comparison of the spectra in the other centrality classes can be found in the Appendix A. The methods are in agreement within uncertainties.

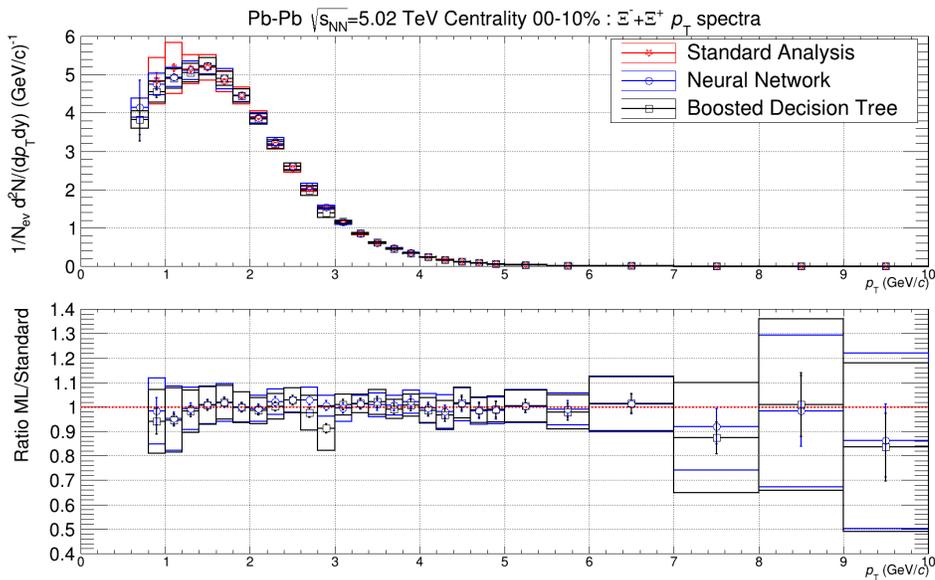


Figure 5.4: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 0-10% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

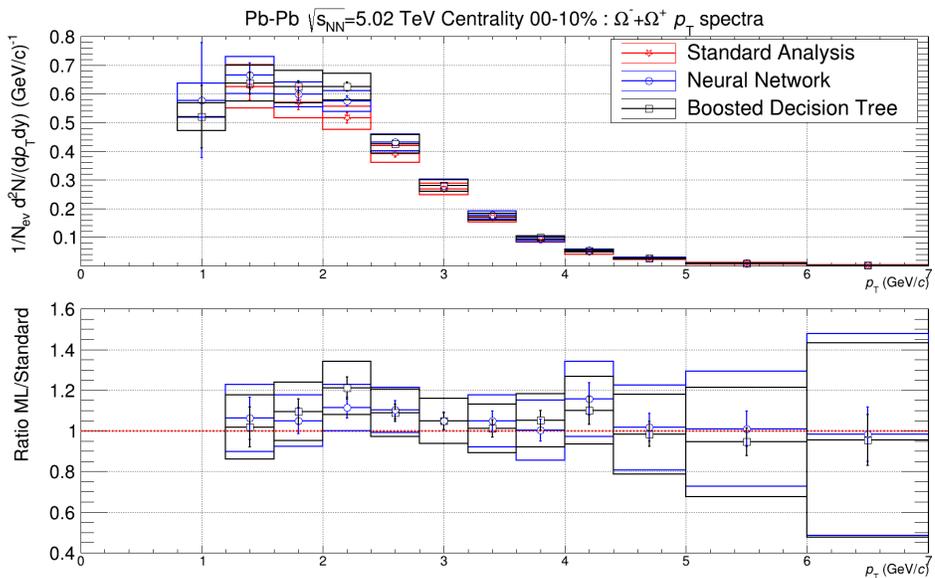


Figure 5.5: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 0-10% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

5.2.4 Extrapolation to low transverse momentum

The extrapolation to low p_T is a necessary technique to estimate the particle production yield from $p_T=0$ GeV/c to the lowest p_T experimental point. This is done by performing a fit of a theoretical function in the experimental points that extends itself all the way back to 0 GeV/c. The yield was estimated using the Blast-Wave function.

The systematic uncertainty of such procedure accounts for the total production yield of particles and not for the spectra itself. It is calculate using two separated methods and summing their uncertainties in quadrature.

High-Low variation fitting

The first source of systematic is estimated by dislocating every point in the corrected p_T spectra by its own total uncertainty in two ways. Let $f(p_T)$ be the value of each point in the corrected p_T spectra and $\sigma_f(p_T) = \sqrt{\sigma_{stat}(p_T)^2 + \sigma_{syst}(p_T)^2}$ the respective total uncertainty, that is the quadratic sum of the statistical and systematic uncertainties. The High variation is the spectra given by $f_{high}(p_T) = f(p_T) + \sigma_f(p_T)$ and the Low variation is is the spectra given by $f_{low}(p_T) = f(p_T) - \sigma_f(p_T)$. Both spectra were fitted with the Blast-Wave function and the one that gave the maximum difference in the extrapolated yield was chosen for the uncertainty.

Figures 5.6 to 5.9 shows the described procedure comparing the ML models with the standard analysis in the 0-10% centrality class.

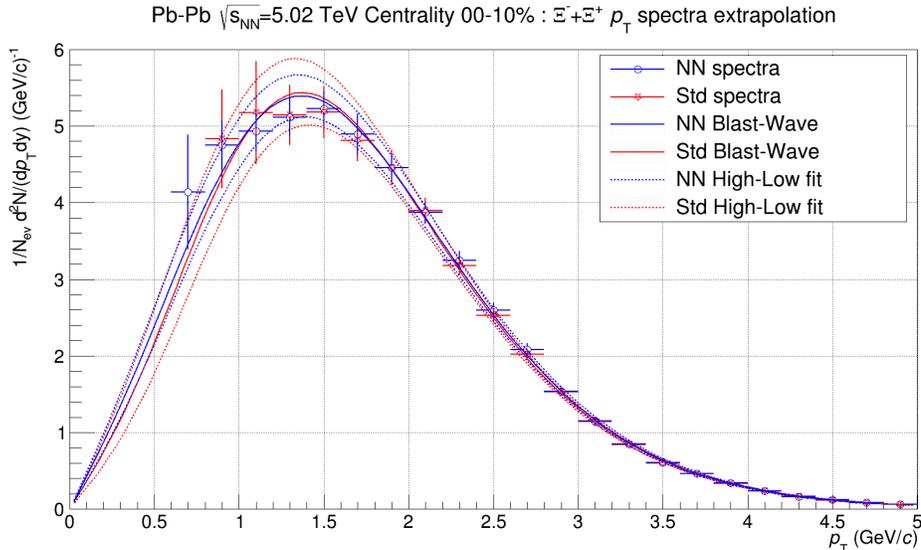


Figure 5.6: High-Low fit extrapolation to low p_T of $\Xi^- + \Xi^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the High-Low fits used to estimate the systematic uncertainty. Neural network analysis is shown in blue and the standard analysis in red.

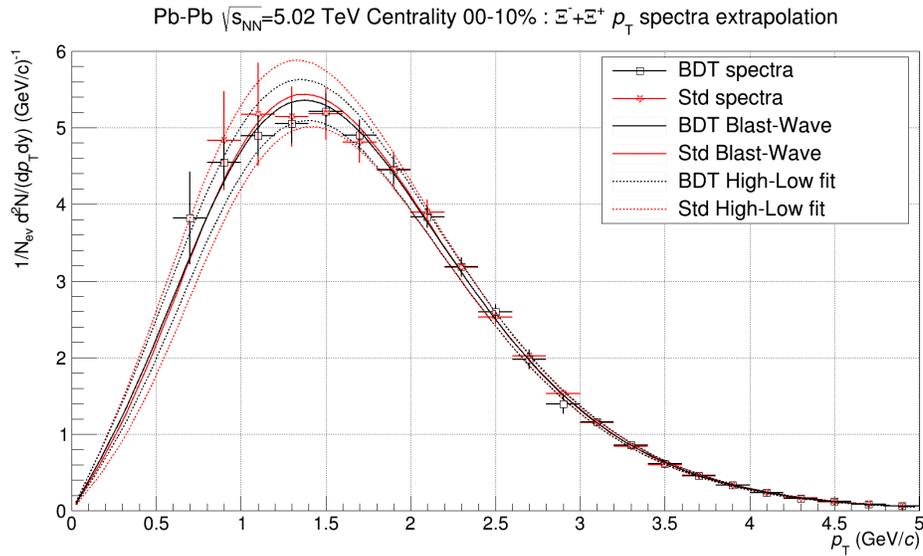


Figure 5.7: High-Low fit extrapolation to low p_T of $\Xi^- + \Xi^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the High-Low fits used to estimate the systematic uncertainty. BDT analysis is shown in black and the standard analysis in red.

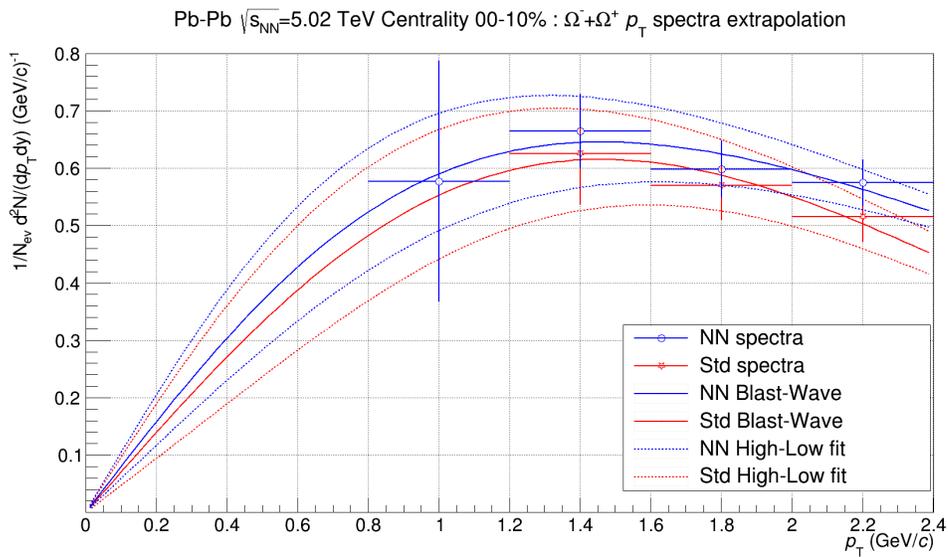


Figure 5.8: High-Low fit extrapolation to low p_T of $\Omega^- + \Omega^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the High-Low fits used to estimate the systematic uncertainty. Neural network analysis is shown in blue and the standard analysis in red.

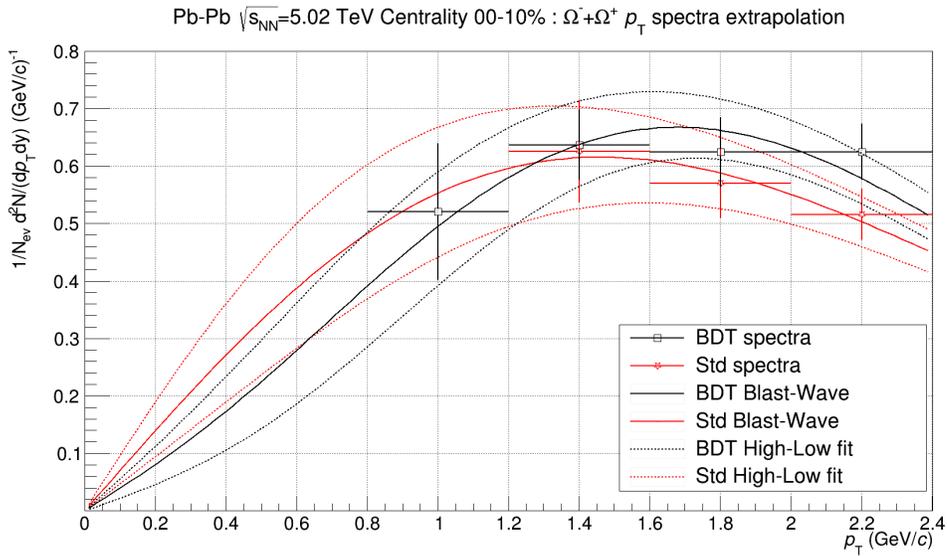


Figure 5.9: High-Low fit extrapolation to low p_T of $\Omega^- + \Omega^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the High-Low fits used to estimate the systematic uncertainty. BDT analysis is shown in black and the standard analysis in red.

Secondary function fitting

The second source of systematic is the choice of the function to perform the fit since other functions can be used in the extrapolation procedure. The systematic uncertainty here is calculated by fitting the spectra with three other functions: m_T exponential, Boltzmann and PHENIX modified Hagedorn and choosing the one that gives the maximum difference in the extrapolated yield for the uncertainty.

Figures 5.10 to 5.13 shows the described procedure comparing the ML models with the standard analysis in the 0-10% centrality class. And tables 5.5 to 5.7 presents the contributions of both sources of low p_T extrapolation systematic uncertainties as well as the uncertainties propagated from the spectra for the production yield of particles.

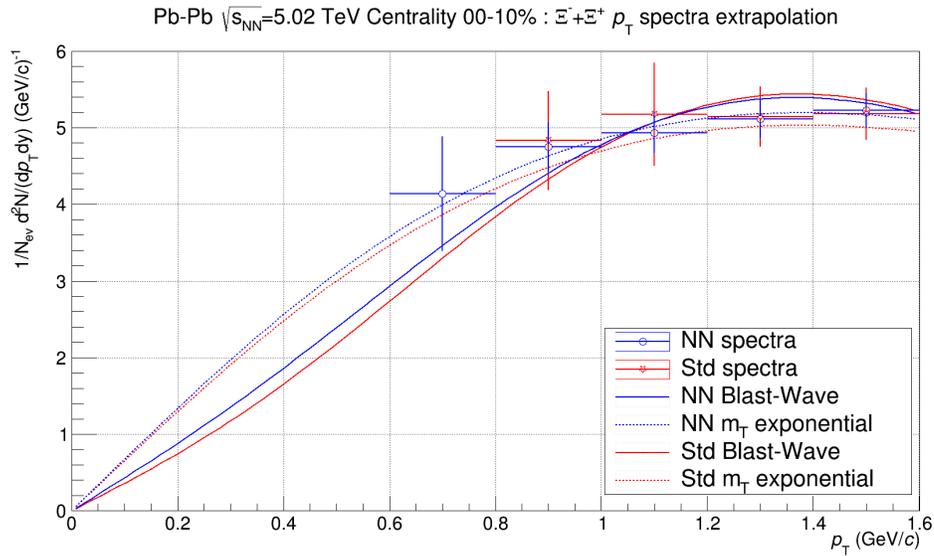


Figure 5.10: Secondary function fit extrapolation to low p_T of $\Xi^- + \Xi^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the m_T exponential fits used to estimate the systematic uncertainty. Neural network analysis is shown in blue and the standard analysis in red.

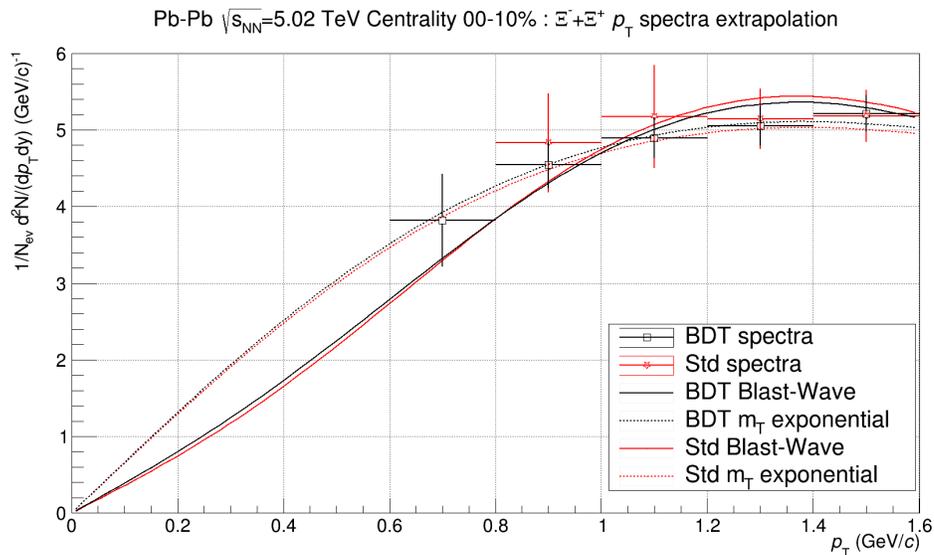


Figure 5.11: Secondary function fit extrapolation to low p_T of $\Xi^- + \Xi^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra and dashed lines are the m_T exponential fits used to estimate the systematic uncertainty. BDT analysis is shown in black and the standard analysis in red.

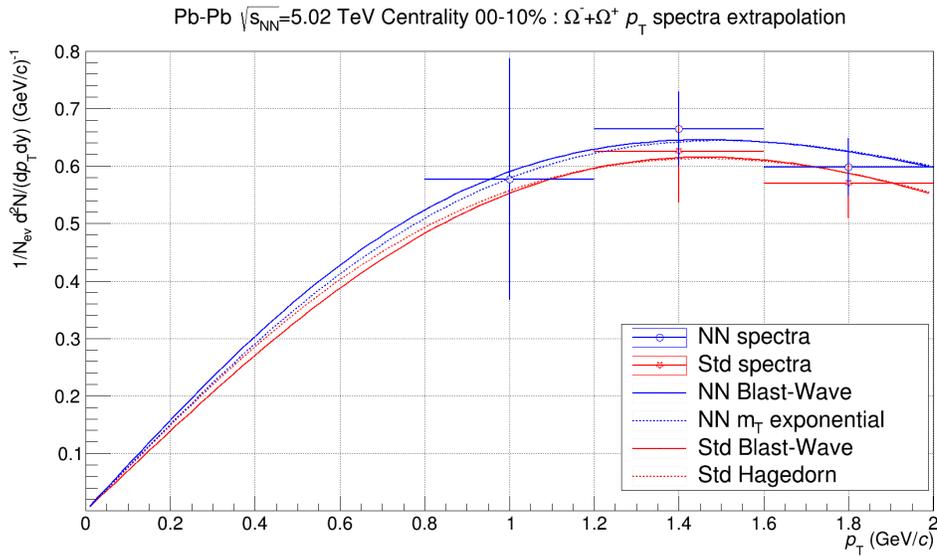


Figure 5.12: Secondary function fit extrapolation to low p_T of $\Omega^- + \Omega^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra. Dashed line for Neural Network is the m_T exponential fit and dashed line for the standard analysis is the Hagedorn fit used to estimate the systematic uncertainty. Neural network analysis is shown in blue and the standard analysis in red.

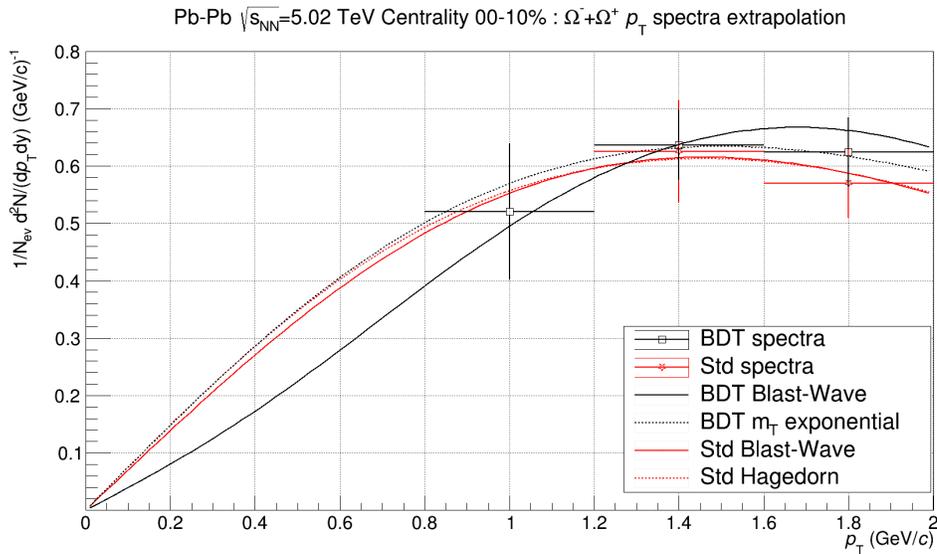


Figure 5.13: Secondary function fit extrapolation to low p_T of $\Omega^- + \Omega^+$ spectra in 0-10% centrality class. Markers are the spectra, lines are the Blast-Wave fit of the spectra. Dashed line for BDT is the m_T exponential and dashed line for the standard analysis is the Hagedorn fit used to estimate the systematic uncertainty. BDT analysis is shown in black and the standard analysis in red.

Table 5.5: Standard analysis relative uncertainties of production yield of $\Xi^- + \Xi^+$ ($\Omega^- + \Omega^+$)

Centrality class	Spectra Uncert. %	High-Low Uncert. %	Secondary Function Uncert. %
00-10%	2.14 (3.27)	2.49 (7.10)	6.61 (0.73)
10-20%	1.81 (3.11)	2.14 (5.80)	2.11 (1.90)
20-30%	1.81 (3.13)	2.28 (6.02)	1.63 (1.85)
30-40%	1.84 (2.98)	2.69 (7.14)	3.89 (1.06)
40-50%	1.84 (3.25)	2.96 (6.37)	1.35 (4.12)
50-60%	1.94 (4.02)	2.83 (4.71)	3.72 (19.1)
60-70%	1.94 (4.21)	2.95 (3.50)	2.66 (21.6)
70-80%	2.08 (3.59)	3.85 (15.2)	1.32 (4.62)
80-90%	2.48 (6.33)	4.96 (17.8)	2.46 (13.1)

Table 5.6: Neural network analysis relative uncertainties of production yield of $\Xi^- + \Xi^+$ ($\Omega^- + \Omega^+$)

Centrality class	Spectra Uncert. %	High-Low Uncert. %	Secondary Function Uncert. %
00-10%	1.81 (5.68)	0.85 (3.61)	2.79 (0.53)
10-20%	1.59 (3.96)	0.81 (3.07)	2.58 (6.31)
20-30%	1.37 (3.31)	0.88 (3.31)	2.06 (1.26)
30-40%	1.39 (3.42)	1.02 (2.69)	1.52 (6.07)
40-50%	1.37 (2.03)	1.02 (4.99)	1.12 (2.22)
50-60%	1.28 (2.76)	1.07 (5.23)	0.58 (17.04)
60-70%	1.48 (2.94)	1.13 (5.25)	0.22 (2.18)
70-80%	1.66 (3.36)	1.49 (15.4)	0.96 (4.46)
80-90%	2.62 (6.83)	2.68 (14.08)	0.66 (3.57)

Table 5.7: BDT analysis relative uncertainties of production yield of $\Xi^- + \Xi^+$ ($\Omega^- + \Omega^+$)

Centrality class	Spectra Uncert. %	High-Low Uncert. %	Secondary Function Uncert. %
00-10%	1.67 (4.17)	0.91 (3.24)	3.20 (4.81)
10-20%	1.46 (3.35)	0.78 (2.59)	2.02 (2.28)
20-30%	1.26 (3.93)	0.85 (3.81)	1.45 (2.34)
30-40%	1.32 (3.19)	0.98 (2.32)	1.35 (6.90)
40-50%	1.29 (2.17)	0.98 (3.87)	1.17 (6.39)
50-60%	1.28 (2.67)	0.92 (4.43)	0.47 (16.93)
60-70%	1.48 (2.73)	1.18 (5.44)	0.02 (11.36)
70-80%	1.87 (3.59)	1.73 (15.73)	0.81 (2.10)
80-90%	2.53 (5.56)	2.51 (15.42)	0.86 (5.56)

Chapter 6

Results and discussion

This chapter presents the main physical results obtained using the methods described in the previous chapter: the corrected p_T spectra and total yield production of multi-strange baryons in Pb-Pb collisions at the center of mass energy of $\sqrt{s_{NN}} = 5.02$ TeV, as well as the strangeness enhancement, defined as the ratio of multi-strange baryons to pions as a function of charged particle multiplicity. Which is the QGP observable of interest in this work.

6.1 Transverse momentum spectra

Figure 6.1 presents the corrected p_T spectra for $\Xi^- + \Xi^+$ and $\Omega^- + \Omega^+$ across all the centrality classes, obtained with each method described in this work.

The spectra of different methods are in agreement within uncertainties and the Machine Learning based methods presents an overall uncertainty improvement. An unprecedented experimental point in the low p_T region is presented for $\Xi^- + \Xi^+$ across all centrality classes and for $\Omega^- + \Omega^+$ in the centrality classes 0-40%. For higher centrality classes it was not possible to perform a reliable signal extraction due to low counts in the invariant mass spectra.

6.2 Particle production yield

The particle production yield is computed by integrating the corrected p_T spectra and its respective extrapolation to low p_T using a blast-wave function. Tables 6.1 to 6.3 presents the results for $\Xi^- + \Xi^+$ and $\Omega^- + \Omega^+$ also across all the centrality classes, obtained with each method described in this work. The yields obtained with each methods are in agreement within uncertainties.

Both the uncertainty improvement of the spectra and the presence of the additional point, when it is present, due to Machine Learning reflects as improvement in the yield uncertainties. Figures 6.2 and 6.3 presents the comparison of yields of multi-strange baryon production, as well, as the ratio of uncertainties from Machine Learning based methods to the standard analysis, showing an overall reduction of 40% in the $\Xi^- + \Xi^+$ measurement and 20% in the $\Omega^- + \Omega^+$, across centrality classes.

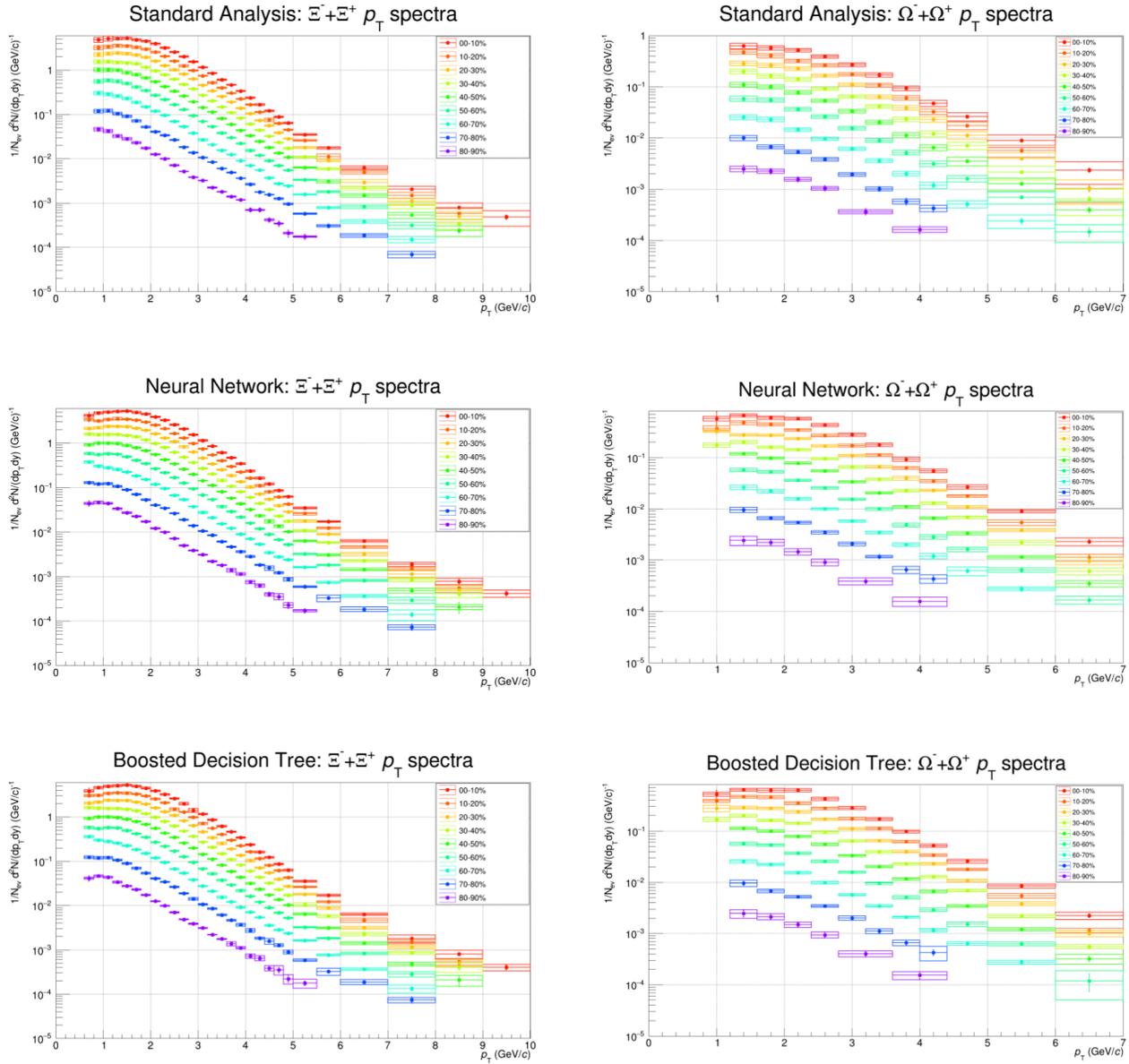


Figure 6.1: Transverse momentum spectra of $\Xi^- + \Xi^+$ (left) and $\Omega^- + \Omega^+$ (right) in Pb-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV for all centrality classes. Obtained with the standard analysis (top), a Neural Network analysis (center) and a Boosted Decision Tree analysis (bottom). Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra.

Table 6.1: Standard analysis: Particle production yield

Centrality class	$\Xi^- + \Xi^+$	$\Omega^- + \Omega^+$
00-10%	10.80 ± 0.80	1.53 ± 0.12
10-20%	7.21 ± 0.25	1.11 ± 0.08
20-30%	4.88 ± 0.16	0.68 ± 0.05
30-40%	3.11 ± 0.16	0.47 ± 0.04
40-50%	1.97 ± 0.07	0.24 ± 0.02
50-60%	1.04 ± 0.05	0.11 ± 0.02
60-70%	0.54 ± 0.02	0.044 ± 0.0099
70-80%	0.21 ± 0.01	0.022 ± 0.0035
80-90%	0.074 ± 0.004	0.0047 ± 0.0011

Table 6.2: Neural Network: Particle production yield

Centrality class	$\Xi^- + \Xi^+$	$\Omega^- + \Omega^+$
00-10%	11.06 ± 0.38	1.64 ± 0.11
10-20%	7.50 ± 0.24	1.03 ± 0.08
20-30%	5.02 ± 0.13	0.75 ± 0.04
30-40%	3.29 ± 0.08	0.42 ± 0.03
40-50%	1.97 ± 0.04	0.27 ± 0.02
50-60%	1.10 ± 0.02	0.11 ± 0.02
60-70%	0.55 ± 0.01	0.056 ± 0.0036
70-80%	0.215 ± 0.005	0.021 ± 0.0034
80-90%	0.073 ± 0.003	0.0049 ± 0.0008

Table 6.3: Boosted Decision Tree: Particle production yield

Centrality class	$\Xi^- + \Xi^+$	$\Omega^- + \Omega^+$
00-10%	10.81 ± 0.40	1.54 ± 0.11
10-20%	7.34 ± 0.19	1.08 ± 0.05
20-30%	4.96 ± 0.10	0.69 ± 0.04
30-40%	3.31 ± 0.07	0.41 ± 0.03
40-50%	1.96 ± 0.04	0.24 ± 0.02
50-60%	1.09 ± 0.02	0.11 ± 0.02
60-70%	0.54 ± 0.01	0.0497 ± 0.0064
70-80%	0.212 ± 0.006	0.021 ± 0.0035
80-90%	0.072 ± 0.003	0.00559 ± 0.00096

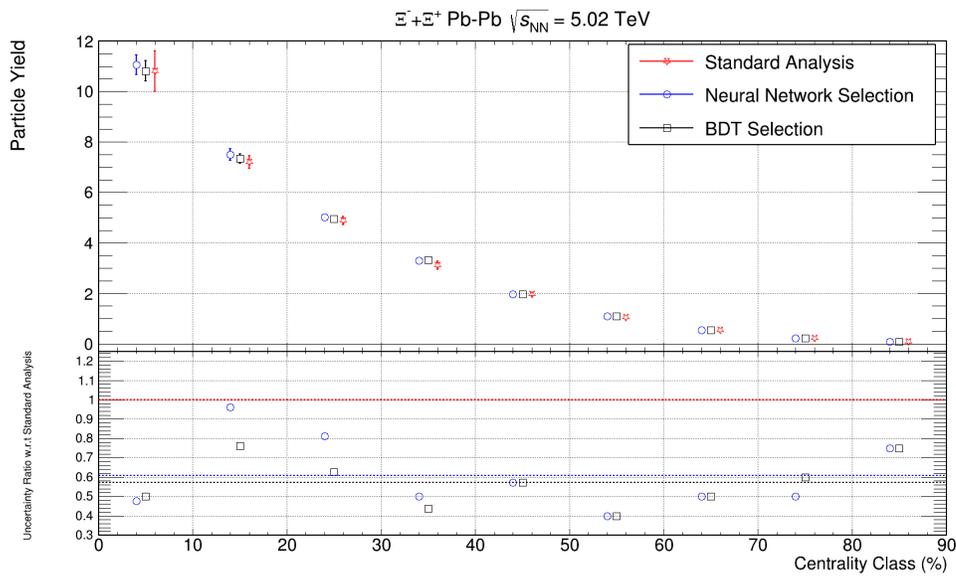


Figure 6.2: $\Xi^- + \Xi^+$ production yield as a function of centrality class. The standard analysis is shown in red, Neural Network analysis in blue and BDT analysis in black. Lower plot presents the ratio of uncertainties from Machine Learning based methods to the standard analysis.

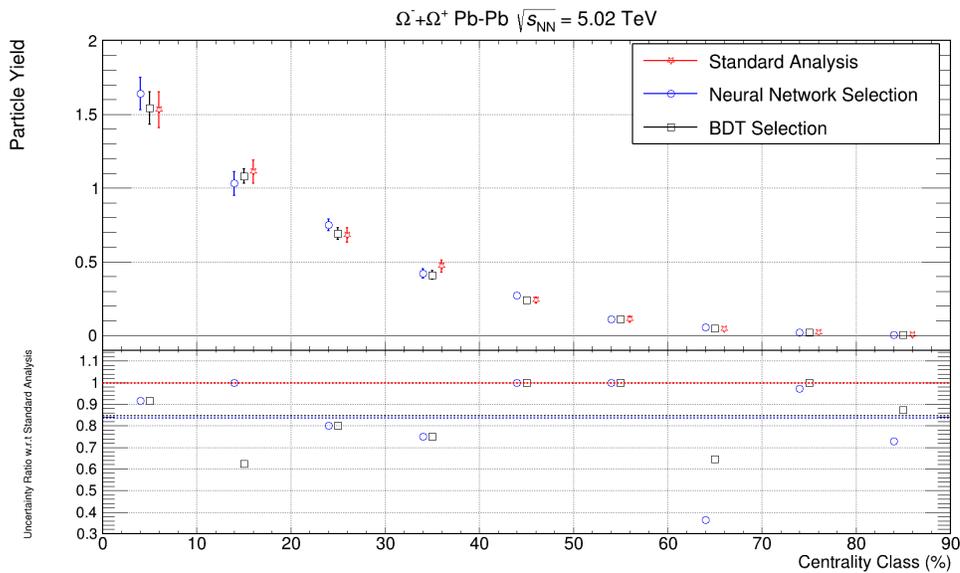


Figure 6.3: $\Omega^- + \Omega^+$ production yield as a function of centrality class. The standard analysis is shown in red, Neural Network analysis in blue and BDT analysis in black. Lower plot presents the ratio of uncertainties from Machine Learning based methods to the standard analysis.

6.3 Strangeness Enhancement

The ratio between multi-strange baryons and pions as a function of charged particle multiplicity, that is mapped into centrality classes using table 3.1, defines the observable known as strangeness enhancement. Which is one possible observable for probing the QGP. The lower the centrality class higher the number of interacting particles forming the QGP in the collision which implies in higher thermal production of strange quarks given the expected temperature of the phase transition to QGP and the mass of the strange quark. Therefore the proportion of strange particles should increase with the number of colliding particles.

Figure 6.4 shows the obtained results with all three methods together with the already published results in different collision systems.

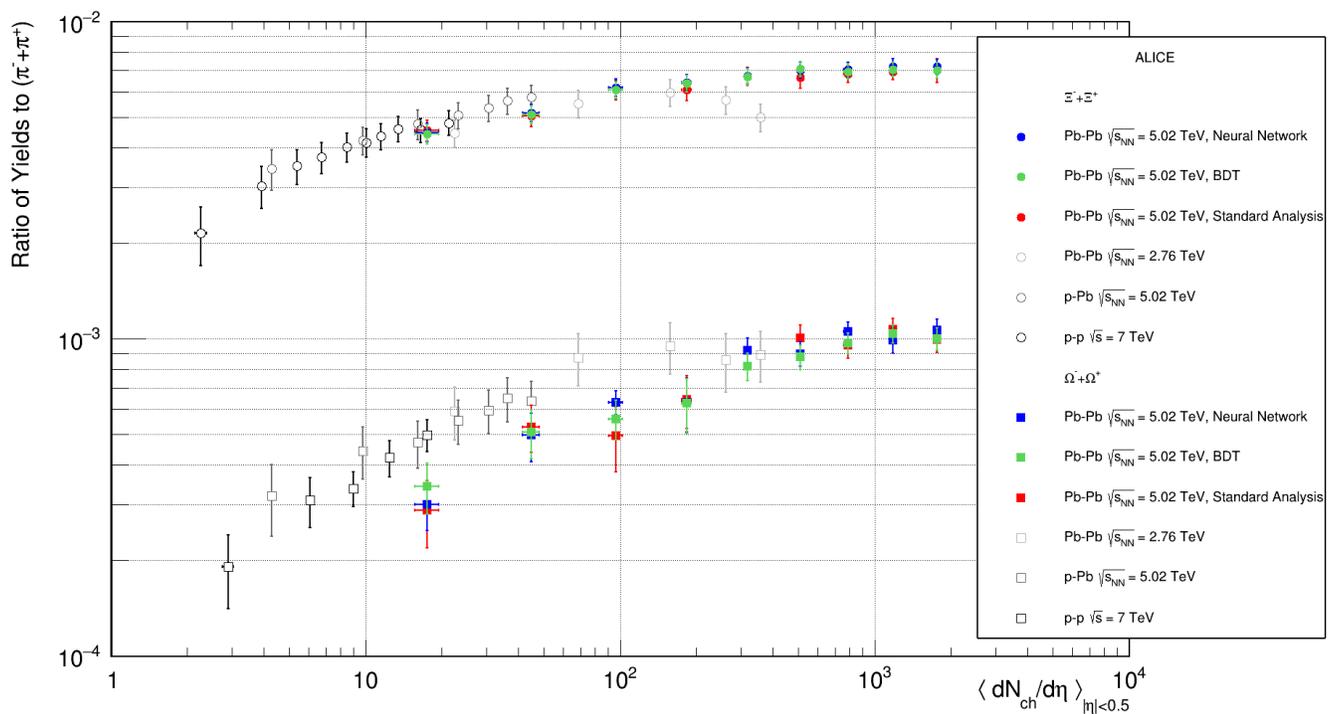


Figure 6.4: Ratio of multi-strange baryons to pions as a function of charged particle multiplicity compared with previous measurements already presented in figure 2.6.

Chapter 7

Conclusion

In this work we developed a novel methodology, using Machine Learning techniques, for measuring multi-strange baryons in Pb-Pb collisions at the center of mass energy $\sqrt{s_{NN}} = 5.02$ TeV with the ALICE apparatus at the LHC. This was possible since the data analysis procedure of strangeness reconstruction is, essentially, a classification problem, which is a class of problems known to have applicable tools based on the paradigm that defines Machine Learning. The models used were artificial neural networks, in the form of a multi-layer perceptron, and boosted decision trees.

A new code was developed, and it is now part of the official ALICE repository, in order to allow compatibility between the collaboration's analysis framework (in C++) and the Machine Learning models built in python. The framework is necessary in order to perform the analysis in the complete data set, accessing the computational power of the Worldwide LHC Computing Grid (WLCG).

The analysis was also performed in the standard way in order to evaluate the impact of the Machine Learning models. The efficiency and significance of machine learning based methods presents a simultaneous overall improvement of around 20% with respect to the standard analysis. Such improvement led to the extension of the inferior p_T limit of the measured particles. The presence of an additional p_T bin directly impacts the total production yield computation, since it requires a low p_T extrapolation of the measured spectra, resulting in an overall reduction of around 40% in the $\Xi^- + \Xi^+$ yield uncertainty and around 20% in the $\Omega^- + \Omega^+$ yield uncertainty.

The physical results of all methods we worked with are in agreement within uncertainties. The ratio of multi-strange baryons to pions as a function of charged particle multiplicity are presented and show the expected increase in the relative production of strangeness. Therefore verifying the strangeness enhancement observable for QGP. The Machine Learning improvements also reflect in the overall precision of such measurement.

The results regarding the methodology comparison in this work were presented at the XLIV Brazilian Workshop on Nuclear Physics in 2021 as a poster, and at the III SBF Spring Meeting in 2023 as the talk "Strangeness reconstruction with machine learning in heavy-ion collisions". The results of multi-strange baryon production in the collision system and energy that we worked with were still not published by the collaboration. We are seeking internal approval for the obtained results and working with the collaboration towards a publication.

As part of the ALICE collaboration during the period of this project I also participated in the data taking of pp collisions in 2022, that is the so called Run3 of the ALICE experiment that started after the upgrades during the last long shutdown. The data taking is performed by the shift crew inside the ALICE Run Control Center (ARC), I did a total of 26 shifts of 8 hours, totaling more than 200 hours of work. Six of those were training shifts, in twelve shifts I was assigned at the Experiment Control System (ECS) position, being responsible by the creation of the environments to acquire data and detector calibration, in the other eight shifts I was assigned at the Shift Leader (SL) position, being responsible for the entire crew: ECS, DCS (Detector Control System) and QC (Quality Control), and coordinating the team with the objective of performing the data taking in the best way possible. I also worked in a service task for the collaboration being part of the team that did the multiplicity calibration of the Run3 data in the Data Preparation Group (DPG).

This work started in 2019 and was, like everything else, affected by the COVID-19 pandemic. During such a difficulty period we had the opportunity to explore the potential of Machine Learning outside the field of physics, and contribute with the understanding of the impact of the disease in patients at the University's Hospital. In collaboration with the School of Medical Sciences of UNICAMP we developed a study that was later published [45] in the Informatics in Medicine Unlocked (<https://doi.org/10.1016/j.imu.2022.101138>).

In this work I had the incredible opportunity to participate in an international collaboration at the frontier of physics research and experienced how challenging it is to work in a high energy physics experiment like ALICE. It requires knowledge of physics, engineering and computer sciences. Professionals from various fields worldwide gathered to make this level of research possible. It improved my overall skills as a scientist and collaborator. I learned what is to work as a group to achieve goals that are far too big to be done alone. It made me realise that even though physics is an exact science, to do science is a human activity. Working as a group, scientists expand the boundaries of knowledge and I am very grateful to be a part of it.

I developed an understanding of the fundamentals of heavy-ion physics, detectors, data acquisition and analysis. Explored a new field of knowledge, Machine Learning, and was pioneer in the study of those techniques in my research group, that is now working with it in a variety of projects. During this project I was amazed by the potential and applicability of Machine Learning tools and it was very interesting to apply it in the strangeness reconstruction, look to the data analysis through a new lens, develop new tools and a new method for investigating such particles that resulted in the same physics with higher precision.

Bibliography

- [1] F. Chollet. *Deep Learning with Python*. Manning, 2017. ISBN: 9781638352044. URL: <https://books.google.com.br/books?id=wzozEAAAQBAJ>.
- [2] T.M. Mitchell. *Machine Learning*. McGraw-Hill international editions - computer science series. McGraw-Hill Education, 1997. ISBN: 9780070428072. URL: <https://books.google.com.br/books?id=x0GAngEACAAJ>.
- [3] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2016. ISBN: 9780262035613. URL: <https://books.google.com.br/books?id=-s2MEAAAQBAJ>.
- [4] Warren S. McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259. URL: <https://doi.org/10.1007/BF02478259>.
- [5] D. Griffiths. *Introduction to Elementary Particles*. Physics textbook. Wiley, 2008. ISBN: 9783527406012. URL: <https://books.google.com.br/books?id=w9Dz56myXm8C>.
- [6] B. Pontecorvo. “Neutrino Experiments and the Problem of Conservation of Leptonic Charge”. In: *Zh. Eksp. Teor. Fiz.* 53 (1967), pp. 1717–1725.
- [7] John N. Bahcall. “Solar Neutrinos. I. Theoretical”. In: *Phys. Rev. Lett.* 12 (11 Mar. 1964), pp. 300–302. DOI: 10.1103/PhysRevLett.12.300. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.12.300>.
- [8] Q. R. Ahmad et al. “Measurement of the Rate of $\nu_e + d \rightarrow p + p + e^-$ Interactions Produced by ^8B Solar Neutrinos at the Sudbury Neutrino Observatory”. In: *Phys. Rev. Lett.* 87 (7 July 2001), p. 071301. DOI: 10.1103/PhysRevLett.87.071301. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.87.071301>.
- [9] Peter W. Higgs. “Broken Symmetries and the Masses of Gauge Bosons”. In: *Phys. Rev. Lett.* 13 (16 Oct. 1964), pp. 508–509. DOI: 10.1103/PhysRevLett.13.508. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.13.508>.
- [10] M.D. Schwartz. *Quantum Field Theory and the Standard Model*. Quantum Field Theory and the Standard Model. Cambridge University Press, 2014. ISBN: 9781107034730. URL: <https://books.google.com.br/books?id=HbdEAgAAQBAJ>.
- [11] E. Noether. “Invariante Variationsprobleme”. ger. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1918 (1918), pp. 235–257. URL: <http://eudml.org/doc/59024>.

- [12] Cush. *Standard Model of Elementary Particles*. Available online at: https://en.wikipedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg, last accessed on 12.10.2023. 2019.
- [13] W. Rossmann. *Lie Groups: An Introduction Through Linear Groups*. Oxford graduate texts in mathematics. Oxford University Press, 2006. ISBN: 9780199202515. URL: <https://books.google.com.br/books?id=bAjulQ65W-UC>.
- [14] J Beringer et al. “Review of Particle Physics, 2012-2013. Review of Particle Properties”. In: *Phys. Rev. D* 86.1 (2012). The 2012 edition of Review of Particle Physics is published for the Particle Data Group as article 010001 in volume 86 of Physical Review D. This edition should be cited as: J. Beringer et al. (Particle Data Group), *Phys. Rev. D* 86, 010001 (2012)., p. 010001. DOI: 10.1103/PhysRevD.86.010001. URL: <http://cds.cern.ch/record/1481544>.
- [15] ALICE Collaboration. *The ALICE experiment – A journey through QCD*. 2022. arXiv: 2211.04384 [nucl-ex].
- [16] J. 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 (2005). First Three Years of Operation of RHIC, pp. 102–183. ISSN: 0375-9474. DOI: <https://doi.org/10.1016/j.nuclphysa.2005.03.085>. URL: <https://www.sciencedirect.com/science/article/pii/S0375947405005294>.
- [17] Kenneth G. Wilson. “Confinement of quarks”. In: *Phys. Rev. D* 10 (8 Oct. 1974), pp. 2445–2459. DOI: 10.1103/PhysRevD.10.2445. URL: <https://link.aps.org/doi/10.1103/PhysRevD.10.2445>.
- [18] Johann Rafelski and Berndt Müller. “Strangeness Production in the Quark-Gluon Plasma”. In: *Phys. Rev. Lett.* 48 (16 Apr. 1982), pp. 1066–1069. DOI: 10.1103/PhysRevLett.48.1066. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.48.1066>.
- [19] P Koch, B Müller, and J Rafelski. “Strangeness in relativistic heavy ion collisions”. In: *Physics Reports* 142.4 (1986), pp. 167–262. ISSN: 0370-1573. DOI: [https://doi.org/10.1016/0370-1573\(86\)90096-7](https://doi.org/10.1016/0370-1573(86)90096-7). URL: <https://www.sciencedirect.com/science/article/pii/0370157386900967>.
- [20] Antonin Maire. “Production des baryons multi-étranges au LHC dans les collisions proton-proton avec l’expérience ALICE”. Presented 13 Oct 2011. 2011. URL: <https://cds.cern.ch/record/1490315>.
- [21] Johann Rafelski. “Melting hadrons, boiling quarks”. In: *The European Physical Journal A* 51.9 (Sept. 2015). DOI: 10.1140/epja/i2015-15114-0. URL: <https://doi.org/10.1140/2Fepja%2Fi2015-15114-0>.
- [22] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. “A brief introduction to PYTHIA 8.1”. In: *Computer Physics Communications* 178.11 (June 2008), pp. 852–867. DOI: 10.1016/j.cpc.2008.01.036. URL: <https://doi.org/10.1016%2Fj.cpc.2008.01.036>.

- [23] T. Pierog et al. “EPOS LHC: Test of collective hadronization with data measured at the CERN Large Hadron Collider”. In: *Physical Review C* 92.3 (Sept. 2015). DOI: 10.1103/physrevc.92.034906. URL: <https://doi.org/10.1103%2Fphysrevc.92.034906>.
- [24] Christian Bierlich and Jesper Roy Christiansen. “Effects of color reconnection on hadron flavor observables”. In: *Physical Review D* 92.9 (Nov. 2015). DOI: 10.1103/physrevd.92.094010. URL: <https://doi.org/10.1103%2Fphysrevd.92.094010>.
- [25] The ALICE Collaboration. “Multi-strange baryon production at mid-rapidity in Pb–Pb collisions at $\sqrt{s_{\text{NN}}}=2.76$ TeV”. In: *Physics Letters B* 728 (2014), pp. 216–227. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2013.11.048>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269313009544>.
- [26] The ALICE Collaboration. “Multi-strange baryon production in pPb collisions at $\sqrt{s_{\text{NN}}}=5.02$ TeV”. In: *Physics Letters B* 758 (2016), pp. 389–401. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2016.05.027>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269316301745>.
- [27] The ALICE Collaboration. “Multiplicity dependence of pion, kaon, proton and lambda production in p–Pb collisions at $\sqrt{s_{\text{NN}}}=5.02$ TeV”. In: *Physics Letters B* 728 (2014), pp. 25–38. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2013.11.020>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269313009234>.
- [28] The ALICE Collaboration. “Enhanced production of multi-strange hadrons in high-multiplicity proton–proton collisions”. In: *Nature Physics* 13 (2017), pp. 535–539. DOI: <https://doi.org/10.1038/nphys4111>. URL: <https://www.nature.com/articles/nphys4111>.
- [29] The ALICE Collaboration. “The ALICE experiment at the CERN LHC”. In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08002. DOI: 10.1088/1748-0221/3/08/S08002. URL: <https://dx.doi.org/10.1088/1748-0221/3/08/S08002>.
- [30] *ALICE: Technical proposal for a Large Ion collider Experiment at the CERN LHC*. LHC technical proposal. Geneva: CERN, 1995. URL: <https://cds.cern.ch/record/293391>.
- [31] G Dellacasa et al. *ALICE time projection chamber: Technical Design Report*. Technical design report. ALICE. Geneva: CERN, 2000. URL: <https://cds.cern.ch/record/451098>.
- [32] Elena Botta. *Particle identification performance at ALICE*. *Particle identification performance at ALICE*. Tech. rep. 6 pages, 5 figures, Proceeding of the Fifth Annual Conference on Large Hadron Collider Physics, May 15-20, 2017, Shanghai, Cina. 2017. arXiv: 1709.00288. URL: <http://cds.cern.ch/record/2282027>.
- [33] “Centrality dependence of the charged-particle multiplicity density at midrapidity in Pb-Pb collisions at $\sqrt{s_{\text{NN}}} = 5.02$ TeV”. In: (2015). URL: <https://cds.cern.ch/record/2118084>.
- [34] K. Aamodt et al. “Centrality Dependence of the Charged-Particle Multiplicity Density at Midrapidity in Pb-Pb Collisions at $\sqrt{s_{\text{NN}}} = 2.76$ TeV”. In: *Phys. Rev. Lett.* 106 (3 Jan. 2011), p. 032301. DOI: 10.1103/PhysRevLett.106.032301. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.106.032301>.

- [35] J. Adam et al. “Centrality Dependence of the Charged-Particle Multiplicity Density at Midrapidity in Pb-Pb Collisions at $\sqrt{s_{NN}} = 5.02$ TeV”. In: *Phys. Rev. Lett.* 116 (22 June 2016), p. 222302. DOI: [10.1103/PhysRevLett.116.222302](https://doi.org/10.1103/PhysRevLett.116.222302). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.116.222302>.
- [36] The ALICE Collaboration. “Centrality dependence of the pseudorapidity density distribution for charged particles in Pb–Pb collisions at sNN=5.02 TeV”. In: *Physics Letters B* 772 (2017), pp. 567–577. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2017.07.017>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269317305646>.
- [37] S. Acharya et al. “Production of charged pions, kaons, and (anti-)protons in Pb-Pb and inelastic pp collisions at $\sqrt{s_{NN}} = 5.02$ TeV”. In: *Phys. Rev. C* 101 (4 Apr. 2020), p. 044907. DOI: [10.1103/PhysRevC.101.044907](https://doi.org/10.1103/PhysRevC.101.044907). URL: <https://link.aps.org/doi/10.1103/PhysRevC.101.044907>.
- [38] Jan Fiete Grosse-Oetringhaus et al. “The ALICE online-offline framework for the extraction of conditions data”. In: *Journal of Physics: Conference Series* 219.2 (Apr. 2010), p. 022010. DOI: [10.1088/1742-6596/219/2/022010](https://doi.org/10.1088/1742-6596/219/2/022010). URL: <https://dx.doi.org/10.1088/1742-6596/219/2/022010>.
- [39] R Brun, F Rademakers, and S Panacek. “ROOT, an object oriented data analysis framework”. In: (2000). URL: <https://cds.cern.ch/record/491486>.
- [40] Danilo Silva De Albuquerque. “Multi-strange hadrons in Pb–Pb collisions at the LHC with ALICE. Hádrõns multi-estranhos em colisões Pb–Pb no LHC com o ALICE”. Presented 16 Aug 2019. Campinas State U., 2019. URL: <http://cds.cern.ch/record/2690627>.
- [41] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [42] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [43] 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*. KDD ’16. San Francisco, California, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785). URL: <http://doi.acm.org/10.1145/2939672.2939785>.
- [44] Ekkard Schnedermann, Josef Sollfrank, and Ulrich Heinz. “Thermal phenomenology of hadrons from 200A GeV S+S collisions”. In: *Phys. Rev. C* 48 (5 Nov. 1993), pp. 2462–2475. DOI: [10.1103/PhysRevC.48.2462](https://doi.org/10.1103/PhysRevC.48.2462). URL: <https://link.aps.org/doi/10.1103/PhysRevC.48.2462>.
- [45] Daniela M.H. Padilha et al. “Construction of a nomogram for predicting COVID-19 in-hospital mortality: A machine learning analysis”. In: *Informatics in Medicine Unlocked* 36 (2023), p. 101138. ISSN: 2352-9148. DOI: <https://doi.org/10.1016/j.imu.2022.101138>. URL: <https://www.sciencedirect.com/science/article/pii/S2352914822002751>.

Appendix A

Comparison of transverse momentum spectra

This appendix presents the comparison of the p_T spectra for $\Xi^- + \Xi^+$ and $\Omega^- + \Omega^+$ obtained with Machine Learning based methods and the standard analysis in centrality classes from 10% to 90%, omitted in Chapter 5. All the results are in agreement within uncertainties.

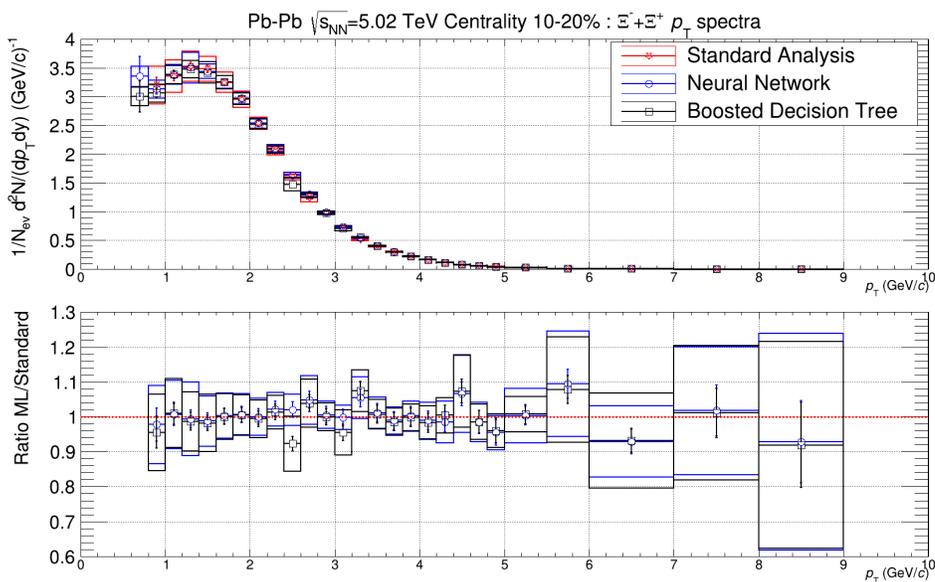


Figure A.1: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 10-20% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

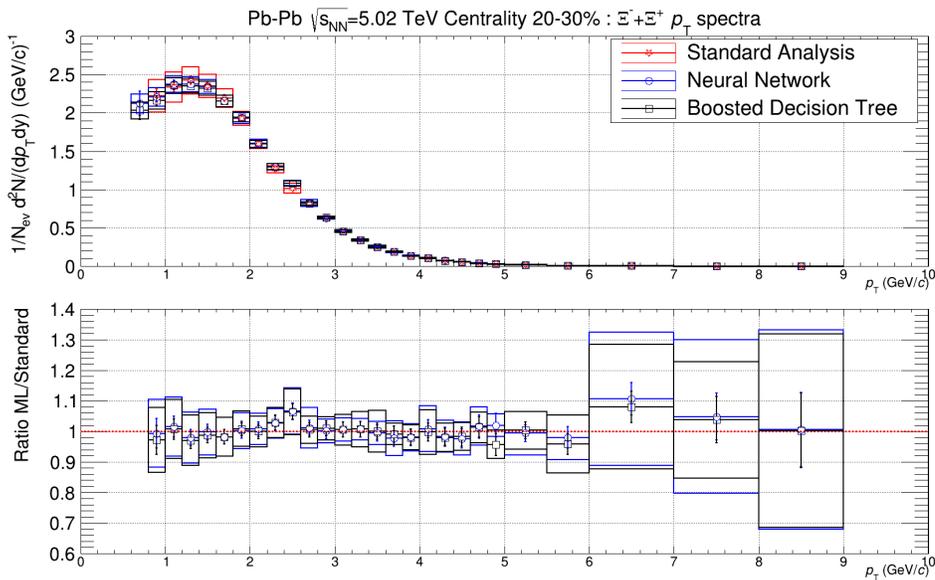


Figure A.2: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 20-30% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

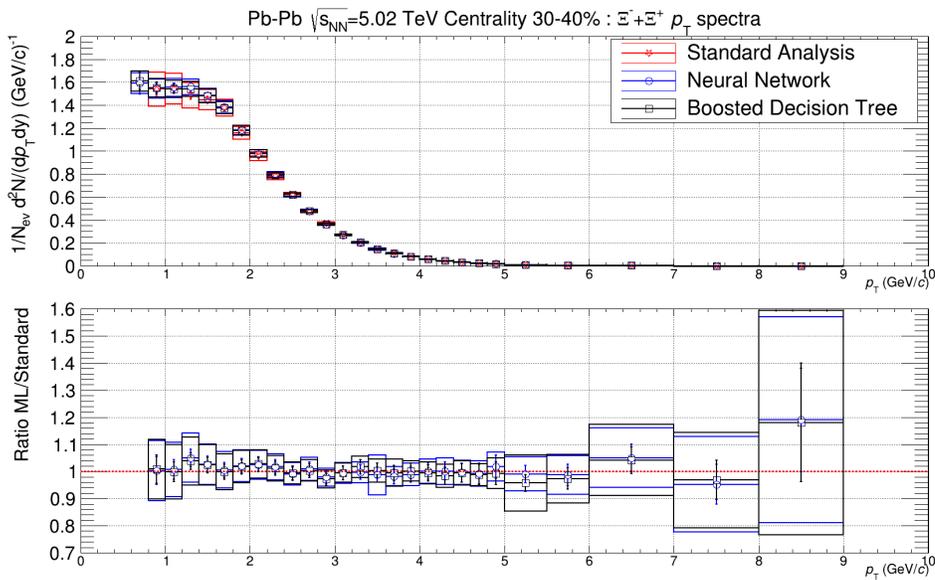


Figure A.3: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 30-40% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

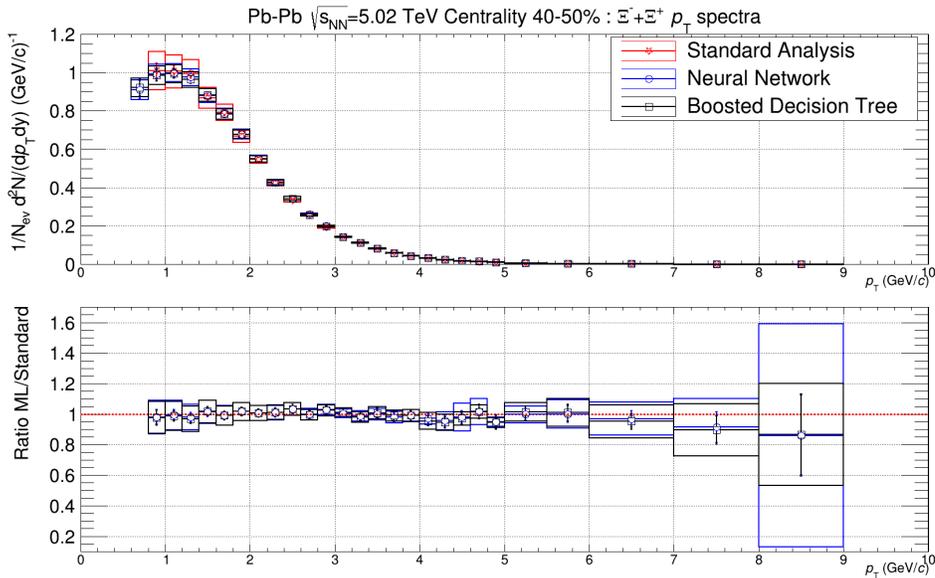


Figure A.4: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 40-50% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

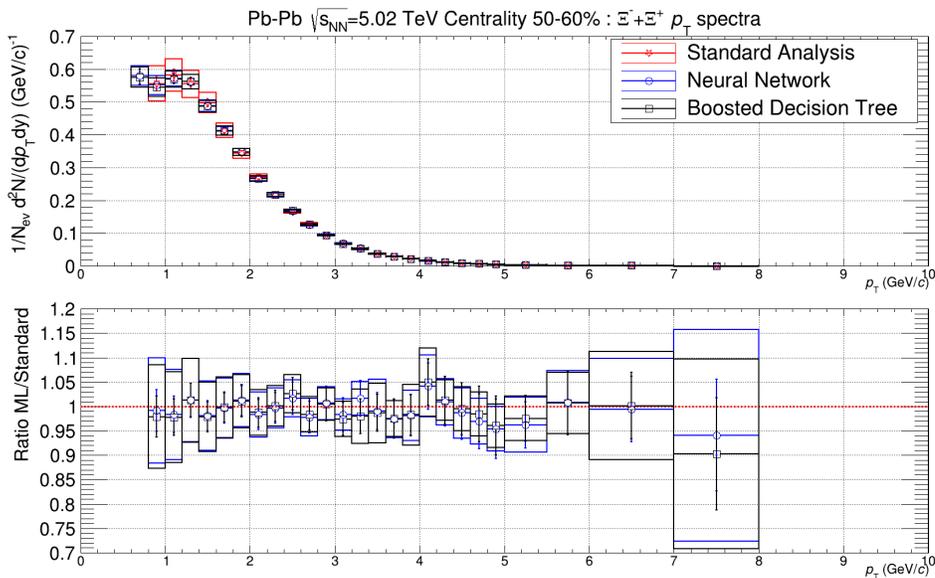


Figure A.5: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 50-60% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

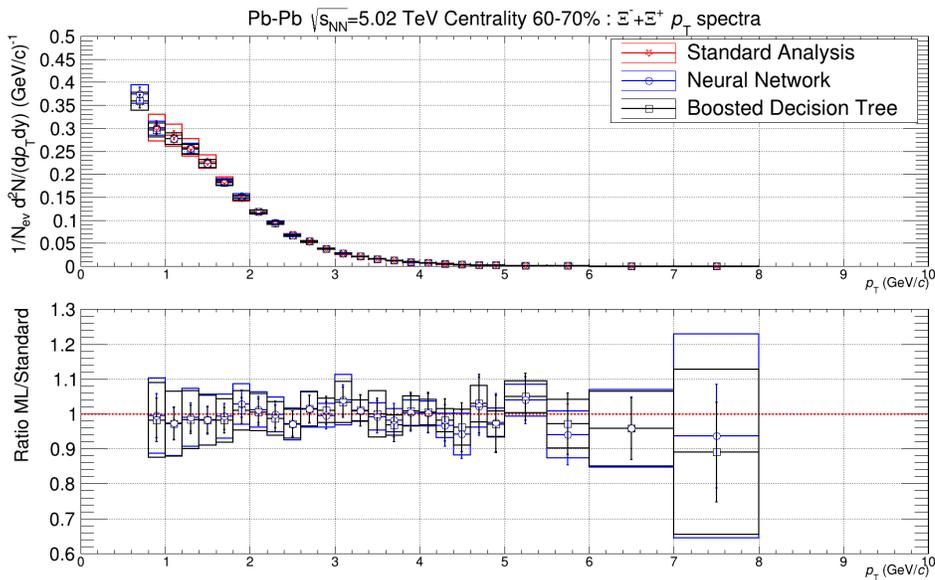


Figure A.6: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 60-70% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

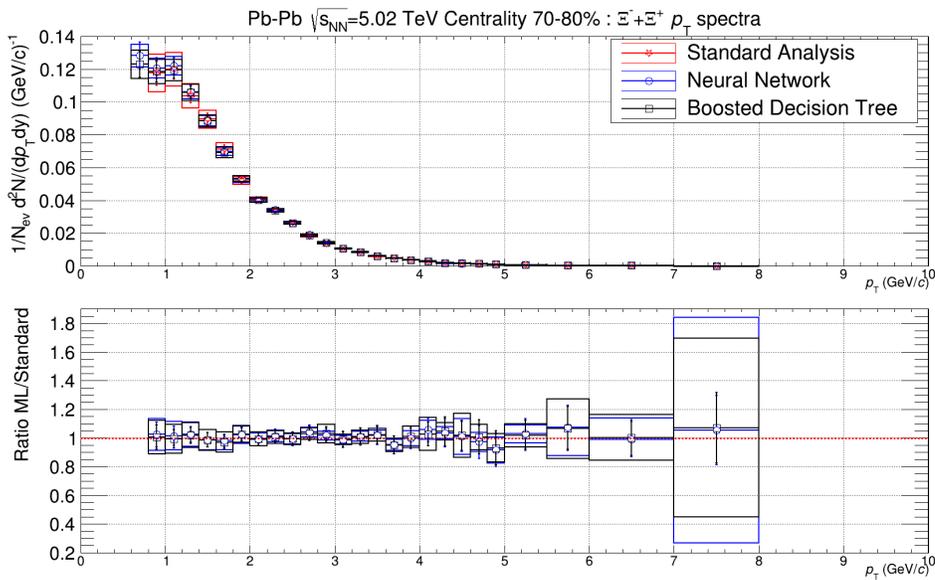


Figure A.7: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 70-80% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

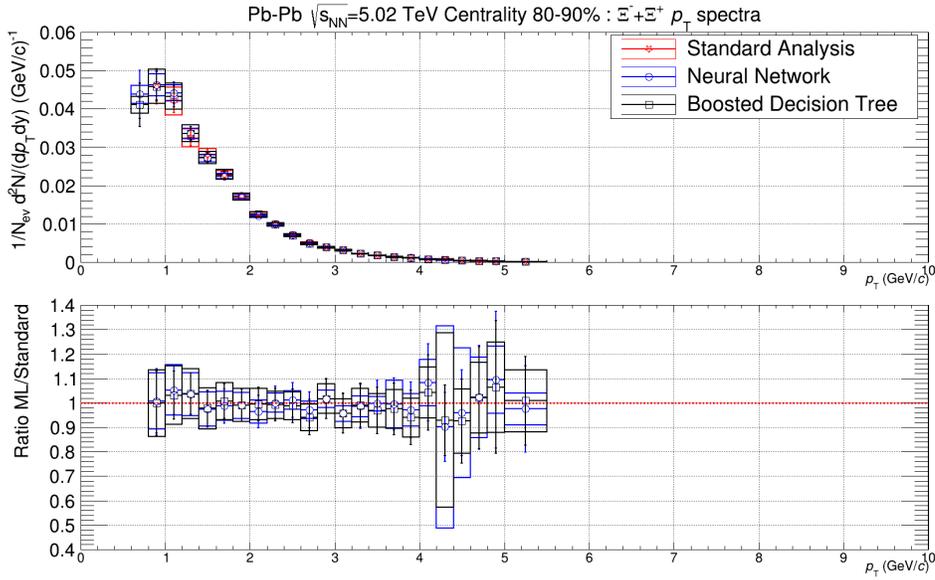


Figure A.8: Comparison of obtained corrected p_T spectra of $\Xi^- + \Xi^+$ in 80-90% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

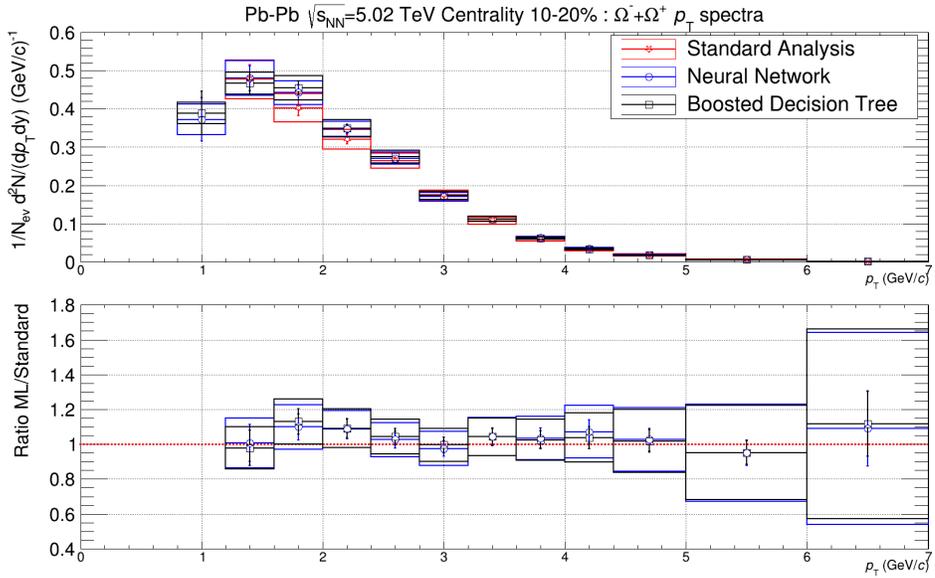


Figure A.9: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 10-20% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

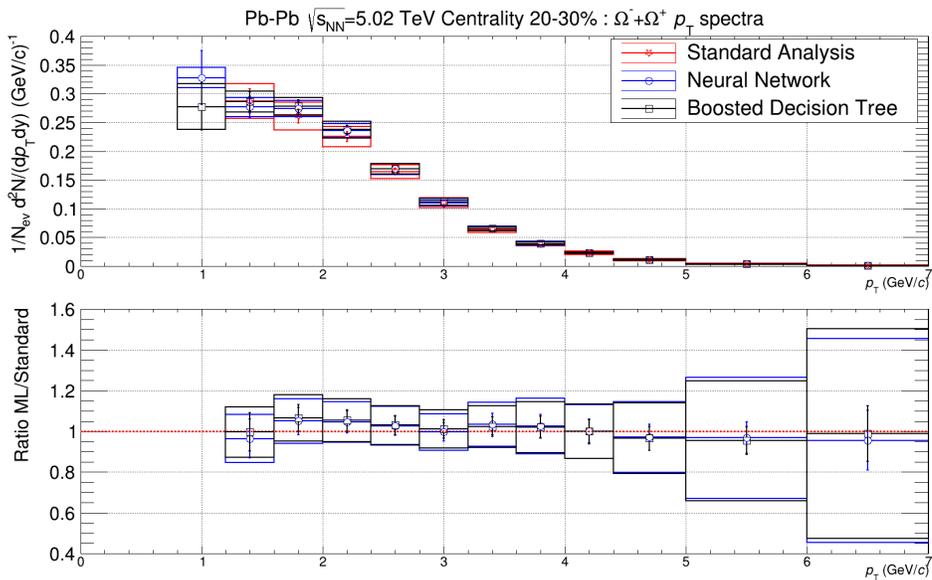


Figure A.10: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 20-30% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

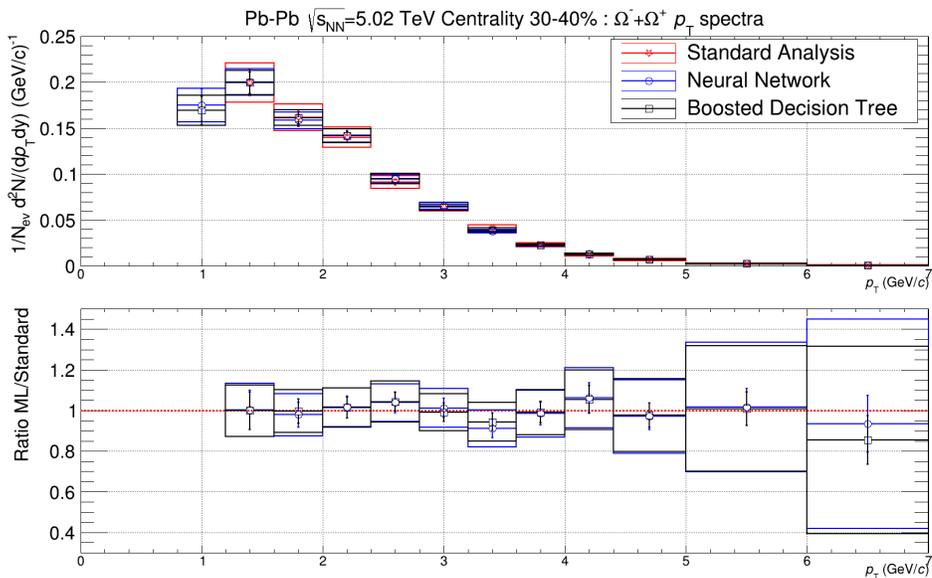


Figure A.11: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 30-40% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

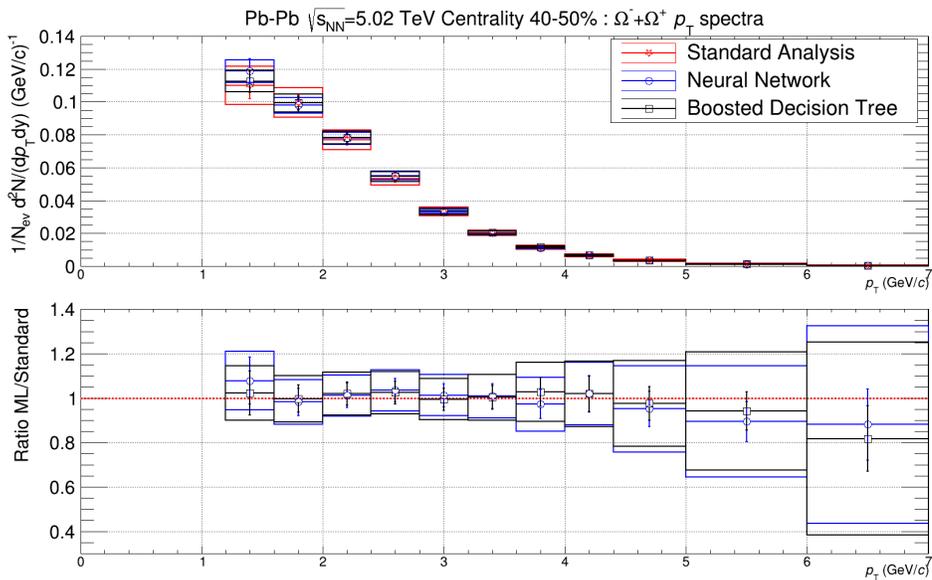


Figure A.12: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 40-50% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

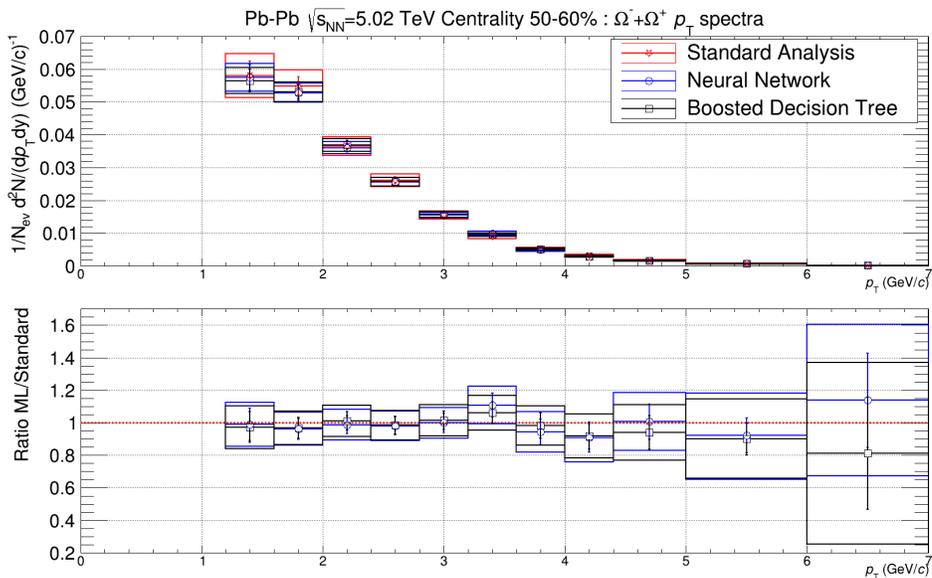


Figure A.13: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 50-60% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

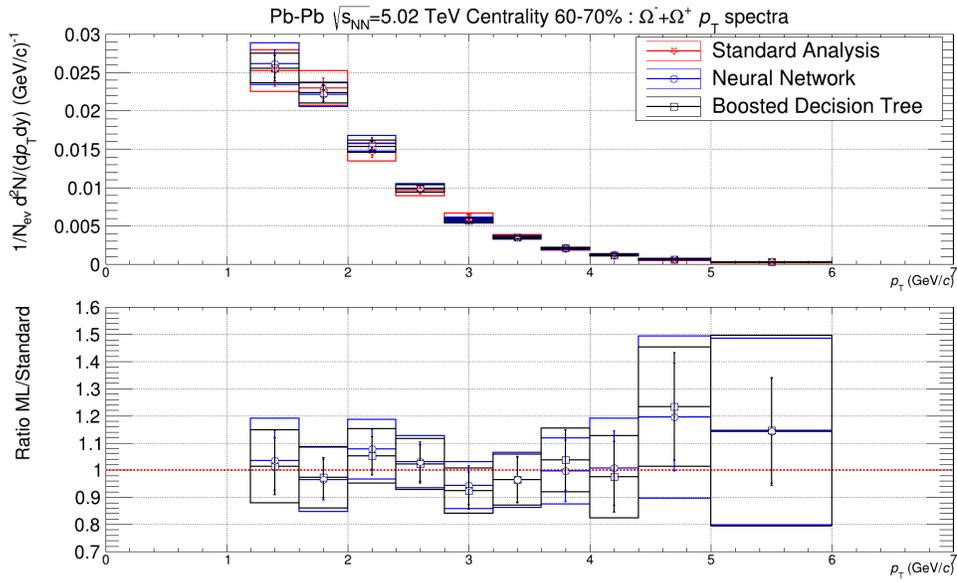


Figure A.14: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 60-70% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

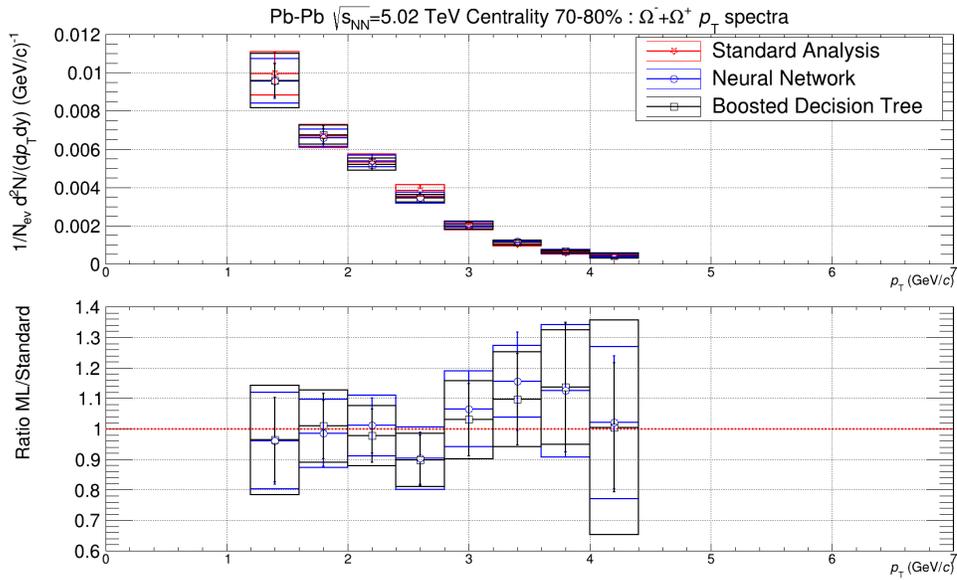


Figure A.15: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 70-80% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

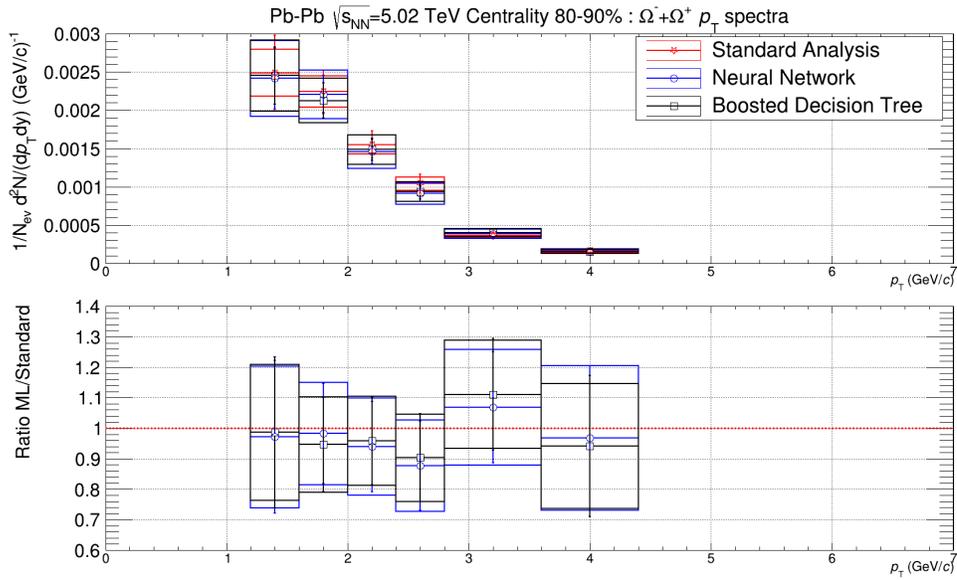


Figure A.16: Comparison of obtained corrected p_T spectra of $\Omega^- + \Omega^+$ in 80-90% centrality class. Top panel show the spectra of the standard analysis in red, of the neural network in blue and of the BDT in black. Line error bars are the statistical uncertainties and box errors are the total systematic uncertainties of each spectra. Bottom panel presents the ratio between the ML based analysis and the standard analysis.

Appendix B

Codes developed for the ALICE framework

Código B.1: AliMachineLearning.h

```

1  #ifndef AliMachineLearning_H
2  #define AliMachineLearning_H
3  #include <TNamed.h>
4
5  //+++++
6  // TObject to hold Machines
7  // virtual class: other specific classes will derive from this
8  // but share the interface!
9  //+++++
10
11 class AliMachineLearning : public TNamed {
12
13 public:
14     //Simple constructor
15     AliMachineLearning();
16
17     //TNamed-inspired constructor
18     AliMachineLearning(const char * name, const char * title = "Machine_Learning");
19
20     //Simple destructor
21     ~AliMachineLearning();
22
23     void Clear(Option_t* = "") {}; //dummy
24
25     virtual void LoadModel(TString lModelName) {};
26
27     virtual double Predict(double *X, int K) { return 0.0; };
28
29     //virtual void Print(Option_t *option="") {};
30
31 private:
32     ClassDef(AliMachineLearning, 1);
33 };
34 #endif

```

Código B.2: AliMachineLearning.cxx

```

1  //+-----+
2  // TObject to hold Machines
3  // virtual class: other specific classes will derive from this
4  // but share the interface!
5  // This is a base class for AliNeuralNetwork and AliBDT
6  //+-----+
7
8  #include "AliMachineLearning.h"
9  #include <iostream>
10 #include <TROOT.h>
11 using namespace std;
12
13 ClassImp(AliMachineLearning);
14
15 //-----
16 AliMachineLearning::AliMachineLearning() : TNamed()
17 {
18     // Dummy Constructor - not to be used!
19 }
20
21 //-----
22 AliMachineLearning::AliMachineLearning(const char * name, const char * title) : TNamed(name, title)
23 {
24     // TNamed-inspired constructor
25 }
26
27 //-----
28 AliMachineLearning::~AliMachineLearning()
29 {
30     // A boring class. Nothing to delete.
31 }

```

Código B.3: AliNeuralNetwork.h

```

1  #ifndef AliNeuralNetwork_H
2  #define AliNeuralNetwork_H
3  #include <TNamed.h>
4  #include "AliMachineLearning.h"
5  #include "TFile.h"
6  #include "TH2.h"
7  #include "TH1.h"
8
9  //+++++
10 // TObject to hold Neural Network
11 //+++++
12
13 class AliNeuralNetwork : public AliMachineLearning {
14
15 public:
16 //Dummy Constructor
17 AliNeuralNetwork ();
18
19 //Standard Constructor
20 AliNeuralNetwork(const char * name, const char * title = "Neural_Network");
21
22 //Simple destructor
23 ~AliNeuralNetwork ();
24
25 //Interface to configure parameters of the machine
26 void LoadModel(TString lModelName);
27
28 double Predict(double* X, int K);
29
30 //void Print(Option_t *option="");
31
32 private:
33
34 //Histograms to store synaptic weights of the Neural Network
35 TH2D* fW1;
36 TH1D* fB1;
37
38 TH2D* fW2;
39 TH1D* fB2;
40
41 TH2D* fW3;
42 TH1D* fB3;
43
44 TH2D* fW4;
45 TH1D* fB4;
46
47 TH2D* fW5;
48 TH1D* fB5;
49 //-----
50
51 ClassDef(AliNeuralNetwork, 1)
52 };
53 #endif

```

Código B.4: AliNeuralNetwork.cxx

```

1  //+-----+
2  // TOBject to hold V0 configuration + results histogram
3  //+-----+
4
5  #include <TNamed.h>
6  #include "AliMachineLearning.h"
7  #include "AliNeuralNetwork.h"
8  #include "TFile.h"
9  #include "TH2.h"
10 #include "TH1.h"
11 #include "TMath.h"
12 #include <iostream>
13 #include <TROOT.h>
14 using namespace std;
15
16 ClassImp(AliNeuralNetwork);
17 //-----
18 AliNeuralNetwork::AliNeuralNetwork() : AliMachineLearning(),
19     fW1(0), fB1(0), fW2(0), fB2(0), fW3(0), fB3(0), fW4(0), fB4(0), fW5(0), fB5(0)
20 {
21     // default constructor, don't allocate memory here!
22     // this is used by root for IO purposes, it needs to remain empty
23 }
24 //-----
25 AliNeuralNetwork::AliNeuralNetwork(const char * name, const char * title) : AliMachineLearning(name, title),
26     fW1(0), fB1(0), fW2(0), fB2(0), fW3(0), fB3(0), fW4(0), fB4(0), fW5(0), fB5(0)
27 {
28     // Named constructor
29 }
30
31 }
32
33 //-----
34 /*
35 AliNeuralNetwork::AliNeuralNetwork(const AliNeuralNetwork& lCopyMe, TString lNewName) : AliMachineLearning(lCopyMe),
36 {
37     SetName( lNewName.Data() );
38 }
39 */
40 //-----
41 AliNeuralNetwork::~AliNeuralNetwork()
42 {
43
44     // Proper destructor: delete pointer data member
45     if(fW1){delete fW1; fW1 = 0x0;}
46     if(fB1){delete fB1; fB1 = 0x0;}
47     if(fW2){delete fW2; fW2 = 0x0;}
48     if(fB2){delete fB2; fB2 = 0x0;}
49     if(fW3){delete fW3; fW3 = 0x0;}
50     if(fB3){delete fB3; fB3 = 0x0;}
51     if(fW4){delete fW4; fW4 = 0x0;}
52     if(fB4){delete fB4; fB4 = 0x0;}
53     if(fW5){delete fW5; fW5 = 0x0;}
54     if(fB5){delete fB5; fB5 = 0x0;}
55 }
56
57 //-----
58 /*
59 AliNeuralNetwork& AliNeuralNetwork::operator=(const AliNeuralNetwork& lCopyMe)
60 {
61     if (&lCopyMe == this) return *this;
62     // Careful with names
63     SetName(lCopyMe.GetName());
64     SetTitle(lCopyMe.GetTitle());
65
66     return *this;
67 }
68 */
69 //-----
70
71 void AliNeuralNetwork::LoadModel(TString lModelName)
72 //Function to load synaptic weights of neural network and store them in the data members
73 {
74     TFile* FileModel = 0x0;
75     FileModel = TFile::Open(lModelName.Data());
76
77     //Check existence, please
78     if(!FileModel) cout << Form("Cannot open requested Model file %s", lModelName.Data());
79     if(!FileModel->IsOpen()) cout << Form("Cannot open Model file %s", lModelName.Data());
80
81     TH2D* IW1 = (TH2D*)FileModel->Get("W1");
82     if(!IW1) cout << Form("File %s does not contain valid W1 layer", lModelName.Data());
83     IW1->SetName("W1_to_copy");
84     fW1 = (TH2D*) IW1->Clone("W1");
85
86     TH1D* IB1 = (TH1D*)FileModel->Get("B1");
87     if(!IB1) cout << Form("File %s does not contain valid B1 layer", lModelName.Data());

```

```

88  lB1->SetName("B1_to_copy");
89  fB1 = (TH1D*) lB1->Clone("B1");
90
91  TH2D* lW2 = (TH2D*)FileModel->Get("W2");
92  if(!lW2) cout << Form("File_%s_does_note_contain_valid_W2_layer", lModelName.Data());
93  lW2->SetName("W2_to_copy");
94  fW2 = (TH2D*) lW2->Clone("W2");
95
96  TH1D* lB2 = (TH1D*)FileModel->Get("B2");
97  if(!lB2) cout << Form("File_%s_does_note_contain_valid_B2_layer", lModelName.Data());
98  lB2->SetName("B2_to_copy");
99  fB2 = (TH1D*) lB2->Clone("B2");
100
101  TH2D* lW3 = (TH2D*)FileModel->Get("W3");
102  if(!lW3) cout << Form("File_%s_does_note_contain_valid_W3_layer", lModelName.Data());
103  lW3->SetName("W3_to_copy");
104  fW3 = (TH2D*) lW3->Clone("W3");
105
106  TH1D* lB3 = (TH1D*)FileModel->Get("B3");
107  if(!lB3) cout << Form("File_%s_does_note_contain_valid_B3_layer", lModelName.Data());
108  lB3->SetName("B3_to_copy");
109  fB3 = (TH1D*) lB3->Clone("B3");
110
111  TH2D* lW4 = (TH2D*)FileModel->Get("W4");
112  if(!lW4) cout << Form("File_%s_does_note_contain_valid_W4_layer", lModelName.Data());
113  lW4->SetName("W4_to_copy");
114  fW4 = (TH2D*) lW4->Clone("W4");
115
116  TH1D* lB4 = (TH1D*)FileModel->Get("B4");
117  if(!lB4) cout << Form("File_%s_does_note_contain_valid_B4_layer", lModelName.Data());
118  lB4->SetName("B4_to_copy");
119  fB4 = (TH1D*) lB4->Clone("B4");
120
121  TH2D* lW5 = (TH2D*)FileModel->Get("W5");
122  if(!lW5) cout << Form("File_%s_does_note_contain_valid_W5_layer", lModelName.Data());
123  lW5->SetName("W5_to_copy");
124  fW5 = (TH2D*) lW5->Clone("W5");
125
126  TH1D* lB5 = (TH1D*)FileModel->Get("B5");
127  if(!lB5) cout << Form("File_%s_does_note_contain_valid_B5_layer", lModelName.Data());
128  lB5->SetName("B5_to_copy");
129  fB5 = (TH1D*) lB5->Clone("B5");
130
131
132  fW1->SetDirectory(nullptr);
133  fB1->SetDirectory(nullptr);
134  fW2->SetDirectory(nullptr);
135  fB2->SetDirectory(nullptr);
136  fW3->SetDirectory(nullptr);
137  fB3->SetDirectory(nullptr);
138  fW4->SetDirectory(nullptr);
139  fB4->SetDirectory(nullptr);
140  fW5->SetDirectory(nullptr);
141  fB5->SetDirectory(nullptr);
142  FileModel->Close();
143 }
144
145 //-----
146 double AliNeuralNetwork::Predict(double* X, int K)
147 //Function to return neural network prediction
148 {
149
150  if(!fW1||!fB1||!fW2||!fB2||!fW3||!fB3||!fW4||!fB4||!fW5||!fB5)
151  {
152      cout << "Cannot Predict: One or more of the NN layers are empty!" << endl;
153      return(0);
154  }
155
156  //PERFORM FORWARD FUNCTION THROUGH NEURAL NETWORK
157
158  //1st: [1][K]x[K][256] = [1][256]
159  double v1[256] = {0};
160  for(Int_t j=0; j<256; j++)
161  {
162      //Layer product
163      for(Int_t k=0; k<K; k++)
164          v1[j] += X[k]*fW1->GetBinContent(k+1,j+1);
165      //Bias sum
166      v1[j]+=fB1->GetBinContent(j+1);
167      //Activation function: Sigmoid
168      v1[j] = 1.0/(1.0+TMath::Exp(-1.0*v1[j]));
169  }
170
171  //2nd: [1][256]x[256][64] = [1][64]
172  double v2[64] = {0};
173  for(Int_t j=0; j<64; j++)
174  {
175      //Layer product

```

```

176     for (Int_t k=0; k<256; k++)
177         v2[j] += v1[k]*fW2->GetBinContent(k+1,j+1);
178     //Bias sum
179     v2[j]+=fB2->GetBinContent(j+1);
180     //Activation function: Sigmoid
181     v2[j] = 1.0/(1.0+TMath::Exp(-1.0*v2[j]));
182 }
183
184 //3rd: [1][64]x[64][16] = [1][16]
185 double v3[16] = {0};
186 for (Int_t j=0; j<16; j++)
187 {
188     //Layer product
189     for (Int_t k=0; k<64; k++)
190         v3[j] += v2[k]*fW3->GetBinContent(k+1,j+1);
191     //Bias sum
192     v3[j]+=fB3->GetBinContent(j+1);
193     //Activation function: Sigmoid
194     v3[j] = 1.0/(1.0+TMath::Exp(-1.0*v3[j]));
195 }
196
197 //4th: [1][16]x[16][4] = [1][4]
198 double v4[4] = {0};
199 for (Int_t j=0; j<4; j++)
200 {
201     //Layer product
202     for (Int_t k=0; k<16; k++)
203         v4[j] += v3[k]*fW4->GetBinContent(k+1,j+1);
204     //Bias sum
205     v4[j]+=fB4->GetBinContent(j+1);
206     //Activation function: Sigmoid
207     v4[j] = 1.0/(1.0+TMath::Exp(-1.0*v4[j]));
208 }
209
210 //5th: [1][4]x[4][1] = [1][1]
211 double v5 = 0;
212 //Layer product
213 for (Int_t k=0; k<4; k++)
214     v5 += v4[k]*fW5->GetBinContent(k+1,1);
215 //Bias sum
216 v5+=fB5->GetBinContent(1);
217 //Activation function: Sigmoid
218 v5 = 1.0/(1.0+TMath::Exp(-1.0*v5));
219
220 return (v5);
221 }
222
223 //-----
224 /*
225 void AliNeuralNetwork::Print (Option_t *option="")
226 {
227     cout<<"=====*<<endl;
228     cout<<"    AliNeuralNetwork Configuration    *<<endl;
229     cout<<"=====*<<endl;
230
231     return;
232 }
233 */

```

Código B.5: AliBDT.h

```

1  #ifndef AliBDT_H
2  #define AliBDT_H
3  #include <TNamed.h>
4  #include "AliMachineLearning.h"
5  #include "TFile.h"
6  #include "TH1.h"
7
8  //+++++
9  // TObject to hold Boosted Decision Trees
10 //+++++
11
12 class AliBDT : public AliMachineLearning {
13
14 public:
15     //Dummy Constructor
16     AliBDT();
17
18     //Standard Constructor
19     AliBDT(const char * name, const char * title = "BDT");
20
21     //Simple destructor
22     ~AliBDT();
23
24     //Interface to configure parameters of the machine
25     void LoadModel(TString lModelName);
26
27     double Predict(double* X, int K);
28
29     //void Print(Option_t *option="");
30
31 private:
32
33     //Histograms to store BDT parameters
34     TH1* fFt; //Feature to avaiate
35     TH1D* fSoS; //Split of corresponding Feature or Score on the leaf
36     //-----
37
38     ClassDef(AliBDT, 1)
39 };
40 #endif

```

Código B.6: AliBDT.cxx

```

1  #include <TNamed.h>
2  #include "AliMachineLearning.h"
3  #include "AliBDT.h"
4  #include "TFile.h"
5  #include "TH1.h"
6  #include "TMath.h"
7  #include <iostream>
8  #include <TROOT.h>
9  using namespace std;
10
11  ClassImp(AliBDT);
12  //-----
13  AliBDT::AliBDT() : AliMachineLearning(),
14                  fFt(0), fSoS(0)
15  {
16      // default constructor, don't allocate memory here!
17      // this is used by root for IO purposes, it needs to remain empty
18  }
19  //-----
20  AliBDT::AliBDT(const char * name, const char * title) : AliMachineLearning(name, title),
21                  fFt(0), fSoS(0)
22  {
23      // Named constructor
24  }
25
26  }
27
28  //-----
29  /*
30  AliBDT::AliBDT(const AliBDT& lCopyMe, TString lNewName) : AliMachineLearning(lCopyMe),
31  {
32      SetName( lNewName.Data() );
33  }
34  */
35  //-----
36  AliBDT::~AliBDT()
37  {
38
39      // Proper destructor: delete pointer data member
40      if(fFt){delete fFt; fFt = 0x0;}
41      if(fSoS){delete fSoS; fSoS = 0x0;}
42  }
43
44  //-----
45  /*
46  AliBDT& AliBDT::operator=(const AliBDT& lCopyMe)
47  {
48      if (&lCopyMe == this) return *this;
49      // Careful with names
50      SetName(lCopyMe.GetName());
51      SetTitle(lCopyMe.GetTitle());
52
53      return *this;
54  }
55  */
56  //-----
57  void AliBDT::LoadModel(TString lModelName)
58  //Function to load synaptic weights of neural network and store them in the data members
59  {
60      TFile* FileModel = 0x0;
61      FileModel = TFile::Open(lModelName.Data(), "READ");
62
63      //Check existence, please
64      if(!FileModel) cout << Form("Cannot open requested Model file %s", lModelName.Data());
65      if(!FileModel->IsOpen()) cout <<Form("Cannot open Model file %s", lModelName.Data());
66
67      TH1* lFt = (TH1*)FileModel->Get("FeatureIndex");
68      if(!lFt) cout << Form("File %s does not contain valid data member", lModelName.Data());
69      lFt->SetName("FeatureIndex_to_copy");
70      fFt = (TH1*)lFt->Clone("FeatureIndex");
71
72      TH1D* lSoS = (TH1D*)FileModel->Get("Split_or_Score");
73      if(!lSoS) cout << Form("File %s does not contain valid data member", lModelName.Data());
74      lSoS->SetName("Split_or_Score_to_copy");
75      fSoS = (TH1D*)lSoS->Clone("Split_or_Score");
76
77      fFt->SetDirectory(nullptr);
78      fSoS->SetDirectory(nullptr);
79      FileModel->Close();
80  }
81
82  //-----
83  double AliBDT::Predict(double* X, int Nt)
84  //Function to return BDT prediction
85  //Nt = number of trees
86  {
87

```

```

88  if(!fFt||!fSoS)
89  {
90      cout << "Can not Predict : One or more of the data members are empty!" << endl;
91      return(0);
92  }
93
94  //PERFORM FORWARD FUNCTION THROUGH BDT
95
96  double P = 0.0;
97
98  int nodes = int(fFt->GetNbinsX()/Nt); //nodes per tree = 2^(tree_depth + 1) - 1
99
100 for(int i=0; i<Nt; i++)
101 {
102     //booster i structure
103     int Lidx = 1; //local index of a node in each tree [1,nodes]
104     int Gidx = (nodes*i)+Lidx; //global index of a node in Nt trees
105
106     while( Lidx < (nodes+1)/2 )
107     {
108         if( X[int(fFt->GetBinContent(Gidx))] < fSoS->GetBinContent(Gidx) ) Lidx = 2*Lidx;
109
110         else Lidx = 2*Lidx+1;
111
112         Gidx = (nodes*i)+Lidx;
113     }
114
115     //Sum leaf score
116     P += fSoS->GetBinContent(Gidx);
117 }
118
119 /*
120 //100 boosters
121 for(int i=0; i<100; i++)
122 {
123     //Booster variables
124     int idx0 = int( fFt->GetBinContent((7*i)+1) );
125     double Split0 = fSoS->GetBinContent((7*i)+1);
126
127     int idx1 = int( fFt->GetBinContent((7*i)+1+1) );
128     double Split1 = fSoS->GetBinContent((7*i)+1+1);
129
130     int idx2 = int( fFt->GetBinContent((7*i)+2+1) );
131     double Split2 = fSoS->GetBinContent((7*i)+2+1);
132
133     //3 to 6 should be leaves
134     int idx3 = int( fFt->GetBinContent((7*i)+3+1) );
135     double Score3 = fSoS->GetBinContent((7*i)+3+1);
136
137     int idx4 = int( fFt->GetBinContent((7*i)+4+1) );
138     double Score4 = fSoS->GetBinContent((7*i)+4+1);
139
140     int idx5 = int( fFt->GetBinContent((7*i)+5+1) );
141     double Score5 = fSoS->GetBinContent((7*i)+5+1);
142
143     int idx6 = int( fFt->GetBinContent((7*i)+6+1) );
144     double Score6 = fSoS->GetBinContent((7*i)+6+1);
145
146     if(idx3+idx4+idx5+idx6!=-4)
147         cout << "somethin is wrong with a leaf!" << endl;
148
149     //booster i structure
150     if( X[idx0]<Split0 )
151     {
152         //0:yes
153         if(X[idx1]<Split1)
154             //1:yes
155             P+=Score3;
156         else
157             //1:no
158             P+=Score4;
159     }
160     else
161     {
162         //0:no
163         if(X[idx2]<Split2)
164             //2:yes
165             P+=Score5;
166         else
167             //2:no
168             P+=Score6;
169     }
170 }
171 */
172 P = 1.0/(1.0+TMath::Exp(-1.0*P));
173
174 return(P);
175 }

```

```
176
177 //-----
178 /*
179 void AliBDT::Print (Option_t *option="")
180 {
181     cout<<"=====*<<endl;
182     cout<<"    AliBDT Configuration    "<<endl;
183     cout<<"=====*<<endl;
184
185     return;
186 }
187 */
```