Aplicação de métodos de árvore de decisão na identificação de mésons D^0 em colisões PbPb a $\sqrt{s_{NN}} = 2{,}76\ \mathrm{TeV}$ no detector CMS do LHC-CERN

Trabalho de Diplomação em Engenharia Física

Universidade Federal do Rio Grande do Sul - Instituto de Física - Escola de Engenharia

Guilherme Hoss

Orientador: César Augusto Bernardes

Agosto de 2023

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

Quando laboratórios de física de partículas, como o CERN (*European Organization for Nuclear Research*), realizam seus experimentos através de um acelerador de partículas, a quantidade de partículas sub-atômicas gerada por uma colisão entre íons pesados - por exemplo - é monumental. Além disso, cada colisão precisa ser processada pelos detectores em uma escala de tempo da ordem de poucos microssegundos.

As colisões entre íons pesados, especificamente, são realizadas com o intuito de recriar condições similares às do início do universo que ocorreram alguns microssegundos após o Big Bang. Tais medidas são cruciais na compreensão da natureza das interações fortes em um regime de alta densidade de energia. Para tal, são utilizados núcleos pesados como o de ouro (Au) ou de chumbo (Pb), que são acelarados por um acelerador de partículas, como o LHC (*Large Hadron Collider*). Essas partículas podem alcançar trilhões de eV de energia durante o experimento, o que se prova intenso o suficiente para que se crie uma "bola de fogo" que "derrete" todos os prótons e nêutrons dos núcleos durante a colisão. Nesta "bola de fogo" criada, a densidade de energia existente é grande o suficiente para observarmos quarks e glúons desconfinados dos prótons e nêutrons dos núcleos [4].

Na sequência desse processo, tudo que havia sido "derretido" se mistura em uma substância composta majoritariamente de quarks e glúons. Essa substância possui propriedades mais próximas de um fluido do que de um gás, por isso, recebe o nome de QGP (*Quark–Gluon Plasma*). A bola de fogo então esfria de maneira quase imediata (a duração do fluido é da ordem de 10^{-21} segundo). Após esse resfriamento, todos os quarks e glúons se recombinam em diversos bárions (a exemplo dos prótons) e mésons - como o D^0 (ver Fig. 1).

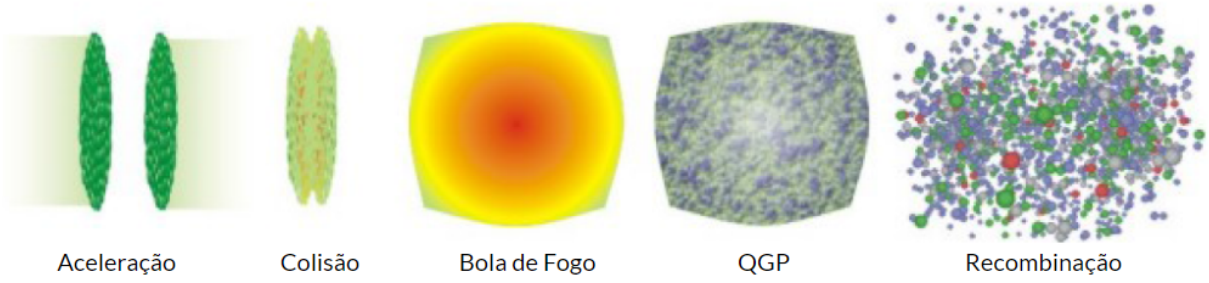

Figura 1 – A sequência de processos resultantes de uma colisão entre íons pesados (fonte: [1]).

Para uma análise precisa desses fenômenos, as milhares de partículas que resultam do resfriamento passam por detectores acoplados a sistemas de leitura e coleta de sinais. Em um estágio posterior, utilizando clusters de computadores, a combinação desses sinais é utilizada para obtermos propriedades das partículas, como trajetória espacial e energia.

Para que se torne viável a análise da maior parte possível da informação detectada pelos dispositivos presentes no acelerador, em muitos casos, faz-se necessária a utilização de

métodos de machine learning que podem processar rapidamente grandes quantidades de dados e nos prover com as informações que gostaríamos de extrair a partir da colisão realizada. Neste trabalho testamos diferentes métodos de árvore de decisão para identificação de partículas em amostras de dados de simulações de colisões entre núcleos de Pb que acontecem no LHC. Mais especificamente, na identificação de mésons D^0 (compostos por um quark charm c e um quark anti-up $\overline{u}$), através de seu decaimento em um káon e um píon carregados. Utilizamos uma amostra de teste com simulações de Monte Carlo (MC) de colisões PbPb com energia no centro de massa por par de núcleons de $\sqrt{s_{NN}} = 2{,}76$ TeV produzida com o gerador HYDJET [5] (responsável pela modelagem das colisões entre íons pesados) acoplado ao gerador PYTHIA8 [6] (responsável pela geração dos mésons D^0). A simulação do detector Compact Muon Solenoid (CMS) do LHC-CERN e os dos efeitos das interações das partículas produzidas nas colisões com o material do detector é feita utilizando o GEANT4 [7]. Esta amostra foi feita apenas para o intervalo de centralidade de 10–30% (onde centralidade é o grau de sobreposição entre os núcleos, sendo 0% a configuração com maior sobreposição). O número de eventos gerados (número de colisões simuladas) nessa configuração foi de 1004240.

Neste trabalho descrevemos o processo de escolha de um método de árvore de decisão adequado ao nosso problema de identificação de mésons D^0. Para isso, determinamos quais das variáveis relacionadas ao méson D^0 e suas partículas "filhas" (produto de seu decaimento) são mais relevantes para o treinamento do método; Analisamos o tempo que é necessário para fazer o treinamento da árvore de decisão utilizando diferentes variáveis; Treinamos e aplicamos o método de árvore de decisão com alta significância estastística de sinal (mésons D^0 produzidos nas colisões) em comparação ao ruído (partículas produzidas na colisão que não são mésons D^0). Comparamos a significância estatística de sinal de nosso método escolhido com métodos desenvolvidos em artigos da colaboração CMS do LHC.

2 MÉSON D^0

A partícula que será o objeto de interesse deste trabalho é o méson D^0. Conforme o QGP resfria, quarks de "sabores pesados" (como o quark charm, de massa $1{,}27 \pm 0{,}02$ GeV [8]) atravessam esse fluido e auxiliam na formação do méson D^0 ao se juntarem com um anti-quark up presente no plasma. Após formado, o D^0 viaja - em média - apenas alguns poucos milímetros antes de decair em outras partículas. Dado que as primeiras camadas de material sensível do detector CMS estão a uma distância de alguns poucos centímetros do ponto de colisão, isso signifca que os mésons D^0 não chegam longe o suficiente para serem detectados diretamente, apenas as partículas nas quais os mesmos decaíram (píons π^+ e káons K^-) são. Portanto, a fim de detectarmos essas partículas, uma espécie de engenharia reversa deve ser feita através de diversos parâmetros que serão melhor explorados na seção dedicada à metodologia do trabalho. Como o quark charm "carrega" consigo todas essas interações com o plasma, o méson D^0 - por consequência - também possui essa informação. Por isso o interesse deste estudo em identificá-lo

em meio a um mar de outras partículas produzidas na colisão.

Na imagem 2 foi realizada uma ilustração desse processo e de um corte transversal do detector CMS e suas camadas com as trajetórias de um píon e káon representadas. Como mostrado nessa figura, o processo indicado (formação e decaimento do D^0) acontece praticamente no ponto central de colisão e fora do alcance das camadas do detector.

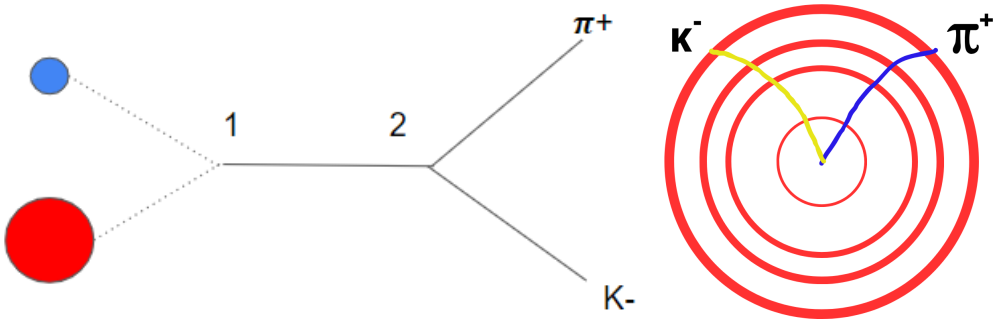

Figura 2 – Na imagem à esquerda está ilustrado o processo de formação e decaimento do méson D^0. O círculo vermelho representa um quark charm e o círculo azul representa um anti-quark up. O ponto 1 indica o momento de formação do méson através da união dos quarks. O ponto 2 indica o momento quando o D^0 decai em suas duas partículas filhas: o píon e o káon. À direita, um esquema representando um corte transversal das camadas do detector de traços do CMS e as trajetórias das partículas filhas representadas em relação às camadas do detector.

Resumidamente, o estudo destas partículas é de grande interesse quando desejamos entender melhor o comportamento do QGP que - como dito na introdução - dura pouquíssimo tempo após uma colisão entre íons pesados.

2.1 Distribuições das variáveis

Com o propósito de ter uma noção do poder de separação entre sinal e ruído das variáveis que serão apresentadas na seção 4.1, a primeira etapa deste trabalho foi baseada na geração de histogramas das mesmas utilizando uma macro em C++ do ROOT (seção 3.3) [9]. Em razão de suas capacidades ilustrativas, são exemplificados 2 histogramas das 24 variáveis analisadas: D3DPointingAngle e DDca.

A variável denominada de D3DPointingAngle apresenta uma grande diferença nas curvas de sinais verdadeiros e falsos, o que sugere um grande potencial para a separação de ruído e sinal, como pode ser visto na Fig. 3. Por outro lado, em ambos os histogramas do DDca, Fig. 4, normalizado ou não, como os mésons D^0 simulados são produzidos em estágios iniciais da colisão, os mesmos devem apresentar um baixo valor de DCA, o que difere um pouco do ruído, que possui valores maiores de DDca em geral.

Em ambas as imagens os mésons D^0 estão representados em azul (Sinal), enquanto as partículas consideradas ruído são representadas em vermelho (Ruído) e a soma das duas

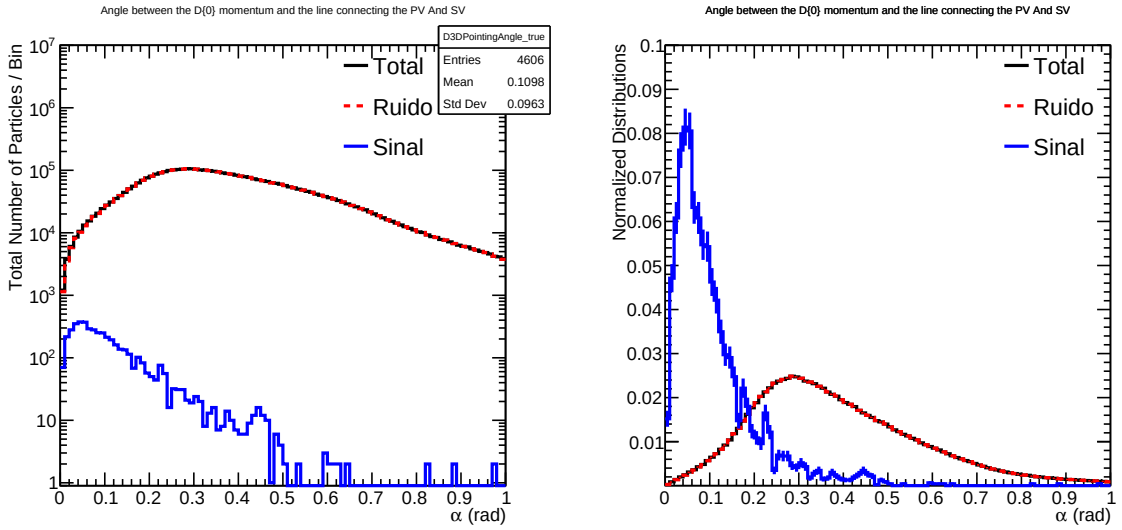

Figura 3 – Gráficos representando o ângulo (no espaço 3D) entre o momentum do méson D^0 e a linha conectando os vértices primários e secundários. Em azul: partículas verdadeiras. Em vermelho: partículas falsas. Em preto: o somatório de todas as partículas. O gráfico da esquerda representa o número total de partículas produzidas nas colisões, mostrando a proporção entre sinal e ruído. No gráfico da direita as distribuições foram normalizadas pela respectiva integral, ou seja, área sob a curva igual a 1.

distribuições está representada em preto (Total). Além disso, podemos ver em cada uma das imagens mencionadas a quantidade total de partículas de sinal, sua média e seu desvio padrão.

O restante das distribuições das variáveis pode ser encontrado no apêndice 6. Note que, embora a comparação entre sinal e ruído de uma dada variável muitas vezes não mostre nenhum ganho em termos de separação de sinal, podem existir correlações entre variáveis, de modo que ao fazer um corte em uma dada variável, as distribuições de outras variáveis podem mudar consideravelmente, consequentemente mudando o grau de separação entre sinal e ruído.

3 MACHINE LEARNING

Diferentemente de um código programado de maneira explícita, os métodos de machine learning (ou aprendizado de máquina) são coleções de algoritmos que permitem que os computadores identifiquem padrões através de uma grande quantidade de dados sem instruções explícitas. Dentro do campo do machine learning, são empregados diferentes critérios para a classificação dos métodos. Uma maneira comum de dividi-los é através das nomenclaturas de *White Box Machine Learning* e *Black Box Machine Learning*. Modelos simples como regressão linear ou árvore de decisão singular são considerados como *White Box*, pois são transparentes e facilmente interpretáveis. Dessa maneira, eles provêm diversos *insights* acerca de como decisões e previsões são feitas pelos mesmos. Por outro lado, algoritmos como as redes neurais com

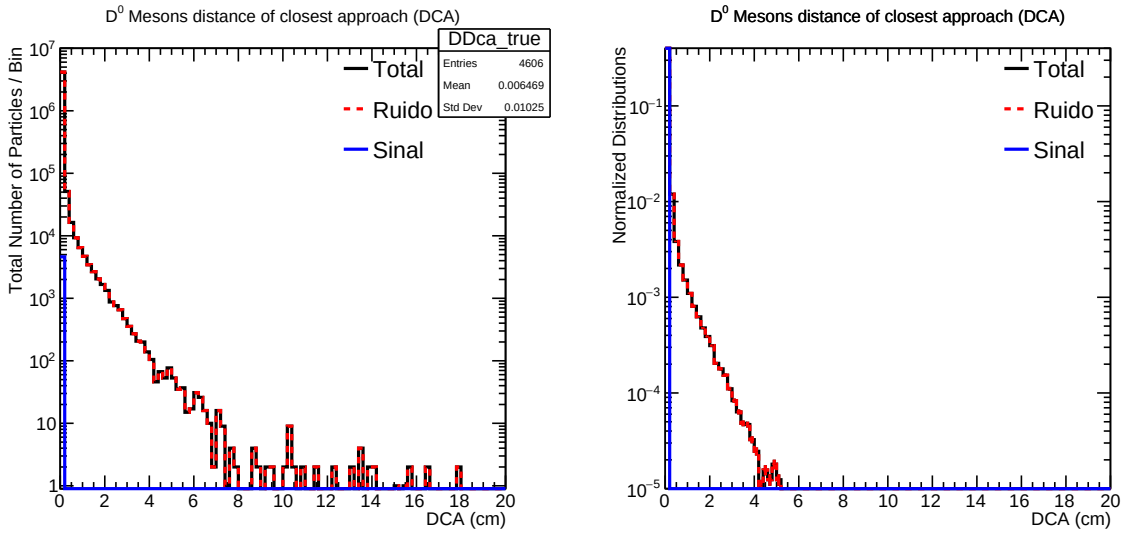

Figura 4 – Gráficos representando a distância mínima do méson D^0 em relação ao vértice primário, *distance of closest approach* (DCA). Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas. A gráfico da esquerda representa o número total de partículas produzidas nas colisões, mostrando a proporção entre sinal e ruído. No gráfico da direita as distribuições foram normalizadas pela respectiva integral, ou seja, área sob a curva igual a 1.

diversas camadas "escondidas" e métodos de *ensemble* - como árvores de decisões aprimoradas - são considerados *Black Box*. Estes métodos possuem uma capacidade maior de interpretar padrões de informação mais complexos, no entanto sacrificam sua transparência e facilidade de interpretação por performance [10].

Devido ao seu grande potencial de aprender padrões e resolver problemas complexos, métodos de alta performance e *black box* são pioneiros e amplamente utilizados em diversas áreas de tecnologia, por isso estão presentes entre os mais diversos tópicos do nosso dia a dia. Alguns exemplos são:

- Reconhecimento de imagens e objetos: as redes neurais convolucionais (*Convolutional Neural Networks* - CNNs) são populares métodos utilizados para fins de classificação e predição de imagens e objetos [11].
- Jogos: unindo aprendizado por reforço (*Reinforcement learning*) e *Deep Learning*, a técnica de aprendizado Deep Q-Network, por exemplo, é utilizada para alcançar níveis superhumanos de performance em jogos como Xadrez, Go e muitos outros [12, 13].
- Reconhecimento de fala: sistemas conhecidos e largamente utilizados como Siri e Alexa usam métodos de aprendizado como o *Deep Learning* para evoluírem e realizarem o reconhecimento de voz de seus usuários [14, 15].

Para este trabalho escolhemos o método da árvore de decisão que é um ingrediente importante em outros métodos mais sofisticados, como a árvore de decisão aprimorada, um método *black box* com alto poder de predição e classificação.

3.1 O Método da Árvore de Decisão

Uma árvore de decisão funciona da seguinte maneira: quando um processo de tomada de decisões complexo é realizado, este método "quebra" essas decisões em uma série de decisões mais simples. Neste método, vários pontos de decisão (nós) são criados e - em cada um deles - será tomada a decisão de seguir por um caminho ou outro (ramos). A partir de um conjunto de dados pré-selecionados para o treinamento, essas decisões são tomadas através da comparação de um atributo numérico com um valor limite ou um atributo nominal com um conjunto de valores possíveis. No final de todas essas subdivisões, os ramos levam às folhas da árvore, que representam o resultado da previsão ou classificação processada pela árvore de decisão [16]. Neste trabalho, este método de machine learning foi utilizado para classificação. Isto é, o trabalho dele - após realizado um treinamento - foi de prever e separar as partículas de nosso interesse do resto.

A fim de explicar brevemente o funcionamento de um algoritmo de classificação, tomemos como exemplo a seguinte situação: dado um conjunto de emails, o algoritmo deve classificá-los em duas categorias: "spam" ou "não spam". Considerando as características de um email como: título, assunto, remetente e corpo, é realizado um treinamento com um conjunto de emails com tais atributos bem definidos e sabendo se os emails são - ou não - spam. Dessa maneira, este método aprende padrões e as ligações que existem entre as características citadas e o fato do email ser considerado spam ou não. Finalmente, o algoritmo poderia ser utilizado para classificar novos emails na categoria de "spam" ou "não spam".

De maneira similar, este método de machine learning foi utilizado para classificar as particulas entre "méson D^0 (sinal)" e "não méson D^0 (ruído)" ao invés de "spam" e "não spam". Ao invés de utilizarmos características de email, foram utilizadas diversas variáveis associadas ao méson D^0. Mais especificamente para o atual trabalho, tendo uma amostra de sinal e ruído, para a definição de uma árvore de decisão, é feito um ranqueamento das variáveis em termos de quais variáveis separam sinal de ruído de forma mais efetiva. Depois, utilizando a variável mais efetiva, aplica-se um corte selecionando dois conjuntos: um com alta taxa de ruído e outro com alta taxa de sinal (isso é feito definindo uma métrica, como por exemplo, o número de partículas de sinal dividido pelo número de partículas de sinal mais ruído, pureza). Em seguida, o processo é repetido para cada um desses conjuntos (o número de repetições do processo é um parâmetro do método, conhecido como profundidade da árvore). Para cada conjunto é associado um *score*, sendo maior o *score* quanto maior a pureza do conjunto em termos de sinal. Depois do treinamento acima, quando aplicarmos o método em uma amostra, cada partícula terminará em um subconjunto com um determinado *score*.

Com o propósito de exemplificar o processo descrito anteriormente e aplicando-o a este trabalho, foi ilustrado na Fig. 5 uma árvore de decisão com profundidade 2 que mostra as tomadas de decisão e os cortes feitos dada a situação de exemplo a seguir: consideremos candidatos a sinal (s_i) e ruído (r_j) descritos por apenas 3 variáveis: x, y e z. O primeiro passo realizado é o ordenamento de cada partícula em termos da variável analizada como, por exemplo:

- $x^{s_4} \leq x^{r_{33}} \leq \cdots \leq x^{r_6} \leq x^{s_{51}}$
- $y^{r_5} \leq y^{r_{24}} \leq \cdots \leq y^{s_7} \leq y^{s_{11}}$
- $z^{r_1} \leq z^{r_{42}} \leq \cdots \leq z^{r_{30}} \leq z^{s_{67}}$

Em seguida, para cada variável, seleciona-se o corte com o maior poder de separação possível entre os candidatos. Em nosso exemplo (com valores e unidades arbitrários), consideremos os seguintes cortes:

- $x < 15$, com separação = 5.
- $y > 7$, com separação = 10.
- $z < -5$ com separação = 3.

Com a informação de que a variável que apresenta maior separação para o corte inicial é y, a árvore de decisão irá separar os candidatos em dois ramos: os que possuem um valor de variável y maior que 7 e os que possuem um valor de y menor que 7. Na sequência, todos os candidatos de cada um desses ramos irá passar por mais um corte e o processo de seleção de cortes é repetido com os novos conjuntos de valores. Por exemplo, para os conjuntos de valores do ramo da esquerda e da direita, respectivamente:

- $x < 8$, com separação = 12.
- $y > 11$, com separação = 2.
- $z < 1$ com separação = 7.

- $x < -2$, com separação = 5.
- $y > 7$, com separação = 1.
- $z < 3$ com separação = 6.

Em nosso exemplo, os candidatos do primeiro ramo serão agora separados pelo corte em $x < 8$, ao mesmo tempo que - no segundo ramo - os outros candidatos serão separados pelo corte realizado em $z < 3$, de acordo com as variáveis que apresentaram maior poder de

separação. Finalmente, as folhas dessa árvore resultam no *score* de cada um dos quatro conjuntos de candidatos resultantes. O *score* está associado diretamente com a pureza do sinal e é calculado através da simples equação $\frac{S}{S+R}$, sendo S a quantidade de partículas consideradas sinal pela árvore e R a quantidade de partículas consideradas ruído pela árvore.

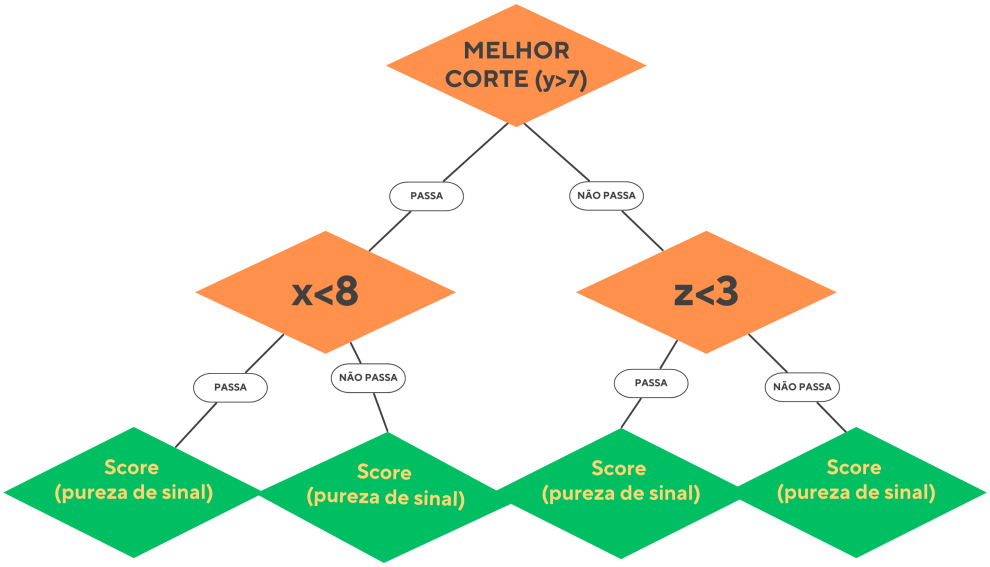

Figura 5 – Uma árvore de decisão ilustrando o processo de classificação neste trabalho.

Como uma singular árvore de decisão é um método relativamente simples, ele desfruta de uma grande interpretabilidade e praticidade no seu treinamento. No entanto, métodos de tamanha simplicidade enfrentam problemas de *overfitting*, que ocorre quando a árvore apresenta um desempenho ótimo utilizando os dados de treinamento, mas apresenta um desempenho abaixo do esperado ao tentar generalizar suas previsões em dados novos. Isto é, o modelo aprendeu somente os ruídos e flutuações estatísticas do treinamento e não os padrões que lhe ajudariam a fazer previsões [17]. A fim de contornar esse problema, utilizamos métodos *ensemble*, que criam múltiplas árvores de decisão com o objetivo de generalizar e criar uma consistência entre os resultados das mesmas, dessa maneira evitando possíveis flutuações estatísticas.

3.2 O Método da Árvore de Decisão Aprimorado com AdaBoost

A técnica de aprendizado da árvore de decisão aprimorado (boosted decision trees, ou BDT) combina os pontos positivos da árvore singular com os benefícios do aprimoramento, ou boosting, resultando em um método robusto e de alta precisão. Neste trabalho, o "tipo" de BDT utilizado é o Adaptive Boosting (ou AdaBoost) [3], frequentemente usado em problemas de classificação e que consiste no treinamento de uma grande quantidade de árvores (da ordem de centenas) de decisão singulares no qual cada árvore irá - separadamente - se destacar positiva ou negativamente em diferentes aspectos. Em seguida neste processo de aprimoramento, é colocada uma ênfase maior no treinamento e na melhoria dos erros que foram encontrados ao longo da árvore. Essa tarefa é então repetida múltiplas vezes a fim de eliminar a presença de erros de

classificação no treinamento das árvores anteriores, dessa maneira produzindo um método de aprendizado eficaz e confiável.

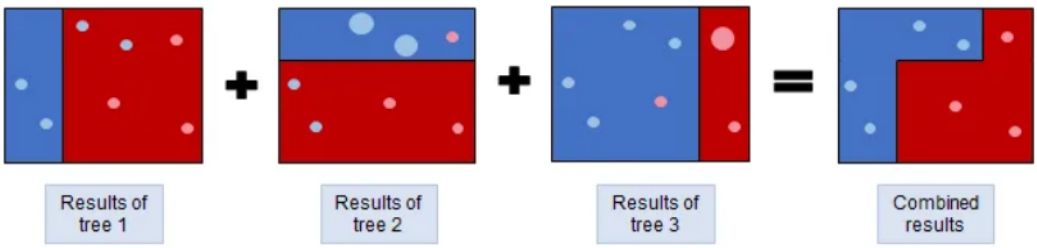

Figura 6 – Sumário e visualização do algoritmo AdaBoost para problemas de classificação. Pontos maiores indicam que as amostras foram previamente classificadas incorretamente e recebem, portanto, pesos de aprimoramento maiores. Fonte: [2]

Mais especificamente, cada árvore subsequente é treinada utilizando um modificador no peso de treinamento de cada erro que aconteceu previamente. Esse modificador faz com que esses erros sejam multiplicados por um "peso de aprimoramento" (boost weight) α. O boost weight é derivado do $erro$ da árvore de decisão anterior, definido como o número de partículas classificadas erroneamente dividido pelo número total de partículas testadas:

$$\alpha = \frac{1 - erro}{erro} \tag{1}$$

Tomando o resultado de uma árvore individual como $h(x)$ (sendo x um vetor das variáveis utilizadas) e definindo sinal como $h(x) = +1$ e ruído como $h(x) = -1$, a classificação do método de aprendizado aprimorado é descrita por:

$$y_{Boost}(x) = \frac{1}{N_{conjunto}} \cdot \sum_{i}^{N_{conjunto}} \ln(\alpha_i) \cdot h_i(x), \tag{2}$$

onde o somatório passa por todas as árvores do conjunto. Valores baixos de $y_{Boost}(x)$ sugerem uma classificação de ruído, enquanto valores altos indicam a classificação de sinal.

O método de aprendizado BDT aprimorado vem sendo utilizado com sucesso em diversas áreas, como as seguintes:

- Taxas de cliques (CTR - Click-Through Rates): a CTR é o número de cliques recebidos por um anúncio dividido pelo número de vezes que ele foi exibido. Neste contexto, as árvores de decisão aprimoradas são utilizadas para prever a probabiidade de um usuário clicar em um anúncio online [18].
- Diagnósticos relacionados a saúde: métodos de machine learning, assim como o BDT, são utilizados para prever a probabilidade e o risco do desenvolvimento de diferentes doenças em pacientes através de dados de imagens provenientes de exames [19].

- Classificação de partículas em colisões de aceleradores de partículas: o BDT surgiu como uma forte alternativa às redes neurais na identificação e classificação de diferentes partículas proveninentes de colisões de íons em aceleradores de partículas [20].

3.3 ROOT e TMVA

O software ROOT [21] - desenvolvido no CERN e que foi utilizado ao longo deste trabalho - proporciona uma estrutura para análise de dados. Como algoritmos de machine learning são de suma importância para estudos semelhantes ao sendo apresentado, o ROOT oferece suporte nativo para diversos métodos de machine learning através de suas bibliotecas. Uma das bibliotecas mais importantes presentes no software é a TMVA (Toolkit for Multivariate Analysis) [22], que oferece, além de uma interface própria na forma de um GUI (*Graphical User Interface*, ver Fig. 7), implementações de diferentes técnicas de machine learning. São algumas delas:

- Redes Neurais;
- Perceptron multicamadas;
- Árvores de decisão aprimoradas.

Através da interface apresentada pelo TMVA ao final dos treinamentos, é possível gerar gráficos que permitem análises de diversos aspectos do treinamento realizado. Incluso está - por exemplo - o teste de Kolmogorov–Smirnov, um teste estatístico que verifica o grau de compatibilidade entre duas distribuições [23]. No TMVA é utilizado para verificar se está ocorrendo *overtraining* em um treinamento de um método de machine learning. Podem ser geradas também matrizes que mostram a correlação entre as diversas variáveis utilizadas e gráficos de rejeição de ruído vs eficiência de sinal, que mostram o desempenho do método em relação à separação entre “sinal” e “ruído” [24].

4 METODOLOGIA

A produção da amostra para o treinamento da técnica de árvore de decisão aprimorada foi feita da seguinte maneira: eventos de simulações de Monte Carlo foram gerados por um procedimento nos quais os mésons D^0 (resultados de colisões entre prótons e nêutrons) que foram gerados pelo PYTHIA 8 [6] foram embutidos em eventos do HYDJET [5], que simula a colisão entre íons pesados. A amostra é completa quando utilizado o GEANT4 [7] para a simulação das interações entre as partículas produzidas na colisão e o material do CMS. Como essas simulações estão fora do escopo do trabalho, o processo de produção da amostra não receberá maiores detalhamentos.

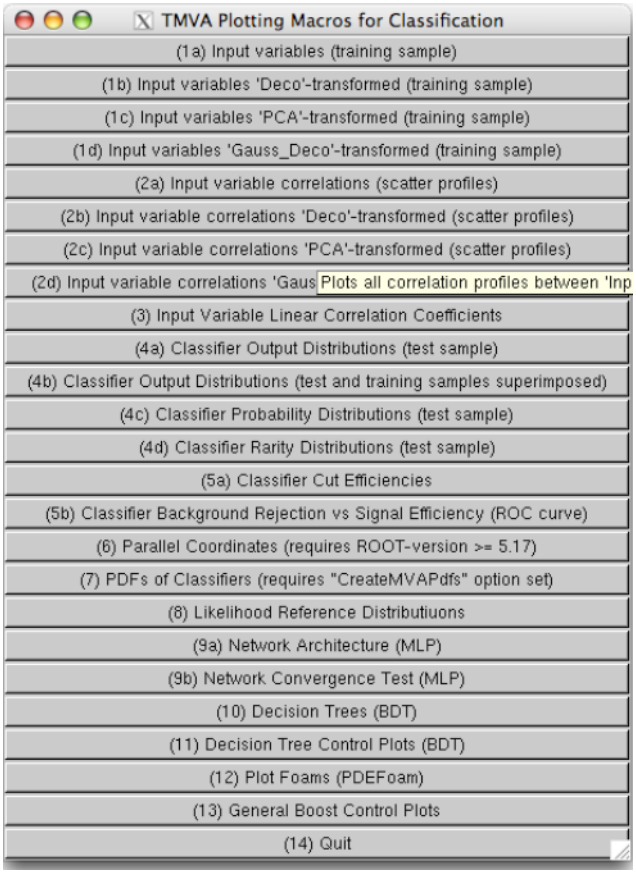

Figura 7 – Interface na qual é possível executar macros que mostram treinamentos, testes e resultados de availiações em problmeas de classificação. Fonte: [3].

O detector utilizado para as simulações foi o CMS. O mesmo tem como um dos principais componentes um solenóide supercondutor, que é responsável por criar um campo magnético praticamente uniforme de $3{,}8$ Tesla [25]. Todas as variáveis citadas na seção 4.1 foram descritas com base em um sistema de coordenadas cartesiano com origem no centro do detector. O eixo z está na direção do feixe de partículas, o eixo x aponta para o centro do acelerador LHC e y aponta para a superfície da caverna na qual o acelerador foi construído. Foram utilizadas coordenadas esféricas, conforme descrito na figura 8.

Como sugerido anteriormente, para estudarmos o méson D^0 produzido em uma colisão de íons pesados é necessário reconstruí-lo, já que não é possível detectá-lo diretamente devido ao seu curto tempo de vida. De maneira geral, isso é realizado através da formação de pares de partículas com cargas opostas resultantes do decaimento da partícula $D^0 \rightarrow \pi^+ + K^-$ e através da reconstrução do vértice secundário do decaimento do méson D^0. Para a reconstrução são utilizados as *tracks* (partículas carregadas reconstruídas no CMS, nesse caso píons e káons carregados provenientes dos D^0s) que possuem a massa esperada das partículas provenientes do decaimento da nossa partícula de interesse. Este trabalho foca na etapa seguinte a esta reconstrução: a seleção dos melhores candidatos a méson D^0, pois mesmo após a reconstrução a grande maioria das partículas não são mésons D^0. Esse processo foi baseado em diversas

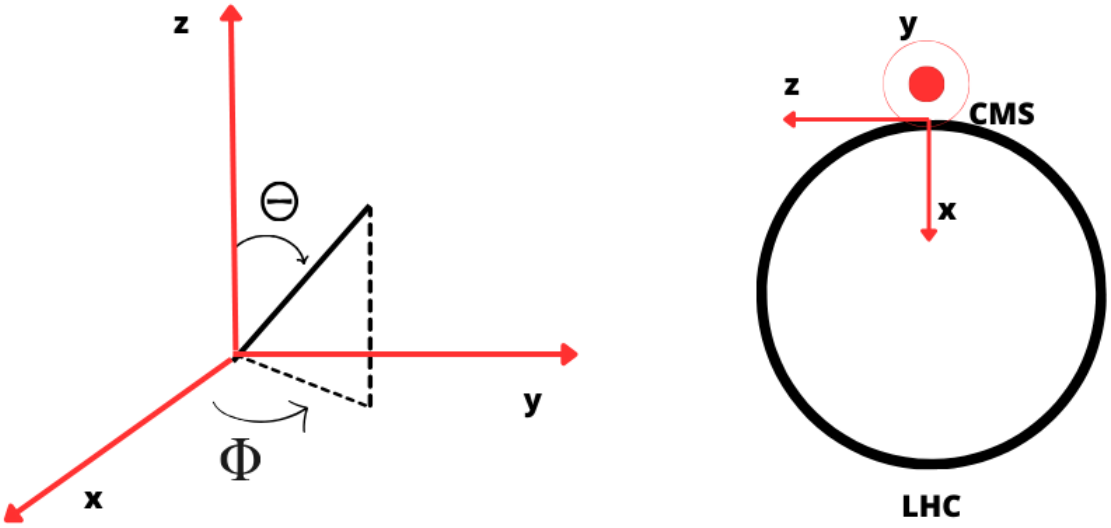

Figura 8 – Esquema com o sistema de coordenadas esféricas do CMS.

variáveis relacionadas diretamente com o méson D^0 além de variáveis relacionadas com seus produtos de decaimento, descritas na seção a seguir.

4.1 Variáveis Associadas ao Méson D^0

A fim de separarmos da melhor maneira as partículas consideradas sinal (mésons D^0) das partículas consideradas ruído (mésons D^0 "falsos", reconstruídos erroneamente), as variáveis citadas a seguir atuam como critérios de seleção dos possíveis candidatos a partícula D^0 e foram selecionadas com base em [26] para que usássemos o máior número possível de variáveis que apresentem algum poder discriminativo entre sinal e ruído. Inicialmente, para o treinamento da técnica BDT, foram utilizadas 24 variáveis relacionadas ao méson D^0 e às partículas resultantes do decaimento, também conhecidas como "filhas" do D^0, píon π^+ e káon K^-:

1. Momentum transversal dos mésons D^0: $p_T = \sqrt{{p_x}^2 + {p_y}^2}$. Em nossa amostra de teste usamos apenas partículas com $3{,}5 < p_T < 4{,}2$ GeV.

2. Nome da variável: DPt

Rapidez dos mésons D^0: $y = \frac{1}{2} \ln \frac{E+cp_z}{E-cp_z}$. Em nossa amostra de teste usamos apenas partículas com $-0{,}8 < y < 0{,}8$.

- Nome da variável: DRapidity

Ângulo azimutal ϕ dos mésons D^0.

- Nome da variável: DPhi

Massa invariante dos mésons D^0: $m_{inv} = \sqrt{E^2 - \vec{p}^2}$, onde E é a energia e $\vec{p}$ o vetor trimomento das partículas. Consideramos a velocidade da luz no vácuo $c = 1$.

- Nome da variável: DMass

Distância entre vértice primário (PV) e secundário (SV) em três dimensões (ver Fig. 9).

- Nome da variável: D3DDecayLength

Significância de D3DDecayLength.

- Nome da variável: D3DDecayLengthSignificance

Projeção de "D3DDecayLength" no plano xy (2D).

- Nome da variável: D2DDecayLength

Significância de "D2DDecayLength".

- Nome da variável: D2DDecayLengthSignificance

Probabilidade de χ^2 do ajuste do vértice secundário.

- Nome da variável: DVtxProb

Ângulo (no espaço 3D) entre o momentum do méson D^0 e a linha conectando os vértices primário e secundário.

- Nome da variável: D3DPointingAngle

Projeção de D3DPointingAngle no plano xy (Fig. 10).

- Nome da variável: D2DPointingAngle

A distância mais próxima do méson D^0 em relação ao vértice primário, *distance of closest approach* (DCA), Fig. 10.

- Nome da variável: DDca

Momentum transversal da primeira partícula filha (trk1), Fig. 11.

- Nome da variável: DTrk1Pt

Incerteza de DTrk1Pt.

- Nome da variável: DTrk1PtErr

Momentum transversal da segunda partícula filha (trk2).

- Nome da variável: DTrk2Pt

Incerteza de DTrk2Pt.

- Nome da variável: DTrk2PtErr

Pseudorapidez (η) da partícula trk1: $\eta = -\ln(\tan\frac{\theta}{2})$.

- Nome da variável: DTrk1Eta

Pseudorapidez (η) da partícula trk2.

- Nome da variável: DTrk2Eta

χ^2 do ajuste da trajetória para a partícula trk1.

- Nome da variável: DTrk1Chi2n

χ^2 do ajuste da trajetória para a partícula trk2.

- Nome da variável: DTrk2Chi2n

Significância do DCA na direção z da partícula 1, Fig. 12.

- Nome da variável: DzDCASignificanceDaugther1

Significância do DCA na direção z da partícula 2.

- Nome da variável: DzDCASignificanceDaugther2

Significância do DCA no plano xy da partícula 1.

- Nome da variável: DxyDCASignificanceDaugther1

Significância do DCA no plano xy da partícula 2.

- Nome da variável: DxyDCASignificanceDaugther2

Na figura 11, trk1 e trk2 indicam candidatos a káons e píons. Além disso, PV (*Primary Vertex* ou vértice primário) é o vértice no qual é formado o méson D^0 e SV (*Secondary Vertex* ou vértice secundário) é o vértice no qual o D^0 decai em suas partículas filhas. A significância de todas as variáveis apresentadas é calculada pelo valor da variável dividido pela sua incerteza. Ou seja, levando em conta uma variável X e sua incerteza ΔX, a significância é de $\frac{X}{\Delta X}$.

Tendo essas variáveis em mãos, é possível iniciar o processo de teste e ranqueamento das mesmas para a seleção das variáveis mais relevantes e para o treinamento do método BDT, como realizado na seção 4.2.

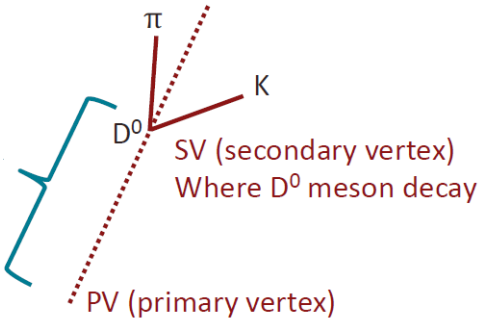

Figura 9 – Comprimento de decaimento de um méson D^0 em um π e um K e os dois vértices (PV e SV) relevantes.

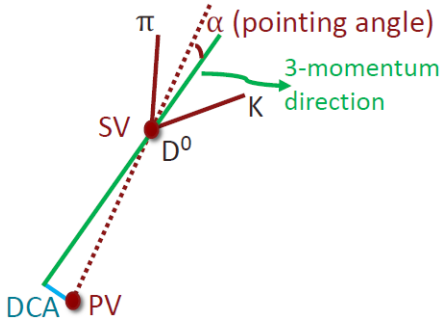

Figura 10 – Ângulo (α) entre o momentum 3D do méson D^0 e a linha que liga os dois vértices PV e SV, além da ilustração do DCA.

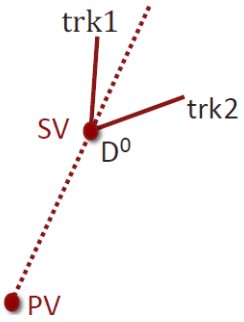

Figura 11 – Ilustração das partículas resultantes do decaimento do méson D^0 com nomes de trk1 e trk2.

4.2 Escolha das Variáveis para Treinamento

A partir de um template do TMVA, criou-se uma macro de classificação usando o método de BDT com AdaBoost [27] com um ensemble de 850 árvores, cada árvore com um número de ramos igual a 3. A escolha deste método de machine learning específico ocorreu pois, após realizada uma comparação da eficiência de rejeição de ruído e a eficiência de sinal entre cinco das opções de BDT disponibilizadas pelo TMVA (BDT com Adaptive Boosting (AdaBoost), BDT com Gradient Boost (BDTG), BDT com Bagging (BDTB), BDT com Decorrelation e Adaptive Boosting (BDTD) e BDT com discriminante de Fisher na divisão de nós (BDTF)), o AdaBoost se destacou como o método com maior rejeição de ruído e eficiência de seleção de

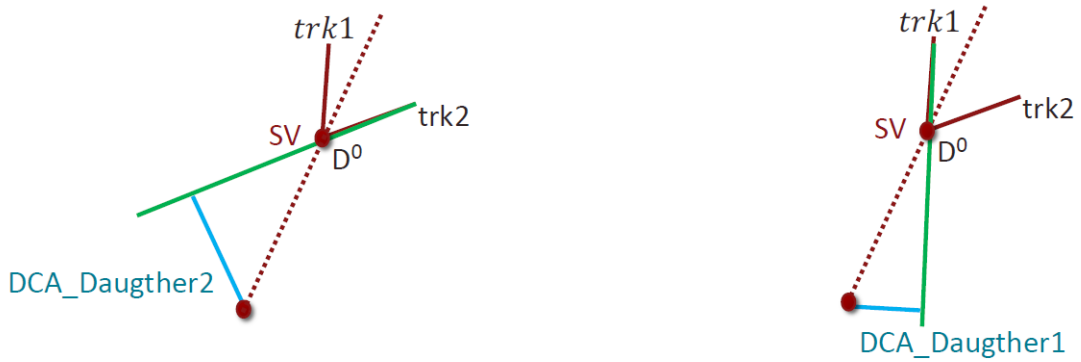

Figura 12 – Ilustração do DCA para ambas as partículas filhas do D^0.

sinal.

Além disso, os parâmetros citados anteriormente (850 árvores e profundidade 3) foram escolhidos pois o algoritmo do AdaBoost possui uma performance superior quando utilizado com grandes quantidades de árvores rasas (com profundidade de ramo igual a 2 ou 3) e que possuem pouco poder de discriminação individualmente. Isto ocorre pois, dessa maneira, as árvores possuem uma baixa propensão de overtraining. Ademais, foi realizada a comparação de nove diferentes configurações variando os dois parâmetros (número de árvores e profundidade de ramo) acima e a configuração de melhor performance (sem que houvesse overtraining) foi, de fato, a configuração original com 850 árvores de decisão com profundidade igual a 3.

Primeiramente, aplicou-se o algoritmo de BDT em dois cenários de teste: no primeiro, foram utilizadas 3 variáveis e no segundo 10, com o intuito de verificar o tempo de processamento. O tempo de treinamento necessário foi de aproximadamente 43 minutos. O TMVA automaticamente também mostra o tempo para teste do método. Neste caso, levou 222 segundos aplicando o método treinado em uma amostra independente com o mesmo número de partículas que a amostra de treino. Os graus de importância das variáveis (em percentagem de uso) foram registrados no terminal de comando do computador e podem ser vistos na tabela 1 abaixo.

Rank	Variável	Importância da Variável
1	D3DPointingAngle	0,162
2	DMass	0,160
3	DPhi	0,142
4	DRapidity	0,142
5	DPt	0,136
6	DVtxProb	0,132
7	D2DPointingAngle	0,126
8	D3DDecayLength	0,000
9	D2DDecayLength	0,000
10	DDca	0,000

Tabela 1 – Informações acerca da importância das variáveis em um treinamento do método BDT com 10 variáveis. Tempo levado para o treinamento: 43 minutos.

Com o intuito de verificar quais variáveis seriam as mais relevantes para o continuamento do trabalho, o mesmo código foi atualizado e executamos o treinamento com 20 das 24 variáveis mencionadas previamente. As quatro variáveis não utilizadas são p_T, y, ϕ e m_{inv} (DPt, DRapidity, DPhi e DMass) dos mésons D^0, pois são utilizadas para apresentação de resultados nas análises de dados e não queremos introduzir distorções ou *bias* nessas variáveis. Posteriormente, analisamos as seguintes informações:

- Ranqueamento e tempo de treinamento Tabela 2.
- Matrizes de correlação entre as variáveis em eventos de sinal e ruído Fig. 13.
- Teste de overtraining em amostras de sinal e ruído usando o teste de Kolmogorov-Smirnov para comparar as distribuições Fig. 14.

Rank	Variável	Importância da Variável
1	D3DPointingAngle	0,116
2	DVtxProb	0,102
3	D2DPointingAngle	0,098
4	DTrk1Pt	0,094
5	DTrk1Eta	0,090
6	DTrk2Pt	0,088
7	DTrk2Eta	0,087
8	DTrk2PtErr	0,077
9	DxyDCASignificanceDaughter2	0,057
10	DxyDCASignificanceDaughter1	0,055
11	DTrk1PtErr	0,054
12	DzDCASignificanceDaughter2	0,042
13	DzDCASignificanceDaughter1	0,038
14	D3DDecayLength	0,000
15	D2DDecayLength	0,000
16	D3DDecayLengthSignificance	0,000
17	D2DDecayLengthSignificance	0,000
18	DTrk1Chi2n	0,000
19	DTrk2Chi2n	0,000
20	DDca	0,000

Tabela 2 – Ranqueamento de importância entre as 20 variáveis utilizadas no treinamento do BDT. Tempo levado para o treinamento: 30 minutos.

A matriz de correlação citada acima utiliza o coeficiente de correlação de Pearson [28] que é calculado (pelo TMVA) através da seguinte equação:

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}. \tag{3}$$

Para calcular a correlação entre duas variáveis, o TMVA faz um histograma em duas dimensões (2D) colocando uma variável no eixo x e outra no eixo y. Nesse caso, n representa o número

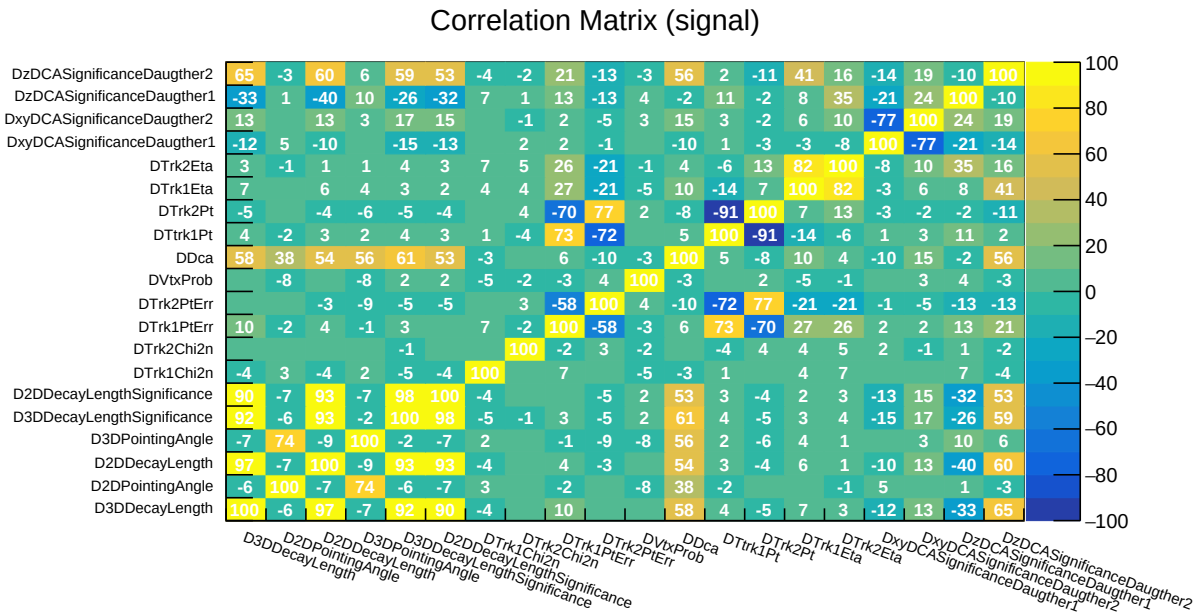

Figura 13 – Matriz que mostra a correlação entre todas as variáveis presentes no treinamento.

total de entradas (número de partículas) nesse histograma 2D, x e y o número de entradas em cada bin no eixo x e y, respectivamente. A somas são realizadas sobre todos os bins do histograma. Dessa maneira, r pode assumir três categorias diferentes: $-100\% \leq r < 0\%$; $r = 0\%$; $0\% < r \leq 100\%$. No primeiro caso ($-100\% \leq r < 0\%$, correlação negativa), podemos interpretar que quando uma variável muda, a outra variável muda na direção contrária. Quando $r = 0\%$, as variáveis simplesmente não apresentam correlação alguma (na imagem acima essa categoria está representada através das lacunas sem números). No último caso ($0\% < r \leq 100\%$, correlação positiva), quando uma variável muda, a outra muda na mesma direção.

O tempo levado para o treinamento representando na tabela 2 foi de aproximadamente 30 minutos. É importante ressaltar que, apesar do número de variáveis ter sido dobrado de um treinamento para o outro, foi adicionada uma linha de código [29] que permitiu que o computador trabalhasse no modo *multi-threading* (que permite a execução em massa de várias threads em um mesmo processo [30]), o que levou a uma diminuição no tempo total de processamento do computador.

Levando em conta as informações presentes na Fig. 13, podemos perceber que existe uma grande correlação positiva entre as seguintes variáveis:

- D3DDecayLengthSignificance e D2DDecayLengthSignificance (98%);
- D2DDecayLength e D3DDecayLength (97%);
- DTrk1Pt e DTrk2Pt (91%);
- DTrk1Eta e DTrk2Eta (82%).

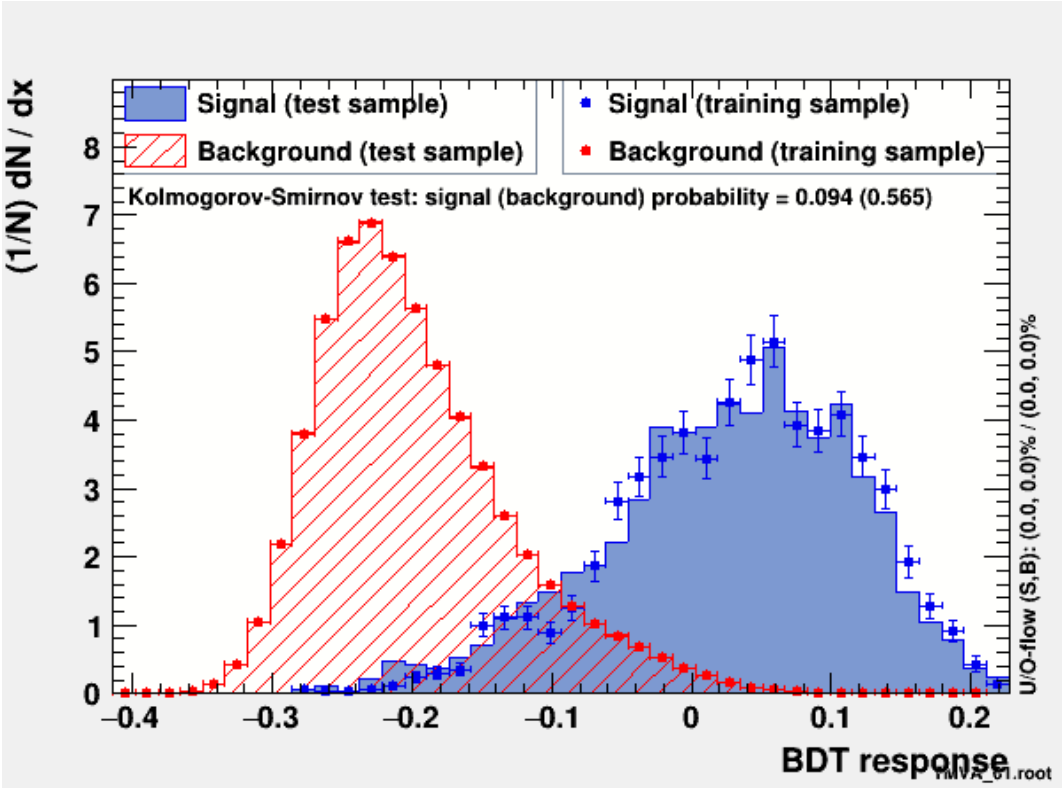

Figura 14 – Teste de Kolmogorov-Smirnov para sinal e ruído do atual treinamento.

Além disso, é possível ver através da tabela 2 que as seguintes variáveis são de pouca importância para o treinamento do método BDT:

- DDca;
- DTrk1Chi2n;
- DTrk2Chi2n.

Reunindo estes dois conjuntos de informações, foram removidas - para as etapas futuras do trabalho - as seguintes variáveis: DTrk1Chi2n; DTrk2Chi2n; DDca; D2DDecayLength e D2DDecayLengthSignificance. A alta correlação entre variáveis não é esperada afetar significativamente a eficiência de selecionar sinal ou rejeitar ruído, mas interfere no tempo de treinamento, pois o mesmo tem mais variáveis para testar. A inclusão de muitas variáveis sem qualquer discriminação pode também custar um preço caro em etapas posteriores das análises de dados, introduzindo incertezas sistemáticas não triviais.

As variáveis com alta correlação em três dimensões receberam preferência em relação às de duas dimensões, pois carregam uma quantidade maior de informação. Além disso, para evitar problemas de assimetria nas distribuições finais, as variáveis filhas (DTrk1Pt, DTrk2Pt, DTrk1Eta e DTrk2Eta) deveriam ser removidas ou mantidas em pares. Como todas receberam uma boa colocação no ranqueamento de importância, optou-se por manter as quatro variáveis

para o treinamento nas próximas etapas. Vale ressaltar que as variáveis D3DDecayLength e D3DDecayLengthSignificance não foram retiradas pois elas possuem alta conexão com o sinal utilizado, como pode ser visto na Tabela 3 e no grau de separação apresentado das mesmas. Ou seja, apesar da importância estar mal ranqueada (pois a variável foi pouco utilizada durante o treinamento), o ranking superior demonstra que as variáveis têm uma separação bastante alta entre sinal e ruído.

Ademais, o testes de KS (Kolmogorov-Smirnov, ver Fig. 14) que são realizados para todos os treinamentos realizados no TMVA são diferentes de um teste convencional de KS, o que os torna um pouco complicados de interpretar. Para este trabalho, os critérios (com base na discussão [31]) utilizados para dizer que não está ocorrendo overtraining, ou seja, que as distribuições do *score* de BDT para as amostras de treino e de teste são compatíveis, foram os seguintes:

- O teste KS não resulte em valores muito pequenos, abaixo de 0,1%.
- A comparação entre as distribuições de teste e treino não possuam diferenças sistemáticas visíveis a olho e com mais de 3 sigmas fora das incertezas estatísticas.

No caso, não observamos overtraining em nosso método. Em seguida, executamos novamente a macro (agora com as variáveis atualizadas) para realizar uma nova análise do tempo de processamento e do ranqueamento de importância das variáveis, como pode ser visto na Tabela 4.

Rank	Variável	Separação
1	D3DPointingAngle	0,589
2	D2DPointingAngle	0,364
3	DxyDCASignificanceDaughter2	0,120
4	DxyDCASignificanceDaughter1	0,117
5	DVtxProb	0,063
6	D3DDecayLengthSignificance	0,062
7	DzDCASignificanceDaughter1	0,058
8	DzDCASignificanceDaughter2	0,058
9	D3DDecayLength	0,023
10	DTrk2Pt	0,016
11	DTrk1Pt	0,015
12	DTrk2Eta	0,014
13	DTrk1PtErr	0,009
14	DTrk2PtErr	0,008
15	DTrk1Eta	0,007

Tabela 3 – Ranqueamento de separação entre as 15 variáveis escolhidas para o treinamento.

Dessa vez, o tempo levado para o treinamento foi de aproximadamente 26 minutos. Como pode ser observado a partir desse fato, a retirada de 5 das variáveis fez com que o processo

Rank	Variável	Importância da Variável
1	D3DPointingAngle	0,117
2	DVtxProb	0,103
3	D2DPointingAngle	0,098
4	DTrk1Pt	0,094
5	DTrk1Eta	0,090
6	DTrk2Pt	0,088
7	DTrk2Eta	0,087
8	DTrk2PtErr	0,076
9	DxyDCASignificanceDaughter2	0,057
10	DxyDCASignificanceDaughter1	0,056
11	DTrk1PtErr	0,054
12	DzDCASignificanceDaughter2	0,042
13	DzDCASignificanceDaughter1	0,038
14	D3DDecayLength	0,000
15	D3DDecayLengthSignificance	0,000

Tabela 4 – Ranqueamento de importância entre as 15 variáveis escolhidas para o treinamento.

de treinamento fosse acelerado em cerca de 4 minutos, ou seja, da ordem de 13% de redução no tempo de execução. As eficiências de sinal e rejeição de ruído foram praticamente as mesmas.

4.3 Aplicação do método de BDT escolhido

Com o objetivo final de identificarmos o méson D^0 e testarmos a pureza de sinal do nosso algoritmo de machine learning, nós aplicamos o nosso método de treinamento BDT em dados de simulações de colisões. Para isso, foi necessário separar a tree utilizada em duas (realizado através do script em Python [32]): metade de todos os eventos foram utilizados para o treinamento do algoritmo e a outra metade foi utilizada para a aplicação em dados de colisões.

Através da nossa macro utilizada para aplicação do método na Ref. [33], foi gerado um gráfico em escala logarítmica (ver Fig. 15) relacionando os diferentes tipos de partículas (total, ruído e sinal) e seus respectivos *scores* do método, que tende a ter valores maiores para candidatos a sinal (eixo "x" dos gráficos), como explicado na seção 3.1.

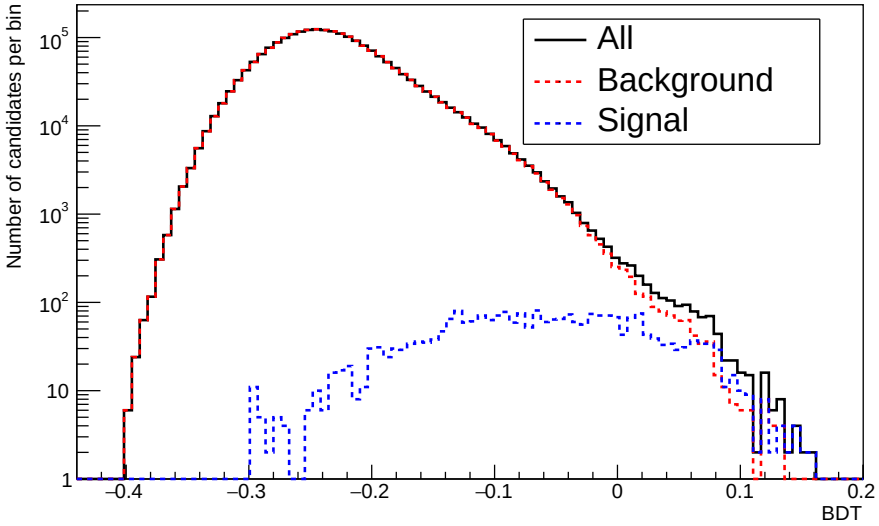

Figura 15 – Gráfico comparando o número de candidatos totais com sinal e ruído de acordo com seus *scores*. Em preto: total de partículas. Em vermelho: partículas consideradas ruído. Em azul: partículas consideradas sinal.

A verificação da efetividade do método é feita analisando a distribuição de massa invariante das partículas selecionadas. No caso dos mésons D^0 é esperado sua massa ser aproximadamente $1{,}865$ GeV [34]. Como o méson D^0 é instável e decai, por exemplo, em um píon e káon carregados, sua massa é calculada somando os quadrimomentos dos píons e káons ($p = p_\pi + p_K$) e depois extraindo a massa invariante usando $p^2 = E^2 - \vec{p}^2 = m^2_{inv}$. A distribuição de massa dos mésons D^0 não é apenas um pico no valor da massa esperado, pois além de efeitos de resolução do detector, existe uma largura física na distribuição associada ao fato do méson D^0 ser uma partícula instável. Dito isso, foram gerados dois histogramas que mostram os resultados da aplicação do nosso treinamento. O primeiro (ver Fig.16) representa a massa invariante de todas as partículas presentes na amostra, por isso não possui nenhum pico identificável do méson D^0, uma vez que nossa amostra é majoritariamente composta por partículas de ruído.

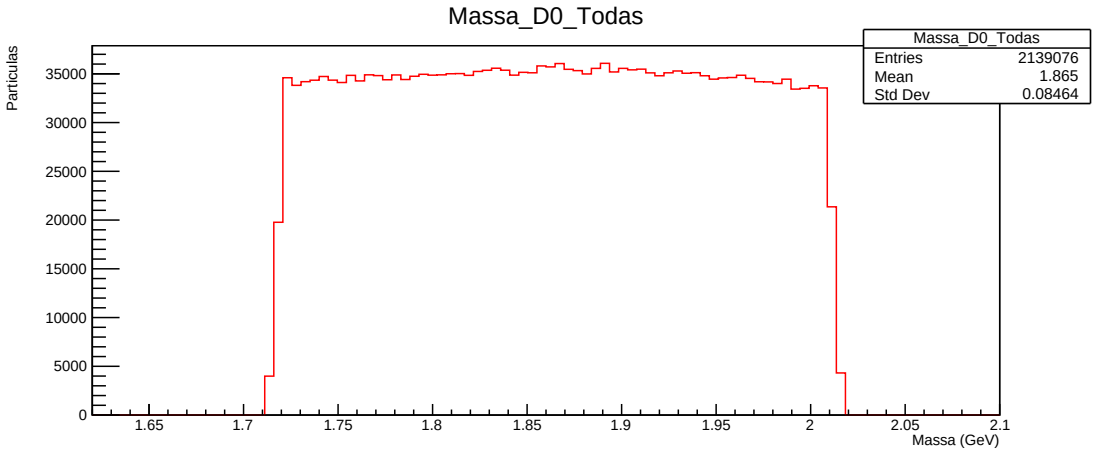

Figura 16 – Histograma que representa as massas invariantes de todas as partículas presentes na amostra de testes na região entre $1{,}71$ e $2{,}02$ GeV.

A fim de obter um histograma que seja capaz de identificar o pico esperado do D^0, realizamos um corte no *score* (eixo x da imagem 15) através do qual a maior parte do ruído foi rejeitado e ao mesmo tempo mantendo uma fração razoável de sinal. O ponto de corte escolhido foi em $BDT > 0{,}0$. Para confirmarmos que a escolha do corte foi feita com sucesso, utilizamos a figura de mérito $\frac{S}{\sqrt{S+R}}$ (onde S é o número de partículas do tipo sinal e R é o número de partículas do tipo rúido) que está relacionada diretamente com a pureza de sinal da nossa distribuição.

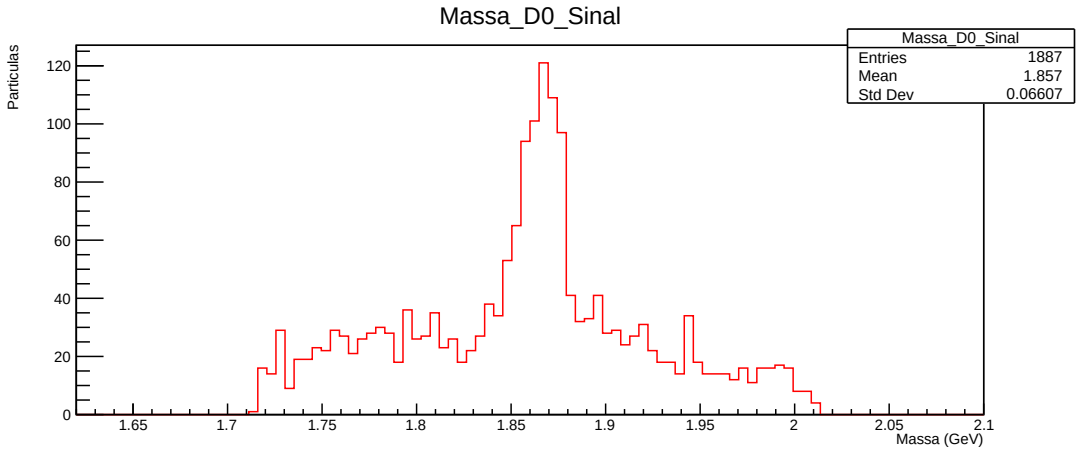

Figura 17 – Histograma representando as massas invariantes das partículas identificadas com o corte $BDT > 0{,}0$.

Analisando a Fig. 17, depois do corte $BDT > 0{,}0$, fica bastante claro que esse histograma apresenta um pico bem definido perto da marca de $1{,}85$ GeV, que se aproxima muito da massa de um méson D^0 ($1{,}865$ GeV, de acordo com a Ref. [34]). Portanto, o méson D^0 foi identificado com uma alta significância estatística, tendo valor de figura de mérito de $S/\sqrt{S+R} = 667/\sqrt{667+1459} \sim 14$. Comparada à seleção com cortes simples (método cut-based) adotadas na Ref. [35], ou seja, com (D3DDecayLenghtSignificance$> 5{,}86$ e DVtxProb$> 0{,}224$ e

D3DPointingAngle$< 0{,}12$), e respectiva significância $S/\sqrt{S+R} = 654/\sqrt{654+3105} \sim 11$, concluímos que nosso método melhorou consideravelmente a significância estatística de sinal vs ruído em relação ao método cut-based.

5 CONCLUSÕES

Este trabalho apresenta um passo a passo de como fazer o treinamento e a aplicação de métodos de árvores de decisão aprimoradas (BDT) para realizar a identificação de mésons D^0 presentes em uma amostra de dados de simulação de colisões de íons pesados no acelerador de partículas LHC utilizando o software ROOT e sua biblioteca TMVA. Em particular, foi realizado o treinamento do método de BDT com diferentes valores de parâmetros e métodos de aprimoramento em amostras de simulações de colisões PbPb com energia no centro de massa por par de núcleons de $2{,}76$ TeV no detector CMS. Como resultado da aplicação deste algoritmo treinado, foram obtidos gráficos que identificam os mésons D^0 através de sua massa invariante.

Ao longo do trabalho foi possível concluir quais as variáveis com maior poder de separação entre ruído e sinal para que as mesmas fossem utilizadas no treinamento do algoritmo BDT. De maneira semelhante, concluímos que - dentre cinco métodos testados de BDT - o de maior interesse para o nosso trabalho foi o BDT com o método de Adaptive Boosting e sua configuração ideal utilizando como parâmetros 850 árvores com profundidade igual a 3.

Observamos uma eficiente separação entre ruído e sinal da amostra de dados utilizada na identificação dos mésons D^0 através do método de BDT escolhido para a amostra que usamos como teste, i.e., colisões de PbPb a $2{,}76$ TeV com intervalo de centralidade de 10–30% (grau de sobreproposição entre os núcleos, 0 % sendo a maior sobreposição), momento transversal (p_T) e rapidez (y) das partículas de $3{,}5 < p_T < 4{,}2$ GeV e $-0{,}8 < y < 0{,}8$, respectivamente. Nosso treinamento com um corte na variável de output de $BDT > 0{,}0$ resultou em uma figura de mérito de $S/\sqrt{S+R} \sim 14$ ao passo que um método otimizado de cut-based (Ref. [35]) teve um valor de $S/\sqrt{S+R} \sim 11$. Indicando que nosso método provê uma significância de sinal consideravelmente maior do que em métodos mais simples de cut-based.

6 APÊNDICE - HISTOGRAMAS DAS VARIÁVEIS

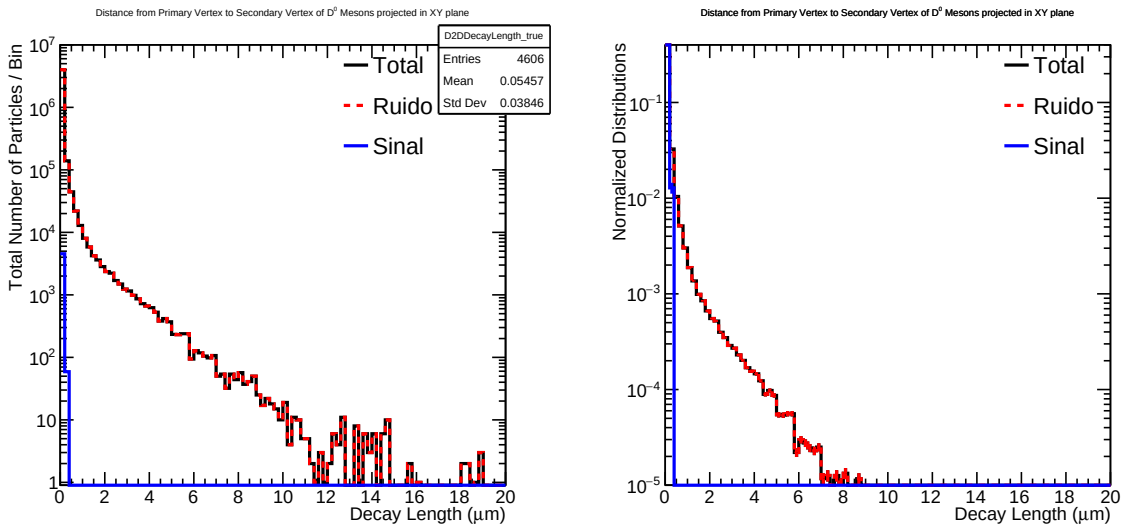

Figura 18 – Gráfico representando a projeção de "D3DDecayLength" no plano xy (2D). Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

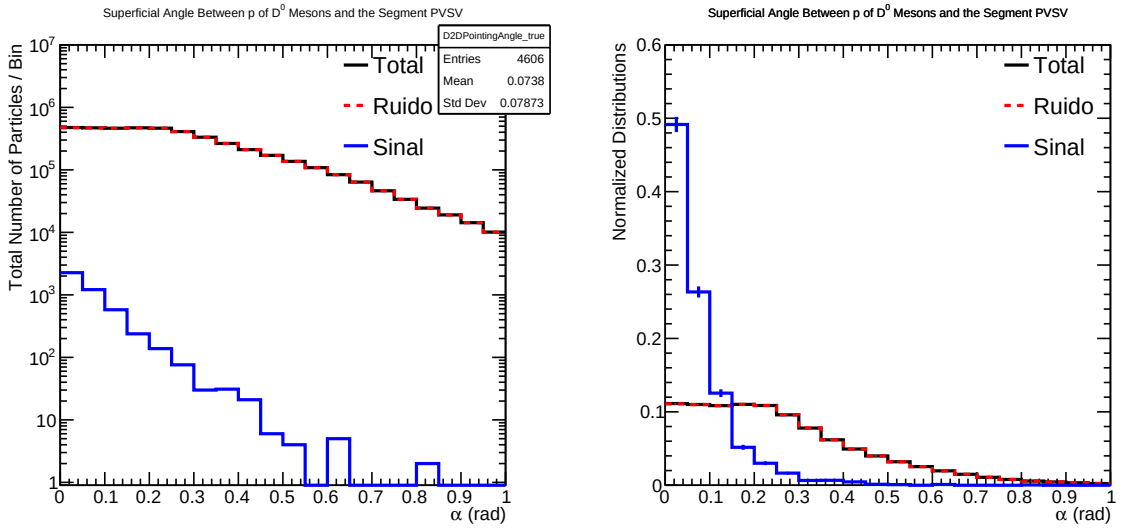

Figura 19 – Gráfico representando a projeção de "D3DPointingAngle" no plano xy (2D). Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

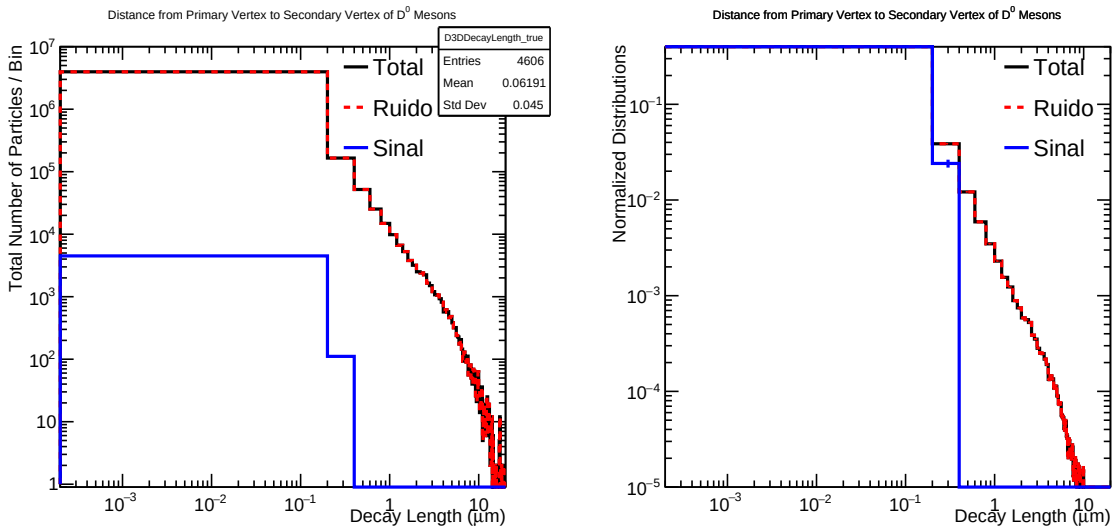

Figura 20 – Gráfico representando a distância entre o vértice primário (PV) e secundário (SV) em três dimensões. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

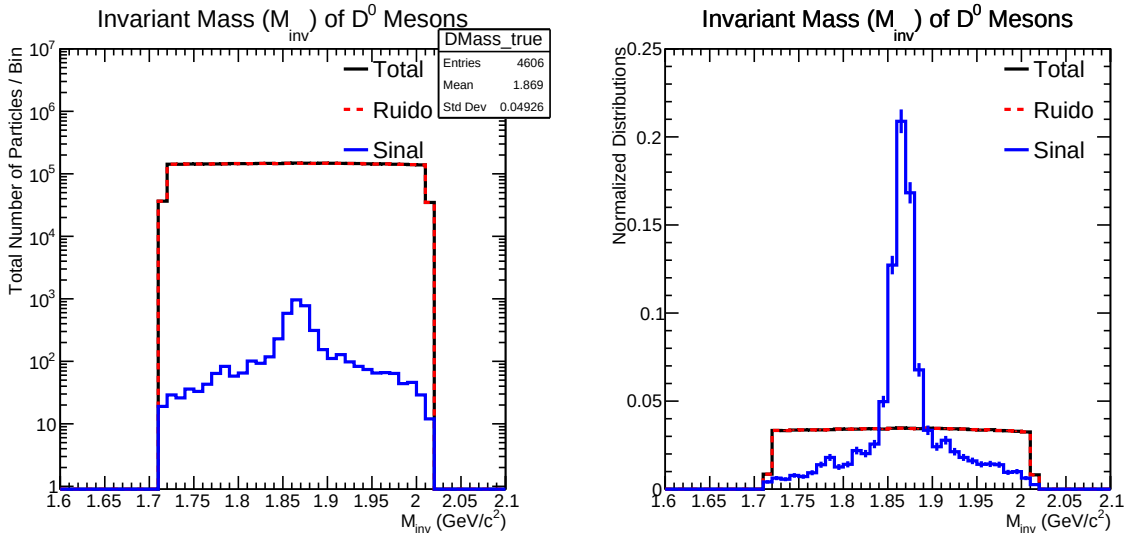

Figura 21 – Gráfico representando a massa invariante dos mésons. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

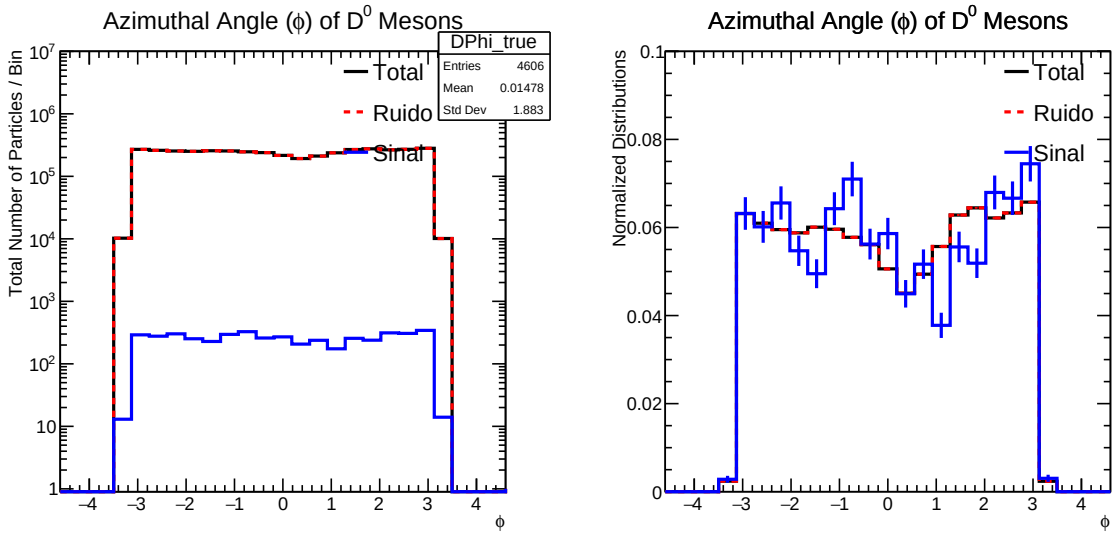

Figura 22 – Gráfico representando o ângulo azimutal dos mésons. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

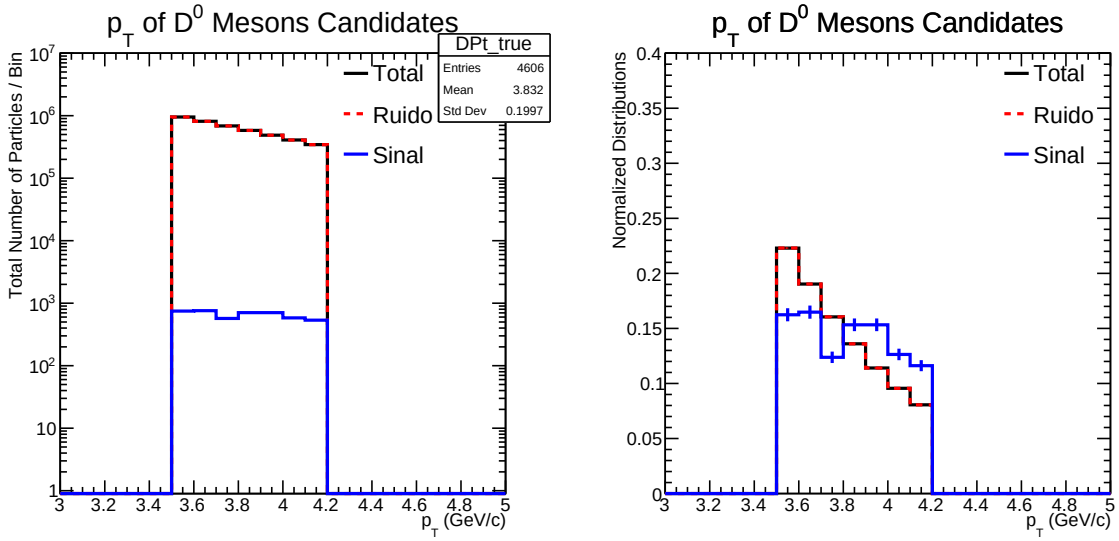

Figura 23 – Gráfico representando o momentum transversal dos mésons. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

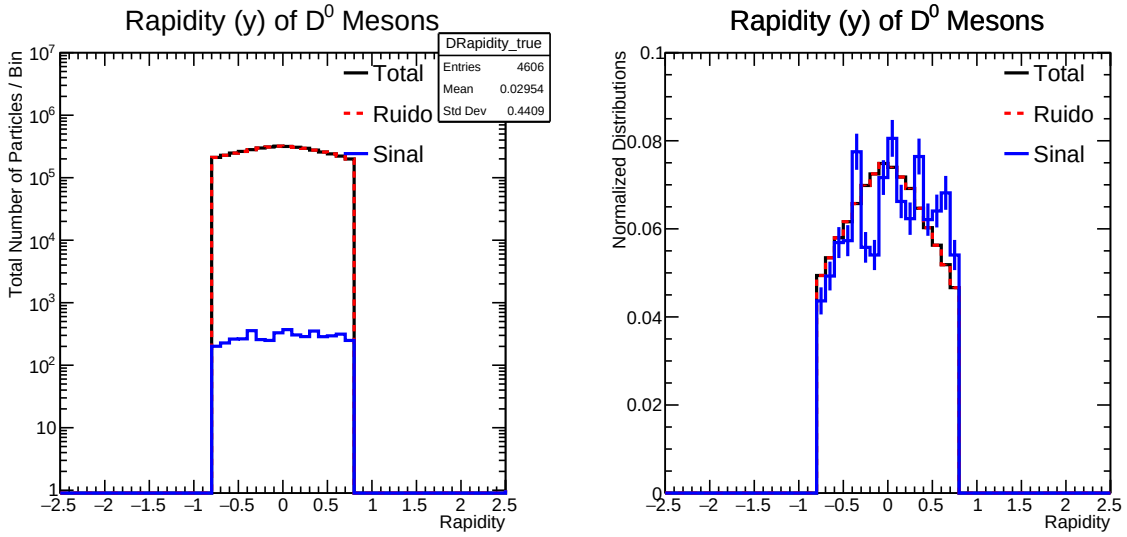

Figura 24 – Gráfico representando a rapidez dos mésons. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

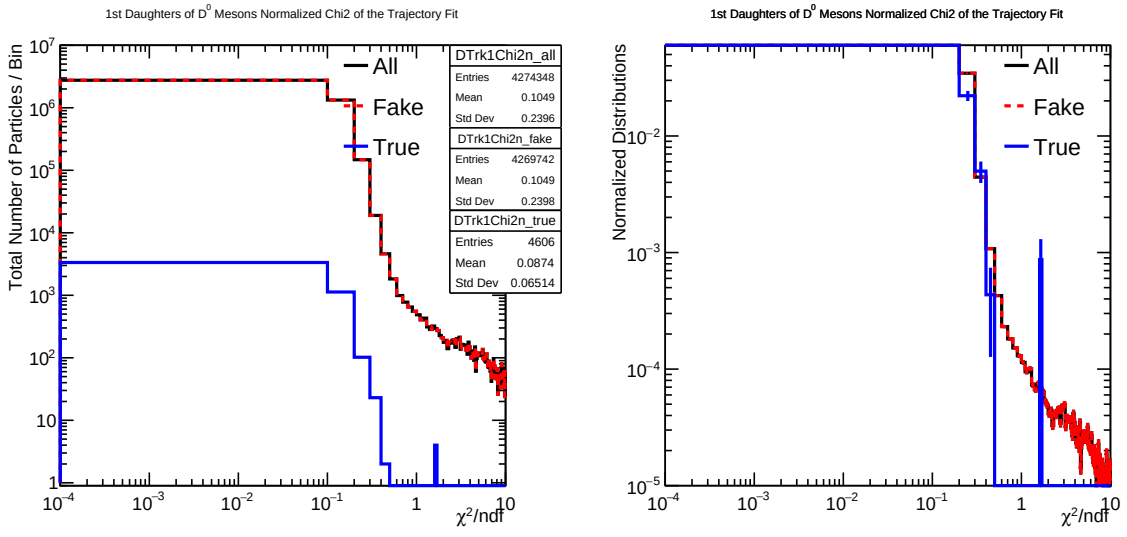

Figura 25 – Gráfico representando χ^2 do ajuste da trajetória para a partícula trk1. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

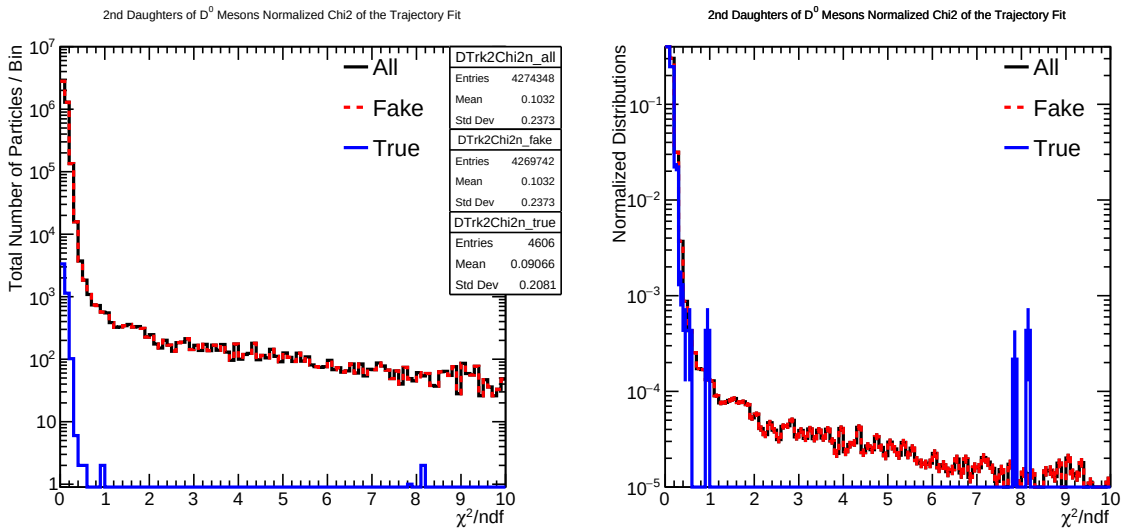

Figura 26 – Gráfico representando χ^2 do ajuste da trajetória para a partícula trk2. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

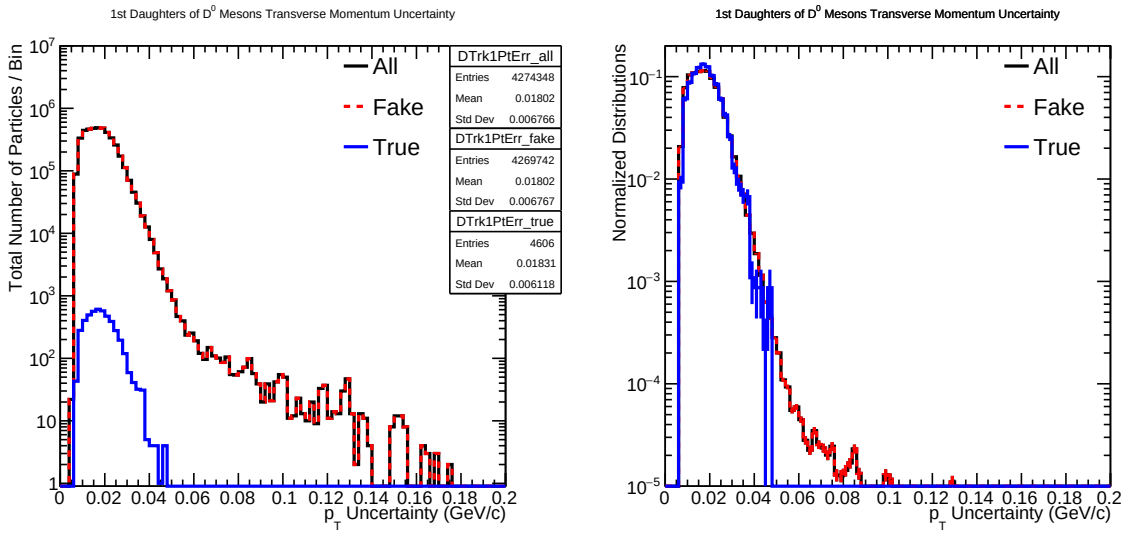

Figura 27 – Gráfico representando a incerteza de DTrk1Pt. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

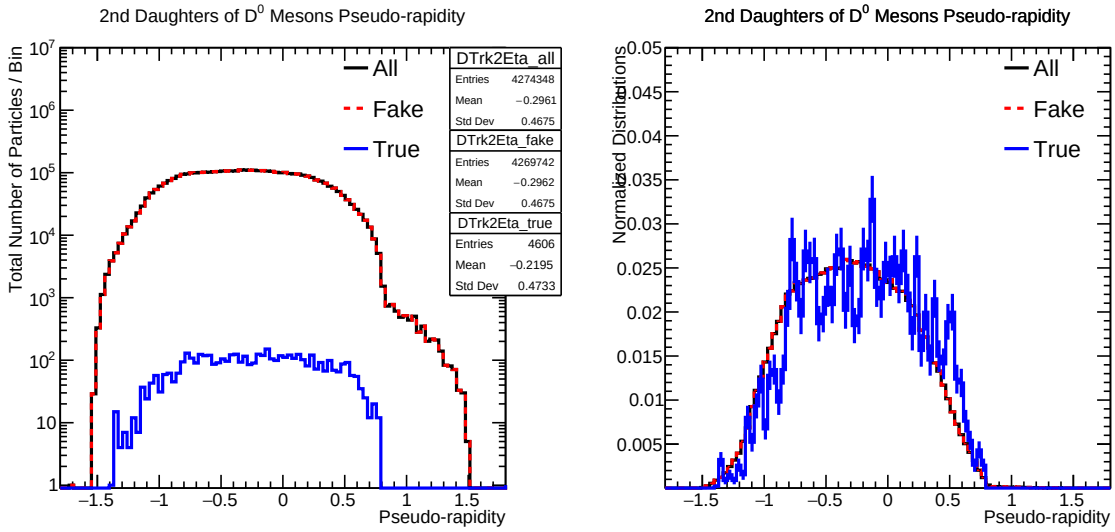

Figura 28 – Gráfico representando a pseudorapidez da partícula trk1. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

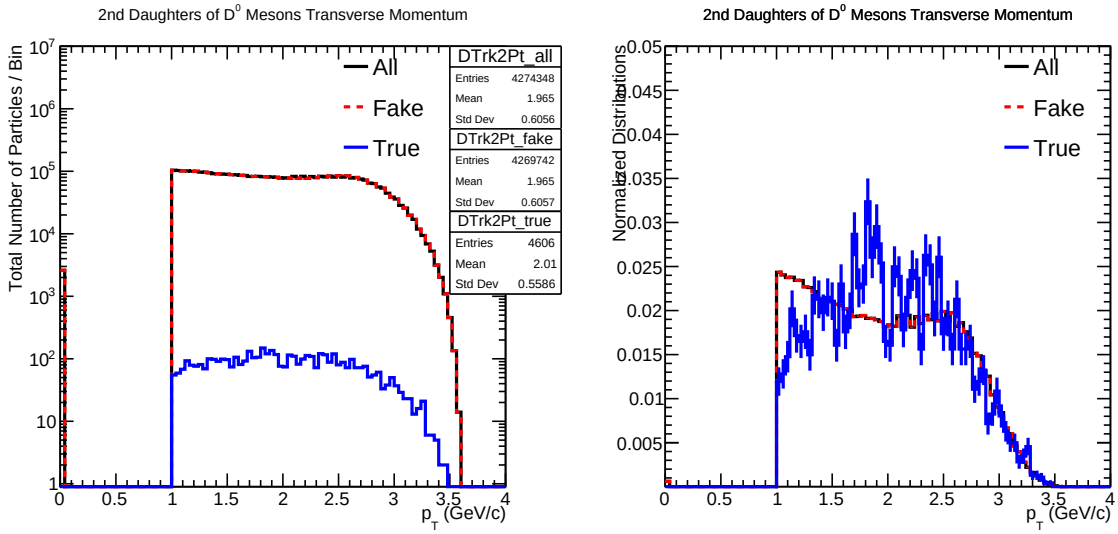

Figura 29 – Gráfico representando o momentum transversal da segunda partícula filha (trk2) Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

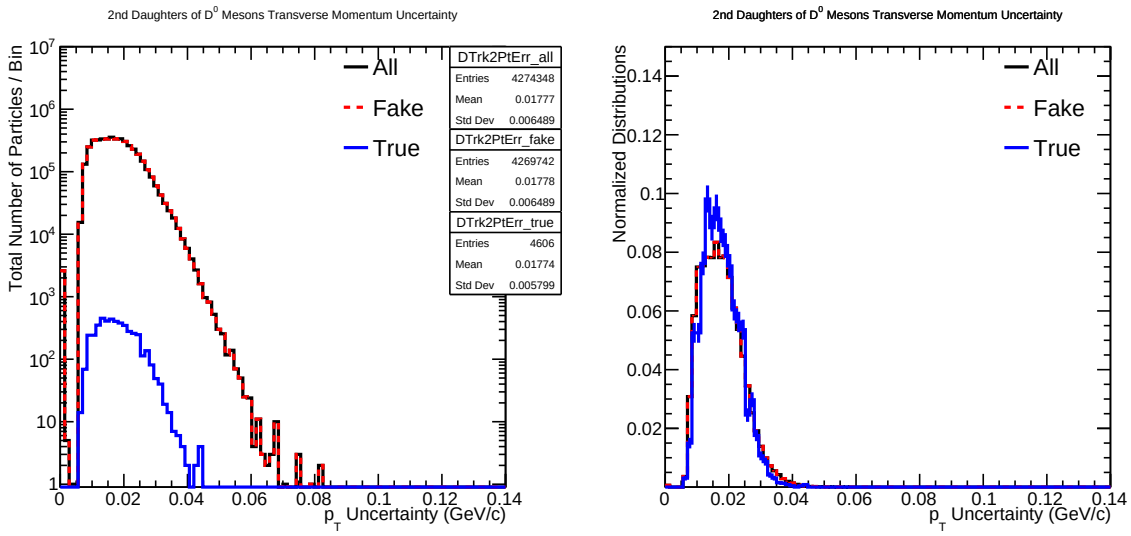

Figura 30 – Gráfico representando a Incerteza de DTrk2Pt. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

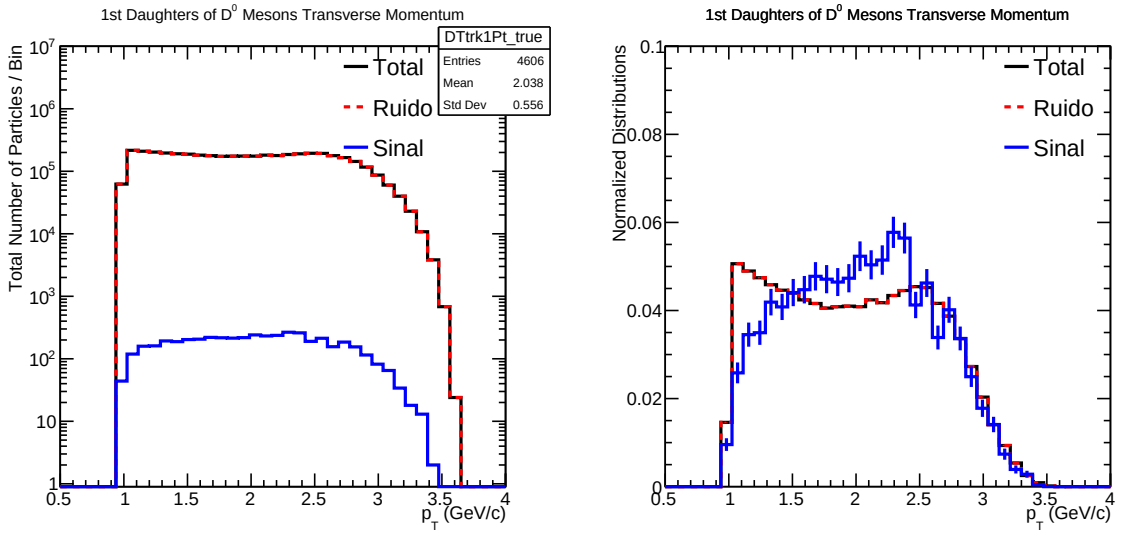

Figura 31 – Gráfico representando o momentum transversal da primeira partícula filha (trk1). Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

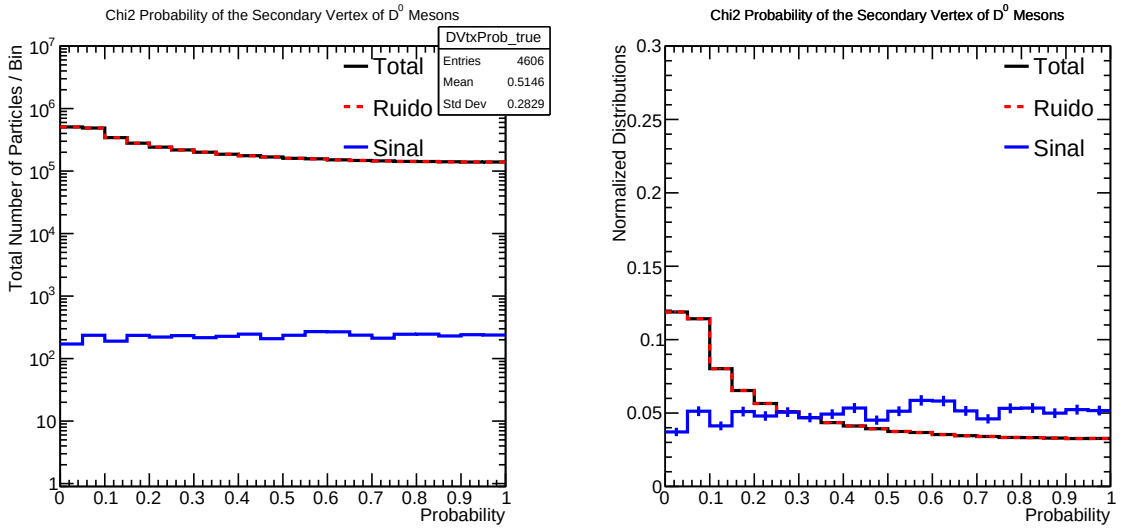

Figura 32 – Gráfico representando a probabilidade de χ^2 do ajuste do vértice secundário. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

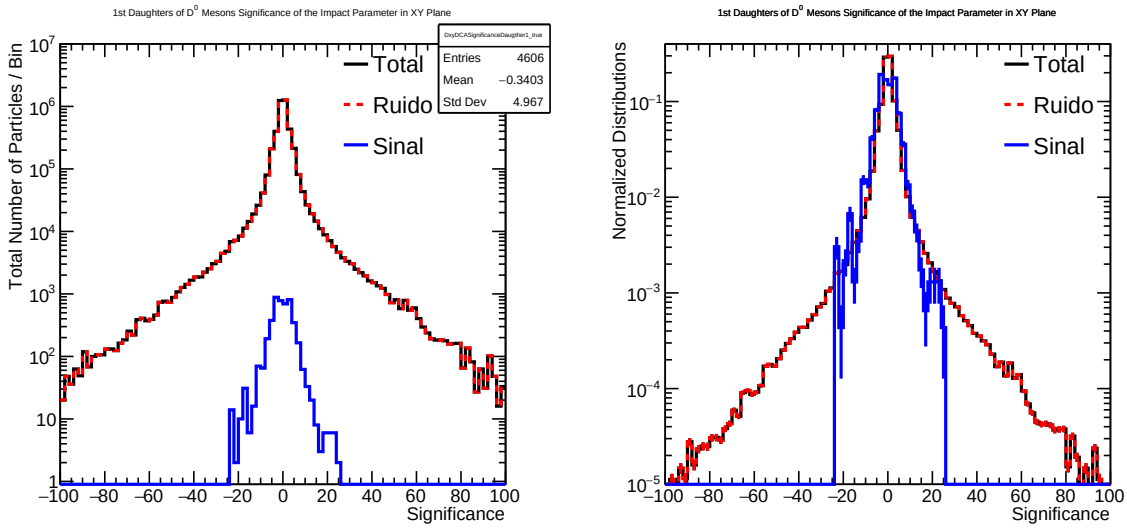

Figura 33 – Gráfico representando a significância do DCA no plano xy da partícula 1. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

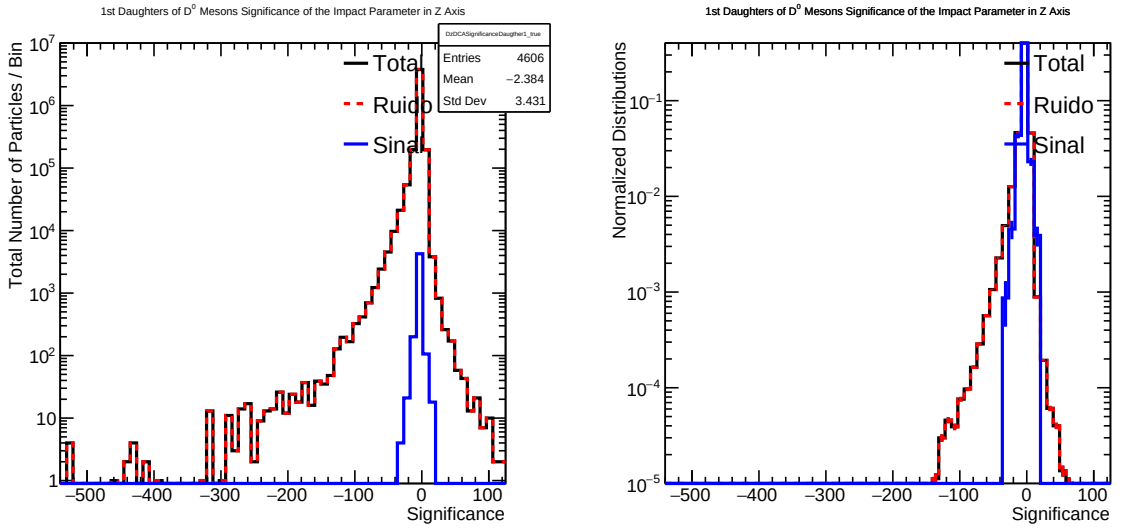

Figura 34 – Gráfico representando a significância do DCA na direção z da partícula 1. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

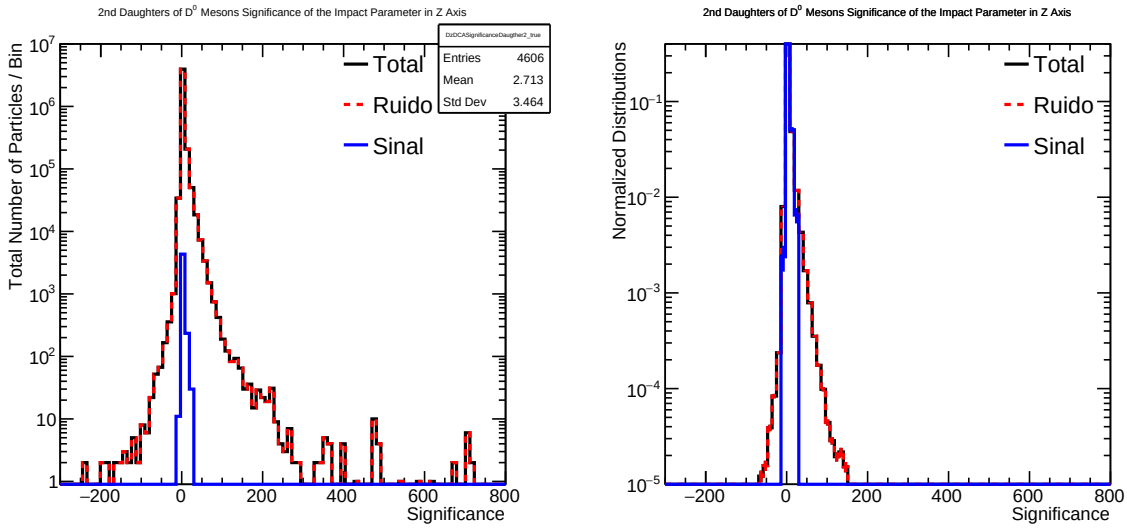

Figura 35 – Gráfico representando a significância do DCA na direção z da partícula 2. Em vermelho: partículas falsas. Em azul: partículas verdadeiras. Em preto: o somatório de todas as partículas.

REFERÊNCIAS

[1] Roman Pasechnik and Michal Šumbera. Phenomenological review on quark–gluon plasma: concepts vs. observations. *Universe*, 3(1):7, 2017. 2, 6

[2] The Ultimate Guide to AdaBoost, random forests and XGBoost. Towards Data Science, Julia Nikulski.Disponível em: https://towardsdatascience.com/the-ultimate-guide-to-adaboost-random-forests-and-xgboost-\7f9327061c4f. Acesso: 20/08/2023. 2, 14

[3] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, H. Voss, M. Backes, T. Carli, O. Cohen, A. Christov, D. Dannheim, K. Danielowski, S. Henrot-Versille, M. Jachowski, K. Kraszewski, A. Krasznahorkay Jr. au2, M. Kruk, Y. Mahalalel, R. Ospanov, X. Prudent, A. Robert, D. Schouten, F. Tegenfeldt, A. Voigt, K. Voss, M. Wolter, and A. Zemla. Tmva - toolkit for multivariate data analysis, 2009. 2, 13, 16

[4] Barbara Jacak and Peter Steinberg. Creating the perfect liquid in heavy-ion collisions. *Physics today*, 63(BNL-93753-2010-JA), 2010. 6

[5] I. P. Lokhtin and A. M. Snigirev. A Model of jet quenching in ultrarelativistic heavy ion collisions and high-p(T) hadron spectra at RHIC. *Eur. Phys. J. C*, 45:211–217, 2006. 7, 15

[6] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An introduction to PYTHIA 8.2. *Comput. Phys. Commun.*, 191:159–177, 2015. 7, 15

[7] S. Agostinelli et al. —a simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250, 2003. 7, 15

[8] Particle Data Group, R. L. Workman, et al. Review of particle physics. *Prog. Theor. Exp. Phys.*, 2022:083C01, 2022. 7

[9] C. A. Bernardes and G. Hoss. macro_doControlPlots.C. Disponível em: https://github.com/CesarBernardes/TCC-HIN-UFRGS/blob/main/D0MesonsID/ControlPlots/macro_doControlPlots.C. 8

[10] Octavio Loyola-Gonzalez. Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE access*, 7:154096–154113, 2019. 10

[11] Jason Brownlee. A gentle introduction to object recognition with deep learning. *Machine Learning Mastery*, 5, 2019. 10

[12] Google DeepMind. AlphaGo - The Movie | Full award-winning documentary. YouTube, 13 de Março de 2020. Disponível em: https://www.youtube.com/watch?v=WXuK6gekU1Y. Acesso: 05/08/2023. 10

[13] Deep Q-Network e Processos de Decisão de Markov. Deep Learning Book. Disponível em: https://www.deeplearningbook.com.br/deep-q-network-e-processos-de-decisao-de-markov/. Acesso: 06/08/2023. 10

[14] What is speech recognition? IBM. Disponível em: https://www.ibm.com/topics/speech-recognition. Acesso: 06/08/2023. 10

[15] O que é Deep Learning? Oracle Brasil. Disponível em: https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-deep-learning/. Acesso: 06/08/2023. 10

[16] Como funciona o algoritmo Árvore de Decisão. Didática Tech, Jul 2022. Disponível em: https://didatica.tech/como-funciona-o-algoritmo-arvore-de-decisao/. Acesso: 10/07/2023. 11

[17] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995. 13

[18] Ilya Trofimov, Anna Kornetova, and Valery Topinskiy. Using boosted trees for click-through rate prediction for sponsored search. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, pages 1–6, 2012. 14

[19] Song Chen and Chuan-Jun Liao. Prediction of the probability and risk factors of early abdominal aortic aneurysm using the gradient boosted decision trees model. *Applied Artificial Intelligence*, 36(1):2014190, 2022. 14

[20] Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2-3):577–584, 2005. 15

[21] ROOT: analyzing petabytes of data, scientifically. https://root.cern/. Acesso: 08/07/2023. 15

[22] Andreas Hocker et al. TMVA - Toolkit for Multivariate Data Analysis. 3 2007. 15

[23] Kolmogorov-Smirnov Goodness-of-Fit Test. Disponível em: https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm. Acesso: 14/08/2023. 15

[24] Andreas Hocker, X Prudent, Jan Therhaag, Y Mahalalel, Moritz Backes, Rustem Ospanov, Maciej Kruk, M Jachowski, Alexander Voight, Arnaud Robert, et al. Tmva-toolkit for multivariate data analysis with root: users guide. Technical report, 2007. 15

[25] The CMS Experiment. Disponível em: https://www.bo.infn.it/gruppo1/en/cms-experiment/. Acesso: 04/09/2023. 16

[26] Albert M Sirunyan et al. Measurement of prompt D^0 and $\overline{\mathrm{D}}^0$ meson azimuthal anisotropy and search for strong electric fields in PbPb collisions at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV. *Phys. Lett. B*, 816:136253, 2021. 17

[27] C. A. Bernardes and G. Hoss. TMVAClassification_01.C. GitHub. Disponível em: https://github.com/CesarBernardes/TCC-HIN-UFRGS/blob/main/D0MesonsID/MLStudies/TMVAClassification_01.C. 20

[28] Pearson Correlation Coefficient (r) | Guide & Examples, Scribbr. Shaun Turney, May 2022. Disponível em: https://www.scribbr.com/statistics/pearson-correlation-coefficient/. Acesso: 07/09/2023. 22

[29] C. A. Bernardes and G. Hoss. TMVAClassification_01.C. GitHub. Disponível em: https://github.com/CesarBernardes/IC-HIN-UFRGS/blob/main/D0MesonsID/MLStudies/TMVAClassification_01.C#L55-L56. 23

[30] O que é multithreading e como a técnica beneficia seu software, Jul 2022. Disponível em: https://blog.tecnospeed.com.br/o-que-e-multithreading-e-como-a-tecnica-beneficia-seu-\software/. Acesso: 08/08/2023. 23

[31] Kolmogorov-Smirnov test values, February 2022. Disponível em: https://root-forum.cern.ch/t/kolmogorov-smirnov-test-values/32868/1. Acesso: 10/08/2023. 25

[32] C. A. Bernardes and G. Hoss. TMVAClassificationApplication.C. GitHub. Disponível em: https://github.com/CesarBernardes/TCC-HIN-UFRGS/blob/main/D0MesonsID/MLStudies/TMVAClassificationApplication.C. 27

[33] C. A. Bernardes and G. Hoss. macro_divideTree_Rdataframe_redefine.py. GitHub. Disponível em: https://github.com/CesarBernardes/IC-HIN-UFRGS/blob/main/D0MesonsID/Skims/macro_divideTree_Rdataframe_redefine.py. 27

[34] PA Zyla and P Eerola. Review of particle physics. 2020. 27, 28

[35] Albert M Sirunyan et al. Measurement of prompt D^0 meson azimuthal anisotropy in Pb-Pb collisions at $\sqrt{s_{NN}}$ = 5.02 TeV. *Phys. Rev. Lett.*, 120(20):202301, 2018. 28, 29