

**IDENTIFICAÇÃO DE PARTÍCULAS EM ALTAS ENERGIAS UTILIZANDO  
MÉTODOS DE APRENDIZADO DE MÁQUINA****João Paulo Seixas Alves e Silva**Instituto Federal Fluminense  
Rua Dr. Siqueira, 273  
jpseixasesilva@hotmail.com**Cristine Nunes Ferreira**Instituto Federal Fluminense  
Rua Dr. Siqueira, 273  
crisnfer@msn.com**Henrique Rego Monteiro da Hora**Instituto Federal Fluminense  
Rua Dr. Siqueira, 273  
dahora@gmail.com**Luiz Gustavo Lourenço Moura**Instituto Federal Fluminense  
Rua Dr. Siqueira, 273  
prof.luizgustavo.iff@gmail.com**RESUMO**

*No campo da física de altas energias são produzidas atualmente massivas quantidades de dados necessitando de ferramentas tecnológicas que apoiem suas pesquisas e descobertas. Este trabalho tem por objetivo a comparação de duas técnicas de aprendizado de máquina: árvores de decisão e redes neurais. A suposição fundamentada em trabalhos correlatos demonstra que as redes neurais desenvolvem melhor em trabalhos de classificação entre outras aplicações. Apoiando-se nisso, a classificação de partículas subatômicas em altas energias estudado a partir de bases de dados do grande colisor de hádrons do CERN, chegou a uma conclusão ímpar ao se utilizar tais métodos e no fim comparando-os. Ao fim, ambos obtiveram resultados muito semelhantes.*

**Palavra-chave:** *Aprendizado de Máquina, Redes Neurais, Árvores de Decisão, Física em Altas Energias*

**ABSTRACT**

*In the field of high energy physics, massive amounts of data are currently being produced, requiring technological tools to support their research and discoveries. This work aims to compare two techniques of machine learning: decision trees and neural networks. The supposition based on related works shows that neural networks develop better in classification work among other applications. Based on this, the classification of subatomic*

*particles at high energies studied from the databases of the great hadrons collider by CERN, came to an unequal conclusion when using such methods and in the end comparing them. In the end, both had very similar results.*

**Keywords:** *Machine Learning, Decision Tree, Neural Network, High Energy Physics*

## 1. INTRODUÇÃO

*Machine Learning*, ou, aprendizado de máquina, é uma das áreas de grande destaque na ciência da computação atualmente. A premissa de se gerar conhecimento a partir de técnicas de computação e matemática aplicada impulsiona esta tecnologia para aplicação em múltiplas áreas do conhecimento. Ferramentas como *Machine Learning* possuem a capacidade de lidar com grandes volumes de dados e gerar análise, conhecimento e conclusões para fins de classificação, regressão, agrupamento entre outras. O mundo passa por uma produção de dados nunca antes vista, gerada pela grande rede de computadores. A produção dessas quantidades massivas de dados advém de inúmeros tipos de dispositivos eletrônico e, ao mesmo tempo, grandes iniciativas de pesquisas científicas internacionais também o fazem (COHEN; FREYTSIS; OSTDIEK, 2018).

Problemas iniciados na física de partículas procuram por soluções por meio de aprendizado de máquina afim de tratar enormes quantidades de dados geradas após colisões de prótons. O *Conseil Européen pour la Recherche Nucléaire* (CERN) é um dos maiores laboratórios para pesquisas em *High Energy Physics* (HEP), ou, física em altas energias. Atualmente o CERN produz cerca de 25 petabytes anualmente, o que torna seus desafios ainda maiores (KUO et al., 2014).

O problema central de experimentos em altas energias é a correta identificação de partículas através de dados coletados por detetores. Para isto, reconstrução de trajetórias, agrupamentos, chuviros (*showers*), jatos (*jets*), anéis (*rings*) e outros, são feitas através de gravações das interações das partículas nos detetores onde depositam energias, direções, tamanho e outras propriedades após colisões de matérias. Existem duas potenciais falhas em classificação de partículas: erro na construção das características das propriedades nos dados brutos acarretando em erro de categorização dos eventos físicos e, além, propriedades utilizadas para caracterização dos eventos são limitadas e não oferecem substancial teórico necessário para a tarefa.

Métodos que utilizam múltiplas variáveis, como *Fisher discriminates* e redes neurais, vem sendo utilizados em físicas de altas energias desde 1990 em experimentos no Fermilab e CERN com notável sucesso. Porém, mais recentemente, com avanços em *Machine Learning*, novas ferramentas como *Boosted Decision Tree* (BDT), *Support Vector Machines* (SVM), *Generative Neural Network* (GNN) dentre outras foram desenvolvidas e estão sendo utilizados para identificação de partículas em diversos experimentos (ADAM-BOURDARIOS et al., 2015).

A utilização de redes neurais e BDT para classificação de partículas demonstram bastante eficiência e são utilizados com frequência em diversas pesquisas. Conforme Zoe et al. (2005), a procura por neutrinos em experimento no Fermilab obteve resultados satisfatórios com ambos algoritmos, porém quando são utilizadas menos variáveis as redes neurais performam ligeiramente piores do que os métodos de impulsionamento (*boosted*).

Em trabalho mais recente, Aurisiano et al. (2016) descreveram uma aplicação de *Convolutional Neural Network* (CNN), ou redes neurais convolucionais, para identificação de interações de neutrinos em calorímetros. Com uma reconstrução mínima do evento, o

algoritmo conseguiu de forma excelente separar elétrons e múons dos neutrinos, classificando assim os neutrinos.

Este trabalho objetiva a classificação de partículas a partir de uma base de dados higienizada. A base de dados possui dados coletados no detetor LHCb (*Large Hadrons Collider beauty*) que possui subdetetores para interagir com a matéria colidida em seu centro. É comum trabalhos desenvolvidos para classificação de partículas em físicas de altas energias utilizarem árvores de decisão e redes neurais conforme exposto acima. Sendo assim, o objetivo proposto é a comparação entre esses dois métodos através de ferramentas que fazem a implementação em alto nível.

A hipótese lançada é de que as redes neurais possuem habilidades maiores para desenvolver classificação quando expostas a grandes quantidades de características (*features*) em comparação a árvores de decisão. Essa hipótese será fundamentada com a citação de trabalhos a seguir.

## 2. REVISÃO DA LITERATURA

### 2.1 Redes neurais profundas, rasas, convolucionais e árvores de decisão

Colisões em altas energias são objetos de estudo que geram grande impacto na ciência, não somente devido a sua tentativa de explicação da síntese da matéria universal, como também todo desafio tecnológico afim de gerar experimentações. Classificação de partículas é a diferenciação do que é partícula, a isso dá-se o nome de *signal*, de tudo aquilo que é percebido (detectado) porém não identificado por alguma similaridade teórica, chamado-o de *background*. A tarefa de identificar partículas é basicamente diferenciar o que é *signal* e o que é *background*, para isso inúmeras soluções foram propostas. Pelo fato de novas partículas serem difíceis de observar em seu momento inicial, o estudo de identificação de partículas é focado no decaimento que acontecem posteriormente, esses detectados com maior propriedade pelos detetores (BALDI; SADOWSKI; WHITESON, 2014).

O recente sucesso das redes neurais profundas, ou *Deep Neural Networks* (DNN), trazem uma nova perspectiva frente aos métodos já empregados em HEP, que, em partes, falham em capturar inúmeras informações disponíveis dos experimentos. Em Baldi et al. (2014), foi demonstrado que DNNs podem melhorar as métricas de classificação em *benchmark datasets*, mesmo comparadas com outros algoritmos como BDT. O estudo foi dividido em dois momentos: o primeiro avança na distinção entre um *signal*, proveniente de uma nova teoria do bóson de Higgs, e *background*, com produtos de decaimento idênticos mas características cinemáticas distintas; o segundo momento tem a tarefa de diferenciar novas partículas supersimétricas levando em consideração seu estado final no qual existem partículas detectáveis e outras invisíveis. O *dataset* foi dividido em 3 níveis de características (*features*), *low level*, *high level* e *Complete*. Ambas divisões foram treinadas por 3 algoritmos diferentes por 5 vezes e manteve-se as mesmas configurações escolhidas, são eles: uma rede neural rasa, ou *shallow neural network* (NN), uma DNN e uma BDT. Assim como todo experimento envolvendo física de altas energias e aprendizado de máquina, a performance dos modelos são descritas pelas curvas ROC (Receiver Operating Characteristic) e a acurácia é medida pela área sob a curva ROC, ou *area under ROC* (AUC). Nas tabelas 1 e 2 estão os resumos das experimentações e a Figura 1 e 2 são exemplos de curvas ROC do primeiro momento (BALDI; SADOWSKI; WHITESON, 2014).

Tabela 1: AUC 1º Momento

Algoritmo	AUC		
	Low Level	High Level	Complete
BDT	73%	78%	81%
NN	73,3%	77,7%	81,6%
DNN	88%	80%	88,5%

Tabela 2: AUC 2º Momento

Algoritmo	AUC		
	Low Level	High Level	Complete
BDT	85%	83,5%	86,3%
NN	86,7%	86,3%	87,5%
NN Dropout	88,6%	85,9%	87,3%
DNN	87,2%	86,5%	87,6%
DNN Dropout	87,6%	86,9%	87,9%

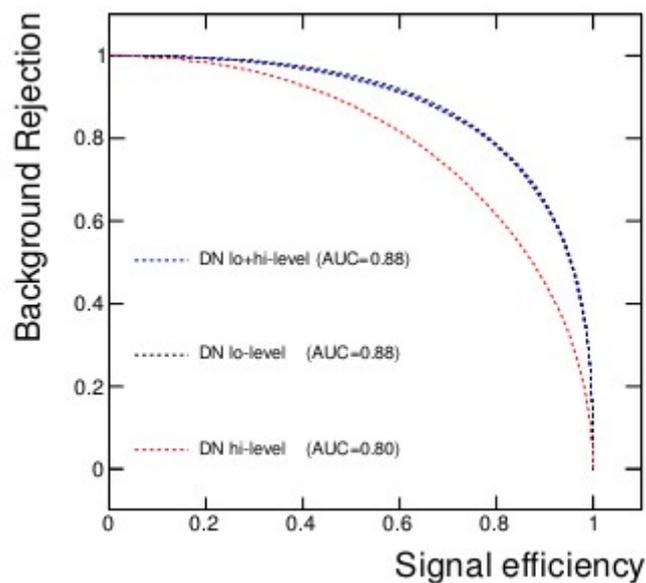


Figura 1: ROC A1 (BALDI; SADOWSKI; WHITESON, 2014)

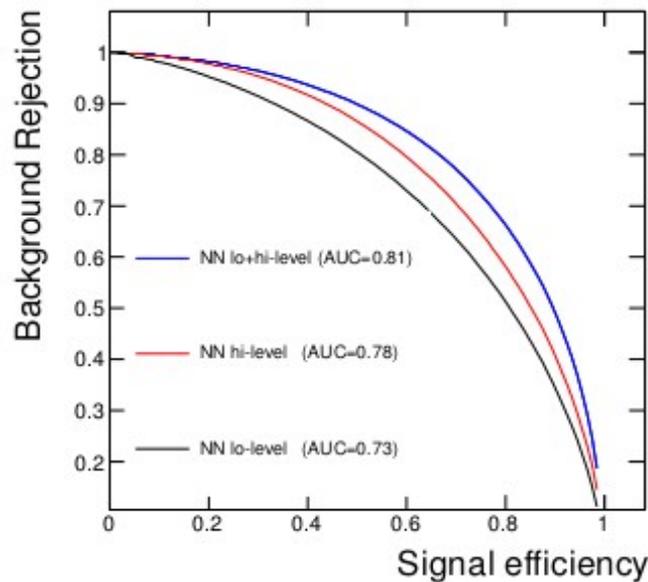


Figura 2: ROC B1 (BALDI; SADOWSKI; WHITESON, 2014)

No primeiro momento, os algoritmos que utilizaram *high level features* performaram abaixo do esperado em comparação daqueles que utilizaram todo o *dataset (Complete)*, sugerindo que, apesar de um alto nível de representatividade das características, o modelo demonstra dificuldade de capturar *insights* (BALDI; SADOWSKI; WHITESON, 2014).

Já no segundo momento, uma nova configuração para as redes neurais foi formada, otimizando-as com regularização do tipo *dropout*. Em ambos experimentos as redes neurais se destacaram, principalmente aquelas que possuíam camadas ocultas (DNN). Aplicando-as em diferentes conjuntos de dados, observou-se que as redes neurais conseguem abstrair as características necessárias para uma classificação com satisfatória acurácia sobrepujando assim as árvores de decisão (BDT), esta última performa de forma semelhante apenas quando comparada as redes neurais rasas (NN). As redes neurais utilizaram configuração de hiperparâmetros os seguintes valores: tangente hiperbólica como função de ativação, inicialização dos pesos através de distribuição normal com zero mean e desvio padrão de 0.1 na primeira camada, 0.001 na camada de saída e 0.05 em todas as camadas ocultas. Os gradientes foram computados com tamanho de mini-batches igual a 100. O modelo foi treinado por 200 épocas com taxa de aprendizado de 1.000002. Quando submetidos a otimização de *dropout*, o valor é de 50% (BALDI; SADOWSKI; WHITESON, 2014).

## 2.2 Redes neurais profundas, rasas e convolucionais

Em experimento desenvolvido por Hertel et al. (2017), foram utilizadas redes neurais convolucionais que têm a finalidade de classificação através de imagens. O trabalho objetivou o estudo de oscilações dos neutrinos baseado na estimação da energia de elétrons neutrinos e elétrons *showers* em detectores do NOvA alcançando resultados de estado da arte.

Neutrinos não possuem massas e raramente interagem com alguma matéria e por isso são de difíceis detecções. O experimento observa a oscilação dos neutrinos pois são uma função da sua energia, e isso consegue-se através de líquido cintilante posto dentro dos calorímetros. Ao interagir com o líquido, forma-se uma imagem 3D dividida em duas direções,  $x - z$  e  $y - z$ . Toda a base de dados gerada a partir da experimentação é dividida em *training set*, *validation set* e *testing set*. Para processamento da base de dados foram utilizados três métodos de normalização: *mean zero unit variance standardization*, *log transformation* e

*constant scaling*. Todos os métodos obtiveram resultados similares (HERTEL et al., 2017).

A *convolutional layer* foi configurada com 32 *filters*, 200 *units* e *Root Mean Square* (RMS) para avaliar a performance. Para *training loss* foi utilizado *Mean Squared Error* (MSE) e *Mean Absolute Error* (MAE), o algoritmo foi otimizado de duas formas, com *dropout* e *L2 weight penalty regularization*. Os modelos são treinados com gradiente estocástico descendente usando *ADAM algorithm*. Foi escolhido o tamanho de *batch size* igual a 128 com *initial rate* 1e-3 e foi treinado por 100 épocas dentre outros hiperparâmetros. Como resultado, anteriormente tinha-se como base uma precisão de 84%, com regularização L2 aumentaram para 85% e por fim 87% utilizando *dropout*. Na Figura 3 estão resultados para energia de neutrinos de elétrons (esquerda) e energia de chuva de elétrons (direita). Para cada parcela, o método de reconstrução tradicional é mostrado em azul e a curva rosa é o método baseado em CNN (HERTEL et al., 2017).

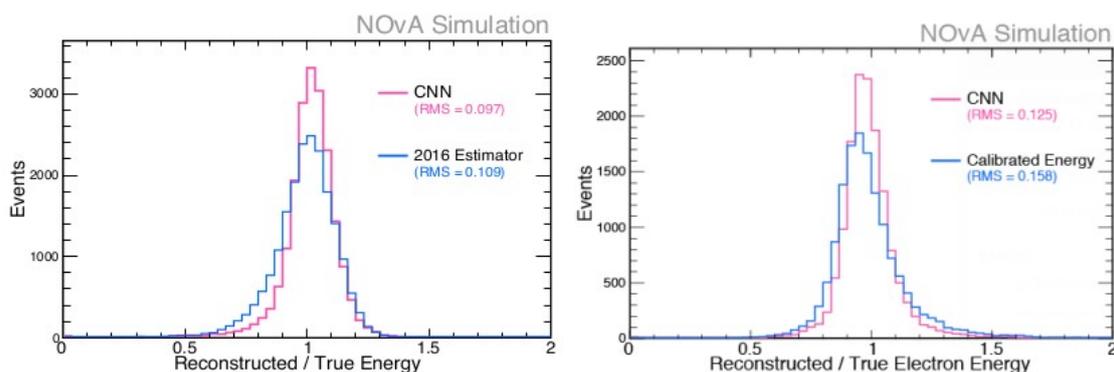


Figura 3: (HERTEL et al., 2017)

O modelo proposto mostrou-se proeminente em predição de *eléctron neutrino energy* e *eléctron shower energy*, porém, um desafio futuro é de fazer predição de *muon neutrino energy* e *muon energy* pelo fato de múons não ter trajetória de fácil detecção necessitando assim uma grande quantidade de imagens (HERTEL et al., 2017).

### 2.3 Redes neurais profundas, convolucionais, generativas e modelos lineares

Em trabalho desenvolvido por Carminati et al. (2017), uma aplicação de redes neurais profundas, convolucionais e gerativas foram utilizadas para classificação, regressão de energia e simulação de partículas produzidas em colisões de altas energias. Para classificação foi utilizada DNN com 4 camadas ocultas com 256 neurônios cada e função de ativação ReLU, ao mesmo tempo que otimizada com dropout 0.5 (50%) e treinada com gradiente estocástico Adam com taxa de aprendizado (*learning rate*) de 0.001. A base de dados foi dividida em 80% de *training set* e 20% destinada a *testing set*. Em paralelo foi construída BDT com 400 *estimators*, com profundidade 5 e taxa de aprendizado 0.5. Figura 4 demonstra os resultados.

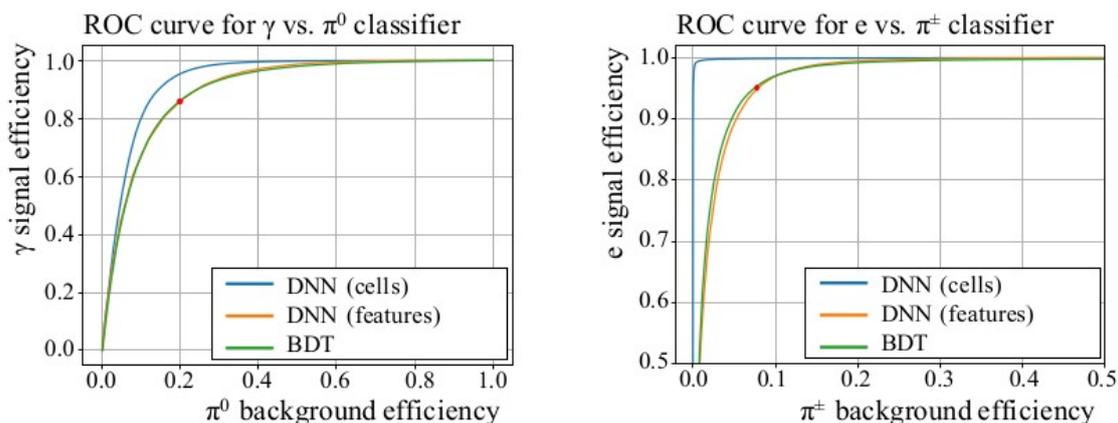


Figura 4: Fonte (CARMINATI et al., 2017)

Tratando-se de regressão, foram construídas duas CNN dedicadas para dados do *Eletromagnetic Calorimeter* (ECAL) e *Hadrons Calorimeters* (HCAL). A saída de ambas as ramificações é linearizada e mesclada, seguida por uma camada totalmente conectada (*fully connected layer*) com 1000 neurônios. O neurônio da camada de saída tem uma função de ativação linear e a função de perda é calculada pelo *Mean-Squared Error* (MSE). A distribuição da base de dados deu-se 50% *training set*, *validation set* 12,5% e *testing set* com 37,5%. Foi feita comparação com modelos lineares e chegando a conclusão que a rede DNN performa melhor como pode ser vista na Figura 5, que compara a dependência energética da resolução do calorímetro para cada tipo de partícula com a rede neural e os modelos de regressão linear simples (CARMINATI et al., 2017).

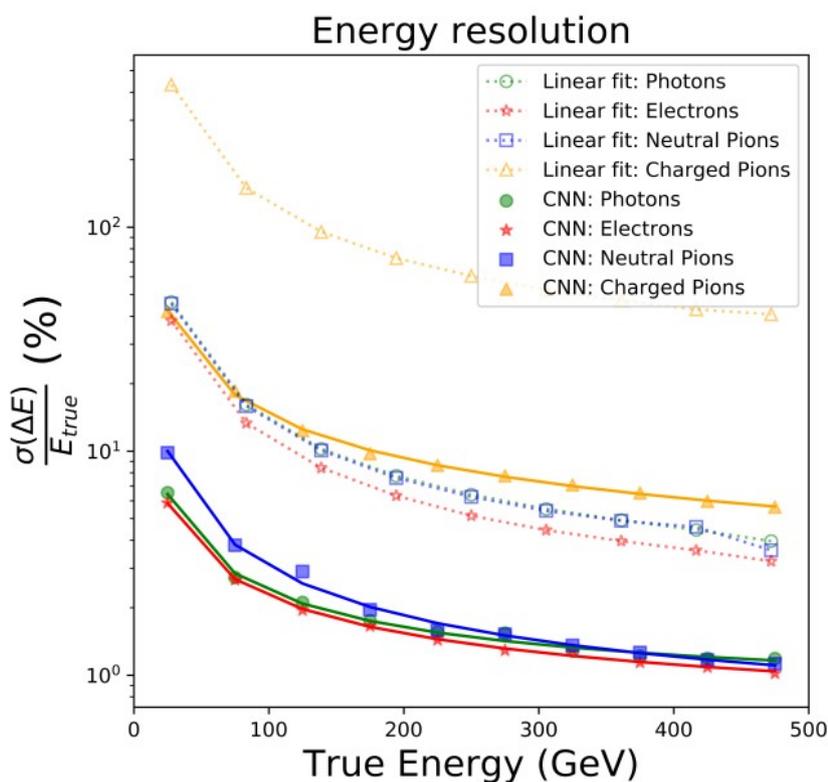


Figura 5: Fonte (CARMINATI et al., 2017)

Continuando o trabalho explorando as potencialidades das redes neurais, foi desenvolvido através de redes generativas, ou Generative Adversarial Networks (GAN) simulações de partículas como uma prova de conceito afim de testar a capacidade de gerar novas simulações. O resultado gerado pela GAN é razoável quando comparado ao sistema GeantV, porém necessita de alguns ajustes para modelar *showers*. O resultado encontra-se na Figura 6.

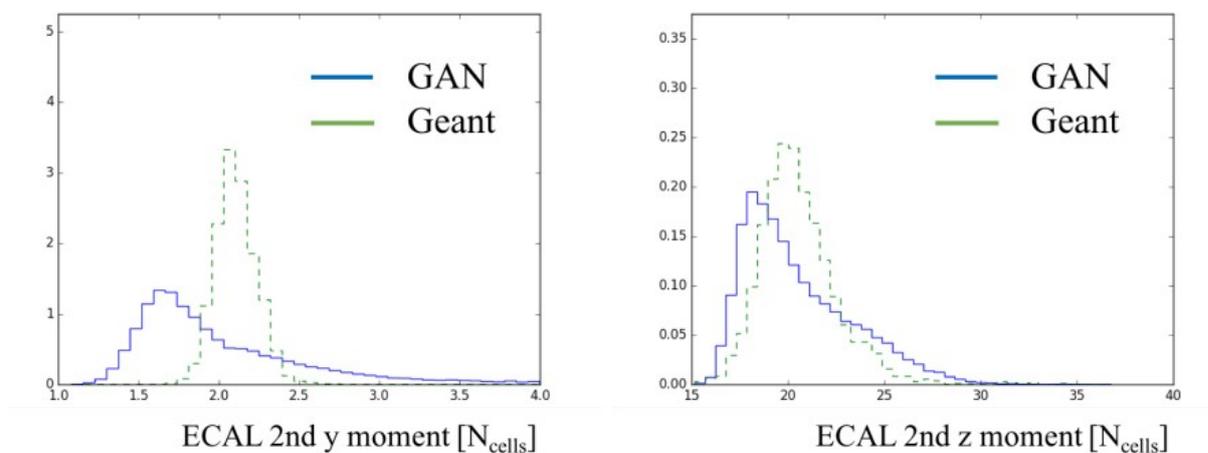


Figura 6: Fonte (CARMINATI et al., 2017)

### 3. METODOLOGIA

#### 3.1 Identificação de Partículas

Classificação de partículas em experimentos de altas energias é uma tarefa que inclui uma grande quantidade de ferramentas, dependendo da solução proposta os meios são diversos. Este experimento concentra-se em utilização de algoritmos de aprendizado de máquina para identificação de partículas em bases de dados provenientes do colisor LHC. Os dados são agerados a partir de colisões *proton-proton* dentro dos detetores e selecionados a partir de um *trigger* com a finalidade de filtrar quais melhores eventos ocorridos (DI MEGLIO et al., 2018).

A atribuição dada a este trabalho é de programação, foi desenvolvido dois algoritmos (classificadores), uma árvore de decisão e uma rede neural. Os classificadores identificam o tipo de partícula. Existem seis tipos de partículas: elétron, próton, múon, kaon, pión e “*ghost*”. O “*ghost*” é uma partícula com outro tipo que não os primeiros cinco ou um ruído do detetor. Diferentes tipos de partículas permanecem diferentes respostas nos sistemas de detecção ou subdetetores. São cinco sistemas: sistema de rastreamento (Tracking System), detetor Cherenkov de imagem em anel (RICH), calorímetros eletromagnéticos e hadrônico (ECAL e HCAL), e a câmara de múons (Múon System) (LIPPMANN, 2012).

As ferramentas e bibliotecas utilizadas para desenvolvimento do código, manipulação da base de dados, compilação dentre outras, serão detalhadas abaixo.

#### 3.1.1 Ferramentas e Bibliotecas

As ferramentas utilizadas para escrita e compilação do código foram a plataforma Colaboratory da Google, ou Colab, e o Jupyter Notebook, base para a plataforma.

Colaboratory é uma nuvem de processamento que disponibiliza uma GPU de forma gratuita.

As bibliotecas para manipulação da base de dados e cálculos matemáticos foi a Pandas e NumPy, respectivamente. Para criação da árvore de decisão e métricas de valoração, a biblioteca Scikit Learn foi empregada. Com a finalidade de utilização de ferramental gráfico, a Matplotlib. As redes neurais foram construídas a partir do Keras tendo em *backend* o Tensorflow.

### 3.2 Base de Dados

A base de dados utilizada para treinamento dos modelos foi extraída do curso *Addressing Large Hadron Collider Challenges by Machine Learning* da plataforma Coursera© e está disponível em <https://github.com/hse-aml>. Suas dimensões são 1.200.000 instâncias e 50 *features*. Algumas colunas podem ser descritas como: reconstrução de trajetória, energia depositada associada a trajetória, delta log-likelihood para partícula candidata, momentum da partícula, momentum transverso e outras. A divisão entre *training set* e *testing set* foi definida em 80% e 20%, respectivamente. Toda base de dados passou por pre processamento transformando seus valores literais (coluna “*Label*”) em numéricos e posteriormente os valores numéricos foram normalizados.

### 3.3 Árvore de Decisão

A árvore de decisão utilizada foi *Gradient Boosting Classifier* dos métodos *ensemble* da biblioteca Scikit Learn cujo objetivo é construir um modelo aditivo de maneira progressiva que permite a otimização de funções de perda de forma arbitrária e diferente. Em cada estágio, as árvores se ajustam ao erro do gradiente negativo da função de perda. A configuração do modelo é a seguinte:

```
.learning_rate=0.04  
.n_estimators=200  
.max_depth=8  
.min_samples_split=1200  
.min_samples_leaf=100  
.subsample=0.85  
.random_state=1231  
.max_features=18  
.min_impurity_decrease=0.001,  
.warm_start=True)
```

### 3.4 Rede Neural

Utilizou-se do alto nível da arquitetura do Keras para facilitar a implementação e desenvolvimento da rede neural, ao mesmo tempo que, utiliza o Tensorflow em *backend*. A configuração da rede neural ficou disposta conforme Figura 7.

Layer (type)	Output Shape	Param #
dense_4 (Dense)	(None, 245)	12250
batch_normalization_3 (Batch Normalization)	(None, 245)	980
dropout_3 (Dropout)	(None, 245)	0
activation_4 (Activation)	(None, 245)	0
dense_5 (Dense)	(None, 245)	60270
dropout_4 (Dropout)	(None, 245)	0
batch_normalization_4 (Batch Normalization)	(None, 245)	980
activation_5 (Activation)	(None, 245)	0
dense_6 (Dense)	(None, 6)	1476
activation_6 (Activation)	(None, 6)	0
Total params: 75,956		
Trainable params: 74,976		
Non-trainable params: 980		

Figura 7: Configuração Rede Neural

O modelo possui função de ativação ‘ReLU’ na camada de entrada e oculta e a ‘Softmax’ na camada de saída. O gradiente utilizado foi o ADAM com *learning rate*=0,001; *beta\_1*=0,999; *epsilon*=1e-08; *decay*=0 e *amsgrad*=False. O modelo treinou por 800 épocas com *batch\_size*=512 e possui regularização L2 com valor de 0,001 e *dropout* de 0,1.

#### 4. RESULTADOS

Os resultados obtidos serão dispostos em formato de curvas ROC, AUC, Matriz de Confusão e Relatório sendo provenientes do conjunto de teste. No Apêndice A encontra-se os modelos gerados nas bases de treino e validação. Na árvore de decisão o *logloss* foi de 0,616 para o conjunto de dados para validação e de 0,620 para testes. A rede neural desempenhou 0,573 na base de validação e 0,575 em testes.

##### 4.1 Árvore de Decisão

Tabela 3: Matriz de Confusão da Árvore de Decisão

Partícula	Electron	Ghost	Kaon	Muon	Pion	Proton
Electron	<b>33597</b>	5121	276	283	412	304
Ghost	1963	<b>34201</b>	663	328	1582	1193
Kaon	529	2643	<b>23874</b>	551	3037	9376
Muon	299	2605	585	<b>35377</b>	886	427
Pion	1050	5104	2537	1017	<b>28350</b>	1982
Proton	476	2967	962	286	2645	<b>23853</b>

Tabela 4: Relatório da Rede Neural

Partícula	Precisão	Recall	F1 Score
<i>Electron</i>	0,89	0,84	0,86
<i>Ghost</i>	0,65	0,86	0,74
<i>Kaon</i>	0,64	0,60	0,62
<i>Muon</i>	0,93	0,88	0,91
<i>Pion</i>	0,77	0,71	0,74
<i>Proton</i>	0,64	0,60	0,62
<b>Average</b>	<b>0,75</b>	<b>0,75</b>	<b>0,75</b>

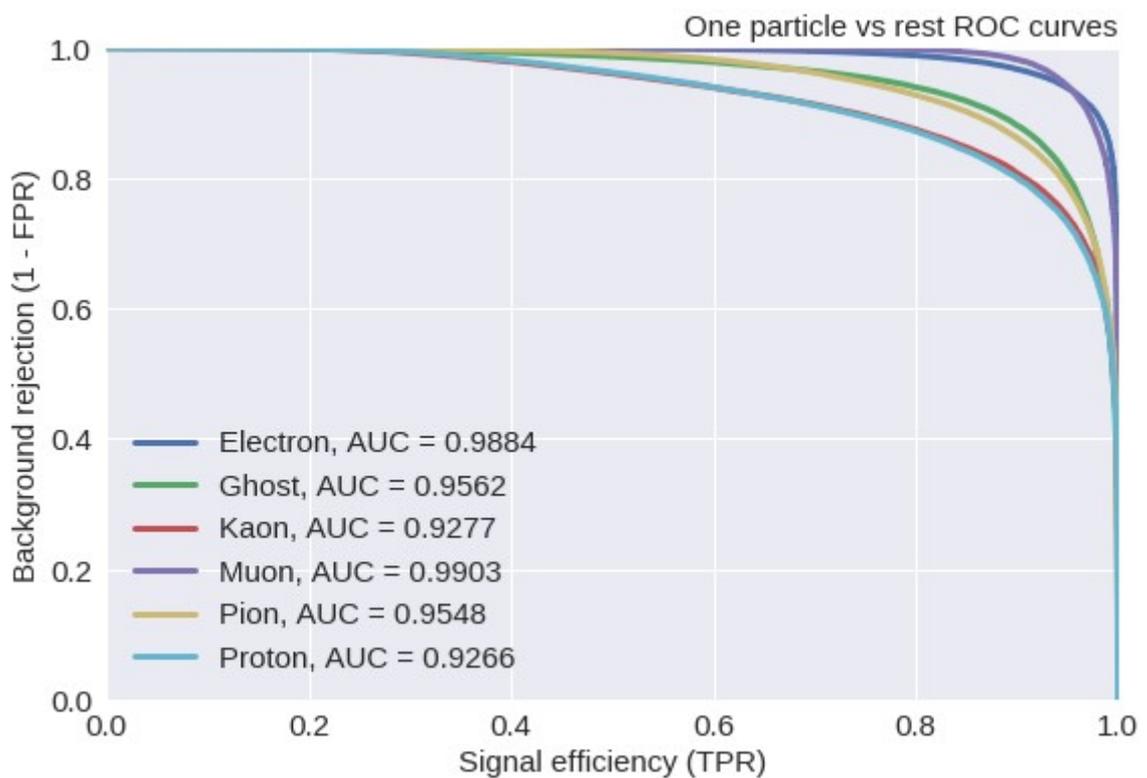


Figura 8: Curva ROC e AUC da árvore de decisão.

#### 4.2 Rede Neural

Tabela 5: Matriz de Confusão da Rede Neural

Partícula	<i>Electron</i>	<i>Ghost</i>	<i>Kaon</i>	<i>Muon</i>	<i>Pion</i>	<i>Proton</i>
<i>Electron</i>	<b>35436</b>	2674	298	348	798	439
<i>Ghost</i>	3334	<b>31387</b>	875	778	2381	1175
<i>Kaon</i>	452	1394	<b>26453</b>	665	4012	7034
<i>Muon</i>	178	1091	495	<b>36407</b>	1635	373
<i>Pion</i>	762	2539	2131	1309	<b>31663</b>	1636
<i>Proton</i>	410	1696	12294	417	3462	<b>21569</b>

Tabela 6: Relatório da Rede Neural

Partícula	Precisão	Recall	F1 Score
Electron	0,87	0,89	0,88
Ghost	0,77	0,79	0,78
Kaon	0,62	0,66	0,64
Muon	0,91	0,91	0,91
Pion	0,72	0,79	0,75
Proton	0,67	0,54	0,60
<b>Average</b>	<b>0,76</b>	<b>0,76</b>	<b>0,76</b>

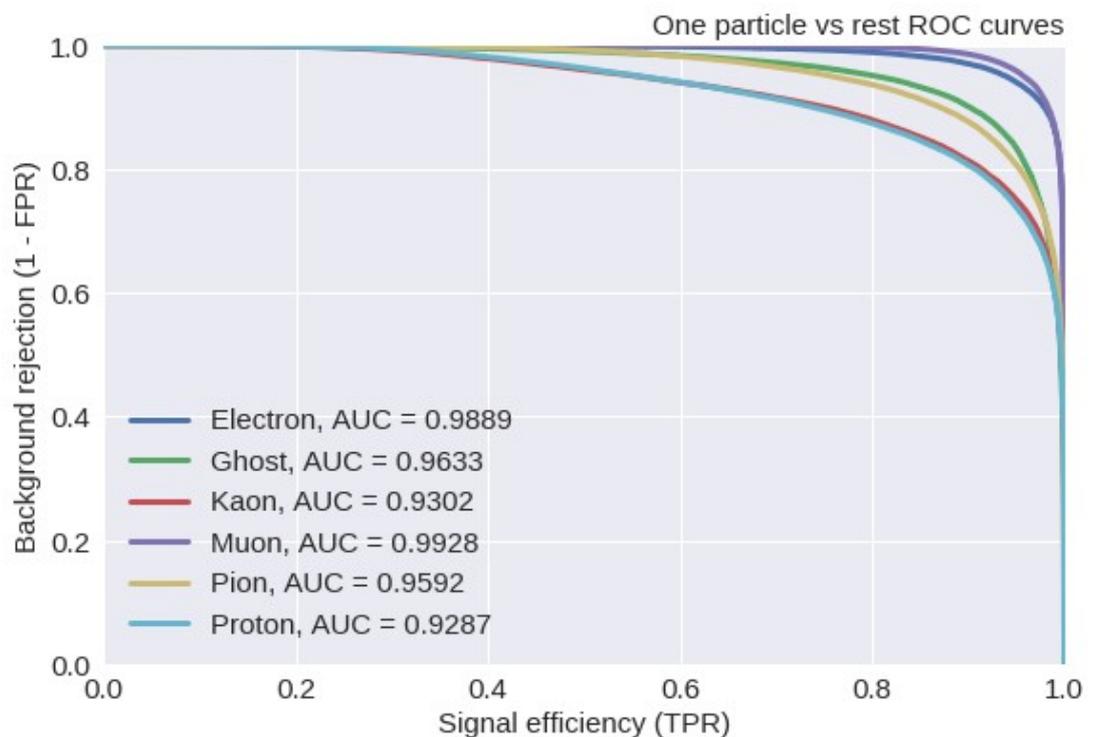


Figura 9: Curva ROC e AUC da rede neural.

## 5. CONCLUSÕES

O método se mostrou eficaz na tentativa de classificação de partículas ao passo que corresponde com as expectativas da hipótese.

Porém, em certo contraponto à hipótese preterida, as redes neurais performaram de forma muito semelhante às árvores de decisão embora trabalhos correlatos demonstrarem sua superioridade. A razão para tal fenômeno poderia derivar-se da natureza da base de dados, em que é notável a limitação de ambos modelos de exceder a precisão de 75% em suas bases de testes conforme relatórios. Como a base de dados fora retirada de um curso online, em seu fórum, os instrutores enunciavam o fato de que o logloss, ou perda, não conseguiria ser ultrapassada para menos de 0,525, levando a entender que a precisão do modelo também será limitada.

Para fins de investigação e trabalhos futuros se pode considerar novos modelos e algoritmos que se façam eficientes junto com utilização de plataformas automatizadas livres no mercado. Modelos de regressão lineares também podem ser utilizados e são razoavelmente eficiente para este contexto ao se valerem de função de saída softmax ideais para classificação em mais de duas classes.

## 6. REFERÊNCIAS

- ADAM-BOURDARIOS, C. et al. The Higgs Machine Learning Challenge. **Journal of Physics: Conference Series**, v. 664, n. 7, p. 072015, 23 dez. 2015.
- AURISANO, A. et al. A Convolutional Neural Network Neutrino Event Classifier. **Journal of Instrumentation**, v. 11, n. 09, p. P09001–P09001, 1 set. 2016.
- BALDI, P.; SADOWSKI, P.; WHITESON, D. Searching for Exotic Particles in High-Energy Physics with Deep Learning. **Nature Communications**, v. 5, n. 1, dez. 2014.
- CARMINATI, F. et al. Calorimetry with Deep Learning: Particle Classification, Energy Regression, and Simulation for High-Energy Physics. p. 6, 2017.
- CARON, S. et al. The BSM-AI project: SUSY-AI–generalizing LHC limits on supersymmetry with machine learning. **The European Physical Journal C**, v. 77, n. 4, abr. 2017.
- COHEN, T.; FREYTSIS, M.; OSTDIEK, B. (Machine) learning to do more with less. **Journal of High Energy Physics**, v. 2018, n. 2, fev. 2018.
- DI MEGLIO, A. et al. **Cern Openlab White Paper On Future Ict Challenges In Scientific Research**. [s.l.] Zenodo, 21 set. 2017. Disponível em: <<https://zenodo.org/record/998694>>. Acesso em: 13 dez. 2018.
- DI MEGLIO, A. et al. **CERN openlab white paper on future ICT challenges in scientific research**. [s.l.: s.n.].
- HERTEL, L. et al. Convolutional Neural Networks for Electron Neutrino and Electron Shower Energy Reconstruction in the NOvA Detectors. p. 5, 2017.
- KUO, M. H. et al. Health big data analytics: current perspectives, challenges and potential solutions. **International Journal of Big Data Intelligence**, v. 1, n. 1/2, p. 114, 2014.
- LIPPMANN, C. Particle identification. **Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment**, v. 666, p. 148–172, fev. 2012.
- ROE, B. P. et al. 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**, v. 543, n. 2–3, p. 577–584, maio 2005.
- Whitepaper\_brochure\_ONLINE.pdf**. , [s.d.]. Disponível em: <[http://cds.cern.ch/record/2301895/files/Whitepaper\\_brochure\\_ONLINE.pdf](http://cds.cern.ch/record/2301895/files/Whitepaper_brochure_ONLINE.pdf)>. Acesso em: 13 dez. 2018
- XIONG, W.; JIN, Y. Hybrid multi-selection based on swarm intelligence and decision tree. **The Journal of China Universities of Posts and Telecommunications**, v. 19, p. 128–134, out. 2012.

YANG, H.-J.; ROE, B. P.; ZHU, J. Studies of stability and robustness for artificial neural networks and boosted decision trees. **Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment**, v. 574, n. 2, p. 342–349, maio 2007.

APÊNDICE A

Tabela 7: Matriz de Confusão da Rede Neural (Training\_data)

<b>Partícula</b>	<i>Electron</i>	<i>Ghost</i>	<i>Kaon</i>	<i>Muon</i>	<i>Pion</i>	<i>Proton</i>
<i>Electron</i>	<b>107187</b>	5785	1108	958	3107	1509
<i>Ghost</i>	12567	<b>89084</b>	3355	2744	8201	3978
<i>Kaon</i>	1168	3541	<b>80935</b>	1854	11971	20768
<i>Muon</i>	492	2648	1672	<b>108996</b>	4776	1315
<i>Pion</i>	2133	6460	7359	4215	<b>95005</b>	4813
<i>Proton</i>	1152	4420	37604	1153	10823	<b>65144</b>

Tabela 8: Relatório da Rede Neural (Training\_data)

<b>Partícula</b>	<i>Precisão</i>	<i>Recall</i>	<i>F1</i>
	<i>n</i>		<i>Score</i>
<i>Electron</i>	0,86	0,90	0,88
<i>Ghost</i>	0,80	0,74	0,77
<i>Kaon</i>	0,61	0,67	0,64
<i>Muon</i>	0,91	0,91	0,91
<i>Pion</i>	0,71	0,79	0,75
<i>Proton</i>	0,67	0,54	0,60
<b>Average</b>	<b>0,76</b>	<b>0,76</b>	<b>0,76</b>

Tabela 9: Matriz de Confusão da Rede Neural (Training\_data)

<b>Partícula</b>	<i>Electron</i>	<i>Ghost</i>	<i>Kaon</i>	<i>Muon</i>	<i>Pion</i>	<i>Proton</i>
<i>Electron</i>	<b>35764</b>	2674	291	339	817	468
<i>Ghost</i>	3285	<b>31442</b>	898	723	2555	1238
<i>Kaon</i>	424	1326	<b>26422</b>	625	4024	6932
<i>Muon</i>	165	1079	519	<b>36167</b>	1573	419
<i>Pion</i>	748	2464	2231	1348	<b>31539</b>	1645
<i>Proton</i>	413	1660	12338	373	3493	<b>21579</b>

Tabela 10: Relatório da Rede Neural (Validation\_data)

<b>Partícula</b>	<i>Precisão</i>	<i>Recall</i>	<i>F1</i>
	<i>n</i>		<i>Score</i>
<i>Electron</i>	0,88	0,89	0,88

<i>Ghost</i>	0,77	0,78	0,78
<i>Kaon</i>	0,62	0,66	0,64
<i>Muon</i>	0,91	0,91	0,91
<i>Pion</i>	0,72	0,79	0,75
<i>Proton</i>	0,67	0,54	0,60
<b>Average</b>	<b>0,76</b>	<b>0,76</b>	<b>0,76</b>

Tabela 11: Matriz de Confusão da Rede Neural (Training\_data)

<b>Partícula</b>	<i>Electron</i>	<i>Ghost</i>	<i>Kaon</i>	<i>Muon</i>	<i>Pion</i>	<i>Proton</i>
<i>Electron</i>	<b>107591</b>	7758	876	771	1724	934
<i>Ghost</i>	7667	<b>98184</b>	2401	1698	6873	3106
<i>Kaon</i>	1232	4849	<b>79602</b>	1802	11203	21549
<i>Muon</i>	547	3018	1084	<b>110636</b>	3848	766
<i>Pion</i>	2047	7729	5380	3854	<b>96673</b>	4302
<i>Proton</i>	1229	5709	26020	1193	10470	<b>75675</b>

Tabela 12: Relatório da Árvore de Decisão (Training\_data)

<b>Partícula</b>	<i>Precisio n</i>	<i>Recall</i>	<i>F1 Score</i>
<i>Electron</i>	0,89	0,90	0,90
<i>Ghost</i>	0,77	0,82	0,79
<i>Kaon</i>	0,69	0,66	0,68
<i>Muon</i>	0,92	0,92	0,92
<i>Pion</i>	0,74	0,81	0,77
<i>Proton</i>	0,71	0,63	0,67
<b>Average</b>	<b>0,79</b>	<b>0,79</b>	<b>0,79</b>

Tabela 13: Matriz de Confusão da Árvore de Decisão (Validation\_data)

<b>Partícula</b>	<i>Electron</i>	<i>Ghost</i>	<i>Kaon</i>	<i>Muon</i>	<i>Pion</i>	<i>Proton</i>
<i>Electron</i>	<b>33985</b>	5122	295	289	392	270
<i>Ghost</i>	1919	<b>34380</b>	687	319	1685	1151
<i>Kaon</i>	535	2617	<b>23851</b>	567	2951	9232
<i>Muon</i>	298	2489	561	<b>35261</b>	886	427
<i>Pion</i>	1069	5033	2524	1010	<b>28399</b>	1940
<i>Proton</i>	446	2931	9763	271	2643	<b>23802</b>

Tabela 14: Relatório da Árvore de Decisão (Validation\_data)

<b>Partícula</b>	<i>Precisio n</i>	<i>Recall</i>	<i>F1 Score</i>
<i>Electron</i>	0,89	0,84	0,86
<i>Ghost</i>	0,65	0,86	0,74
<i>Kaon</i>	0,63	0,60	0,62
<i>Muon</i>	0,93	0,88	0,91
<i>Pion</i>	0,77	0,71	0,74
<i>Proton</i>	0,65	0,60	0,62
<b>Average</b>	<b>0,75</b>	<b>0,75</b>	<b>0,75</b>