ESTUDO DE SEPARAÇÃO SINAL/RUÍDO EM EXPERIMENTOS USANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA

Palavras-Chave: SEPARAÇÃO SINAL/RUÍDO, APRENDIZADO DE MÁQUINA, SENSORES

Autores:
HENRIQUE DE MIRANDA LIMA DOS SANTOS, IFGW – UNICAMP
Dr. LUIS FERNANDO GOMEZ GONZALEZ, IC – UNICAMP
Prof(a). Dr(a). JULIANA FREITAG BORIN, IC – UNICAMP

1 Introdução

Desde o início do século XX, a computação tem sido transformadora para o progresso científico. Na ciência experimental, é uma ferramenta que permitiu avanços em planejamento, escalabilidade e reprodutibilidade, características essenciais para a formulação de descobertas científicas. Além disso, se tornou instrumento conveniente para armazenamento e análise de conjuntos de dados relevantes.

Na física experimental de altas energias, a computação científica é particularmente crucial, devido a extensos conjuntos de medidas e à complexidade das análises necessárias. Muitas vezes, a principal tarefa do processamento de dados é classificar a discrepância entre sinal e ruído nos experimentos. Isso pois, em tal contexto, o ruído nos dados de tais experimentos podem significar a diferença entre um novo entendimento da matéria no universo e a não compreensão de um evento [1].

Tendo isso em vista, neste projeto buscou-se desenvolver métodos de seleção de sinal em conjuntos de dados de um experimento de detecção de partículas usando radiação *Cherenkov* [2] em água. Os sinais de ruído e partículas foram obtidos do experimento Neutrinos Angra, cujo objetivo é detectar antineutrinos gerados no reator nuclear da usina de Angra II, na Central Nuclear Almirante Álvaro Alberto, localizada em Angra dos Reis, RJ [3].

Os modelos desenvolvidos foram baseados em algoritmos de aprendizado de máquina (*machine learning*), especificamente os modelos supervisionados [4]: Árvores de Decisão (*Decision Tree*), *Support Vector Machine* [5] e *Multilayer Perceptron* [6]. Além desses, o modelo não-supervisionado *Kmeans* também foi testado [7]. Tais aplicações podem representar um avanço promissor na área, por não terem sido exploradas em muitos trabalhos.

Assim, após o desenvolvimento dos modelos, foi possível realizar a comparação com os métodos tradicionais de análise dos dados. Seguindo a análise, verificou-se que os algoritmos desenvolvidos foram superiores em valor estatístico de predição aos métodos elaborados anteriormente.

2 Metodologia

Para a criação dos métodos baseados em aprendizado de máquina, foi utilizada a linguagem de programação Python (versão 3). Além disso, foram importadas as bibliotecas *scikit-learn* [8], *numpy*, *pandas*, *matplotlib*, dentre outras. Assim, a metodologia pode ser divida em quatro etapas:

1. ***Feature Engineering*:** tanto os modelos supervisionados quanto os não-supervisionados trabalham com características que serão utilizadas para definir a classe de cada pulso (ruído ou partícula). Assim, foram escolhidas amplitude, largura à meia altura, valor eficaz (ou RMS, *root mean squared*), carga total (obtida através da integração do pulso ao longo do intervalo de tempo), desvio padrão e as derivadas mínima, média e máxima. A Figura 1 apresenta um exemplo de pulso obtido no experimento es suas características.

2. **Classificação dos Dados:** foi utilizada a função *Langauss* para o ajuste para os dados. O parâmetro R^2 foi utilizado para a classificação. Além disso, foi utilizado um conjunto de dados especial, obtido a partir do fenômeno de decaimento do múon, para definição dos dados.

3. **Implementação dos Modelos:** Os modelos treinados foram: Árvores de Decisão, *Support Vector Machine*, *Multilayer Perceptron* e *Kmeans*.

4. **Avaliação dos Resultados:** com intuito de analisar a qualidade das previsões dos modelos, foi utilizado o parâmetro F1 (ou *F1 score*), assim como o tempo de processamento para um conjunto de dados padrão (em torno de 3 milhões de pulsos). Além da comparação entre os modelos desenvolvidos, foram utilizados como referência três métodos tradicionais: ajuste dos dados pela função *Langauss*, o *threshold* (ou limite) de amplitude, e o *threshold* por carga (ou integração) maior que zero - método utilizado atualmente pela equipe do experimento Neutrinos Angra. O processamento dos dados foi realizado em um computador pessoal com a seguinte configuração: placa Gigabyte B5m Aorus Elite, processador AMD Ryzen 9 5900X (12 core × 24), 48 GB de RAM, placa gráfica Nvidea GTX 1050 TI 4GB e sistema operacional Ubuntu 22.04.4 LTS (64 bits).

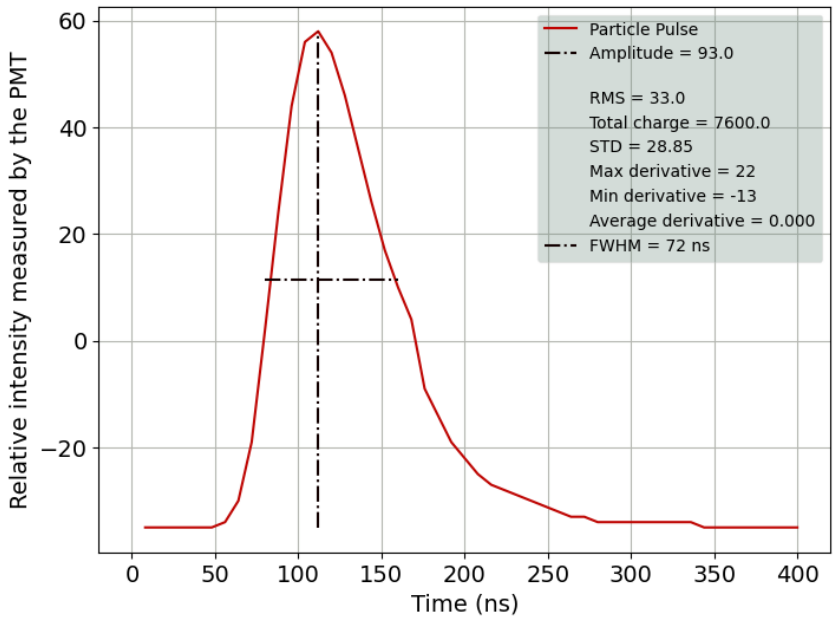

Figura 1: Exemplo de pulso obtido no experimento e as características utilizadas na fase de *feature engineering*

3 Resultados e Discussão

A Tabela 1 apresenta os resultados obtidos com os modelos de aprendizado de máquina e com os métodos tradicionais. Ao observar os dados, é possível verificar que os modelos de aprendizado de máquina retornam resultados superiores em relação à qualidade estatística das previsões. Quanto ao tempo de execução, os modelos de aprendizado de máquina foram superiores ao método de ajuste pela função *Langauss*, por essa ser uma função complexa, mas em geral foram inferiores aos métodos de *threshold*.

Como pode ser observado na Figura 2, foi possível desenvolver três modelos superiores ao método tradicionalmente utilizado na análise de dados do experimento. Assim sendo, foi demonstrado o potencial que os algoritmos de aprendizado de máquina, principalmente os supervisionados, têm para distinguir pulsos de sinal e ruído com base em suas características. Os modelos implementados têm baixo nível de complexidade e de otimização, mostrando que ainda há espaço para evolução em relação a esses resultados.

O maior desafio desse projeto foi a classificação dos dados. O ajuste pela função Langauss é (CITAR) utilizado tradicionalmente na física de partículas. No entanto, possui algumas limitações, por realizar

	F1 Score	**Tempo de execução**
Decision Tree	0,9741	3,3 segundos
Multilayer Perceptron	0,9736	1701,2 segundos
Threshold Ideal	0,9556	25,7 segundos
Support Vector Machine	0,9104	789,0 segundos
Threshold Atual	0,8149	199,9 segundos
KMeans	0,2715	10,0 segundos
Ajuste *Langauss*	1	87,2 horas

Tabela 1: F1 (*F1 score*) e tempo de execução da predição para os métodos testados.

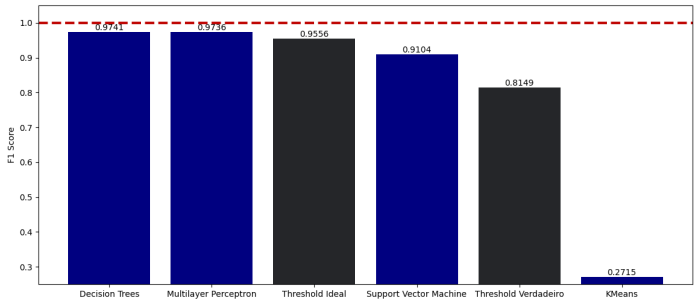

Figura 2: Resultados do valor F1 para cada modelo desenvolvido (colunas azuis) assim como para os modelos tradicionais (colunas cinzas). Em F1 = 1, se encontra o valor máximo (ideal) para o *F1 score*.

bons ajustes a pulsos nulos ou quase nulos, que deveriam ser classificados como ruído. Para solucionar tal problema, idealmente, deveriam ter sido utilizados pulsos provenientes da calibração dos sensores. No caso de experimentos de detecção de partículas, calibrar significa utilizar fontes de radiação bem conhecidas para coletar dados de contagem e deposição de energia no detector. Como isso não foi feito no experimento Neutrinos Angra, a utilização de sinais gerados pelo fenômeno de decaimento do múon foi elegante e conveniente.

Isso se mostra também outro ponto em que esse trabalho pode obter resultados melhores. Com a calibração do experimento e a utilização desses dados, seria possível treinar os modelos de inteligência artificial com conjuntos de dados com maior distinção entre ruído e fenômenos de partículas. Os algoritmos seriam, então, mais bem sucedidos e teriam maior qualidade nas suas predições.

Além disso, existe a possibilidade de aplicar os algoritmos desenvolvidos em sistemas embarcados do experimento. Dessa forma, o modelo funcionaria como uma porta lógica de "aprovação"do sinal obtido, classificando-o como sinal de partícula ou de ruído. Como visto na análise dos resultados, o experimento possui atualmente um método que aceita muitos falsos positivos. Assim, seria possível descartar os pulsos de ruído que antes seriam armazenados, permitindo uma taxa de obtenção de dados maior do que a atual do experimento.

4 Conclusões

O principal objetivo do projeto é demonstrar o potencial dos métodos de *machine learning* na classificação de *datasets* que apresentam ruído. O trabalho foi bem sucedido e dos diversos modelos testados, foi possível implementar modelos superiores aos convencionais. Em detectores de partículas, a tarefa é contraintuitiva do ponto de vista computacional, e os métodos utilizados até então falham por excluir um grande conjunto de sinais de partículas, ou por incluir uma grande quantidade de sinal proveniente de ruído. Além disso, quando comparados com o método mais preciso, de ajuste pela função *Langauss*, os modelos se mostraram até 5 ordens de magnitude mais econômicos computacionalmente. Apesar de ser necessário o ajuste para treinar os modelos, tal treinamento precisa ocorrer apenas uma vez, sem demandar tanto tempo de processamento para as aplicações futuras dos algoritmos.

Assim sendo, demonstrou-se que o investimento de tempo e esforço na aplicação de técnicas baseadas em aprendizado de máquina pode trazer frutos positivos na análise de dados de experimentos que envol-

vem classificação de sinal e ruído. Os modelos de aprendizado de máquina que se destacaram foram os modelos Árvore de Decisão e *Multilayer Perceptron*. Com a possibilidade da aplicação dos modelos em sistemas embarcados, potencialmente terão mais eficiência computacional, assim como permitirão maior fluxo de dados. Isso contribuirá para a qualidade dos dados do experimento, proporcionando resultados físicos mais significativos.

Referências Bibliográficas

[1] W. R. Leo, *Techniques for Nuclear and Particle Physics Experiments: A How-to Approach*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. [Online]. Available: https://link.springer.com/10.1007/978-3-642-57920-2

[2] G. F. Knoll, *Radiation detection and measurement*, 4th ed. Hoboken, N.J: John Wiley, 2010, oCLC: ocn612350364.

[3] E. Kemp, W. V. Santos, J. C. Anjos, P. Chimenti, L. F. G. Gonzalez, G. P. Guedes, H. P. Lima, R. A. Nóbrega, I. M. Pepe, and D. B. S. Ribeiro, "Results from ON-OFF analysis of the Neutrinos-Angra detector," 2024, version Number: 1. [Online]. Available: https://arxiv.org/abs/2407.20397

[4] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. [Online]. Available: https://link.springer.com/10.1007/978-0-387-30164-8

[5] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: http://link.springer.com/10.1007/BF00994018

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/

[7] A. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 2016. [Online]. Available: https://books.google.com.br/books?id=1-4lDQAAQBAJ

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html