

# Análise comparativa de métricas de avaliação em modelos de aprendizado de máquina na detecção de ataques de spoofing de GPS

Ana Carla Rodrigues<sup>1</sup>, Jessica Fileto<sup>1</sup>

<sup>1</sup> Centro de Matemática, Computação e Cognição  
Universidade Federal do ABC (UFABC)  
Av. dos Estados, 5001 – 09210-580 – Santo André – SP – Brasil

{carla.rodrigues, jessica.fileto}@ufabc.edu.br

**Resumo.** *O Veículo Aéreo Não Tripulado (VANT), comumente conhecido como drone, é uma plataforma que está revolucionando indústrias ao redor do mundo, oferecendo soluções de alta flexibilidade e riscos reduzidos. O Global Positioning System (GPS) fornece ao VANT navegação precisa, rastreamento e voo autônomo. Os VANTs podem sofrer ataques de spoofing nas coordenadas de GPS, comprometendo a segurança da informação e demonstrando que a precisão na análise dos dados são cruciais na tomada de decisões. A inteligência artificial pode ser uma aliada na detecção destes ataques através de técnicas sofisticadas, reduzindo a quantidade de falsos alarmes e fornecendo informações confiáveis por meio do aprendizado de máquina que exerce papel fundamental na classificação destes ataques. A proposta deste projeto é realizar uma análise comparativa das métricas de avaliação dos seguintes modelos de aprendizado de máquina: Random Forest, XGBoost, SVM, K-NN, Gaussian Naive Bayes e Extremely Randomized Trees, visando identificar aquele que melhor equilibre a detecção correta de spoofing e viabilidade de implementação do modelo.*

## 1. Introdução

Os veículos aéreos não tripulados (VANTS), também conhecidos como drones, são um tipo de tecnologia fundamental tanto em áreas civis quanto militares. Seu uso reduz a exposição humana a tarefas repetitivas, de longa duração e em alguns casos, de alto risco [Brasil 2025]. Desempenham papel importante nas seguintes áreas: gerenciamento de desastres, vigilância aérea, fotografia aérea de rastreamento, pesquisa e resgate, monitoramento de gado, dentre outros. [Titouna e Naït-Abdesselam 2021]

Por serem operados de maneira autônoma e remota suas coordenadas são essenciais para o sucesso de suas missões, para isso são usados sinais de Sistema Global de Navegação por Satélite (GNSS), sendo o mais conhecido o *Global Positioning System* (GPS), esses sinais podem ser criptografados ou não [A. Faria1 et al. 2018]. Sendo os não criptografados mais suscetíveis a ataques conhecidos como *Spoofing* de GPS [Srinivasan S e Sathyadevan 2023].

O ataque de *spoofing* de GPS é caracterizado por um atacante que faz uso de antenas terrestres emitindo sinais falsos. [Jafarnia-Jahromi, Broumandan e Nielsen 2012] Estes ataques são classificados em: simples, intermediários e sofisticados [Aissou et al. 2021].

Para garantir um voo seguro do VANT é essencial ter medidas para detecção dos ataques de *Spoofing de GPS*. A identificação precisa desses ataques é extremamente importante, pois técnicas sofisticadas de *spoofing* podem causar disrupção no funcionamento do VANT, portanto são necessários métodos robustos e adaptativos [İşleyen e Bahtiyar 2024]. Nesse contexto algoritmos de aprendizado de máquina tem se mostrado promissores, pois são capazes de analisar grandes volumetrias de dados, identificando padrões e anomalias. Tais abordagens oferecem uma resposta robusta e adaptativa a essas ameaças contribuindo para as operações dos VANTs [İşleyen e Bahtiyar 2024].

Neste trabalho será feita a reimplimentação dos algoritmos *RandomForest* e *XGBoost* e seus respectivos experimentos do artigo [Aissou et al. 2021], com o objetivo de comparar esses algoritmos com outros que não são baseados em árvore, sendo eles: *Support Vector Machines* (SVM), *K-NN*, *Gaussian Naive Bayes* e *Extremely Randomized Trees*.

## 2. Trabalhos relacionados

Diversos estudos recentes tem utilizado aprendizado de máquina para detecção de ataques de *spoofing* de GPS. Podemos mencionar algumas abordagens utilizadas:

- **Redes neurais profundas (DNN):** Extração de padrões complexos em séries temporais dos sinais GNSS. [İşleyen e Bahtiyar 2024]
- **Máquinas de vetores de suporte (SVM):**  
Abordagem teve como objetivo a identificação de discrepâncias entre as posições determinadas pelos sinais de GPS e aquelas medidas pelo sistema de navegação inercial (INS). Especificamente, o modelo SVM foi treinado através dos erros obtidos das diferenças posicionais, permitindo que o sistema detecte inconsistências que poderiam indicar um ataque de *spoofing* de GPS. [Panice et al. 2017]
- **Redes neurais convolucionais (CNN):** Em comparação aos modelos tradicionais de aprendizagem de máquina, os de aprendizagem profunda obtiveram alta acurácia na detecção dos ataques de *spoofing*. A grande vantagem desses modelos baseados em aprendizagem profunda, é pelo fato de aprenderem automaticamente a extrair as *features* sem precisarem de intervenção humana. Além de se adaptarem melhor a *datasets* mais complexos. [Sun et al. 2023]
- **Aprendizagem supervisionada (Baseado em Árvores):** Aissou et al. (2021) utilizou os seguintes modelos baseados em árvores: Random Forest, Gradient Boost, XGBoost e LightGBM para fazer um comparativo de qual seria melhor na detecção dos ataques de *spoofing* de GPS. Sendo que o XGBoost obteve a melhor acurácia (95,52%).
- **IA Generativa:** Abordagem que em comparação com outros modelos de aprendizagem de máquina, se destacou pela eficácia na detecção dos ataques de *spoofing* e *jamming*. [El alami e Rawat 2024]

## 3. Delimitação e Escopo

O artigo [Aissou et al. 2021] não cita o *dataset* usado para o treinamento dos modelos, então para as análises decorrentes deste artigo será utilizado o *Mendeley Data* [Aissou 2022], que foi disponibilizado pelos autores do artigo. Esse *dataset* é uma versão simplificada, pois é representada em duas dimensões. Esses dados foram gerados de sinais GPS autênticos e contém aproximadamente 500,000 dados.

Nas próximas seções se estabelecerá a execução do trabalho. Na seção 4, faremos nossa proposta de mudança ao trabalho de [Aissou et al. 2021], discutindo a metodologia e modelos de aprendizado de máquina a serem implementados. Logo, os resultados e discussões serão apresentados na seção 5. Finalmente, nossas conclusões sobre o trabalho serão expostos na seção 6.

## 4. Metodologia

A metodologia adotada e os passos seguidos foram inspirados no artigo de [Aissou et al. 2021], que implementou os algoritmos baseados em árvore: *Random Forest*, *Gradient Boost*, *XGBoost* e *LightGBM* com o objetivo de explorar os dados obtidos.

Será realizada a reimplementação dos algoritmos (i) *Random Forest* e (ii) *XGBoost*, ambos obtiveram o pior e melhor desempenho, respectivamente, ponderando as métricas de tempo de execução e uso de memória, no comparativo dos modelos baseados em árvore. Ambos algoritmos serão comparados com outros modelos que não são baseados em árvore, sendo eles: (iii) SVM, (iv) *K-NN* e (v) *Gaussian Naive Bayes*.

Será incluído o modelo baseado em árvore conhecido como (vi) *Extremely randomized trees* que é um modelo melhorado do *Random Forest*, que se difere pelo fato que no *Extremely randomized trees* não existe a fase de *bagging* e no momento da separação dos nós, esta escolha é feita aleatoriamente [Geurts, Ernst e Wehenkel 2006].

Será feita a análise comparativa entre os algoritmos baseados em árvore e os demais algoritmos, com o objetivo de identificar aquele que melhor equilibre a detecção correta de *spoofing* e viabilidade de implementação do modelo. Também ocorrerá a análise entre os modelos baseados em árvore para averiguar se o *Extremely Randomized Trees* obtém um desempenho melhor que o *Random Forest*. O algoritmo *XGBoost* será mantido, pois obteve o melhor desempenho no artigo de [Aissou et al. 2021]. Portanto ele será usado como base para comparação com os demais algoritmos.

Para a avaliação do desempenho dos modelos será utilizada a biblioteca *PyCM* do Python, que é uma biblioteca de matriz de confusão multiclasse, que implementa diversas métricas de classificação. [Haghighi et al. 2018]

Serão analisadas a acurácia, precisão e a quantidade de falsos negativos. Além disso, será feita a análise do tempo de treinamento, predição e uso de memória de cada modelo, com o objetivo de identificar o mais viável para implementação em VANTs.

### 4.1. Divisão dos dados

A base será dividida em porções diferentes para testes e o treinamento do modelo, sendo 70% para treino e 30% para testes conforme realizado no artigo. [Aissou et al. 2021]

### 4.2. Matthews Correlation Coefficient (MCC)

O *Matthews Correlation Coefficient* (MCC) é uma métrica de avaliação de modelos de classificação binária, que posteriormente foi extrapolada para classificação multiclasse. [Jurman, Riccadonna e Furlanello 2012]

Neste trabalho será utilizado o MCC, juntamente com a acurácia para avaliar a eficácia dos modelos de aprendizado de máquina na detecção de ataques de *spoofing* de GPS.

### 4.3. Pré-Processamento dos dados

Será aplicada a técnica de pré processamento *Principal Component Analysis* (PCA) para melhorar a precisão dos resultados finais. Ela consiste em reduzir a dimensão do dataset para deixar apenas as informações mais relevantes, removendo assim as redundâncias [Pearson 1901]. Assim como serão utilizadas outras duas técnicas adicionais conhecidas como *Standard Scaler* para melhorar o desempenho do processamento [Pearson 1894] e o *Random Under Sampler* para reduzir o desbalanceamento das classes que dificultam o treinamento e a análise dos dados [Japkowicz 2000].

Essas técnicas são diferentes da usada no artigo de [Aissou et al. 2021] que se chama *Spearman Correlation Coefficient*.

### 4.4. Análise do desempenho

Para analisar a eficácia do algoritmo através dos dados disponíveis é necessário avaliar os comportamentos dos modelos e para isso serão utilizadas as seguintes métricas: (i) probabilidade de detecção, (ii) acurácia, (iii) probabilidade de detecção falsa, (iv) probabilidade de alarme falso, (v) taxa de erro, (vi) coeficiente MCC, (vii) F1 score, (viii) especificidade.

## 5. Resultados e discussões

Com base nos dados da Tabela 1 podemos avaliar o desempenho dos algoritmos utilizados para classificar ataques de *spoofing* aos VANTS.

**Tabela 1. Métricas de desempenho**

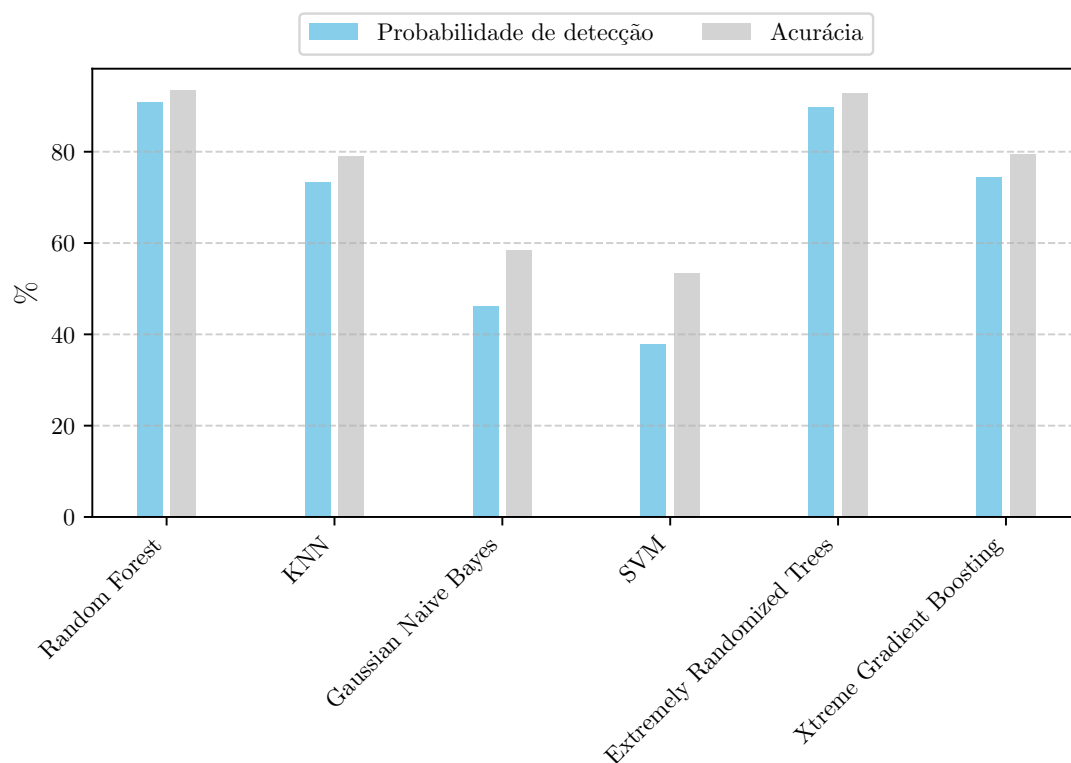
| Métrica                         | Random Forest | Extremely Randomized Trees | XGBoost | KNN    | Gaussian Naive Bayes | SVM    |
|---------------------------------|---------------|----------------------------|---------|--------|----------------------|--------|
| Probabilidade de detecção       | 90.76%        | 89.71%                     | 74.31%  | 73.32% | 46.21%               | 37.68% |
| Acurácia                        | 93.51%        | 92.65%                     | 79.29%  | 78.96% | 58.31%               | 53.25% |
| Probabilidade de detecção falsa | 9.24%         | 10.29%                     | 25.69%  | 26.68% | 53.79%               | 62.32% |
| Probabilidade de alarme falso   | 1.63%         | 1.74%                      | 3.79%   | 5.37%  | 26.45%               | 15.95% |
| Taxa de Erro                    | 6.49%         | 7.35%                      | 20.71%  | 21.04% | 41.69%               | 46.75% |
| Coeficiente MCC                 | 0.81          | 0.79                       | 0.57    | 0.55   | 0.16                 | 0.18   |
| F1-Score                        | 0.91          | 0.90                       | 0.77    | 0.76   | 0.53                 | 0.45   |
| Especificidade                  | 0.98          | 0.98                       | 0.96    | 0.95   | 0.74                 | 0.84   |

### 5.1. Probabilidade de detecção

Os algoritmos com a maior probabilidade de detecção foram o *Random Forest* (90.76%) e o *Extremely Randomized Trees* (89.71%), enquanto os que apresentaram pior desempenho foram o *SVM* (37.68%) e o *Gaussian Naive Bayes* (46.21%). A figura 1 mostra a probabilidade de detecção e a acurácia dos modelos.

### 5.2. Acurácia

A acurácia do *Random Forest* foi de 93.51%, seguido pelo *Extremely Randomized Trees* com 92.65%, enquanto o *XGBoost* obteve 79.29% e o *SVM* 53.25%. Esses resultados indicam que o *Random Forest* e o *Extremely Randomized Trees* são os mais eficazes na detecção de ataques de *spoofing*. O *XGBoost* obteve uma acurácia inferior ao *Random Forest*, o que indica que as técnicas de pré-processamento não foram eficazes para esse algoritmo. O *SVM* obteve a menor acurácia, indicando que não é adequado para a detecção de ataques de *spoofing*.



**Figura 1. Gráfico de probabilidade de detecção e acurácia.**

### 5.3. Probabilidade de detecção falsa

Os algoritmos que não são baseados em árvore obtiveram as maiores probabilidades de detecção falsa, sendo o *Gaussian Naive Bayes* (53.79%) e o *SVM* (62.32%). Esses resultados indicam que os algoritmos não baseados em árvore são ineficazes na detecção de ataques de *spoofing*.

### 5.4. Probabilidade de alarme falso

O *Random Forest* obteve a menor probabilidade de alarme falso (1.63%). Em comparação com o estudo de [Aissou et al. 2021], na qual obteve uma probabilidade de alarme falso de (8.53%), o *Random Forest* obteve uma melhora significativa. Possivelmente devido as técnicas de pré-processamento implementadas neste trabalho.

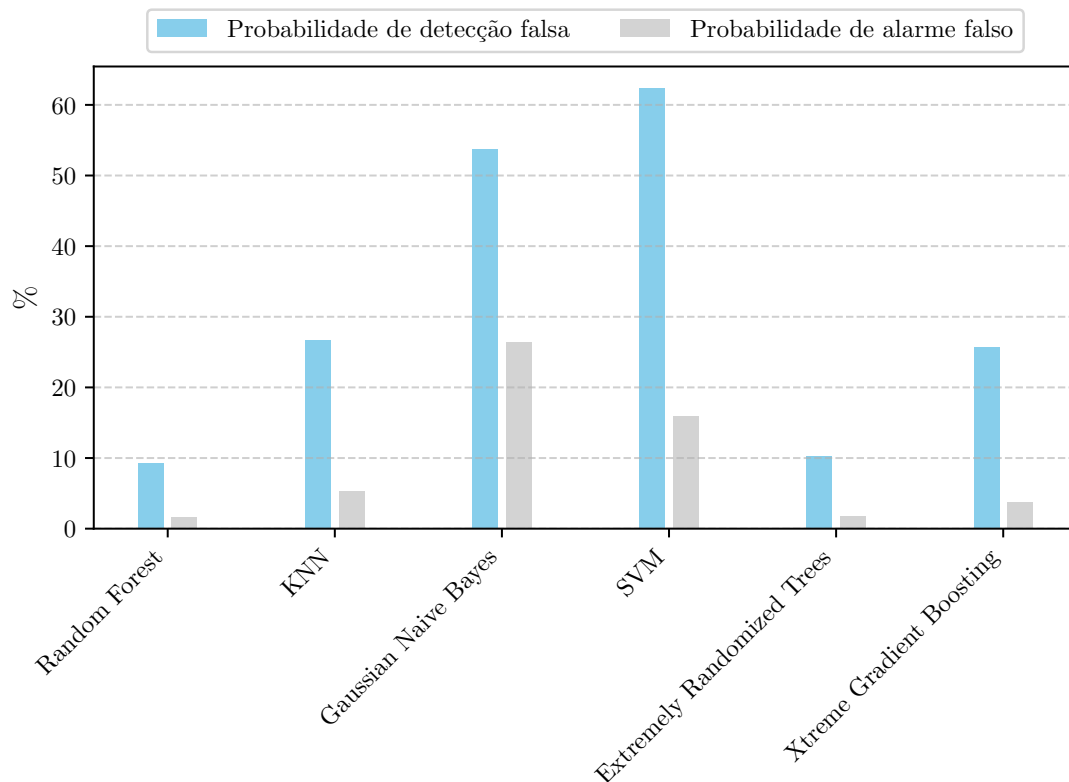
A figura 2 mostra a probabilidade de detecção falsa e alarme falso dos modelos.

### 5.5. Taxa de erro

Os algoritmos com a menor taxa de erros são: *Random Forest* (6.49%) e o *Extremely Randomized Trees* (7.35%), enquanto os que apresentaram pior performance foram *SVM* (46.75%) e *Gaussian Naive Bayes* (41.66%).

### 5.6. Coeficiente MCC

O *Random Forest* apresentou o maior MCC (0.81) provando ser confiável na detecção, enquanto o *SVM* o pior (0.45).



**Figura 2. Gráfico de probabilidade de detecção falsa e alarme falso.**

### 5.7. F1 score

Novamente o *Random Forest* (0.91) e o *Extremely Randomized Trees* (0.90) provaram sua eficácia, e SVM (0.45) como pior na avaliação de recall e precisão.

### 5.8. Especificidade

O *Random Forest* tem a maior especificidade prevendo um verdadeiro negativo para cada categoria disponível (0.98) e o menor SVM (0.84).

### 5.9. Métricas de performance

Com base na Tabela 2 podemos avaliar as métricas de performance dos modelos de aprendizado de máquina.

**Tabela 2. Métricas de performance dos modelos**

| Algoritmo                  | Tempo de treinamento | Uso de memória (treinamento) | Tempo de predição | Uso de memória (predição) |
|----------------------------|----------------------|------------------------------|-------------------|---------------------------|
| Gaussian Naive Bayes       | 209.44 ms            | 8.92 MiB                     | 48.67 ms          | 33.95 MiB                 |
| KNN                        | 335.34 ms            | 4.18 MiB                     | 9129.37 ms        | 25.39 MiB                 |
| SVM                        | 772.69 ms            | 1.45 MiB                     | 15.19 ms          | 9.41 MiB                  |
| Extremely Randomized Trees | 6180.53 ms           | 10.12 MiB                    | 2237.85 ms        | 18.13 MiB                 |
| Random Forest              | 43128.56 ms          | 11.84 MiB                    | 1475.68 ms        | 18.13 MiB                 |
| Xtreme Gradient Boosting   | 229203.56 ms         | 18.13 MiB                    | 793.14 ms         | 23.96 MiB                 |

Os algoritmos que não são baseados em árvore, foram os que apresentaram melhores métricas de performance, com destaque para o *Gaussian Naive Bayes* que obteve o

menor tempo de treinamento (209.44 ms), e o *SVM* que obteve o menor uso de memória no treino e na predição (1.45 MiB e 9.41 MiB), além de ser o mais rápido na predição (15.19 ms). Apesar da boa performance em tempo de execução e uso de memória, as métricas de desempenho foram ruins, com alta taxa de erro (41.69% e 46.75%) e baixa acurácia (58.31% e 53.25%).

## 6. Conclusão

Neste trabalho foi realizada a análise de diferentes algoritmos de aprendizagem de máquina para detectar ataques do tipo *spoofing* a sinais de GPS em VANTs.

Dentre os algoritmos implementados o que apresentou as melhores métricas de desempenho foi o *Random Forest*. Em relação as métricas de performance, o *Random Forest* obtém melhores resultados que o *XGBoost* nos aspectos de tempo de treinamento (43128.56 ms) e uso de memória (18.13 MiB), porém o *XGBoost* ainda é mais rápido na predição (793.14 ms).

Os demais algoritmos que não são baseados em árvore, *SVM* e *Gaussian Naive Bayes* apresentaram resultados ruins, com altas taxas de erro (46.75% e 41.69% respectivamente), e baixa acurácia (53.25% e 58.31% respectivamente). O *K-NN* também não se destacou, com taxa de erro de 21.04% e acurácia de 78.96%. Apesar de sua acurácia estar próxima ao *XGBoost*, as probabilidades de detecção falsa e alarme falso foram altas (26.68% e 5.37% respectivamente). Esses resultados indicam que esses algoritmos não são adequados para a detecção de ataques de *spoofing* em VANTs.

Ao longo deste artigo o objetivo era buscar melhorar a eficiência do *Random Forest* comparado com o *XGBoost*, que ponderado com as métricas de performance foi o que obteve melhor eficiência no artigo de [Aissou et al. 2021]. Após os experimentos o objetivo foi alcançado com sucesso, através das técnicas de pré-processamento implementadas.

Apesar das métricas serem positivas, seria necessário melhorar as métricas de performance para que o algoritmo fosse viável para implementação em VANTs.

Em comparação ao artigo de [Aissou et al. 2021], o *Random Forest* obteve uma acurácia menor (93.51% contra 94.07%), o que indica que ainda há espaço para melhorar. Já o *XGBoost* obteve uma acurácia significativamente menor (79.29% contra 95.52%), indicando que as técnicas de pré-processamento utilizadas não foram eficazes para esse algoritmo. Portanto, é necessário explorar outras técnicas de pré-processamento ou ajustar os hiperparâmetros do modelo para melhorar seu desempenho. As probabilidades de detecção falsa e alarme falso do *Random Forest* foram de 9.24% e 1.63% respectivamente, enquanto no trabalho de [Aissou et al. 2021] foram de 3.77% e 8.53%. Isso indica que o *Random Forest* ainda pode ser melhorado para reduzir esses valores, aumentando a confiabilidade do modelo. Há a possibilidade que a escolha do *dataset* tenha influenciado negativamente nos resultados do *XGBoost*.

O algoritmo *Extremely Randomized Trees* não provou ser mais eficiente que o *Random Forest*, mas suas métricas de performance são ligeiramente melhores, sendo mais rápido no treinamento (6180.53 ms) e consumindo (10.12 MiB) de memória no treinamento e (18.13 MiB) na predição. O único ponto negativo é que é mais lento na predição (2237.85 ms). Portanto, seria uma alternativa viável ao *Random Forest*, pois no que diz respeito as métricas de desempenho os resultados são similares ao *Random Forest*.

## 7. Trabalhos Futuros

Em trabalhos futuros seria interessante investigar o uso de outros algoritmos de aprendizado de máquina para detecção de ataques de *spoofing* de GPS, como por exemplo, redes neurais convolucionais (CNN) e inteligência artificial generativa, que tem se mostrado eficazes em outras aplicações. [Sun et al. 2023]; [El alami e Rawat 2024]

Realizar o *cross-validation* seria interessante, pois poderia fornecer uma avaliação mais robusta do desempenho dos modelos, evitando o overfitting e garantindo que os resultados obtidos sejam generalizáveis para novos dados.

Por fim, buscar a melhora do *Extremely Randomized Trees* pode ser um caminho viável e econômico em custos operacionais do que o *XGBoost* e o *Random Forest*.

## Referências

- A. Faria1, Lester de et al. (2018). “GPS Jamming Signals Propagation in Free-Space, Urban and Suburban Environments”. Em: *Journal Aerospace Technology Management*. DOI: 10.5028/jatm.v10.870.
- Aissou, Ghilas (2022). *A DATASET for GPS Spoofing Detection on Unmanned Aerial System*. Mendeley Data, V3. DOI: 10.17632/z7dj3yyzt8.3.
- Aissou, Ghilas et al. (2021). “Tree-based Supervised Machine Learning Models For Detecting GPS Spoofing Attacks on UAS”. Em: *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0649–0653. DOI: 10.1109/UEMCON53757.2021.9666744.
- Brasil, Dialogos Uniao Europeia (2025). *Estudo sobre a Indústria Brasileira e Europeia de Veículos aéreos não tripulados*. Rel. técn. Dialogos Uniao Europeia Brasil. DOI: [https://www.gov.br/mdic/pt-br/images/publicacaoa\\_DRONES-20161130-20012017-web.pdf](https://www.gov.br/mdic/pt-br/images/publicacaoa_DRONES-20161130-20012017-web.pdf).
- El alami, Hassan e Danda B. Rawat (2024). “DroneDefGANt: A Generative AI-Based Approach for Detecting UAS Attacks and Faults”. Em: *ICC 2024 - IEEE International Conference on Communications*, pp. 1933–1938. DOI: 10.1109/ICC51166.2024.10622524. (Acesso em 23/07/2025).
- Geurts, Pierre, Damien Ernst e Louis Wehenkel (2006). “Extremely Randomized Trees”. Em: *Machine Learning* 63.1, pp. 3–42. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/s10994-006-6226-1. URL: <http://link.springer.com/10.1007/s10994-006-6226-1> (acesso em 06/08/2025).
- Haghighi, Sepand et al. (2018). “PyCM: Multiclass confusion matrix library in Python”. Em: *Journal of Open Source Software* 3.25, p. 729. DOI: 10.21105/joss.00729. URL: <https://doi.org/10.21105/joss.00729>.
- İşleyen, Emre e Şerif Bahtiyar (2024). “GPS Spoofing Detection on Autonomous Vehicles with XGBoost”. Em: *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pp. 500–505. DOI: 10.1109/UBMK63289.2024.10773593. (Acesso em 23/07/2025).
- Jafarnia-Jahromi, Ali, Ali Broumandan e Gérard Nielsen J.and Lachapelle (2012). “GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques”. Em: *International Journal of Navigation and Observation*. DOI: 10.1155/2012/127072.
- Japkowicz, Nathalie (2000). “Learning from Imbalanced Data Sets”. Em: *Proceedings of the AAAI Workshop on Learning from Imbalanced Data Sets*. AAAI Press.



- Jurman, Giuseppe, Samantha Riccadonna e Cesare Furlanello (2012). “A Comparison of MCC and CEN Error Measures in Multi-Class Prediction”. Em: *PLOS ONE* 7.8, pp. 1–8. DOI: 10.1371/journal.pone.0041882. URL: <https://doi.org/10.1371/journal.pone.0041882>.
- Panice, G. et al. (2017). “A SVM-based detection approach for GPS spoofing attacks to UAV”. Em: *2017 23rd International Conference on Automation and Computing (ICAC)*, pp. 1–11. DOI: 10.23919/IconAC.2017.8081999.
- Pearson, Karl (1894). “Contributions to the Mathematical Theory of Evolution”. Em: *Philosophical Transactions of the Royal Society of London. A* 185, pp. 71–110. DOI: 10.1098/rsta.1894.0003.
- (1901). “On Lines and Planes of Closest Fit to Systems of Points in Space”. Em: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572. DOI: 10.1080/14786440109462720.
- Srinivasan S, Prasanna e Shiju Sathyadevan (2023). “GPS Spoofing Detection in UAV Using Motion Processing Unit”. Em: *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–4. DOI: 10.1109/ISDFS58141.2023.10131729. (Acesso em 23/07/2025).
- Sun, Yichen et al. (2023). “A Deep-Learning-Based GPS Signal Spoofing Detection Method for Small UAVs”. Em: *Drones* 7.6. ISSN: 2504-446X. DOI: 10.3390/drones7060370. URL: <https://www.mdpi.com/2504-446X/7/6/370>.
- Titouna, Chafiq e Farid Naït-Abdesselam (2021). “A Lightweight Security Technique For Unmanned Aerial Vehicles Against GPS Spoofing Attack”. Em: *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 819–824. DOI: 10.1109/IWCMC51323.2021.9498734. (Acesso em 23/07/2025).