



ACADEMIA MILITAR

Impacto do *Machine Learning* nos Sistemas de Aquisição de Alvos da Artilharia Antiaérea

**Aspirante de Artilharia Gregory António Portillo Dourado de
Carvalho**

Trabalho de Investigação Aplicada

Ciências Militares na Especialidade de Artilharia

Orientador: Tenente – Coronel de Artilharia José Miguel Sequeira Maldonado

Júri

Presidente: Professora Auxiliar Convidada Sandra Luzia Esteves Oliveira de Almeida

Arguente: Tenente – Coronel de Artilharia Nuno Calhaço

Orientador: Tenente – Coronel de Artilharia José Miguel Sequeira Maldonado

Diretor de Curso: Tenente – Coronel de Artilharia Bruno Filipe Simões Ladeiro

Junho 2025



ACADEMIA MILITAR

Impacto do *Machine Learning* nos Sistemas de Aquisição de Alvos da Artilharia Antiaérea

**Aspirante de Artilharia Gregory António Portillo Dourado de
Carvalho**

Trabalho de Investigação Aplicada

Ciências Militares na Especialidade de Artilharia

Orientador: Tenente – Coronel de Artilharia José Miguel Sequeira Maldonado

Júri

Presidente: Professora Auxiliar Convidada Sandra Luzia Esteves Oliveira de Almeida

Arguente: Tenente – Coronel de Artilharia Nuno Calhaço

Orientador: Tenente – Coronel de Artilharia José Miguel Sequeira Maldonado

Diretor de Curso: Tenente – Coronel de Artilharia Bruno Filipe Simões Ladeiro

Junho 2025

RESUMO

A presente dissertação tem como objetivo analisar o impacto da aplicação do *Machine Learning* nos Sistemas de Aquisição de Alvos da Artilharia Antiaérea. Numa época marcada pela transformação digital e pelo surgimento de novas ameaças aéreas, como os Sistemas Aéreos Não Tripulados e mísseis hipersónicos, o uso de técnicas de ML representa uma oportunidade estratégica para aumentar a eficiência, precisão e tempo de resposta dos sistemas de defesa.

O principal objetivo consistiu em avaliar o impacto da aprendizagem automática nos sistemas referidos, sendo definidos como objetivos específicos: identificar sistemas que já integram esta tecnologia, analisar as suas potencialidades e limitações, e aferir a sua aplicabilidade no contexto nacional. Para responder a estes objetivos, foi adotada uma metodologia de base mista, conjugando com uma análise comparativa entre algoritmos com e sem *Machine Learning*, desenvolvidos no *software MatLab and Simulink* e a realização de uma entrevista semiestruturada a um especialista do sector da empresa *Thales Group*. Os dados recolhidos foram tratados por via estatística.

Os resultados obtidos evidenciam melhorias significativas ao nível da precisão na classificação de alvos, da previsão de trajetórias e da redução de falsos positivos. Contudo, foram também identificadas limitações, como a dependência de dados de alta qualidade e os requisitos computacionais elevados.

Conclui-se que a aprendizagem automática representa uma ferramenta promissora para o reforço das capacidades da Artilharia Antiaérea, sendo a sua integração dependente da adequação tecnológica e da formação especializada dos operadores.

Palavras-chave: *Machine Learning*; Artilharia Antiaérea; Sistemas de Aquisição de Alvos; Defesa Aérea; Inteligência Artificial.

ABSTRACT

This dissertation aims to analyze the impact of applying Machine Learning to Anti-Aircraft Artillery Target Acquisition Systems. In an era marked by digital transformation and the emergence of new aerial threats, such as Unmanned Aerial Systems and hypersonic missiles, the use of ML techniques represents a strategic opportunity to increase the efficiency, precision and response time of defense systems.

The main objective was to assess the impact of machine learning on these systems, with the following specific objectives: to identify systems that already integrate this technology, to analyze its potential and limitations, and to assess its applicability in the national context. To meet these objectives, a mixed methodology was adopted, combining a comparative analysis of algorithms with and without Machine Learning, developed in MatLab and Simulink software, and a semi-structured interview with a specialist in the sector from the Thales Group company. The data collected was processed statistically.

The results obtained show significant improvements in terms of target classification accuracy, trajectory prediction and the reduction of false positives. However, limitations were also identified, such as the dependence on high-quality data and the high computational requirements.

The conclusion is that machine learning is a promising tool for strengthening the capabilities of anti-aircraft artillery, although its integration depends on technological suitability and the specialized training of operators.

Keywords: Machine Learning; Anti-Aircraft Artillery; Target Acquisition Systems; Air Defense; Artificial Intelligence.

ÍNDICE GERAL

INTRODUÇÃO	1
CAPÍTULO 1 - ENQUADRAMENTO TEÓRICO	4
1.1. <i>O Machine Learning</i>	4
1.1.1. Dissimulação do <i>Machine Learning</i> no quotidiano	5
1.2. Técnicas de <i>Machine Learning</i> e os seus Dados	7
1.2.1. Tipos de dados	8
1.2.2. Tipos de técnicas.....	9
1.2.3. Tarefas e algoritmos de aprendizagem automática.....	10
1.2.4. Síntese.....	17
1.3. <i>O Machine Learning</i> e os Sistemas de Aquisição de Alvos	19
CAPÍTULO 2 – EFEITOS DO MACHINE LEARNING.....	24
2.1. Potencialidades	24
2.1.1 Melhoria da deteção e classificação de alvos	24
2.1.2. Redução de falsos positivos.....	25
2.1.3. Integração com sistemas de Comando.....	25
2.1.4. Manutenção preditiva	25
2.2. Limitações.....	25
2.2.1. Dependência de dados fidedignos.....	25
2.2.2. Complexidade computacional.....	26
2.3. Síntese.....	27
CAPÍTULO 3 – APLICAÇÃO DO MACHINE LEARNING	29
3.1. Síntese.....	30
CAPÍTULO 4 – METODOLOGIA.....	32
CAPÍTULO 5 – MÉTODOS E MATERIAIS.....	37
5.1. Algoritmo Base	38
5.2. Algoritmo sem <i>Machine Learning</i>	40

5.3. Algoritmo com <i>Machine Learning</i>	42
5.4. Entrevista	46
CAPÍTULO 6 – RESULTADOS	48
6.1. Resultados do Algoritmo sem <i>Machine Learning</i>	48
6.1.1. Primeira execução	48
6.1.2. Execução intermédia	50
6.1.3. Última execução	52
6.1.4. Síntese	53
6.2. Resultados do Algoritmo com ML	54
6.2.1. Primeira execução	55
6.2.2. Execução intermédia	57
6.2.3. Última execução	59
6.2.4. Síntese	60
6.3. Resultados da Entrevista	60
6.3.1. Impacto do <i>Machine Learning</i> nos Sistemas de Aquisição de Alvos da Artilharia Antiaérea (Pergunta de Partida)	61
6.3.2. Potencialidades e Limitações do <i>Machine Learning</i> na Artilharia Antiaérea (Pergunta Derivada 1 – Objetivo Específico 2)	61
6.3.3. Aplicabilidade do <i>Machine Learning</i> na Artilharia Antiaérea do Exército Português (Pergunta Derivada 2 – Objetivo Específico 1 e 2)	62
6.3.4. Importância da Implementação de Sistemas Aquisição de Alvos com <i>Machine Learning</i> (Pergunta Derivada 3 – Objetivo Específico 3)	63
6.3.5. Síntese	64
CAPÍTULO 7 – DISCUSSÃO DOS RESULTADOS	66
7.1. Impacto do <i>Machine Learning</i> nos Resultados Obtidos	66
7.2. Análise Estatística Comparativa	66
7.3. Análise da Entrevista	67
7.4. Síntese	68
CONCLUSÕES	70
REFERÊNCIAS BIBLIOGRÁFICAS	73

ANEXOS E APÊNDICES.....	I
--------------------------------	----------

ÍNDICE DE FIGURAS

Figura nº 1 – Tipologia do <i>Machine Learning</i>	10
Figura nº 2 – Fluxo lógico de um modelo <i>Machine Learning</i>	12
Figura nº 3 – Exemplo de uma Árvore de Decisão (DT)	13
Figura nº 4 – Análise de classificação vs análise de regressão	15
Figura nº 5 – Diferença entre o ML e DL	17
Figura nº 6 – Componentes e sistemas da AAA	IV
Figura nº 7 - Output 'MetricsLog.txt'	48
Figura nº 8 - Matriz de confusão da primeira execução	49
Figura nº 9 - Previsão vs Real (Trajetória) da primeira execução.....	50
Figura nº 10 - Output 'MetricsLog.txt'	50
Figura nº 11 - Matriz de confusão da execução intermédia	51
Figura nº 12 - Previsão vs Real (Trajetória) da execução intermédia	52
Figura nº 13 - Output 'MetricsLog.txt'	52
Figura nº 14 - Matriz de confusão da última execução	53
Figura nº 15 - Previsão vs Real (Trajetória) da última execução	53
Figura nº 16 – LSTM de criação do modelo ML	54
Figura nº 17 – LSTM de treino de um modelo ML existente	55
Figura nº 18 – Output 'MetricsLog.txt'	55
Figura nº 19 - Matriz de confusão da primeira execução	56
Figura nº 20 - Previsão vs Real (Trajetória) da primeira execução	57
Figura nº 21 - Output 'MetricsLog.txt'	57
Figura nº 22 - Matriz de confusão da execução intermédia	58
Figura nº 23 - Previsão vs Real (Trajetória) da execução intermédia	58
Figura nº 24 – Output 'MetricsLog.txt'	59
Figura nº 25 – Matriz de confusão da última execução	59
Figura nº 26 – Previsão vs Real (Trajetória) da última execução	60
Figura nº 27 – Análise descritiva	I
Figura nº 28 – Resultados dos pressupostos.....	I
Figura nº 29 – Resultados do teste para amostras independentes	I

Figura nº 30 – Regressão linear do erro em distância em função dos RMSE	I
Figura nº 31 - Densidade e Boxplots do RMSE em x	I
Figura nº 32 – Média e mediana do RMSE em x	I
Figura nº 33 - Densidade e Boxplots do RMSE em y	II
Figura nº 34 – Média e mediana do RMSE em y	II
Figura nº 35 - Densidade e Boxplots do RMSE em z	III
Figura nº 36 – Média e mediana do RMSE em z	III
Figura nº 37 - Densidade e Boxplots dos erros em distância	IV
Figura nº 38 – Média e mediana dos erros em distância	IV
Figura nº 39 – Densidade e Boxplots da Identificação da ameaça.....	V
Figura nº 40 – Média e mediana da Identificação de ameaça	V
Figura nº 41 – RMSE dos valores em x	I
Figura nº 42 - RMSE dos valores em y	I
Figura nº 43 – RMSE dos valores em z.....	I
Figura nº 44 – Termo de Consentimento Informado.....	I

ÍNDICE DE TABELAS

Tabela nº 1 – Algoritmos ML e aplicações	18
Tabela nº 2 – Matriz análise SWOT cruzada sobre o impacto do ML.....	22
Tabela nº 3 – Comparação entre Apêndice B e Anexo A	41
Tabela nº 4 – Técnicas de ML – Vantagens vs Desvantagens	43
Tabela nº 5 – Comparação entre Apêndice B e C	45
Tabela nº 6 – Potencialidades e Limitações	61
Tabela nº 7 – Resumo enquadramento da entrevista e objectivos	I
Tabela nº 8 – Questões entrevista.....	I

LISTA DE FÓRMULAS

Regressão linear:

$$y = b_0 + b_1x$$

1

Regressão linear múltipla:

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

2

Equações movimento:

$$x(t + 1) = x(t) + v(x) * \Delta t$$

3

'Drone'

$$\begin{aligned} \text{Velocidade} &\equiv (\text{speedFactor}) \\ &= 0.8 + \text{randn}() * 0.1; \end{aligned}$$

randn() → número aleatório

$$x(t + 1) = 100 + 50 * \cos(\text{speedFactor} * t);$$

$$y(t + 1) = 200 + 50 * \sin(\text{speedFactor} * t);$$

$$z(t + 1) = 50 + \text{randn}() * 5;$$

'Aircraft'

$$x(t + 1) = 50 + 0.2 * t;$$

$$y(t + 1) = 100 + 0.3 * t;$$

$$z(t + 1) = 100 + 0.05 * t + \text{randn}() * 2;$$

'Helicopter'

$$\begin{aligned} x(t + 1) &= 150 + 30 * \sin(0.5 * t) \\ &+ \text{randn}() * 0.1; \end{aligned}$$

$$\begin{aligned} y(t + 1) &= 150 + 30 * \cos(0.3 * t) \\ &+ \text{randn}() * 0.1; \end{aligned}$$

$$\begin{aligned} z(t + 1) &= 70 + 10 * \sin(0.7 * t) + \text{randn}() \\ &* 0.2; \end{aligned}$$

Precisão (idAccuracy):

idAccuracy

$$= \frac{\text{sum}(\text{diag}(\text{confusionMatrix}))}{\text{sum}(\text{confusionMatrix}(:))} * 100$$

4

Precisão trajetória:

Diferença (atual – predita)

$$= x, y, z (\text{actual})$$

$$- x, y, z (\text{previsão}) \equiv$$

$$\equiv \text{diffTrajectory}$$

$$= \text{actualTrajectory}$$

$$- \text{predictedTrajectory};$$

accuracyPercentage

$$\begin{aligned} &= (1 / (1 \\ &+ \text{diffTrajectory})) \\ &* 100; \end{aligned}$$

5

Erro distância:

$$\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2]}:$$

trajectoryError

$$= \text{sqrt}(\text{sum}(\text{diffTrajectory}.^2, 2));$$

6

MSE:

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - x'_i)^2$$

7

RMSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x'_i)^2}$$

8

LISTA DE ANEXOS E APÊNDICES

Anexo A – Algoritmo Base (<i>MathWorks</i>).....	I
Apêndice A – Sistemas de Aquisição de Alvos	I
Apêndice B – Algoritmo sem <i>Machine Learning</i>	I
Apêndice C – Algoritmo com <i>Machine Learning</i>	I
Apêndice D – Resultados Análise Descritiva.....	I
Apêndice E – Resultados do teste Shapiro-Wilk.....	I
Apêndice F – Resultados do teste U de Mann-Whitney.....	I
Apêndice G – Resultados Regressão Linear.....	I
Apêndice H – Gráficos Representativos.....	I
Apêndice I – Gráficos de Dispersão	I
Apêndice J – Guião de Entrevista.....	I
Apêndice K – Termo de Consentimento Informado.....	I

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

4IR – Quarta Revolução Industrial

AAA – Artilharia Antiaérea

ADL – Análise Discriminante Linear

ATR – *Automatic Target Recognition*

C2 – Comando e Controlo

C4ISR – *Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance*

CNN – *Convolutional Neural Network*

DL – *Deep-Learning*

DT – *Decision Tree*

EUA – Estados Unidos da América

FLIR – *Forward Looking Infra-Red*

HI – Hipóteses de Investigação

IA – Inteligência Artificial

ISAR – *Inverse Synthetic Aperture Radar*

ISR – Israel

KNN – *K-nearest neighbors*

LSTM – *Long-Short Term Memory*

ML – *Machine Learning*

MLP – *Multilayer Perceptron*

MSE – *Mean Squared Error*

NB – Naive Bayes

NATO – *North Atlantic Treaty Organization*

OE – Objectivo Específico

OG – Objectivo Geral

PD – Pergunta derivada

PP – Pergunta de Partida

RCS – *Radar Cross Section*

RF – *Random Forest*

RMSE – *Root Mean Squared Error*

RNA – Rede Neural Artificial

RUS – Federação Russa

SAA – Sistema(s) de Aquisição de Alvo(s)

SANT – Sistemas Aéreos Não Tripulados

SIGINT - *Signal Intelligence*

SVM – *Support Vector Machine*

SWE – Suécia

SWOT – *Strengths, Weaknesses, Opportunities and Threats*

UAS – *Unmanned Aerial Systems*

INTRODUÇÃO

Estudos realizados até à data apontam que de certa forma, o *Machine Learning* (ML) enquanto componente da Inteligência Artificial (IA), exerce um impacto considerável nos Sistemas de Aquisição de Alvos (AAA) para a Artilharia Antiaérea (AAA). O tópico do ML nos SAA dos meios da AAA tem vindo a ser discutido e estudado, por vários autores, seguindo o desenvolvimento tecnológico dos meios.

Verly et al., (1989) aborda o desenvolvimento de um sistema ATR¹ que faz uso do ML, resulta num sucesso na deteção e reconhecimento de objetivos específicos. No entanto, como mencionado por Ploumis (2022) o emprego do ML para adquirir conhecimentos táticos em combates aéreos, implica uma criação de normas de aplicação operacionais.

A aplicação de sistemas que fazem uso de ML é demonstrada pelo aumento considerável de precisão em comparação aos meios tradicionais (Kim et al., 2018) e, coadunando com o estudo de Kim et al. (2018) que discute Helland et al. (1981), a evolução de técnicas de compreensão de imagens para aquisição autónoma de alvos, incorpora para isso a análise de movimentos e perseguição inteligente de múltiplos alvos.

O debate do ML nos SAA tem vindo a acompanhar a evolução tecnológica, dos meios e dos sistemas e de acordo com Abeynayake et al. (2019) onde, concentrou-se no desenvolvimento de um algoritmo de reconhecimento automático de alvos baseado no ML, visando a redução de falsos alarmes e aprimorar taxas de deteção com sucesso.

Posto isto, é proposto por Kondaveeti et al. (2024) um sistema ATR para o reconhecimento automático de alvos de aeronaves através do processamento de imagens, designadamente o sistema ISAR². Já Ehrman (2020) explorou a incorporação

¹ O ATR é uma tecnologia que recorre a algoritmos e sensores para identificar alvos sem intervenção humana. Essencialmente através da comparação de imagens capturadas em tempo real com base nas imagens armazenadas para identificar e classificar os alvos.

² ISAR é uma técnica radar que cria imagens de alvos em movimento, onde a captura é sintetizada pelo movimento do próprio alvo, em vez de ser criada pelo deslocamento do radar. Ou seja, em vez de o radar se mover para recolher dados, o movimento do alvo em si é usado para criar a imagem.

da orientação da aeronave no reconhecimento automático de alvos por meio de radar passivo.

Em conjunto, os estudos consultados convergem para a mesma linha de pensamento, sendo esta, a evidência do potencial das técnicas de ML para aprimorar os SAA aéreos, desde a detecção e reconhecimento de ameaças aéreas até a classificação automática de alvos, incluindo a integração de informações de orientação para um reconhecimento aprimorado do alvo.

O século XXI trouxe às operações militares um novo paradigma. O ambiente operacional alterou-se significativamente e as operações militares passaram a desenvolver-se essencialmente com o apoio das novas tecnologias. Com esta revolução tecnológica a nível militar, mais propriamente para a AAA, onde as capacidades destes sistemas vão constantemente compensar as evoluções das aeronaves, novas tipologias de mísseis e novos tipos de ameaças aéreas, tais como, os Sistemas Aéreos Não Tripulados (SANT) que “incluem drones, caracterizados por utilizarem rotas de voo pré-programadas, e veículos pilotados remotamente (*Remotely Piloted Vehicles* – RPV), controlados por operadores/pilotos em terra” EME (2016, p. 48).

Diante este contexto, a presente Dissertação que concorre para a obtenção do grau de Mestre em Ciências Militares na especialidade de Artilharia, tem como PP “Qual o impacto do uso do ML nos SAA da AAA na realidade do Exército Português?” e, de forma a ir ao encontro do OG “Compreender o impacto do ML nos SAA na AAA, no contexto do Exército Português”.

Para tal, foram definidos três OE:

OE1: Identificar os sistemas que recorrem ao ML e analisar suas características;

OE2: Avaliar as potencialidades e limitações da utilização de ML em nos SAA;

OE3: Aferir a importância de empregar SAA na AAA que façam uso do ML, no caso do Exército Português.

A dissertação encontra-se estruturada em cinco capítulos, sendo que no Capítulo 1, são apresentados os principais conceitos relacionados ao ML e aos SAA, contextualizando a investigação. O Capítulo 2 analisa as potencialidades e limitações do ML nos SAA, identificando desafios técnicos e operacionais. No Capítulo 3, explora-se a aplicabilidade do ML na AAA do Exército Português, abordando possíveis implementações e benefícios. O Capítulo 4 detalha a metodologia empregue

na investigação, detalhando técnicas de recolha e análise de dados. O Capítulo 5 expõe os métodos usados para ir ao encontro do OG, realçando a análise experimental e estatística entre algoritmos com e sem ML. O Capítulo 6 apresenta os resultados obtidos, verificando a pertinência do ML na AAA. Já no Capítulo 7 encontra-se a discussão dos resultados dos capítulos anteriores, respondendo às Hipóteses de Investigação (HI) levantadas de forma a concorrer para uma resposta à Pergunta de Partida (PP), culminando nas conclusões e recomendações para futuras investigações.

A modernização dos sistemas de defesa aérea e em particular a defesa AA é uma necessidade operacional para as Forças Armadas, e o ML representa uma tecnologia emergente com potencial para transformar a aquisição de alvos, tornando-a mais precisa, eficiente e adaptável às modernas ameaças. Assim, esta dissertação pretende contribuir para a compreensão desse impacto, fornecendo fundamentação teórica e análise experimental sobre a utilização do ML na AAA do Exército Português.

CAPÍTULO 1 - ENQUADRAMENTO TEÓRICO

O presente capítulo tem a finalidade de apresentar e clarificar um conjunto de conceitos que se consideram pertinentes para um melhor entendimento da problemática em questão.

1.1. O *Machine Learning*

O ML tem-se tornado cada vez mais importante nos sistemas de defesa, uma vez que oferece capacidades avançadas de deteção de ameaças, tomada de decisões e operações autónomas (Alcántara Suárez & Monzon Baeza, 2023). Nas diversas aplicações do ML, numa simples nota introdutória, é sabido que na cibersegurança incluem a deteção de *malware*³ permitindo uma prevenção de ameaças e o aumento do risco das redes (Rege et al., 2018). Neste sentido, este tipo de tecnologia pode melhorar significativamente a deteção e análise de *malware*, abordando o desafio crescente de ataques cibernéticos sofisticados (Fraley & Cannady, 2017).

Os métodos de ML podem ser utilizados para melhorar a discriminação de alvos em radares através da análise da sua assinatura radar, ou seja, *Radar Cross Section* (RCS), permitindo diferenciar diversas tipologias de aeronaves (Tunçer, 2019). Além disso, o ML também possibilita a redução do tempo de decisão, algo crítico na defesa AA, onde frações de segundo podem determinar a eficácia de uma resposta a uma ameaça (Rege et al., 2018).

Contudo, o uso do ML nos sistemas de defesa não está isento de desafios. Fraley e Cannady (2017) sugerem algumas preocupações que incluem questões éticas e de segurança, como a falta de transparência dos modelos e o risco de ataques cibernéticos contra sistemas baseados em IA. Além disso, o ML pode ser suscetível a ataques, como tentativas de *jamming*⁴ contra radares, alterando os padrões esperados e confundindo

³ O termo *malware* descreve qualquer algoritmo ou aplicações maliciosas que causam danos ou perturbam a utilização normal de dispositivos eletrónicos.

⁴ *Jamming* ou *Jamming Attack*, é uma técnica utilizada para interromper a comunicação entre os vários dispositivos conectados na mesma rede. Este tipo de ataque é comum em redes sem fio, onde o emissor do ataque emite sinais que interferem na transmissão de dados, tornando a comunicação impossível. O objetivo principal é causar a degradação ou a interrupção do serviço, afetando a disponibilidade da rede.

a detecção automática de ameaças (Alcántara Suárez & Monzon Baeza, 2023) e, para fazer face a estes desafios, recomenda-se uma maior transparência, responsabilização e envolvimento das partes interessadas na conceção e implementação de sistemas de aprendizagem automática (Alcántara Suárez & Monzon Baeza, 2023).

Ao longo deste capítulo serão abordados em detalhe as bases do ML e as suas aplicações no quotidiano de forma a delimitar a criação de ideias deste novo tipo de tecnologia.

1.1.1. Dissimulação do *Machine Learning* no quotidiano

O ML, a aprendizagem da máquina, é uma subárea da IA que utiliza algoritmos para processar grandes volumes de dados, identificar padrões e realizar previsões ou tomar decisões de forma autónoma, tendo por base uma aprendizagem estrutural. Conforme destacado por Agrawal et al. (2018) em *“Prediction Machines: The Simple Economics of Artificial Intelligence”*, o ML pode ser entendido como uma máquina de previsão, na qual os algoritmos são utilizados para estimar resultados futuros, substituindo ou complementando a intuição e o julgamento humano.

Esta capacidade de previsão fundamenta o valor do ML em diferentes áreas de aplicação. Assim sendo, a nível económico e organizacional de acordo com Davenport e Mittal (2023) em *“All In on AI”* e com Davenport et al. (2023) no seu artigo na *Harvard Business Review*, o ML tem sido um elemento essencial para a transformação de processos empresariais. As organizações estão a utilizar o ML para automatizar tarefas repetitivas, melhorar a eficiência operacional e oferecer soluções personalizadas em larga escala. Tal ocorre, especialmente em sectores que lidam com grandes volumes de dados, como a indústria militar, onde a análise rápida e precisa de informações pode representar uma vantagem estratégica.

Coadunando com o exposto previamente, o ML integra também a indústria de defesa, ou seja, desempenha um papel cada vez mais crucial no aprimoramento de capacidades tecnológicas e estratégicas.

De acordo com estudos recentes, o ML é aplicado em diversas frentes, incluindo a detecção e perseguição de alvos, dado que os algoritmos de ML são usados para processar dados de sensores em tempo real, aumentando a precisão na identificação e classificação de alvos em cenários dinâmicos. Incluindo aplicações em sistemas de defesa AA, tal como: i) em sistemas radar e em sistemas míssil; ii) análise

preditiva – o ML é empregue para prever potenciais ameaças com base em padrões históricos e dados em tempo real, permitindo uma resposta proactiva e eficaz; iii) automatização de veículos – tecnologias baseadas no ML permitem que os SANT tomem decisões de navegação, vigilância e combate com maior autonomia; iv) manutenção preditiva – sistemas que possuem como modelo de manutenção baseado no ML, estão capacitados a monitorizar o desempenho do próprio sistema e prever falhas, reduzindo custos e melhorando a disponibilidade operacional; v) cibersegurança – o ML é aplicado para detetar e mitigar ataques cibernéticos em infraestruturas críticas, de forma a proteger redes militares contra ameaças emergentes.

Estas aplicações destacam o potencial transformador do ML na defesa, aumentando a precisão, a eficiência e a capacidade de resposta em cenários críticos.

Já na esfera social, autores como Gunkel (2020), Guzman e Lewis (2020) discutem como o ML está a redefinir as dinâmicas de comunicação pessoas e os sistemas, criando novos formatos de interação protagonizada por algoritmos. Além disso, estudos como o de García-Orosa et al. (2023) destacam o papel do ML na produção, distribuição e consumo de informações, influenciando profundamente as práticas de comunicação à escala global.

No livro *“The Age of Surveillance Capitalism”*, Zuboff (2019) argumenta que o ML tem um papel central no capitalismo, onde os dados são recolhidos em larga escala e utilizados para prever e influenciar comportamentos humanos. De forma similar, Couldry e Mejias (2020) apontam como o ML contribui para a "colonização da vida humana", apropriando-se de dados para fins comerciais e de controlo social. Esposito (2022), por sua vez, analisa o impacto do ML na produção de "inteligência social artificial", questionando os limites entre as tomadas de decisão humanas e automáticas.

Em *“Homo Deus: A Brief History of Tomorrow”*, Harari (2017), apresenta o ML como um dos pilares que moldará o futuro da humanidade, redefinindo conceitos de trabalho, criatividade e autonomia. Hermann (2022), por sua vez, discute os desafios éticos relacionados à personalização de conteúdos por meio de algoritmos de ML, com especial atenção aos impactos na comunicação digital.

Para além das aplicações diretas já mencionadas, no que concerne à investigação, o ML é de certa forma impactante, uma vez que, Xu et al. (2021) demonstram como o ML se tornou num paradigma poderoso para a pesquisa científica,

permitindo a análise e interpretação de grandes volumes de dados. Essa abordagem expandiu significativamente o alcance do conhecimento científico, abrindo novas possibilidades para a resolução de problemas complexos em diversas áreas, incluindo na defesa.

Face ao exposto, para que haja uma integração dos sistemas baseados em ML, nas suas diversas frentes, Han et al. (2000), num artigo publicado na *Harvard Data Science Review*, propõem um quadro de cinco princípios para o uso responsável do ML. Esses princípios incluem transparência, justiça, beneficência, responsabilidade e sustentabilidade, enfatizando a necessidade de um desenvolvimento que considere tanto o impacto social quanto ético das tecnologias baseadas em ML.

Posto isto, é possível compreender que o ML não é apenas uma ferramenta tecnológica, mas também um agente transformador que altera a forma como as informações são processadas, as decisões são tomadas e os sistemas se adaptam ao ambiente. A sua aplicação oferece uma oportunidade sem precedentes para inovação e, ao mesmo tempo apresenta desafios éticos, sociais e técnicos.

Com base nos autores citados, é possível concluir que o ML é um fator-chave para avanços nas áreas organizacional, comunicação e defesa, exigindo uma abordagem consequente e responsável para maximizar seus benefícios enquanto se mitigam seus riscos.

1.2. Técnicas de *Machine Learning* e os seus Dados

De acordo com Sarker (2019), no domínio da Quarta Revolução Industrial (4IR), onde prevalecem grandes quantidades de dados diversos, a aplicação de algoritmos de aprendizagem automática (ML) é crucial para a análise inteligente de dados e o desenvolvimento de sistemas autônomos.

Para tal, é necessário entender o cerne do desenvolvimento do ML nas suas aplicações práticas como na cibersegurança, cuidados de saúde, comércio eletrônico, agricultura e outros domínios (Cao, 2018).

O ML surge como uma tecnologia crucial na 4IR, permitindo que os sistemas aprendam e melhorem de uma forma autónoma sem uma programação explícita (Sarker, 2019; Sarker et al., 2021). É possível então deduzir que: i) o ML, como um componente essencial da 4IR, facilita a análise inteligente de dados e o desenvolvimento de aplicações; ii) os principais tipos de algoritmos de ML incluem a

aprendizagem supervisionada, não supervisionada, semi-supervisionada e *reinforcement*, registando uma tendência de popularidade crescente ao longo dos anos (Han et al., 2011; Sarker & Kayes, 2020); iii) e que os modelos de ML dividem-se em quatro grupos macro [classificação; regressão; agrupamento e Deep-Learning (DL)], consoante o objetivo final e resultados específicos.

1.2.1. Tipos de dados

Os dados são cruciais para a construção de modelos de aprendizagem automática e de sistemas orientados para os dados do quotidiano.

Com a análise do tipo de dados de ML, segundo Sarker (2021), apresentam-se sob várias formas: estruturados, semiestruturados, não estruturados e *metadata*⁵ que fornecem informações sobre os próprios dados. Sendo o ML uma subárea da IA, funciona de uma forma lógica e encadeada e, para tal, necessita de informação para encorpar o seu raciocínio lógico (McCallum, 2005).

Face às informações supracitadas, temos que a informação necessitada do ML, ou seja, os seus dados podem ser:

- **Estruturados** – seguem uma estrutura bem definida em conformidade com um modelo de dados numa ordem padrão, são altamente organizados e normalmente de fácil acesso e são armazenados num formato tabular, tal como numa base de dados. Podendo então, serem dados como o registo da deteção radar, dados meteorológicos e registos de trajetórias de alvos aéreos;
- **Semiestruturados** – os dados não são armazenados tal igual a uma base de dados, no entanto, possuem uma certa organização estrutural de armazenamento que facilita também o seu acesso e análise computacional. São dados do tipo de comunicação entre sistemas, como dados relativos a sensores comunicados para uma rede interna de partilha de informação;
- **Não estruturados** – como o nome indica, não possuem qualquer forma organizada de armazenamento, tornando assim o processo de recolha e análise de dados um processo mais complexo e moroso. São dados de difícil

⁵ Os metadados (*metadata*) são dados que fornecem informações sobre outros dados. A título de exemplo de metadados temos: localização física, números de identificação, etiquetas de documentais e sistemas de classificação numa combinação de algarismos.

categorização, ou seja, dados multimédia como as imagens Eletro-óptico/Infravermelho (EO/IR), vídeos de reconhecimento aéreo e assinaturas acústicas. Devido ao facto de este tipo de modelo de dados não ser organizado implica que seja necessário o uso de modelos ML avançados para compreender e analisar tais dados;

- **Metadata** – representam as informações referentes aos dados, ou seja, não representam qualquer conteúdo palpável sendo somente os “dados sobre os dados”.

Na aprendizagem automática, os investigadores utilizam vários conjuntos de dados para diferentes fins, escolhendo frequentemente conjuntos de dados amplamente utilizados.

Os tipos de dados podem variar entre aplicações no mundo real, o que torna essencial compreender e categorizar os dados de forma eficaz.

1.2.2. Tipos de técnicas

Face ao exposto anteriormente, segundo o estudo de Sarker (2021) e complementando com Han et al. (2011), é necessário compreender como é que as vastas e complexas categorizações dos dados se processam. Daí, os algoritmos de aprendizagem automática são classificados em quatro tipos principais: aprendizagem supervisionada, aprendizagem não supervisionada, aprendizagem semi-supervisionada e aprendizagem *reinforcement* (Naeem et al., 2020). O mesmo é possível observar pela Figura nº 1.

Adequando os estudos dos autores supracitados, a **aprendizagem supervisionada** consiste em treinar um modelo para mapear entradas e saídas com base em *labeled data* (dados rotulados), ou seja, dados que possuem categorias ou valores previamente definidos. Isso permite que o modelo aprenda padrões e realize previsões com base em exemplos já classificados. Esta abordagem é amplamente utilizada em tarefas como a classificação textual para análise e categorização de conteúdos e até mesmo na identificação de padrões em produtos ou serviços. Por sua vez, uma aprendizagem díspar da anterior, a **aprendizagem não supervisionada** funciona com conjuntos de dados não rotulados, tornando-a num processo orientado para os dados sem intervenção humana. Já a **aprendizagem semi-supervisionada** utiliza uma mistura de dados rotulados e não rotulados, ou seja, este tipo de

aprendizagem preenche a lacuna entre as abordagens totalmente supervisionadas e não supervisionadas. O seu objetivo é melhorar os resultados da previsão para além do que os dados rotulados podem alcançar por si só e sendo uma técnica de aprendizagem que mitiga as falhas, entre o limbo da aprendizagem supervisionada e não supervisionada, as suas áreas de aplicação são mais subjectivos, tal como tradução automática, deteção de fraudes e até classificação de textos.

Por último, a **aprendizagem *reinforcement***, para Kaelbling et al. (1996) e Naeem et al. (2020), permite que os *softwares* melhorem o seu desempenho de forma autónoma em ambientes específicos. Esta abordagem centra-se na otimização do desempenho para obter os melhores resultados possíveis num determinado contexto.

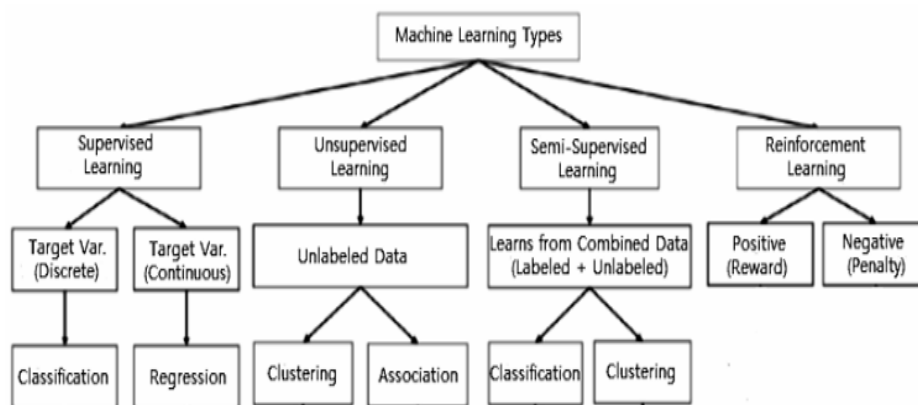


Figura nº 1 – Tipologia do *Machine Learning*

Fonte: Sarker (2021)

Diferentes técnicas de aprendizagem automática desempenham papéis vitais com base na natureza dos dados e nos resultados pretendidos. Estas técnicas possuem a capacidade de melhorar a inteligência e as capacidades das aplicações baseadas em dados. No entanto, consoante a finalidade do modelo, cada técnica possui as suas especificidades devendo a técnica ser escolhida com base na finalidade do modelo, é assim possível observar pela Figura nº 1 que mediante a técnica a utilizar de acordo com os dados adquiridos, há modelos específicos de ML a seguir. O mesmo é exposto adiante.

1.2.3. Tarefas e algoritmos de aprendizagem automática

Esta secção aborda uma variedade de algoritmos de aprendizagem automática que abrangem diferentes tarefas, como a análise de classificação, a análise de regressão, o agrupamento de dados e a aprendizagem pelas regras de associação (Sarker et al., 2021). Não são abordados todos os modelos, uma vez que nem todos se alinham com o propósito da presente dissertação. No entanto, incluem-se também métodos de DL para análise inteligente de dados e como o próprio nome indica, o DL representa uma metodologia mais avançada do ML, pois utiliza camadas profundas de redes neurais para aprender automaticamente padrões complexos a partir dos dados (Han et al., 2011).

De acordo com a conformidade de autores, temos como pontos principais o seguinte:

- **Análise de classificação** – que categoriza os dados em classes ou grupos tendo por base em características específicas bem definidas (Sarker et al., 2018);
- **Análise de regressão** – que é utilizada para prever valores numéricos contínuos com base em variáveis de entrada (Han et al., 2011);
- **Agrupamento de dados** (*clustering*) – ou seja, uma tarefa que envolve o agrupamento de pontos de dados semelhantes (Sarker et al., 2021);
- **Aprendizagem pelas regras de associação**, centrada na descoberta de relações homogéneas em grandes conjuntos de dados (Sarker & Kayes, 2020);
- **DL**, faz parte de um espectro mais vasto de aprendizagem automática, concebido para processar eficazmente conjuntos de dados em grande escala (Han et al., 2011).

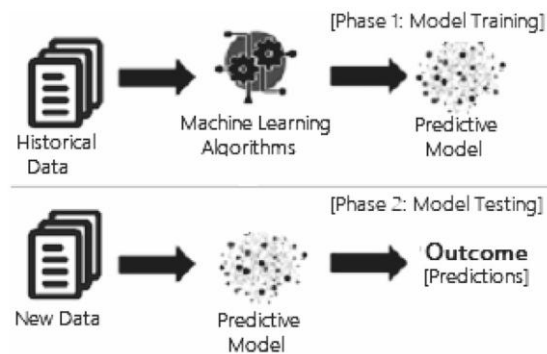


Figura nº 2 – Fluxo lógico de um modelo *Machine Learning*

Fonte: Sarker & Kayes (2020)

Pela Figura nº 2 é possível visualizar o fluxo de trabalho geral de um modelo de previsão baseado no ML, em que o modelo é treinado através de dados históricos numa primeira fase e, em seguida, gera resultados para novos dados de teste na fase subsequente.

Numa abordagem detalhada das menções acima e pertinentes, com base na literatura especializada, a **análise por classificação** na aprendizagem automática envolve a previsão de um rótulo para uma classe de acordo com um determinado exemplo utilizando métodos de aprendizagem supervisionada, como mencionado anteriormente. Visa então mapear as variáveis de entrada para as variáveis de saída, representando rótulos ou categorias do destino.

Desta forma, no espectro da análise de classificação temos a mais comumente usada a classificação binária (Han et al., 2011). Ou seja, para tarefas com dois rótulos de classe como “verdadeiro e falso” ou “sim e não”.

Alguns algoritmos de classificação mais comuns incluem:

- Naive Bayes (NB) – baseado no teorema de Bayes, que é um teorema de probabilidades que caracteriza a possibilidade de um acontecimento *a posteriori* acontecer, conhecendo uma causa *a priori* (Avelar et al., 2023; Sarker, 2019), assume-se assim a independência entre as características (Sarker et al., 2018). Funciona bem para categorias binárias e classes múltiplas em vários cenários do quotidiano, como a classificação de documentos e a filtragem de spam;

- Análise Discriminante Linear (ADL) – cria um classificador de limite de decisão linear ajustando as densidades das classes e aplica também como recurso o teorema de NB, matematicamente, que está relacionada com a análise de regressão (Han et al., 2000; Pedregosa et al., 2012);
- *K-nearest neighbors* (KNN) – um algoritmo de aprendizagem baseado em instâncias que classifica novos pontos de dados com base em medidas de semelhança, tal como descrevem Sarker e Kayes (2020), cria-se assim uma relação entre a vizinhança de dados;
- *Support Vector Machine* (SVM) – constrói patamares categóricos para tarefas de classificação e tem como objetivo uma forte separação entre classes (Sarker et al., 2021);
- *Decision Trees* (DT) – um método de aprendizagem supervisionado não paramétrico eficaz na análise dos comportamentos. É ainda explicado por Breiman (2001) e Han et al. (2011) que as instâncias das DT têm por base os valores atribuídos, sendo assim criadas várias categorias de decisão, Figura nº 3.

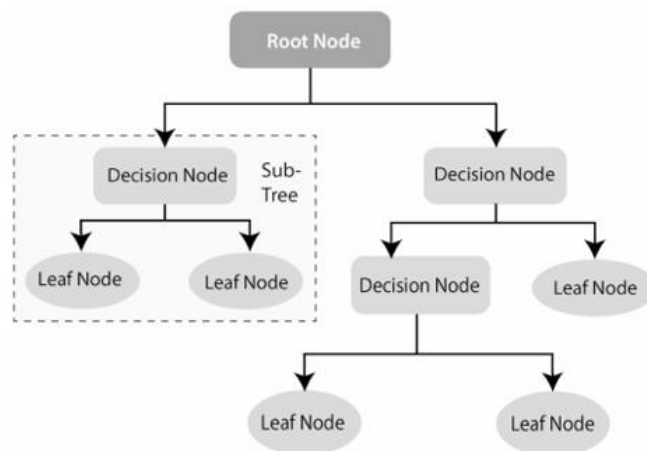


Figura nº 3 – Exemplo de uma Árvore de Decisão (DT)

Fonte: Breiman (2001)

- *Random Forest* (RF) – uma técnica de conjunto que ajusta várias árvores de decisão em paralelo para reduzir o sobreajuste⁶ e aumentar a precisão da previsão. Para isto, utiliza algumas técnicas tais como, o *AdaBoost* e o *Extreme Gradient Boosting*, não sendo pertinente para a presente dissertação, no entanto, é de realçar que são variações que aumentam o desempenho da DT com estratégias diferentes (Sarker & Kayes, 2020).

Em resumo, nesta secção são abordados vários algoritmos de classificação amplamente utilizados em diferentes domínios de aplicação, destacando os seus princípios e aplicações em cenários reais.

Contrapondo a análise por classificação, a **análise de regressão** no ML envolve métodos para prever um resultado contínuo (y) com base numa ou mais variáveis de previsão (x). É assim possível distinguir que a principal diferença entre regressão e classificação é que esta prevê rótulos de classe, enquanto a regressão prevê quantidades contínuas (Han et al., 2011). Tal como a classificação, na regressão existem várias formas que se cingem neste modelo. Temos assim:

- Regressão linear – que estabelece uma relação (linha de regressão) entre a variável dependente (y) e as variáveis independentes (x) através de uma linha reta de melhor ajuste, sendo obtida através da equação nº 1, onde y é a saída, b_0, b_1 são os coeficientes de regressão, e x é a variável de entrada (Han et al., 2000);
- Regressão linear múltipla – que amplia a regressão linear simples ao incorporar duas ou mais variáveis de previsão para modelar a variável de resposta, descrito na equação nº 2;
- Regressão polinomial – adequada quando a relação entre x e y não é linear, mas segue um polinómio de grau n . Sendo derivada da equação de regressão linear, oferece flexibilidade com distribuições de dados não lineares (Han et al., 2011) e, os mesmos autores afirmam que para reduzir o sobreajuste⁶ e a

⁶ Sobreajuste (*overfitting*) e subajuste (*underfitting*): *overfitting* ocorre quando um modelo, neste caso, modelo ML, aprende detalhes excessivos dos dados de treino, incluindo ruídos e variações aleatórias, o que resulta num bom desempenho nos dados de treino, mas pobre desempenho em dados novos (teste). Por outro lado, *underfitting* acontece quando um modelo é muito simples e não consegue capturar as relações importantes nos dados, resultando em baixo desempenho tanto nos dados de treino quanto nos dados de teste.

complexidade do modelo face a numerosas características temos a LASSO e Regressão Ridge.

Em suma, a análise de regressão fornece uma base para a previsão de resultados contínuos e é fundamental em várias aplicações do mundo real em domínios como as finanças, os cuidados de saúde e a análise preditiva. É assim exposta, na Figura nº 4, a diferença entre estas duas últimas técnicas de análise.

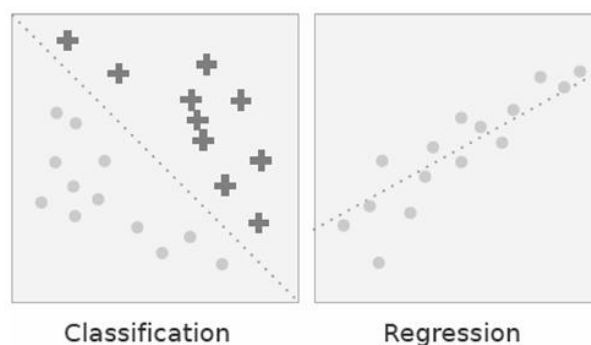


Figura nº 4 – Análise de classificação vs análise de regressão

Fonte: Sarker (2021)

Já a análise de **agrupamento de dados**, ou *clustering*, como MacQueen (1967) expõe, é uma técnica de aprendizagem automática não supervisionada utilizada para agrupar pontos de dados relacionados em grandes conjuntos de dados sem resultados específicos em mente. Os métodos de agrupamento mais populares incluem:

- Agrupamento *K-means*, que é um algoritmo rápido, robusto e simples para agrupar dados bem separados, onde os dados são atribuídos a agrupamentos para minimizar a distância ao valor central (MacQueen, 1967). No entanto o algoritmo é sensível a *outliers*, ou seja, a valores com um grande viés em relação ao desvio padrão; o agrupamento por deslocação da média, que é uma técnica não paramétrica que não necessita de conhecimento prévio dos números ou formas dos agregados. Esta técnica agrupa os dados em secções de informação coincidente numa densidade de amostra simples (Fukunaga & Hostetler, 1975);
- Agrupamento hierárquico aglomerativo, sendo um método comum que agrupa objectos com base na semelhança de uma forma ascendente, criando assim por sua vez uma hierarquia em forma de árvore, oferece estruturas

mais informativas em comparação com os agrupamentos do *K-means* (Sneath, 1957).

Estes algoritmos de agrupamento oferecem diversas abordagens para agrupar dados e têm aplicações em vários domínios, ajudando a tomar melhores decisões com base nas estruturas e semelhanças dos dados.

O DL, um subconjunto das Redes Neurais Artificiais (RNA) (Xu et al., 2021), envolve a aprendizagem de representação através de várias camadas de processamento para análise de dados. Os pontos principais incluem:

- *Multilayer Perceptron* (MLP) – que é a arquitetura base do DL, que cria as RNA *feed-forward*, ou seja, a informação somente vai numa direção de forma a facilitar a resolução de problemas (Alves et al., 2021; Pedregosa et al., 2012). Inclui camadas de entrada, camadas ocultas e camadas de saída, com nós ligados a pesos específicos;
- *Convolutional Neural Network* (CNN) – que melhora a concepção tradicional das RNA com camadas convolucionais, onde matematicamente, uma função não existe sem uma anterior e, onde esta também não funciona sem uma *a priori*, ou seja, a informação encontra-se totalmente ligada (Matlab for Engineers, 2017c; Upreti, 2022). Para isso utiliza a estrutura 2D dos dados de entrada, amplamente aplicada no reconhecimento e processamento de imagens (Lecun et al., 1998). Esta rede é então capaz de detetar automaticamente características cruciais, tornando-a mais potente do que a RNA, apesar da uma maior complexidade computacional;
- *Long-Short Term Memory* (LSTM) – com uma arquitetura de rede neural recorrente no âmbito do DL, adequada para dados sequenciais como a análise de séries temporais, o reconhecimento de voz e o reconhecimento de padrões (Xin et al., 2018).

Para uma aprendizagem mais completa que o ML, é necessário conjugar várias técnicas e modelos de aprendizagem, para que o ponto vital da aprendizagem automática seja a sua eficiência e eficácia, para isso é necessária uma aprendizagem contínua com recurso a diversos tipos de dados.

O exposto previamente, coaduna técnicas e modelos de ML, o que por sua vez, para que seja obtido cada vez mais a eficiência de aprendizagem é necessário conjugar

todos eles. Para isso, de forma a demonstrar visualmente temos a Figura nº 5 que expõem a diferença entre o DL e o ML.

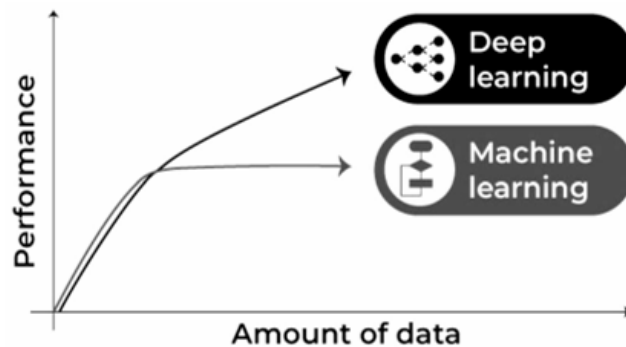


Figura nº 5 – Diferença entre o ML e DL

Fonte: Xin et al. (2018)

1.2.4. Síntese

Os algoritmos de ML apresentam uma ampla versatilidade em diversas tarefas, sendo aplicáveis em domínios que vão desde a classificação e regressão até ao *clustering* e DL. A classificação permite categorizar a informação em rótulos ou grupos distintos, enquanto a regressão é utilizada para prever valores contínuos baseando-se nas variáveis de entrada. O *clustering*, por sua vez, é empregue para identificar padrões e segmentos em conjuntos de dados sem supervisão prévia, e o DL destaca-se pela sua capacidade de processar grandes volumes de informação através de redes neurais profundas, permitindo a extração automática de características complexas.

A análise prática destaca a utilização de diversos algoritmos dentro dessas categorias. Métodos como NB, KNN, DT e SVM são amplamente utilizados em cibersegurança, análise de comportamento do humano e classificação de documentos. Já os modelos de regressão, incluindo Regressão Linear e Polinomial, encontram aplicação em análises financeiras, manutenção preditiva e previsão de necessidades. No contexto de agrupamento, técnicas como *K-Means* e o agrupamento hierárquico são utilizadas para análise de imagens e identificação de anomalias em grandes conjuntos de informações – *big data*.

Com o avanço da capacidade computacional, o DL tornou-se numa área de destaque, elevando aplicações que exigem a extração autónoma de características complexas. Modelos como CNN são fundamentais para reconhecimento facial e análise de imagens, enquanto RNA e LSTM são amplamente utilizadas na análise de séries temporais e previsão de trajetórias.

Embora os algoritmos de ML ofereçam soluções robustas para desafios computacionais, enfrentam obstáculos como a necessidade de grandes volumes de dados, elevada necessidade de processamento e riscos de sobreajuste⁶, especialmente no DL (Matlab for Engineers, 2017b, 2017c). No entanto, sua aplicabilidade continua a expandir-se, tornando-se numa ferramenta essencial em diversos domínios científicos e industriais.

A seguir, apresenta-se um quadro síntese, exposta pela Tabela nº 1 os principais algoritmos de ML e suas aplicações:

Tabela nº 1 – Algoritmos ML e aplicações

Tipo	Descrição	Algoritmos principais	Aplicabilidade
Análise de classificação	Classifica dados em rótulos ou categorias específicas	NB SVM KNN DT	Diagnósticos médicos Filtragem de spam Detecção de fraudes Cibersegurança
Análise de regressão	Prediz valores contínuos com base em variáveis independentes	Regressão Linear Regressão Polinomial	Predição de vendas Análise financeira Manutenção preditiva
Clustering	Organiza dados em grupos com base em semelhanças	K-means Agrupamento hierárquico	Segmentação do mercado Agrupamento de clientes <i>big-data analysis</i>

DL	Subcampo do ML para lidar com grandes volumes de dados e problemas complexos	RNA CNN LSTM MLP	Reconhecimento facial
			Análise de séries temporais
			Cibersegurança
			Previsão de trajetórias

1.3. O *Machine Learning* e os Sistemas de Aquisição de Alvos

O avanço do ML tem proporcionado transformações significativas nos SAA da AAA, particularmente nos radares de vigilância, nos radares de perseguição e conduta de tiro e nos sistemas *Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance* (C4ISR). A aplicação do ML nesses sistemas visa aumentar a precisão, reduzir falsos positivos, otimizar a classificação de ameaças e permitir tomadas de decisão mais rápidas e eficazes em ambientes de combate dinâmicos e complexos (Rudin, 2019; Yang et al., 2020).

Os radares desempenham um papel essencial na aquisição de alvos, uma vez que pesquisam e detetam objectos aéreos hostis, ou não, em tempo real.

Com a integração do ML, estes sistemas podem melhorar substancialmente a sua eficiência em diversas áreas, tais como:

- **Redução de falsos positivos** – aves, SANT comerciais e outros objectos possam gerar falsos alertas em sistemas convencionais, ou seja, os modelos de ML podem ser treinados para reconhecer padrões de voo e RCS específicas, permitindo a distinção entre potenciais ameaças, ou não (Yang et al., 2020);
- **Previsão de trajetória** – algoritmos de ML utilizam dados históricos e dados em tempo real para prever os movimentos futuros das aeronaves e SANT, proporcionando uma melhor capacidade de resposta dos sistemas de defesa (Rudin, 2019);
- **Classificação de ameaças** – O ML pode diferenciar rapidamente entre aeronaves amigas, inimigas ou desconhecidas, passando pelas RCS já

conhecidas e informações adicionais para melhorar a identificação de ameaças (Courtney, 2021);

- **Adaptação a condições ambientais** – Os radares podem ser afetados por interferências climáticas, como chuva e tempestades de areia. No entanto, os modelos de ML ajustam automaticamente os parâmetros do radar para compensar variações nas condições operacionais (Hebert, 2020).

Coadunando com o exposto pelo Apêndice A e pela investigação sobre o ML é possível tecer algumas conclusões, ou seja, cingindo-nos sobre os radares de perseguição e conduta de tiro, o ML desempenha um papel ainda mais crítico ao melhorar a precisão do perseguição dos alvos e a orientação de sistemas de armas. É especialmente relevante para mísseis guiados, onde a precisão da informação fornecida pelo radar influencia diretamente a eficácia da interceção (Rudin, 2019).

Em outro espectro dos SAA, temos ainda os sensores e, como mencionado estes desempenham um papel complementar dos radares. Os sensores EO/IR suplementam os radares ao fornecer imagens nas gamas do espectro visual e térmicas para a identificação de alvos.

A aplicação do ML nestes sensores possibilita o reconhecimento de imagem uma vez que podem analisar imagens térmicas e ópticas em tempo real para detetar características específicas dos alvos, como formato, assinatura térmica e padrões de movimento (Yang et al., 2020). A classificação automática de alvos nos sistemas baseados em ML pode variar entre diferentes tipos de alvos, como SANT, aeronaves de asa fixa e rotativa¹⁰ (ou helicópteros), reduzindo a necessidade de intervenção humana (Courtney, 2021). Através da compensação de variáveis ambientais o ML melhora a eficácia dos sensores ao ajustar automaticamente a deteção conforme mudanças nas condições de iluminação, presença de nuvens e interferências atmosféricas (Courtney, 2021; Hebert, 2020).

Já numa óptica macro dos sistemas, a fusão de dados oriundos de múltiplas fontes [radares, sensores EO/IR e *Signal Intelligence* (SIGINT)] constituem um dos principais desafios dos SAA modernos. O ML aprimora esse processo ao correlacionar dados de diferentes sensores através de algoritmos avançados, dado que integram informações de radares, sensores ópticos e SIGINT para criar uma imagem operacional mais precisa do campo de batalha (Courtney, 2021). Desta forma reduzem

a sobrecarga de dados/informações (Rudin, 2019) filtrando e estabelecendo prioridades das informações de forma a permitir decisões mais céleres e eficazes.

Com esta ampla gama de capacidades de integração de modelos ML o processo de automatizar o Comando e Controlo (C2) coadjuva na tomada de decisão, recomendando estratégias de defesa e alocação de recursos em tempo real (Thales, 2021b, 2024a).

Além de melhorar a eficiência operacional, o ML também pode ser aplicado na manutenção preditiva dos próprios SAA. A monitorização contínua do espaço batalha através de radares e sensores permite prever falhas antes que ocorram, reduzindo o tempo de inatividade e garantindo a disponibilidade operacional (Yang et al., 2020). Outra preocupação emergente é a segurança dos próprios sistemas baseados em ML. Como essas tecnologias dependem de grandes volumes de dados, podem ser alvo de ataques cibernéticos, como manipulação de conjuntos de treino e criação de falsos positivos. Dessa forma, são necessárias estratégias robustas de segurança para proteger os algoritmos contra ameaças externas (Rudin, 2019).

Em síntese, o impacto do ML nos SAA da AAA é significativo, proporcionando maior precisão, eficiência e adaptabilidade em ambientes operacionais complexos. A integração do ML nos radares, sensores EO/IR e sistemas C4ISR melhora a deteção, classificação e o *tracking* de ameaças, enquanto reduz a sobrecarga de dados e automatiza processos de decisão. No futuro, espera-se que essa tecnologia desempenhe um papel ainda mais central na defesa AA, especialmente diante do surgimento de novas ameaças.

Com isto, é então possível categorizar o impacto do ML nos SAA supracitados através de uma matriz SWOT cruzada, referenciada como Tabela nº 2.

A relação entre Forças e Oportunidades (SO) demonstra que o ML tem potencial exponencial para os SAA ao automatizar operações, integrar sensores múltiplos e prever ameaças, permitindo que os sistemas AAA respondam de maneira mais rápida e precisa.

Na relação entre Forças e Ameaças (ST) temos que a aplicação do ML pode mitigar desafios como as interferências eletrónicas, ataques cibernéticos e deteção de aeronaves *stealth*, aumentando a resiliência dos sistemas.

Numa visão relacionada com as fraquezas (W) temos a relação entre Fraquezas e Oportunidades (WO), onde as limitações do ML, como dependência de dados e alta

complexidade computacional, podem ser minimizadas através de bases de dados simuladas, otimização de modelos com tomadas de decisão mais transparentes.

Por fim, temos que a relação entre Fraquezas e Ameaças (WT), através da combinação da redundância operacional, adaptação contínua e mitigação de custos podem reduzir riscos e garantir que os sistemas de ML na AAA sejam eficazes mesmo em cenários adversos.

Esta análise mostra que a integração do ML nos SAA da AAA tem grande potencial para aprimorar a detecção e resposta a ameaças, mas também apresenta desafios operacionais e de implementação. Para que o ML seja eficaz, é fundamental superar a dependência de dados, otimizar modelos para tempo real e garantir segurança contra possíveis ataques cibernéticos. A adoção de estratégias SO e ST permite explorar os benefícios do ML, enquanto WO e WT ajudam a mitigar riscos e adaptar os sistemas de defesa aérea às ameaças emergentes.

Tabela nº 2 – Matriz análise SWOT cruzada sobre o impacto do ML

	Oportunidades (O): Cenário externo favorável para o ML nos SAA da AAA	Ameaças (T): Fatores externos que dificultam a implementação do ML nos SAA da AAA
Forças (S): Vantagens internas do ML aplicadas aos SAA da AAA	SO – Estratégias ofensivas Como potencializar as forças internas para aproveitar as oportunidades externas? Automatização dos sistemas de defesa AA: O ML pode reduzir a necessidade da componente humana na aquisição de alvos, agilizando a resposta e otimiza o uso de recursos Fusão avançada de dados: O ML permite integrar informações de radares, sensores EO/IR e SIGINT para fornecer uma visão operacional do campo de batalha mais precisa	ST – Estratégias de defesa Como utilizar as forças para mitigar ameaças externas? Proteção contra guerra eletrônica: O ML pode ser utilizado para identificar padrões de interferência inimiga e ajustar dinamicamente os algoritmos de rastreamento Resiliência contra-ataques cibernéticos: O ML pode fortalecer SAA ao detectar anomalias

	Desenvolvimento de armamento: Sistemas de defesa antiaérea podem ser aprimorados com ML para aumentar a sua precisão	Adaptação a alvos <i>stealth</i>: Técnicas avançadas de ML podem aprimorar a detecção de aeronaves <i>stealth</i> que minimizam a RCS
Fraquezas (W): Limitações internas do ML aplicadas aos SAA da AAA	WO – Estratégias de reestruturação Como minimizar as fraquezas para aproveitar as oportunidades? Criação de bases de dados simuladas: Para compensar a escassez de dados de combate real, pode utilizar-se técnicas de geração sintética de dados e aprendizagem por <i>reinforcement</i> em treino operacional Otimização de modelos para operação em tempo real	WT – Estratégias defensivas Como reduzir as fraquezas para mitigar ameaças? Adopção da redundância operacional: Complementar o ML com abordagens tradicionais, como algoritmos baseados em regras, para mitigar falhas Treino contínuo e adaptação a novas ameaças: Implementação de ciclos de re-treino para lidar com adversários que explorem vulnerabilidades do ML Mitigação de custos através de parcerias: Cooperação entre governos, indústria militar e centros de pesquisa para desenvolver soluções eficientes e sustentáveis

CAPÍTULO 2 – EFEITOS DO *MACHINE* *LEARNING*

Neste capítulo procura-se responder à PD1 “Quais as potencialidades/limitações do ML nos SAA da AAA?” Para tal fim, procura-se determinar o que implica a integração de algoritmos de ML nos SAA da AAA.

2.1. Potencialidades

2.1.1 Melhoria da deteção e classificação de alvos

A utilização de algoritmos de ML, incluindo o DL, permite uma classificação mais precisa de alvos nos sistemas radar. Por exemplo, a *MathWorks* demonstra um fluxo de trabalho para a classificação de alvos, para tal, utiliza o ML e o DL, onde é destacada a eficácia destas abordagens na distinção de diferentes tipos de alvos (Opti-Num Solutions, 2022; The MathWorks, 2024). O ML vem habilitar a diferenciação entre vários corpos que viajam nos céus, entre SANT, mísseis, aeronaves e aves, reduzindo potenciais falsos alarmes, ou seja, os falsos positivos, ou até fazer uma associação de determinadas características de um possível alvo e, utilizá-las num futuro para melhor classificação e identificação do alvo.

Já na vertente do movimento propriamente dito, temos que é possível prever trajetórias com base nas equações do movimento de cada elemento aéreo, ou seja, os algoritmos de ML podem prever as trajetórias futuras de um alvo, concorrendo para uma reação mais célere da tomada de decisão. Coadunando com Molnar (2019), os modelos ML aprendem pelos dados históricos e generalizam este conhecimento de forma a conseguir fazer previsões para novos dados e, até aos dias de hoje as redes neurais (CNN, RNA) e o DL são comumente utilizados para resolver tarefas complexas como a previsão de trajetórias.

Algoritmos de redes neurais recorrentes, como LSTM têm sido aplicados na previsão de trajetórias de SANT em ambientes urbanos. Um estudo publicado no *Journal of Communications and Information Networks* propõe um modelo de previsão de trajetória em tempo real para pequenos quadricópteros (SANT), demonstrando a eficácia dessas redes na previsão precisa de movimentos futuros de alvos aéreos (Yang et al., 2020).

2.1.2. Redução de falsos positivos

A aplicação de técnicas de DL em sistemas de radar tem mostrado potencialidade na redução de falsos positivos. Uma revisão publicada na *Multidisciplinary Digital Publishing Institute (MDPI) Remote Sensing* discute os avanços no uso de DL para a extração de características e reconhecimento de alvos nos sistemas radar, indicando melhorias na precisão e redução de detecções incorretas (Jiang et al., 2023).

2.1.3. Integração com sistemas de Comando

Os sistemas de C4ISR são beneficiados pela implementação do ML, uma vez que, com a integração de informações provenientes de sensores, combinando dados de radares, sensores EO/IR e outras fontes, possibilita uma imagem operacional mais precisa do campo de batalha. Segundo a NATO Science and Technology Organization (2022b; Thales, 2024b), o ML otimiza a integração de dados no C4ISR, permitindo uma melhor tomada de decisão.

2.1.4. Manutenção preditiva

O ML também pode ser aplicado na manutenção preditiva de sistemas de defesa AA. Através da análise de padrões de funcionamento de radares e sensores, os algoritmos identificam possíveis falhas antes que as mesmas ocorram, aumentando a disponibilidade operacional e reduzindo potenciais custos a juzante com manutenção corretiva (Meng et al., 2020; Thales, 2021b).

2.2. Limitações

2.2.1. Dependência de dados fidedignos

O desempenho dos modelos de ML dependem significativamente da qualidade e quantidade dos dados utilizados para o treino dos modelos. Dados insuficientes ou de baixa qualidade podem resultar em modelos imprecisos, comprometendo a eficácia dos SAA. A MATLAB (2020) e The MathWorks (2024) destacam a importância de dados de alta qualidade no treino dos modelos de classificação de alvos pelo radar. Já Molnar (2019) expõe que para treinar modelos ML são necessárias grandes quantidades de dados, onde a qualidade e diversidade dos mesmos é fulcral. Tal pode

levar a modelos enviesados e incompletos, tornando assim o processo de treino de modelos ML ainda mais moroso e complexo.

Neste sentido, para Goodfellow et al. (2016), para treinar modelos eficazes, é necessário reunir grandes volumes de dados que representem diferentes cenários de combate, tipos de alvos (SANT, mísseis, aeronaves) e condições ambientais. Bases de dados insuficientes ou enviesadas podem levar a falhas na detecção ou na classificação de alvos. Conjugando com Meng et al. (2020) os sistemas de AAA modernos devem integrar dados recolhidos por radares e sensores EO/IR. Contudo, a obtenção de dados em cenários reais de combate ou em treinos simulados pode ser um desafio devido à sensibilidade das informações na esfera militar, no entanto, independentemente da sensibilidade das informações, os sistemas devem ser atualizados com dados novos para acompanhar os avanços tecnológicos das potenciais ameaças, tal como novas técnicas de camuflagem ou aeronaves *stealth* (NATO Science and Technology Organization, 2022a).

2.2.2. Complexidade computacional

Algoritmos avançados de ML, especialmente aqueles baseados em DL, podem exigir recursos computacionais significativos, o que pode ser uma limitação em sistemas que necessitam de processamento em tempo real. Molnar (2019) e The MathWorks (2024) abordam a necessidade de recursos computacionais adequados para implementar técnicas de DL em sistemas de radar e enquanto modelos avançados, oferecem um alto rendimento de *performance*, muitas vezes operam as denominadas “*black boxes*”⁷ o que torna o processo de interpretação mais complexo (MATLAB, 2020; Matlab for Engineers, 2017a). Ou seja, por muito avançados que os modelos sejam, muitas vezes falham na transparência da interpretação na exposição de resultados (Molnar, 2019).

Os especialistas que desenvolvem e operam os modelos precisam ter conhecimentos em ML, estatística, análise de dados, para além de compreender aspectos operacionais e técnicos de sistemas de defesa aérea (Choudhuri, 2024; Chouldechova & Roth, 2020). Destaca-se ainda que os especialistas são fundamentais

⁷ *Black box*: a definição de *black box* no contexto do ML refere-se a um modelo ou sistema que seja de difícil compreensão, mesmo para especialistas na área.

para evitar erros como o *sobreajuste*⁶, o *subajuste*⁶, ou dados enviesados, que podem impactar negativamente a eficácia operacional.

Apesar dos avanços dos modelos de DL, os mesmos são frequentemente criticados pela falta de interpretabilidade. Esta é uma verosímil questão na sua íntegra, uma vez que os especialistas devem ser capazes de explicar como os modelos tomam decisões críticas em situações de combate (Rudin, 2019).

2.3. Síntese

A aplicação do ML nos SAA da AAA representa um avanço tecnológico significativo, redefinindo a eficiência e a precisão desses sistemas. As potencialidades do ML, como a melhoria na detecção e classificação de alvos, a previsão de trajetórias e a redução de falsos positivos, mostram-se indispensáveis num cenário de crescente complexidade nas operações de defesa antiaérea. A capacidade de distinguir diferentes tipos de alvos – SANT, mísseis e aeronaves – com maior exatidão, combinada à previsão de movimentos futuros, permite reações mais rápidas e informadas, otimizando os recursos disponíveis e reduzindo a possibilidade de erros operacionais.

No entanto, as limitações associadas à implementação do ML, como a dependência de bases de dados extensas, a necessidade de recursos computacionais robustos e a exigência de especialistas capacitados, apresentam-se como desafios para a aplicação prática desta tecnologia. A qualidade e a representatividade dos dados utilizados para treinar os modelos continuam a ser fatores críticos, especialmente em contexto militar, onde a recolha de informações pode ser limitada por questões de segurança ou sensibilidade. Além disso, a complexidade dos algoritmos, frequentemente criticada pela falta de interpretação dos resultados, exige esforços adicionais para garantir transparência e confiabilidade nos processos da tomada de decisão.

Apesar das limitações, os avanços recentes demonstram que as soluções baseadas em ML estão a consolidar-se como ferramentas indispensáveis nos sistemas de AAA. Ao combinar a automatização de processos com análises em tempo real, essas tecnologias oferecem vantagens estratégicas, garantindo maior eficiência operacional e contribuindo para o fortalecimento das capacidades de defesa.

Portanto, a adoção de ML nos SAA não é apenas um reflexo do progresso tecnológico, mas também uma necessidade para enfrentar as necessidades de um

ambiente operacional dinâmico e em constante evolução. Para maximizar os benefícios, é essencial investir em pesquisa, desenvolvimento e treino, garantindo a criação de sistemas robustos, confiáveis e capazes de atender às exigências das forças de defesa modernas. Assim, o impacto do ML nestes sistemas deve ser continuamente avaliado, com vista a promover melhorias sustentáveis e acompanhar os avanços no campo da IA e da tecnologia militar.

CAPÍTULO 3 – APLICAÇÃO DO *MACHINE* *LEARNING*

A modernização dos sistemas de defesa aérea exige o aprimoramento contínuo da capacidade de detecção, classificação e neutralização de ameaças. O ML surge como um recurso fundamental para otimizar os SAA na AAA, permitindo a automatização e o aumento da eficiência operacional. A integração do ML pode contribuir significativamente para a melhoria da identificação de alvos, previsão de áreas de empenhamento e tomada de decisão automática no C2.

Este capítulo procura responder à PD2 “Qual a aplicabilidade do ML na AAA do Exército Português?” Para tal, perceber em que aspectos seria possível aplicar o ML na AAA.

No contexto da detecção e identificação de alvos, os algoritmos ML possibilitam a análise eficiente de grandes volumes de dados provenientes de sensores, aprimorando a distinção entre diferentes tipos de alvos, como aeronaves, SANT e mísseis e, reduzindo falsos positivos.

Modelos de DL, como CNN e RNA, têm demonstrado elevada eficácia na classificação de alvos, sendo amplamente utilizados em sistemas de defesa AA. Estudos indicam que as aplicações de redes neurais profundas melhoram a capacidade de distinção de assinaturas térmicas e espectrais de alvos detectados por sensores EO/IR garantindo maior precisão na identificação (Jiang et al., 2023). No estudo de Yang et al. (2020) onde prevê trajetórias de SANT, os modelos baseados nas LSTM são particularmente eficazes na previsão de trajetórias de alvos aéreos, permitindo aos sistemas de defesa AA antever movimentos e ajustar a sua resposta de forma otimizada. Esta capacidade reduz a probabilidade de falsos positivos e melhora a eficácia dos sistemas de defesa AA.

Outra aplicação essencial do ML na AAA reside na previsão das áreas de empenhamento de mísseis superfície-ar. Tradicionalmente, esse processo depende de simulações balísticas e avaliações táticas baseadas no estudo das informações – *intelligence* (Pey, 2022). No entanto, técnicas de ML permitem a automatização desse cálculo, integrando variáveis como condições meteorológicas, velocidade dos alvos e padrões de movimento. Pesquisas recentes demonstram que a utilização de redes neurais profundas para definir zonas de empenhamento possibilita a alocação eficiente

de recursos e a melhoria na resposta a ataques aéreos (Dantas et al., 2023; Feng et al., 2024).

Além da identificação de alvos e da previsão de zonas de empenhamento, o ML desempenha um papel crucial na automatização dos processos de tomada de decisão. Não obstante este âmbito de aplicação não se encontra abrangido pelo objecto do presente estudo.

Nos sistemas de C2, algoritmos de ML possibilitam a fusão de dados provenientes de múltiplos sensores, criando uma imagem operacional mais precisa e reduzindo o tempo necessário para respostas estratégicas. Entre as principais aplicações destaca-se a priorização automática de ameaças, onde sistemas C2 equipados com ML analisam assinaturas de radar, padrões de movimento e outros fatores críticos para classificar alvos e determinar quais devem ser neutralizados primeiro. (NATO Science and Technology Organization, 2022a, 2022b).

A Thales, uma das principais empresas no setor de defesa, tem aplicado técnicas de IA e ML em seus sistemas de radar para aprimorar a detecção, pesquisa e identificação de alvos aéreos (Thales, 2021a). Um exemplo notável é o Ground Master 400 Alpha (GM400α), que utiliza a IA para melhorar a análise SIGINT, reduzir falsos positivos e aumentar a precisão na identificação de ameaças aéreas, como *SANT* e aeronaves *stealth*. Além disso, o Ground Master 200 (GM200), um radar de médio alcance, incorpora algoritmos de ML para analisar padrões de voo em tempo real, prevendo possíveis trajetórias de aeronaves hostis. De forma a permitir uma resposta mais célere dos sistemas de defesa antiaérea, é referido pela Thales (2021b) que esta capacidade reduz o tempo necessário para a tomada de decisão em cenários de alta complexidade.

Outro avanço significativo proporcionado pela IA, nomeadamente o ML, nos SAA diz respeito à sua integração com os sistemas C4ISR. A fusão de dados entre radares, sensores EO/IR e outras fontes de informação permite gerar uma visão mais consolidada do campo de batalha, proporcionando aos operadores informações, em tempo real, para as tomadas de decisão informadas e estratégicas (Oswald, 2018).

A NATO tem investido consideravelmente no desenvolvimento de sistemas de C2 baseados em ML, visando aprimorar a eficiência operacional das suas defesas AA (NATO Science and Technology Organization, 2022b).

3.1. Síntese

Em síntese, a implementação do ML na AAA do Exército Português representa um avanço significativo no aprimoramento da defesa antiaérea, permitindo maior precisão na deteção e classificação de alvos, previsões mais eficazes das áreas de empenhamento e processos de tomada de decisão mais automatizados.

A integração de técnicas avançadas de ML nos SAA reduz a carga cognitiva dos operadores, otimiza a alocação de recursos e amplia a capacidade de resposta diante de ameaças emergentes. Todavia, a componente Humana e a sua relação com a IA/ML não deve ser menosprezada de forma a garantir que as questões éticas e legais sejam cumpridas, tal como, garantir a proporcionalidade da força na alocação de meios face a uma ameaça.

Para garantir a eficácia desses sistemas, é necessário um investimento contínuo em infraestrutura computacional e capacitação de pessoal especializado. Em resposta à PD2, portanto, é evidência que o ML não se apresenta apenas como uma tecnologia viável, mas sim como um recurso essencial para fortalecer a defesa antiaérea do Exército Português, garantindo uma capacidade operacional mais eficiente e adaptável ao cenário contemporâneo de ameaças aéreas.

CAPÍTULO 4 – METODOLOGIA

Seguidamente pretende-se apresentar a orientação metodológica utilizada nesta investigação.

De modo a atingir os objetivos supramencionados, levantaram-se três PD para auxiliar na resposta à PP.

Com a PD n.º 1: “Quais as potencialidades/limitações do ML nos SAA da AAA?” e a PD n.º 2: “Qual a aplicabilidade do ML na AAA do Exército Português?” procurou-se determinar quais são as potencialidades e limitações de recorrer ao ML e se o mesmo é possível introduzir nos SAA. A PD n.º 3: “Qual a importância de empregar SAA que fazem recurso do ML, no caso da AAA, do Exército Português?” procura compreender em síntese a verdadeira importância do emprego de SAA que recorrem ao ML na AAA. A resposta às três PD concorre para a resposta à PP anteriormente identificada, na tentativa de determinar qual o impacto do uso deste tipo de tecnologia nas Forças Armadas, mais propriamente, na AAA.

Com base nas PD e na literatura existente, foram formuladas as seguintes Hipóteses de Investigação (HI):

HI-1: “Algoritmos com base em modelos ML melhoram a precisão da identificação de alvos, reduzindo os falsos positivos e aumentando a percentagem de precisão na classificação de alvos”;

HI-2: “A implementação do ML nos SAA da AAA pode otimizar a tomada de decisão, proporcionando uma redução no tempo de resposta”;

HI-3: “A aplicação do ML enfrenta desafios operacionais significativos, incluindo um grande volume de informação e requisitos computacionais”

A validação destas HI é dada pelo meio de uma análise experimental entre algoritmos, bem como a avaliação qualitativa por meio de uma entrevista com um especialista.

A presente investigação adota uma abordagem mista (qualitativa e quantitativa), combinando análises estatísticas com uma entrevista semiestruturada. Esta opção dá-se pela necessidade de validar empiricamente os benefícios do ML nos SAA, enquanto simultaneamente permite compreender a viabilidade prática da sua implementação a partir da perspetiva de um especialista na área.

A componente quantitativa centra-se na construção e avaliação de um modelo de ML desenvolvido para otimizar a aquisição de alvos na AAA. Para esse fim, foram comparados dois algoritmos o Algoritmo sem ML – Apêndice B – sem a aplicação de ML, representando os métodos convencionais de aquisição de alvos e o Algoritmo com ML – Apêndice C – incorpora técnicas de ML para melhorar a precisão na identificação e classificação de alvos. A comparação entre os dois algoritmos foi feita através de métricas estatísticas, com destaque para o *Root Mean Squared Error* (RMSE) nos eixos x, y e z, a percentagem de acerto da identificação de alvos e o erro em distância. A análise estatística incluiu testes de normalidade (Shapiro-Wilk), testes de comparação de grupos (Mann-Whitney U) e regressão linear para avaliar a correlação entre os erros dos modelos.

A justificação para a escolha do RMSE como métrica principal fundamenta-se na sua ampla utilização em avaliações de desempenho de modelos preditivos, pois penaliza erros maiores de forma mais acentuada, garantindo-se maior precisão da análise (Chai & Draxler, 2014).

Para uma compreensão mais profunda do impacto do ML, foram elaboradas análises complementares, incluindo gráficos de dispersão para visualizar a distribuição dos erros dos modelos, avaliação da variabilidade dos resultados através de medidas descritivas e uma matriz SWOT cruzada, aplicada para identificar forças, fraquezas, oportunidades e ameaças da implementação do ML nos SAA, podendo assim culminar numa resposta mais profunda e fundamentada das três PD e sintetizar o CAPÍTULO 1 - ENQUADRAMENTO TEÓRICO.

Complementarmente à análise quantitativa, foi conduzida uma investigação qualitativa através de uma entrevista semiestruturada com um representante da Thales Group, uma empresa de referência no desenvolvimento de radares e sistemas de defesa que integram técnicas avançadas de ML. A entrevista teve como objetivo a avaliação da viabilidade técnica da implementação de ML nos SAA, compreender desafios operacionais e logísticos relacionados à sua adoção e obter *insights* sobre tendências futuras no desenvolvimento de sistemas inteligentes para defesa AA.

A utilização da entrevista semiestruturada justifica-se pelo seu potencial em fornecer respostas aprofundadas, permitindo explorar questões emergentes que não haviam sido previamente consideradas (Bryman, 2016).

Como método de abordagem, utilizou-se o método dedutivo uma vez que parte de premissas gerais para uma conclusão e como método de procedimento, o método experimental combinado com uma análise quantitativa.

Procurou criar-se uma série de simulações, através da utilização do *Matlab and Simulink* (The MathWorks, 2025). Para o efeito, foi necessário escolher a técnica de ML. Tal escolha baseou-se numa análise criteriosa das diferentes abordagens disponíveis, considerando fatores como: a capacidade de previsão, interpretabilidade, exigências computacionais e aplicabilidade prática nos SAA da AAA.

Assim sendo, a aprendizagem supervisionada foi a abordagem escolhida para a modelação do sistema experimental devido à necessidade de associar entradas a saídas específicas de informação, ou seja, mapear características dos alvos a categorias predefinidas. Neste contexto, a aprendizagem supervisionada permite treinar modelos a partir de dados históricos rotulados, possibilitando uma classificação precisa e previsões confiáveis sobre a trajetória e o comportamento dos alvos aéreos. Esta abordagem mostrou-se mais adequada em comparação com a aprendizagem não supervisionada, que se concentra na identificação de padrões em dados sem rótulos, sendo mais eficaz em tarefas exploratórias, como deteção de anomalias, mas menos apropriada para a necessidade específica deste estudo, que exige categorização precisa dos alvos.

Adicionalmente, considerando o carácter dinâmico da aquisição de alvos, onde estes variam no tempo e exigem previsões contínuas, optou-se pelo uso de RNA, especificamente a arquitetura LSTM. Este tipo de modelo destaca-se na análise de dados sequenciais, tornando-se numa ferramenta essencial para prever a movimentação de alvos em tempo real. A sua capacidade de memorizar padrões ao longo do tempo e de mitigar o problema do desaparecimento do gradiente, comum em redes neuronais recorrentes tradicionais, justifica a sua escolha. Estudos recentes demonstram que modelos baseados em LSTM são altamente eficazes na previsão de trajetórias e na classificação de objectos móveis em sistemas de radar e vigilância aérea (Jiang et al., 2023; Yang et al., 2020).

Embora modelos mais complexos de DL, como CNN, tenham demonstrado desempenho elevado em tarefas de classificação de imagens, a sua aplicabilidade na previsão de trajetórias é mais limitada. As CNN são particularmente eficientes no reconhecimento de padrões e análise de imagens, mas dada a natureza temporal e

sequencial dos dados analisados nesta dissertação, a escolha de LSTM revelou-se mais vantajosa.

Além disso, a aprendizagem *reinforcement*, embora seja uma técnica promissora para a otimização de estratégias de defesa, apresenta desafios significativos na sua implementação operacional, uma vez que requer grandes volumes de interações para treinar os modelos a tomarem decisões eficazes. A sua elevada complexidade computacional e necessidade de longos períodos de treino tornam-na menos prática para a aplicação imediata em sistemas de AAA, razão pela qual não foi adotada neste estudo.

Portanto, a escolha da abordagem baseada em LSTM fundamenta-se na necessidade de analisar padrões temporais, prever movimentos de alvos com alta precisão e possibilitar uma resposta rápida e eficaz na defesa antiaérea. Combinando estas capacidades com a estrutura da aprendizagem supervisionada, garante-se um modelo robusto e adaptável às exigências do campo de batalha, contribuindo para a melhoria da precisão na aquisição de alvos e otimização da alocação de recursos da defesa aérea.

Como mencionado anteriormente, a análise SWOT foi utilizada como ferramenta de síntese e, uma vez que é amplamente utilizada no planeamento estratégico para avaliar fatores internos e externos que impactam as tomadas de decisão (Gürel & Tat, 2017). Neste contexto, a utilização da matriz SWOT cruzada justifica-se como uma ferramenta metodológica essencial para compreender e justificar a relevância do ML e permite-nos identificar as forças e fraquezas do algoritmo escolhido e desenvolvido, bem como as oportunidades e ameaças do seu uso na aquisição de alvos da AAA. No entanto, a simples identificação desses fatores não é suficiente para orientar a decisão metodológica. Por isso, optou-se pela matriz SWOT cruzada, que estabelece relações entre esses fatores, permitindo um entendimento profundo do impacto do ML no desempenho dos SAA (Benzaghta et al., 2021).

Por outro lado, a relevância da SWOT como ferramenta de análise estratégica é amplamente documentada na literatura. Estudos como o de Puyt et al. (2023) ressaltam a aplicabilidade da SWOT para a avaliação de novas tecnologias, especialmente em contextos dinâmicos, como o desenvolvimento de sistemas baseados em ML para aplicações militares. Assim, a abordagem adotada nesta dissertação não apenas fortalece a fundamentação teórica das escolhas metodológicas, como também

permite um alinhamento entre as capacidades do algoritmo e os desafios operacionais dos SAA.

A metodologia adotada permite uma análise abrangente do impacto do ML nos SAA da AAA do Exército Português, combinando rigor estatístico com informação qualitativa. O estudo não apenas avalia o desempenho técnico do ML, mas também considera sua aplicabilidade prática e os desafios operacionais envolvidos.

Assim, a abordagem mista utilizada nesta investigação assegura uma resposta fundamentada à PP, permitindo concluir de forma objetiva sobre os benefícios e limitações da implementação do ML na defesa antiaérea.

A redação deste Trabalho de Investigação rege-se pelo modelo definido pela Norma de Execução Permanente número 522/2.^a que regula a redação dos Trabalhos de Investigação na Academia Militar, datada de 24 de junho de 2024.

CAPÍTULO 5 – MÉTODOS E MATERIAIS

Este capítulo insere-se no campo da aplicação de ML nos SAA da AAA, com foco na análise comparativa de um algoritmo com e sem ML. O objetivo principal é avaliar as potencialidades e limitações da introdução de técnicas de aprendizagem automática nestes sistemas, analisando métricas de desempenho e os impactos na precisão e eficiência operacional. Esta investigação decorreu em ambiente computacional, no qual foram desenvolvidos e testados algoritmos de previsão de trajetória e classificação de alvos. Além disso, complementou-se a análise com uma entrevista com um representante da Thales Group, visando obter *insights* sobre o impacto real da adoção do ML em sistemas militares.

Temos então que a abordagem metodológica adotada combina uma análise experimental, baseada na comparação de dois modelos computacionais, e uma análise qualitativa, por meio de uma entrevista semiestruturada.

Assim sendo, para a análise experimental onde procura-se perceber o impacto direto de um algoritmo de ML num SAA foi utilizado uma abordagem evolutiva constituída em três níveis distintos: i) **Algoritmo Base** – Um modelo inicial utilizado para simular trajetórias de alvos sem técnicas avançadas de previsão. (MATLAB, 2020; The MathWorks, 2022); ii) **Algoritmo sem ML** – Evolução do algoritmo base com a inclusão de técnicas heurísticas, ou seja, técnicas que funcionam como atalhos para simplificar o processo de resolução de problemas quando a existência de informação é escassa, podendo assim usar estas técnicas para a estimativa das trajetórias; iii) **Algoritmo com ML** – Integração de ML para melhorar a identificação e previsão de trajetória.

A análise comparativa entre sistemas convencionais e modelos baseados em ML é uma abordagem recomendada para avaliar o impacto da IA na defesa antiaérea (Choudhuri, 2024). Daí, os sistemas de defesa modernos dependem de métodos computacionais robustos para processar grandes volumes de dados em tempo real, tornando essencial a evolução desses algoritmos (Morteza & Mohammad-Reza, 2023; NATO Science and Technology Organization, 2022a) e, de forma a obter dados verosímeis e que de certa forma demonstrem o impacto do ML, foi utilizado o *software Matlab and Simulink* da The MathWorks (2025), onde o mesmo oferece um leque de informação sobre possíveis algoritmos de ML (Algoritmo Base) e uma rede de apoio

para fundamentar e deduzir outros possíveis algoritmos (Algoritmo sem ML e Algoritmo com ML). Ambos os modelos foram avaliados através de um conjunto de dados previamente normalizado e métricas de desempenho como o RMSE, precisão da identificação de alvos e a precisão da previsão de trajetória através do erro em distância entre pontos.

É importante ainda referir que o RMSE é amplamente utilizado para avaliar a precisão de previsões em modelos de ML, pois penaliza os erros quanto maiores estes forem devido à sua estrutura quadrática (Willmott & Matsuura, 2005). Esta métrica é especialmente relevante em modelos de perseguição de alvos e previsão de trajetórias, onde pequenos desvios podem resultar em grandes imprecisões acumuladas ao longo do tempo (Chai & Draxler, 2014). Além disso, o RMSE permite uma interpretação direta do erro médio em unidades de medida coerentes com a aplicação (Walker et al., 2003). Posto isto, o RMSE é utilizado como a principal métrica de erro entre os algoritmos desenvolvidos.

5.1. Algoritmo Base

O algoritmo base da (MATLAB, 2020; The MathWorks, 2022) tem como objetivo simular trajetórias de alvos aéreos (SANT, aeronaves de asa fixa e asa rotativa)¹⁰, introduzir ruído nos dados, isto é, adaptação do algoritmo à realidade, uma vez que a informação recolhida por sensores/radares não é estritamente “limpa”, onde muitas vezes há informação corrompida tanto pelas condições meteorológicas ou a existência de interferências. Desta forma, é adicionado o ruído aos dados, de modo a prever a trajetória futura e avaliar a precisão da previsão com a maior fidelidade possível (Fukunaga & Hostetler, 1975).

O algoritmo utiliza métodos determinísticos baseados em cálculos de gradiente, indicando que cada posição, com uma determinada direção, levar-nos-á a uma nova posição e, de acordo com o seu gradiente é possível deduzir posições futuras baseadas na anterior, criando assim uma trajetória (The MathWorks, 2022) e estatísticas de erro para prever os movimentos dos alvos (Luenberger & Ye, 2008).

O exposto pelo Anexo A pode ser dividido em **cinco etapas** principais:

- **Configuração do Ambiente** – O algoritmo define um diretório de saída e verifica a existência do mesmo; um arquivo *log.csv*⁸ é criado ou atualizado para armazenar as métricas dos testes e define o número de execuções (*numRuns*);
- **Simular Dados** – São criados 100 alvos com rótulos categóricos (*Drone, Aircraft, Helicopter*) e cada classe segue um modelo de movimento distinto: os SANT possuem um movimento circular perturbado por um fator de velocidade; as aeronaves de asa fixa têm um movimento linear com uma leve variação; e aeronaves de asa rotativa com um movimento oscilatório baseado em funções trigonométricas.

Para maior realismo, e de acordo com a introdução aos processos estocásticos de Brown e Hwang (2012), uma vez que utiliza processos aleatórios para modelar sistemas ao longo do tempo pois, aproxima variações naturais às informações dos sensores onde são adicionados ruídos às coordenadas (x, y, z), uma vez que, o uso de distribuições normais para simular ruído é comum numa modelagem estocástica;

- **Cálculo da Velocidade e Previsão de Trajetória** – para calcular as velocidades (v_x, v_y, v_z) recorre-se ao gradiente (*gradient*), ou seja, um modelo de previsão determinístico é empregue para projetar a trajetória futura, assumindo um movimento constante. A previsão de trajetórias é baseada na extrapolação da velocidade, onde a mesma segue princípios do movimento retilíneo uniforme, onde a posição futura é dada pela equação nº 3. A equação significa que a nova posição $x(t + 1)$ do objecto é igual à posição anterior $x(t)$, somada ao deslocamento causado pela velocidade $v(x)$ ao longo do intervalo de tempo Δt . De forma a facilitar o modelo de previsão, este modelo não considera aceleração, sendo válido apenas para movimentos com velocidade constante (Enders, 2019);
- **Avaliação de Erros** – São calculadas métricas de erro para avaliar a precisão da previsão, ou seja, Erro Quadrático Médio (MSE) – equação nº 7 – Raiz do Erro Quadrático Médio (RMSE) – equação nº 8 – e o Erro em Distância Total, que representa a discrepância global entre a trajetória prevista e a real – equação nº 6;

⁸ Log.csv – ficheiro Excel onde foram guardados os dados das simulações

Coadunando com a informação supracitada temos que o MSE e RMSE são amplamente utilizados para medir o desempenho de previsões numéricas, sendo adotados em análise de séries temporais e modelagem de trajetórias (Goodfellow et al., 2016);

- **Visualização e Registo dos Resultados** – o algoritmo cria gráficos 3D permitindo uma melhor comparação entre a trajetória real e a trajetória predita e as métricas de desempenho são guardadas nos ficheiros designados anteriormente (*log.csv*⁸ e *MetricsLog.txt*⁹).

Embora o algoritmo base ofereça um modelo funcional para prever trajetórias, possui limitações, entre elas temos o facto de não recorrer ao ML, dependendo somente a uma extrapolação das equações de movimento; sensibilidade ao ruído, que pode afetar significativamente os cálculos, pois não há filtragem de dados, por exemplo, os filtros de Kalman que destaca-se dos demais pelo facto de ser aquele que minimiza a variância do erro de estimativa e ser de implementação relativamente simples (Ribeiro, 2004) e, ainda a falta de integração de técnicas de regressão ou DL poderiam mitigar erros resultantes.

Estas limitações motivam a adoção de modelos de ML, que serão introduzidas no Apêndice B e C para aprimorar a previsão e reduzir erros.

5.2. Algoritmo sem *Machine Learning*

O algoritmo não recorre ao ML (Apêndice B) para executar as suas previsões, mas sim, baseia-se somente nas equações do movimento de cada elemento categorizado – *Aircraft*; *Drone*; *Helicopter*. Ou seja, o algoritmo aleatoriamente irá selecionar um elemento e com base nas equações de movimento nos eixos x, y, z desse mesmo elemento escolhido, irá “prever” o próximo ponto da trajetória.

Uma vez que se revê no algoritmo base – Anexo A – a sua estrutura arquitetónica é muito semelhante ao algoritmo providenciado pela (The MathWorks, 2022, 2024, 2025). No entanto, é-lhe introduzido um sistema de classificação baseado numa matriz de confusão, permitindo avaliar a identificação correta dos alvos, ou seja, através das trajetórias já categorizadas o algoritmo irá identificar qual a melhor ameaça

⁹ MetricsLog.txt – ficheiro de texto onde foram guardadas informações para serem usados nesta dissertação.

aérea que se identifica com a tal trajetória. É também adicionado um maior número de simulações a executar uma vez que a repetição de execuções em simulações permite análises estatísticas mais confiáveis e minimiza o impacto de variações aleatórias nos resultados (Montgomery, 2017).

Uma vez que para Provost e Kohavi (1998) as matrizes de confusão são amplamente utilizadas na análise de classificadores, para verificar a precisão da classificação de padrões, o algoritmo (Apêndice B) é reforçado com uma matriz de confusão para a classificação de alvos, permitindo assim avaliar a identificação de alvos como precisa ou não, através da comparação entre a classe real e a classe prevista (equação nº 4).

Uma vez que os algoritmos previamente mencionados – Anexo A e Apêndice B – são muito idênticos foi criada a Tabela nº 3, que permite uma melhor compreensão dos dados comparados.

Tabela nº 3 – Comparação entre Apêndice B e Anexo A

Critério	Algoritmo Base (The MathWorks, 2025)	Algoritmo sem ML
Nº de execuções	1	150
Amostras por execução	100	400
Ruído	Sim	Sim
Previsão de Trajectórias	Simples	Sim, baseado nas velocidades
Classificação e Identificação de alvos	Não	Sim, através da matriz de confusão
Cálculo de precisão	MSE e RMSE	MSE, RMSE e <i>idAccuracy</i> (4)

Sabendo que o algoritmo não recorre ao ML possui limitações idênticas ao do algoritmo base exceto no que concerne à matriz de confusão, onde apenas compara as previsões baseadas no modelo físico, sem inteligência adaptativa.

Estas limitações motivam a introdução de ML que será abordado no próximo algoritmo.

5.3. Algoritmo com *Machine Learning*

A escolha da técnica de ML para o desenvolvimento do algoritmo (Apêndice C) nos SAA da AAA exige uma fundamentação estruturada e cimentada em critérios técnicos e operacionais. Para isso, a utilização de um quadro comparativo das diferentes abordagens de ML permite uma análise objetiva das opções disponíveis, considerando fatores como precisão, complexidade computacional, interpretabilidade e aplicabilidade ao domínio militar.

Autores como Han et al. (2011) e Sarker (2021) destacam que a comparação sistemática de algoritmos de ML é essencial para selecionar a técnica mais adequada a um problema específico. Segundo esses estudos, os critérios mais relevantes para a escolha de um algoritmo incluem:

- **Precisão** – Capacidade de prever corretamente a categoria de um alvo;
- **Robustez** – Como o modelo responde a ruído e incerteza nos dados;
- **Complexidade Computacional** – Necessidade de recursos para processamento em tempo real;
- **Interpretação** – Compreensão do processo de decisão do modelo;
- **Capacidade de Generalização** – Como o modelo se comporta em dados não trabalhados.

No contexto do SAA da AAA, a escolha da técnica de ML impacta diretamente a capacidade de detecção, classificação e *tracking* de alvos, sendo necessário comparar as abordagens referentes à Tabela nº 1 do Capítulo 1.2 (pg. 7), ou seja, o quadro comparativo, exposto pela Tabela nº 4 fornece uma visão clara sobre quais técnicas atendem melhor às exigências operacionais da AAA, considerando que a LSTM mostrou-se adequada para prever trajetórias de alvos em tempo real, uma vez sendo referente ao DL e recorrer às DT que é útil para classificar e identificar alvos.

Tabela nº 4 – Técnicas de ML – Vantagens vs Desvantagens

Técnica ML	Vantagens	Desvantagens
Classificação	Preciso nas tarefas de identificação Fácil interpretabilidade (dependendo do algoritmo) Escalável para grandes volumes de dados	Pode exigir dados rotulados extensos Sensível a dados enviesados
Regressão	Fácil implementação Interpretação direta Útil para análise quantitativa	Baixa capacidade de modelar relações não lineares Pode ser influenciado por <i>outliers</i>
Clustering	Não requer dados rotulados Identifica padrões ocultos Útil para grandes conjuntos de dados	Difícil avaliar a qualidade do agrupamento Pode ser sensível a <i>outliers</i>
DL	Alto desempenho em tarefas complexas Capacidade de aprendizagem a partir de grandes volumes de dados Reconhecimento de padrões não lineares	Elevado emprego computacional Necessidade de grandes bases de dados Baixa interpretabilidade

Por sua vez, este algoritmo recorre ao ML para identificar e classificar o alvo e prever o próximo da trajetória. Este algoritmo recorre a uma rede de LSTM raíz do

software, onde é capaz de treinar os modelos de ML com base nas escolhas aleatórias do algoritmo e como utiliza LSTM consegue adequar as “suas respostas” a um historial já reconhecido.

Sabe-se ainda, que a aplicação de ML nos SAA representa um avanço significativo na classificação e previsão de trajetória de alvos aéreos. Estudos demonstram que algoritmos de ML e DL melhoram a precisão e reduzem falsos positivos em radares de defesa aérea (Jiang et al., 2023). Este algoritmo pode ser dividido em duas principais abordagens, a classificação de alvos através das DT, técnica amplamente validada na literatura para reconhecimento automático de padrões (Han et al., 2011; Matlab for Engineers, 2017a, 2017b); e a previsão de trajetórias pela LSTM, modelo que se destaca na análise de séries temporais e previsão de movimento de aeronaves (Matlab for Engineers, 2017c; Yang et al., 2020).

Esta abordagem reflete-se numa maior precisão na aquisição de alvos, redução da carga cognitiva do operador e tomadas de decisão mais rápidas e automatizadas.

A fundamentação da estrutura do algoritmo, descrito no Apêndice C, reflete as melhores práticas descritas na literatura sobre sistemas de defesa aérea e AA alienados à IA (Haenlein & Kaplan, 2019). A seguir, cada etapa do algoritmo é relacionada com os conceitos teóricos estabelecidos.

O algoritmo inicia com a simulação de trajetórias de diferentes alvos aéreos (SANT, aeronaves de asa fixa e asa rotativa)¹⁰, refletindo abordagens de simulação de guerra empregues em defesa AA (Meng et al., 2020).

A normalização dos dados (x, y, z, vx, vy, vz) segue as recomendações matemáticas e computacionais para melhorar a estabilidade do treino do modelo (Sarker & Kayes, 2020), sendo fundamental para evitar resultados enviesados, garantindo que todos os atributos tenham o mesmo peso no treino do modelo de ML (Han et al., 2011). Seguindo o mesmo raciocínio descrito, o uso de bases de dados sintéticas para treinar os SAA é um método recomendado para testar e validar modelos de ML antes da sua implementação operacional (NATO Science and Technology Organization, 2022b).

A componente de ML que integra o algoritmo (Apêndice C) pode ser descrita em duas componentes chave, como descrito anteriormente, ou seja, a utilização de árvores de decisão e redes LSTM.

A classificação dos alvos é feita pelas DT, técnica comprovadamente eficaz para identificação de aeronaves em ambientes onde o recurso de detecção seja pelo radar (Sarker et al., 2018).

Neste caso em específico o modelo é treinado para diferenciar SANT, aeronaves de asa fixa e asa rotativa, reduzindo falsos positivos e melhorando a identificação de ameaças. Jiang et al. (2023) destacam que as DT são eficazes na diferenciação de alvos aéreos, especialmente quando combinadas com a aprendizagem supervisionada, exposta no enquadramento teórico.

A previsão da trajetória futura dos alvos é realizada por uma rede neural LSTM, capaz de reconhecer padrões temporais e prever deslocamentos com alta precisão (Yang et al., 2020) e para o mesmo autor as redes LSTM demonstram alta eficácia na previsão de trajetória em tempo real, reduzindo o tempo de reação dos sistemas de defesa. Como qualquer outro modelo de ML é necessário que o mesmo seja alvo de validações e este modelo não é exceção. O mesmo é avaliado através de métricas de erro e de validação cruzada *k-fold* (k=10), ou seja, em cada execução geral do algoritmo, na reprodução aleatória de dados o algoritmo executa uma verificação definida por “k” sendo ele o mesmo número de vezes que vai realizar garantindo assim confiabilidade na aplicação prática reduzindo o risco de ocorrer *sobreajuste*⁶ por parte do modelo ML (Han et al., 2011; Matlab for Engineers, 2017a, 2017b, 2017c; The MathWorks, 2024), ou seja, o modelo já treinado vai perdendo a precisão dos resultados pois, recorre aos resultados passados como se fossem um novo resultado.

De forma a perceber as diferenças entre os algoritmos sem ML e com ML, nomeadamente, Apêndice B e C, recorre-se à tabela seguinte.

Tabela nº 5 – Comparação entre Apêndice B e C

Critério	Algoritmo sem ML (Apêndice B)	Algoritmo com ML (Apêndice C)
Classificação de alvos	Recorre à determinação aleatória	Recorre às DT

Previsão de trajetória	Baseada na velocidade inicial das equações de movimento	Recorre às redes LSTM para prever com base em dados históricos.
Precisão	Menor	Maior
Adaptação a novos cenários	Não se ajusta	Aprende e melhora com os novos dados
Redução de erros	Sujeito a falsos positivos	Filtra falsos positivos

Existe ainda uma componente do algoritmo que é o cerne da questão da aprendizagem autónoma do algoritmo que se baseia na melhoria do modelo ML já existente. Ou seja, o algoritmo é executado e cria um modelo ML (*lstm3D.mat*). Caso o algoritmo seja executado novamente, uma vez que já existe um modelo ML e de forma a reduzir o risco de *sobreajuste*⁶ como exposto anteriormente, o mesmo irá treinar o modelo existente para novos cenários com novos dados. Podendo assim, garantir que os modelos não são enviesados por dados já existentes.

5.4. Entrevista

Para complementar a investigação e garantir uma visão mais viável e atual sobre a aplicabilidade do ML nos SAA da AAA no Exército Português, foi realizada uma entrevista semiestruturada com um especialista do setor de defesa, representante da *Thales Group*, uma das empresas líderes em tecnologia de radar e defesa antiaérea.

A entrevista teve como principal objetivo compreender o impacto prático da integração de ML nos SAA, indo ao encontro dos objetivos específicos da investigação. Procurou-se com a entrevista identificar perspetivas técnicas e operacionais, incluindo os desafios e benefícios da implementação do ML em contextos militares.

Foi aplicado o guião de entrevista, referente ao Apêndice J, com o principal foco em esclarecer as principais lacunas dos capítulos acima, de forma que conjugando com a análise experimental e estatística, com a revisão de literatura, a entrevista seja o

ponto esclarecedor e exemplo prático deste estudo. Foi ainda aplicado, por questões de ética, um Termo de Consentimento Informado, referente ao Apêndice K.

As respostas foram posteriormente analisadas qualitativamente, identificando padrões e correlações com os resultados experimentais.

CAPÍTULO 6 – RESULTADOS

6.1. Resultados do Algoritmo sem *Machine Learning*

Com o algoritmo do Apêndice B foi possível obter os seguintes resultados expostos da Figura nº 6 à Figura nº 14, sendo que foram executadas 150 execuções do algoritmo e por ser insustentável a apresentação de todos os resultados, optou-se por apresentar três dos 150 resultados. Um sendo o primeiro resultado, outro o resultado referente à execução intermédia e outro referente à execução final. Não sendo o intuito desta fase demonstrar por completo os resultados, mas sim o resultado estatístico de todo o algoritmo nas 150 execuções, foi utilizado o *software* jamovi (The jamovi Project, 2022) de forma que fosse possível obter uma análise estatística fidedigna.

6.1.1. Primeira execução

São acrescentadas ainda duas métricas de percepção de precisão. A precisão de acerto, entre a posição de trajetória real vs a prevista, é dada pela equação nº 5 e o erro em distância entre estas duas posições já mencionadas pela equação nº 6.

```
Run 1
Accuracy: -515.19%
Erro Distancia: 2899.622
Precisao de ID: 34.2500%
RMSE (X, Y, Z): 1742.4261, 145.9296, 32.1056
```

Figura nº 6 - Output 'MetricsLog.txt'

Pela primeira execução do algoritmo é possível determinar pela equação nº 5 que a “*Accuracy*” apresentada é um valor impactante inferior a zero. No entanto, o valor sendo negativo ou positivo, neste caso é indiferente, uma vez que somente importante o valor absoluto representante pois, através da equação mencionada é subtraído uma posição (x, y, z) a outra posição, caracterizando assim um afastamento. Por sua vez, quanto maior for este valor absoluto, maior será o afastamento entre a previsão e o valor real.

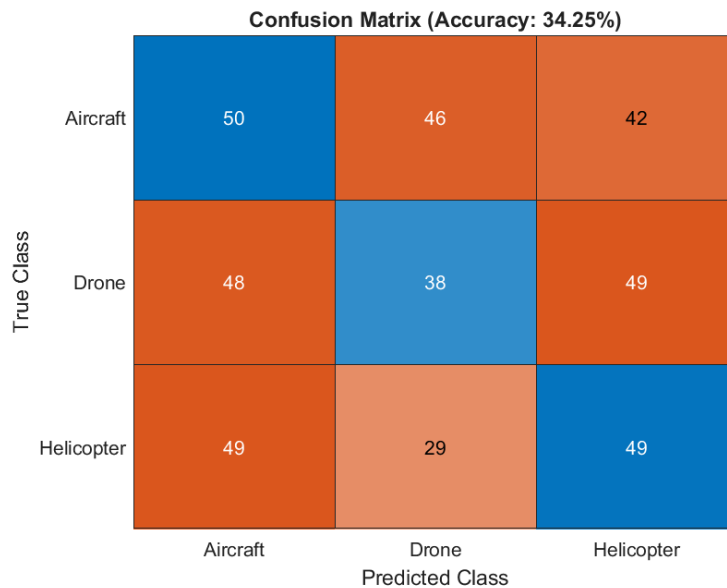


Figura nº 7 - Matriz de confusão da primeira execução

Com a primeira execução do algoritmo, com a formulação total de 400 amostras por execução, na matriz de confusão de identificação de alvos é possível observar que no cruzamento da aba “*Predicted Class*” com a aba “*True Class*” é possível extrair do resultado que o algoritmo não consegue distinguir um SANT¹⁰ de uma aeronave de asa rotativa de uma aeronave de asa fixa. Observa-se que em 147 exemplos aleatórios formulados de uma aeronave de asa fixa pelo algoritmo, deduz que 50 diz ser uma aeronave de asa fixa – estando correto – 48 sendo um SANT e 49 sendo uma aeronave de asa rotativa– sendo falso; em 113 formulações de um SANT deduz que 38 são um SANT– corresponde à verdade – 29 uma aeronave de asa rotativa e 46 uma aeronave de asa fixa – sendo falso; em 140 formulações de uma aeronave de asa rotativa corresponde a 49 deduções corretas; 49 falsas deduções de um SANT e 42 deduções erradas dizendo ser uma aeronave de asa fixa. Totalizando assim, com as 400 amostras uma percentagem de acerto 34,25%.

Com recurso de grafismo 3D que o *software* The MathWorks (2025) disponibiliza, é possível observar a disposição do último ponto da localização real da trajetória com a prevista.

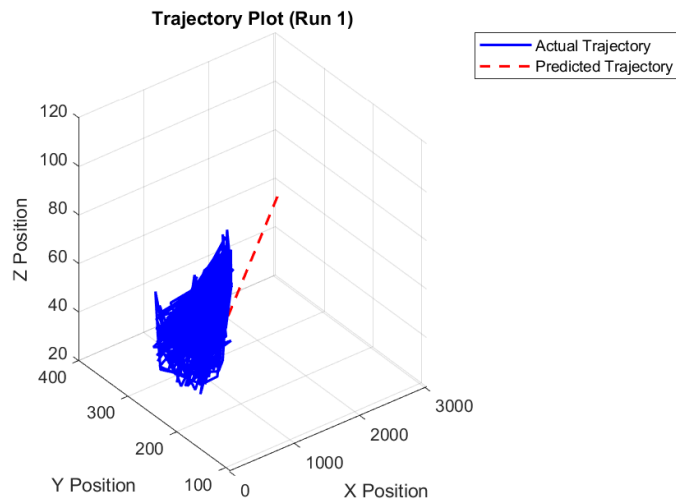


Figura nº 8 - Previsão vs Real (Trajetória) da primeira execução

Coadunando com a Figura nº 6 e Figura nº 8, acima representadas, é possível observar uma previsão díspar com a realidade, com um afastamento de 2899 metros, ou seja, o erro em distância dada pela equação nº 6.

6.1.2. Execução intermédia

É possível constatar que nesta execução, equivalente à execução número 75, temos os resultados divergentes da primeira execução, sustentando que as previsões feitas não recorrem às previsões prévias. Assim sendo, coaduna com as afirmações feitas até à presente análise.

Pela Figura nº 9 é possível observar que o erro em distância é grande e constata-se ainda que a “Accuracy” mantém-se inferior a zero, no entanto, como referido anteriormente, somente é relevante o valor em absoluto, ou seja, 10.42%.

```
Run 75
Accuracy: -10.42%
Erro Distancia: 492.671
Precisao de ID: 33.5000%
RMSE (X, Y, Z): 222.8138, 55.5239, 216.6386
```

Figura nº 9 - Output 'MetricsLog.txt'

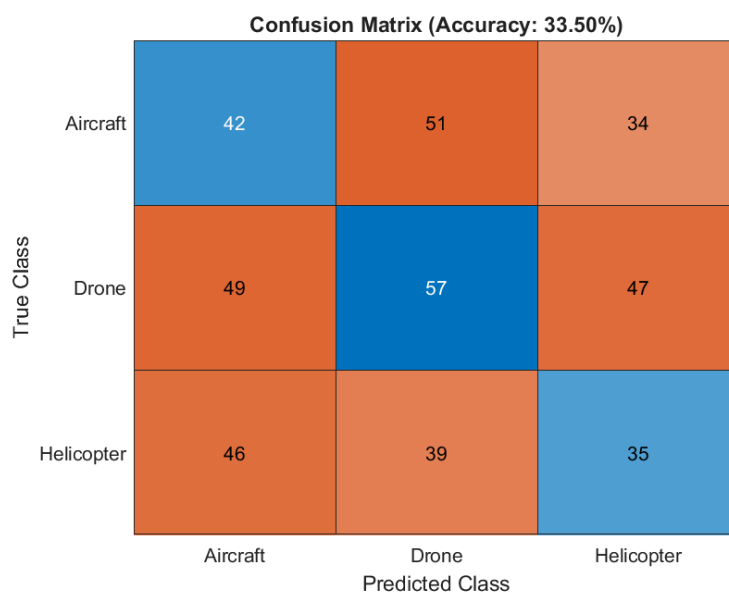


Figura nº 10 - Matriz de confusão da execução intermédia

Com a execução intermédia do algoritmo, com a formulação total de 400 amostras por execução, na matriz de confusão de identificação de alvos é possível observar que no cruzamento da aba “*Predicted Class*” com a aba “*True Class*” é possível extrair do resultado que o algoritmo não consegue distinguir um SANT de uma aeronave de asa rotativa de uma aeronave de asa fixa. Observa-se que em 137 exemplos aleatórios formulados de uma aeronave de asa fixa pelo algoritmo, deduz que 42 diz ser uma aeronave de asa fixa – estando correto – 49 um SANT e 46 sendo uma aeronave de asa rotativa – sendo falso; em 147 formulações de um SANT deduz que 57 são um SANT– corresponde à verdade – 39 uma aeronave de asa rotativa e 51 uma aeronave de asa fixa – sendo falso; em 119 formulações de uma aeronave de asa rotativa corresponde a 35 deduções corretas; 47 falsas deduções de um SANT e 34 deduções erradas dizendo ser uma aeronave de asa fixa. Totalizando assim, com as 400 amostras uma percentagem de acerto 33,50%.

A exposição gráfica do desfasamento métrico entre as posições da trajetória é observável pela Figura nº 11 com um erro de 492 metros.

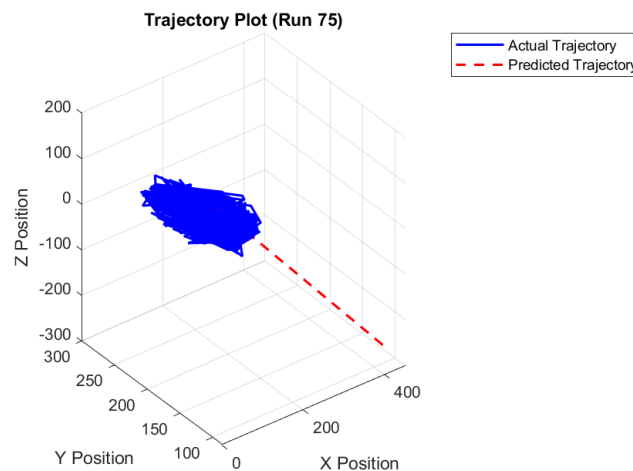


Figura nº 11 - Previsão vs Real (Trajetória) da execução intermédia

6.1.3. Última execução

Como é possível observar pela Figura nº 12, não existe qualquer relação dos valores obtidos pelas execuções do algoritmo, observa-se então que esta execução apresenta um afastamento de precisão de 1094,64 %, através da mesma forma supra referenciada e ainda com um valor de 5518 metros de erro em distância.

```
Run 150
Accuracy: -1094.64%
Erro Distancia: 5518.863
Precisao de ID: 31.2500%
RMSE (X, Y, Z): 2591.4318, 1927.9287, 560.0179
```

Figura nº 12 - Output 'MetricsLog.txt'

A matriz de confusão, Figura nº 13, expõe que com a formulação total de 400 amostras por execução, na matriz de confusão de identificação de alvos é possível observar que no cruzamento da aba “*Predicted Class*” com a aba “*True Class*” constata-se que o algoritmo não consegue distinguir um SANT de uma aeronave de asa rotativa de uma aeronave de asa fixa. Observa-se que em 121 exemplos aleatórios formulados de uma aeronave de asa fixa pelo algoritmo, deduz que 40 diz ser uma aeronave de asa fixa – estando correto – 38 um SANT e 43 sendo uma aeronave de asa rotativa– sendo falso; em 142 formulações de um SANT deduz que 50 são um SANT– corresponde à verdade – 49 uma aeronave de asa rotativa e 43 uma aeronave de asa

fixa – sendo falso; em 137 formulações de uma aeronave de asa rotativa corresponde a 35 deduções corretas; 49 falsas deduções de um SANT e 53 deduções erradas dizendo ser uma aeronave de asa fixa. Totalizando assim, com as 400 amostras uma percentagem de acerto 31,25%.

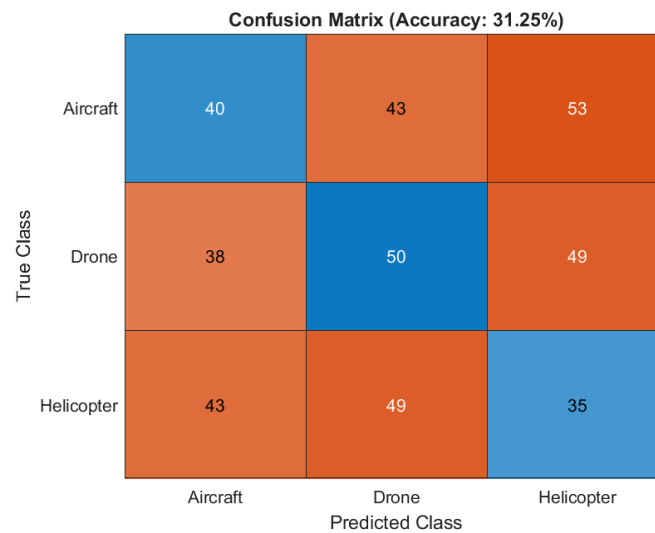


Figura nº 13 - Matriz de confusão da última execução

Sendo a representação gráfica exposta pela abaixo, Figura nº 14.

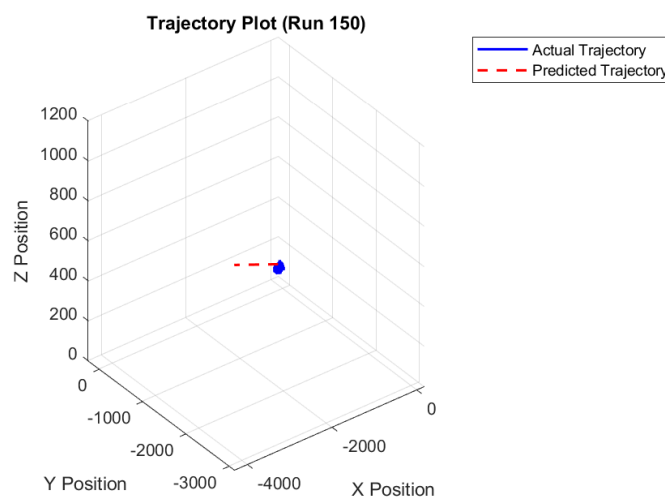


Figura nº 14 - Previsão vs Real (Trajetória) da última execução

6.1.4. Síntese

É notória a aleatoriedade dos resultados deste algoritmo, que não dispõe de recurso ao ML, uma vez que não absorve conhecimento prévio de formulações anteriores para deduzir uma melhor resposta.

Com recurso ao *software* jamovi (The jamovi Project, 2022) foi possível expor os dados obtidos, pelas execuções do algoritmo na forma de análise estatística do desempenho do algoritmo sem ML. Os mesmos encontram-se discutidos no CAPÍTULO 7 – DISCUSSÃO DOS RESULTADOS.

6.2. Resultados do Algoritmo com ML

Com o algoritmo do Apêndice C foi possível obter os seguintes resultados expostos desde a Figura nº 17 à Figura nº 25 sendo que, foram executadas 100 execuções do algoritmo, uma vez que para recorrer às redes LSTM é necessária uma grande capacidade de processamento do computador, como exposto no Capítulo 2.2 (p.25). É limitada assim a amostra a 100 execuções, e de forma que a comparação seja coerente, recorro à demonstração da primeira execução, da intermédia e da última. Não sendo o intuito desta fase demonstrar por completo os resultados, mas sim o resultado estatístico de todo o algoritmo nas 100 execuções, foi utilizado o *software* jamovi (The jamovi Project, 2022) de forma que fosse possível obter uma análise estatística fidedigna. Uma vez que algoritmo recorre as redes LSTM do *software* MathWorks (The MathWorks, 2025), o mesmo apresenta a aprendizagem do modelo numa forma visual podendo ser observado pela Figura nº 15 a primeira interação da rede LSTM e criação do modelo e pela Figura nº 16 o aperfeiçoamento do modelo ML.

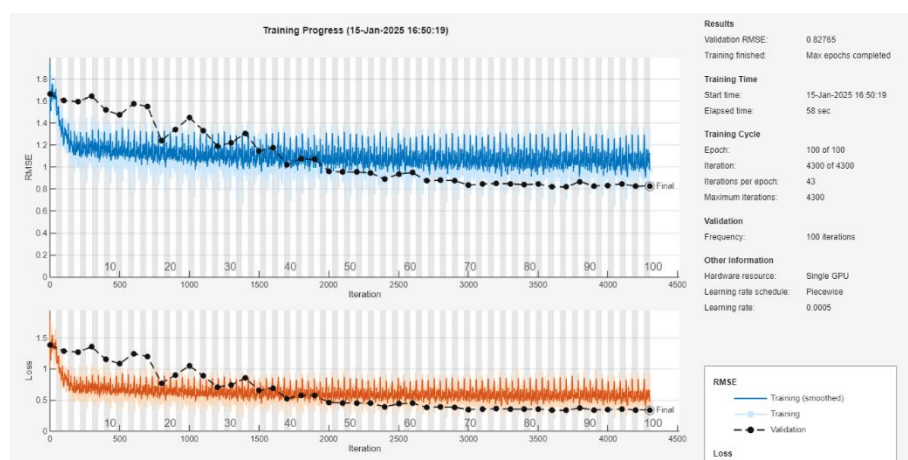


Figura nº 15 – LSTM de criação do modelo ML

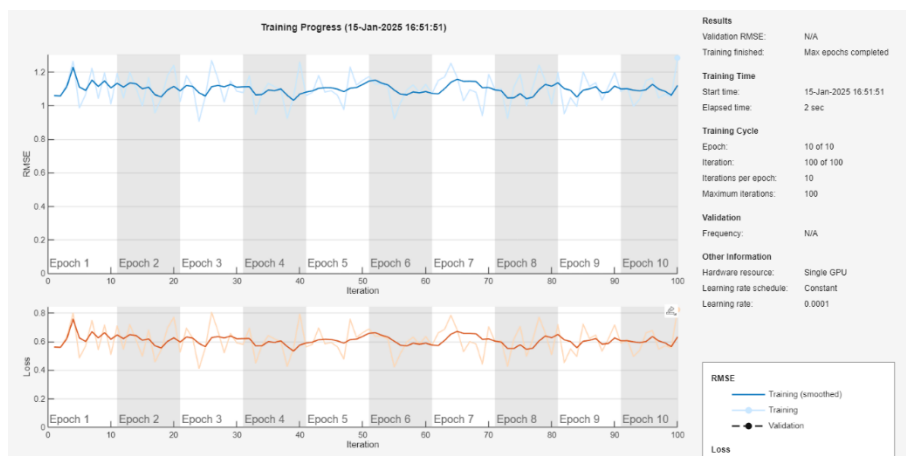


Figura nº 16 – LSTM de treino de um modelo ML existente

6.2.1. Primeira execução

Com a primeira execução e criação do modelo ML foi possível obter os seguintes resultados expostos pela figura abaixo, Figura nº 17. Observa-se uma melhoria de desempenho em concorrência direta com o algoritmo previamente exposto – Algoritmo sem ML. Aliando-me da justificação supra sobre a “Accuracy” temos que o desempenho do algoritmo nesta primeira execução é significativo face ao anterior, uma vez que o afastamento de precisão é somente de 2.55 % e, temos ainda que o erro em distância é de 38 metros.

```
Run 1
Accuracy: 2.55%
Erro Distancia: 38.258
Precisao de ID: 98.6667%
RMSE (X, Y, Z): 51.3441, 43.8662, 23.2151
```

Figura nº 17 – Output 'MetricsLog.txt'

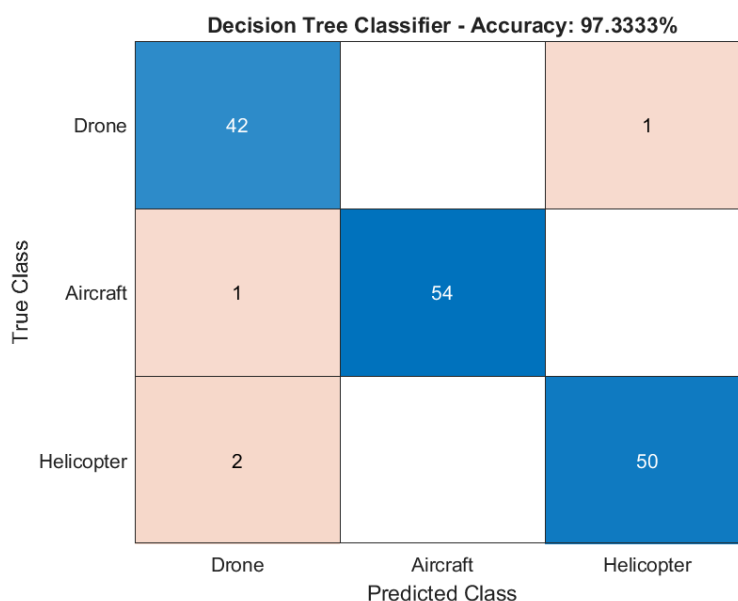


Figura nº 18 - Matriz de confusão da primeira execução

Uma vez mais devido à limitação computacional, as amostras por execução foram limitadas de 400 para 150 face ao algoritmo sem ML.

Com a primeira execução do algoritmo, com a formulação total de cento e cinquenta (150) amostras por execução, a matriz de confusão de identificação de alvos é possível observar que no cruzamento da aba “*Predicted Class*” com a aba “*True Class*” é possível retirar do resultado que o algoritmo consegue distinguir um SANT de uma aeronave de asa rotativa de uma aeronave de asa fixa. Observa-se que em 45 exemplos aleatórios formulados de um SANT pelo algoritmo, deduz que 42 diz ser um SANT– estando correto – um uma aeronave de asa fixa e dois sendo uma aeronave de asa rotativa– sendo falso; em 54 formulações de uma aeronave de asa fixa deduz que os 54 são uma aeronave de asa fixa – estando correto; em 51 formulações de uma aeronave de asa rotativa corresponde a 50 deduções corretas; uma falsa dedução de um SANT. Totalizando assim, com as 150 amostras uma percentagem de acerto 97,3%. E é ainda notório que a melhor dedução do algoritmo é diferenciar uma aeronave de asa fixa de um SANT e uma aeronave de asa rotativa, sendo que este último ainda traz alguma confusão para o algoritmo.

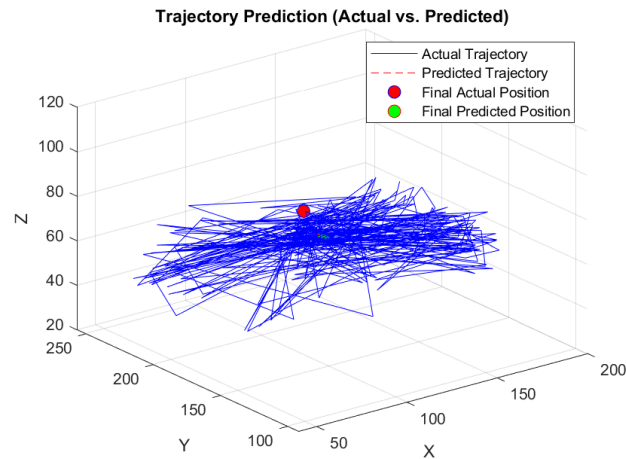


Figura nº 19 - Previsão vs Real (Trajetória) da primeira execução

Através da Figura nº 19, é exposto que o erro em distância da previsão da trajetória é ínfimo, sendo ele de 38 metros – Figura nº 17.

6.2.2. Execução intermédia

Passando à execução número 50, ou seja, a execução intermédia, onde o algoritmo já não cria um novo modelo ML, mas sim volta a treinar o modelo – Figura nº 16 – com novos dados e novos cenários de forma a manter a confiabilidade do modelo. Ao contrário do que foi observado pelo algoritmo sem ML, onde não existia relação entre os dados obtidos, pela Figura nº 20 vemos que os resultados obtidos são idênticos aos resultados equivalentes da primeira execução, onde foi então criado o modelo ML.

```
Run 50
Accuracy: 3.04%
Erro Distancia: 31.920
Precisao de ID: 100.0000%
RMSE (X, Y, Z): 54.0641, 49.9919, 45.5699
```

Figura nº 20 - Output 'MetricsLog.txt'

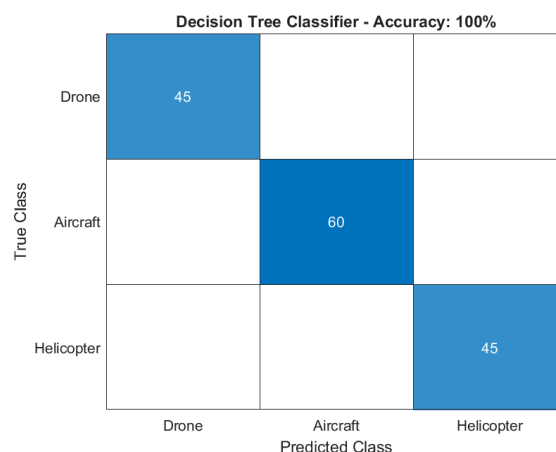


Figura nº 21 - Matriz de confusão da execução intermédia

A execução do algoritmo número 50, com a formulação total de 150 amostras por execução, a matriz de confusão de identificação de alvos é possível observar que no cruzamento da aba “*Predicted Class*” com a aba “*True Class*” é possível retirar do resultado que o algoritmo consegue distinguir um SANT de uma aeronave de asa rotativa de uma aeronave de asa fixa. Observa-se que em 45 exemplos aleatórios formulados de um SANT pelo algoritmo, deduz que 45 diz ser um SANT – estando correto; em 60 formulações de uma aeronave de asa fixa deduz que 60 são uma aeronave de asa fixa – estando correto; em 45 formulações de uma aeronave de asa rotativa corresponde a 45 deduções corretas; Totalizando assim, com as 150 amostras uma percentagem de acerto 100%. É exposta ainda pela Figura nº 22 a representação 3D da previsão de trajetória.

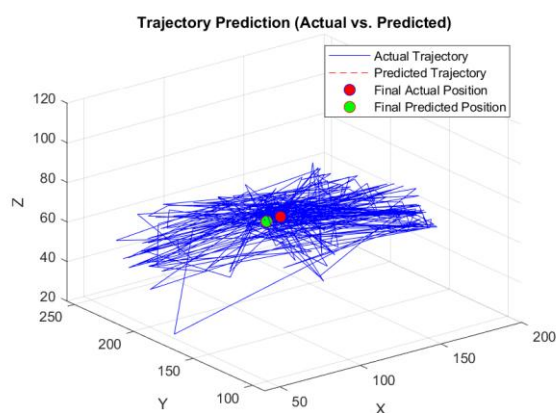


Figura nº 22 - Previsão vs Real (Trajetória) da execução intermédia

6.2.3. Última execução

Com esta execução realça-se a capacidade de melhoria de desempenho do algoritmo, com um erro em distância de 12 metros.

```
Run 100
Accuracy: 7.61%
Erro Distancia: 12.148
Precisao de ID: 99.3333%
RMSE (X, Y, Z): 72.7518, 68.9045, 37.7268
```

Figura nº 23 – Output 'MetricsLog.txt'

Por fim, a última execução do algoritmo do Apêndice C, com a formulação total de 150 amostras por execução, a matriz de confusão de identificação de alvos é possível observar que no cruzamento da aba “*Predicted Class*” com a aba “*True Class*” é possível retirar do resultado que o algoritmo consegue distinguir um SANT de uma aeronave de asa rotativa de uma aeronave de asa fixa. Observa-se que em 57 exemplos aleatórios formulados de um SANT pelo algoritmo, deduz que 57 diz ser um SANT – estando correto; em 38 formulações de uma aeronave de asa fixa deduz que 38 são uma aeronave de asa fixa – estando correto; em 55 formulações de uma aeronave de asa rotativa corresponde a 54 deduções corretas; Totalizando assim, com as 150 amostras uma percentagem de acerto 99,3%.

Decision Tree Classifier - Accuracy: 99.3333%			
True Class	Drone	Aircraft	Helicopter
	Drone	57	1
	Aircraft		38
Helicopter			54
	Predicted Class		

Figura nº 24 – Matriz de confusão da última execução

E com uma representação gráfica exposta pela Figura nº 25.

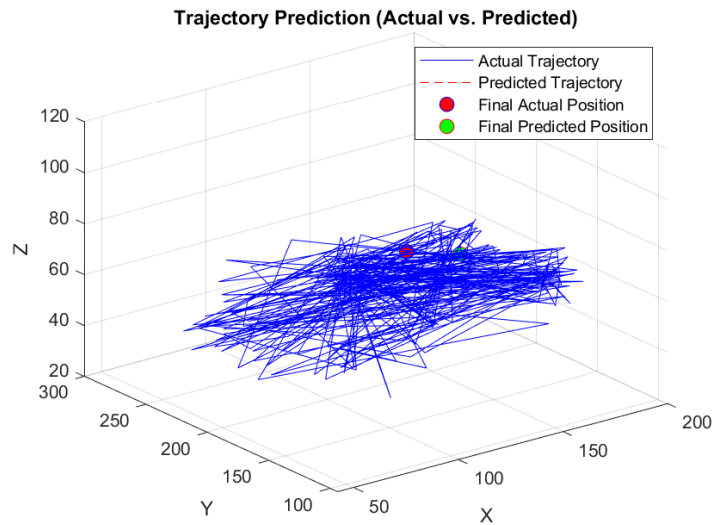


Figura nº 25 – Previsão vs Real (Trajetória) da última execução

6.2.4. Síntese

Com este algoritmo desenvolvido com recurso ao ML, e contrapondo os resultados obtidos do algoritmo sem ML, consegue-se compreender a divergência de resultados obtidos e consequentemente os níveis de desempenho. No Capítulo adiante é ainda feita a devida análise estatística.

6.3. Resultados da Entrevista

A entrevista conduzida permitiu reunir *insights* valiosos sobre o impacto do ML nos SAA da AAA, uma vez que foram abordados potenciais desafios, benefícios e limitações desta tecnologia no contexto do Exército Português. A exposição da entrevista será estruturada com base na relação entre as respostas obtidas e os objetivos específicos, bem como às perguntas derivadas estabelecidas.

É de realçar que a pessoa entrevistada tem de nome Tiago Guedes e o mesmo está empregue na Thales Edisoft Portugal, a desempenhar a função de Arquiteto de Sistemas na unidade *Land and Air Defence Systems* e possui experiência em projetos de C4ISR. No entanto, é realçado pelo mesmo que todas as respostas referem a sua perceção pessoal e não refletem nenhuma posição da Thales Edisoft Portugal nem do Grupo Thales e de acordo o pedido do Sr. Tiago Guedes onde pede exclusividade nas

informações divulgadas e de acordo com o Termo de Consentimento Informado assinado pelas duas partes, a entrevista não será transcrita na sua plenitude de forma a cumprir com o requisitado.

6.3.1. Impacto do *Machine Learning* nos Sistemas de Aquisição de Alvos da Artilharia Antiaérea (Pergunta de Partida)

A entrevista evidencia que o ML está a transformar os SAA, proporcionando avanços na deteção, rastreio e classificação de ameaças. A integração do ML melhora a precisão dos radares e dos sensores, reduzindo falsos positivos e otimizando a tomada de decisão nos sistemas C4ISR.

No entanto, são também destacadas limitações que devem ser consideradas, como a necessidade de intervenção humana na decisão final e as dificuldades na obtenção de *datasets* para treino dos modelos. Fatores estes que são fundamentais para compreender o impacto do ML no contexto da AAA, especialmente no Exército Português, onde a implementação ainda está numa fase inicial.

6.3.2. Potencialidades e Limitações do *Machine Learning* na Artilharia Antiaérea (Pergunta Derivada 1 – Objetivo Específico 2)

Foi constatado no desenrolar da entrevista que existem inúmeras potencialidades e limitações desta tipologia de tecnologia por si só e que ainda sofre uma agravante na defesa, nomeadamente na defesa antiaérea. Expõe-se assim, na tabela abaixo, Tabela nº 6, as potencialidades e limitações.

Tabela nº 6 – Potencialidades e Limitações

Argumento	Justificação
Potencialidades	
1. Melhoria na deteção e classificação de alvos	O ML permite a identificação precisa de alvos, diferenciando SANT, aeronaves tripuladas, mísseis e até objectos não hostis, como aves. Isso reduz falsos positivos e aumenta a eficácia operacional

2. Previsão de trajetórias e modo de Ataque	Algoritmos baseados no ML, incluindo as redes neurais, permitem prever trajetórias futuras e padrões de ataque, tornando a resposta defensiva mais eficiente
3. Otimização da resposta a ameaças	Sistemas de C2 com ML podem automatizar a priorização de ameaças, garantindo que alvos críticos sejam neutralizados rapidamente
4. Redução da carga cognitiva dos operadores	A automatização de processos analíticos reduz a carga de trabalho dos operadores, permitindo maior rapidez na tomada de decisão
Limitações	
5. Dependência de dados de qualidade	O desempenho do ML depende da qualidade e diversidade dos dados utilizados no treino dos modelos. A escassez de <i>datasets</i> detalhados pode comprometer a eficácia do sistema
6. Risco de viés e erros na classificação	A precisão do ML está diretamente relacionada à disponibilidade e representatividade dos dados. Novos tipos de ameaças podem não ser reconhecidos, reduzindo a eficácia do sistema
7. Complexidade computacional e necessidade de especialistas	Modelos de DL exigem elevada capacidade computacional, tornando sua implementação desafiadora em ambientes militares onde o processamento deve ocorrer em tempo real
8. Limitação na tomada de decisão	O ML não substitui a supervisão humana, pois opera com base em padrões estatísticos e pode apresentar falhas em situações imprevistas

6.3.3. Aplicabilidade do *Machine Learning* na Artilharia Antiaérea do Exército Português (Pergunta Derivada 2 – Objetivo Específico 1 e 2)

A Thales tem sido um dos principais intervenientes na integração de ML nos SAA, desenvolvendo radares avançados como o GM400α e o GM200, que utilizam IA para melhorar a identificação e a resposta a ameaças. Principais aplicações do ML na AAA:

Sistemas radar e Sensores – ou seja, radares com ML, como o GM200, permitem identificar padrões de voo e prever movimentos hostis com maior precisão e com a integração de sensores EO/IR acresce uma melhoria na fusão de dados e caracterização de alvos;

Sistemas de C2 – A automatização de processos permite uma resposta mais rápida e eficiente com uma crescente capacidade de priorizar ameaças críticas em cenários complexos;

Interoperabilidade e Adaptação – Sistemas da AAA modernos são projetados para integrar novas tecnologias sem comprometer a arquitetura base, facilitando assim a implementação do ML, ou seja, não é estritamente necessário que o sistema da AAA tenha sido inicialmente concebido para fazer recurso ao ML.

6.3.4. Importância da Implementação de Sistemas Aquisição de Alvos com *Machine Learning* (Pergunta Derivada 3 – Objetivo Específico 3)

A crescente sofisticação das ameaças aéreas, como enxames de SANT e mísseis hipersônicos, exige uma modernização dos sistemas de defesa. O ML apresenta-se como ferramenta essencial para garantir uma resposta rápida e eficiente nesses cenários. Para afirmar esta importância da implementação foram tecidos alguns benefícios da mesma, entre eles temos, uma maior eficácia operacional, uma vez que o tempo entre a deteção e a neutralização de ameaças seria mais reduzido; melhoria na fusão de dados dos diversos sistemas de aquisição da AAA, concorrendo para uma melhoria crescente da aquisição dos alvos; apoio à decisão operacional, permitindo que os operadores atuem com maior confiança e eficiência, e é referido apoio uma vez que é afirmado pelo entrevistado que a relação da tomada de decisão e as “máquinas” seja com a intervenção estrita da componente humana, reduzindo assim potenciais falhas devido a aos erros dos sistemas; capacidade de adaptação a novas ameaças, através da melhoria contínua dos modelos ML.

Apesar dos avanços, há uma preocupação ética e regulatória quanto ao uso de ML em decisões críticas, o que reforça a necessidade da supervisão humana na AAA.

6.3.5. Síntese

A entrevista realizada com um especialista da Thales Edisoft Portugal permitiu reunir contributos relevantes sobre o impacto do ML nos SAA da AAA, especialmente no contexto do Exército Português. A análise das respostas revelou uma forte consonância com os objetivos específicos da investigação e contribuiu para responder às perguntas derivadas propostas.

Relativamente à PP, confirmou-se que o ML representa um vetor transformador nos SAA, com implicações significativas na eficácia operacional, ao permitir melhorias nas capacidades de deteção, classificação, previsão de ameaças e automatização da tomada de decisão em sistemas C2. A análise revela que, embora a aplicação do ML esteja em desenvolvimento no contexto nacional, já há passos concretos, como a aquisição do radar GM200, que incorpora capacidades baseadas em inteligência artificial.

No que respeita à PD 1, foram identificadas potencialidades fundamentais, como a redução de falsos positivos, previsão de trajetórias e alocação eficiente de recursos. Contudo, destacam-se também limitações estruturais, nomeadamente a dependência de dados de treino de alta qualidade, riscos associados ao viés dos modelos, desafios de processamento em tempo real e a necessidade de manutenção da supervisão humana nas decisões críticas.

A PD 2 é diretamente abordada nas considerações sobre a aplicabilidade prática do ML nos sistemas do Exército Português. Verifica-se que os atuais sistemas AAA, mesmo os que não foram originalmente concebidos para integrar ML, podem ser adaptados tecnologicamente, graças à arquitetura modular dos sistemas modernos. A interoperabilidade, a fusão de dados de sensores e a agilidade de resposta são identificadas como ganhos operacionais relevantes.

Por fim, no âmbito da PD 3, a entrevista reforça a importância estratégica da integração do ML na AAA. O ML é visto como uma tecnologia crítica para enfrentar ameaças emergentes, como os enxames de drones e os mísseis hipersónicos. Além de potenciar a capacidade de adaptação e de resposta, o ML surge como catalisador de uma transformação organizacional e tecnológica, exigindo ao mesmo tempo cautela ética, regulamentar e doutrinária quanto à delegação de decisões operacionais a sistemas automatizados.

Neste sentido, os resultados da entrevista oferecem evidência empírica qualificada que valida as diretrizes teóricas e experimentais da dissertação, demonstrando que o emprego do ML nos SAA da AAA poderá representar um salto qualitativo na capacidade de defesa aérea, desde que as limitações técnicas, éticas e operacionais sejam respeitadas e continuamente avaliadas.

CAPÍTULO 7 – DISCUSSÃO DOS RESULTADOS

7.1. Impacto do *Machine Learning* nos Resultados Obtidos

A estatística descritiva revela uma diferença substancial no desempenho dos algoritmos com e sem ML, passível ser constatada pelo Apêndice D, G e H. O RMSE nos eixos x, y e z apresenta valores significativamente menores no algoritmo **com ML** (RMSE_X = 65.9, RMSE_Y = 59.8, RMSE_Z = 30.5) em comparação ao algoritmo **sem ML** (RMSE_X = 1233, RMSE_Y = 1539, RMSE_Z = 743). Além disso, a precisão na identificação de alvos (“%id”) é substancialmente maior no algoritmo **com ML (96.7%)** em relação ao **sem ML (33.4%)**. O erro em distância também é drasticamente reduzido (**com ML: 44.0m; sem ML: 3908m**). Esses resultados indicam que a **integração do ML melhora significativamente a previsão de trajetórias e a classificação de alvos**.

7.2. Análise Estatística Comparativa

De acordo com o Apêndice E, o teste de Shapiro-Wilk foi aplicado para avaliar se os dados seguem uma distribuição normal. Os resultados indicam que todas as variáveis apresentam **$p < 0.001$, rejeitando a hipótese nula de normalidade**. Assim sendo justifica o uso de testes **não paramétricos** para comparação entre os algoritmos, uma vez que a normalidade é um requisito essencial para testes paramétricos como o teste t (Razali & Wah, 2011).

Devido à falta de normalidade nos dados, o teste U de Mann-Whitney foi utilizado para comparar os RMSE entre os dois algoritmos. Os resultados demonstram diferenças altamente significativas ($p < 0.001$), confirmando que o algoritmo **com ML apresenta erros significativamente menores** em comparação ao algoritmo **sem ML**, como observado no Apêndice F. O teste U de Mann-Whitney é apropriado para avaliar diferenças entre grupos independentes quando a suposição de normalidade não é satisfeita (Nachar, 2008).

Conforme explanado no Apêndice G, na regressão linear entre RMSE e erro em distância (“Erro (m)”) demonstra uma forte relação entre as variáveis (**$R = 0.993$, $R^2 = 0.987$**), indicando que **uma redução nos valores de RMSE está diretamente associada a uma redução do erro em distância**. Isto reforça a **eficácia do modelo**

com ML na melhoria da previsão de trajetórias. A regressão linear foi escolhida porque permite quantificar a relação entre duas variáveis contínuas e avaliar a força dessa relação por meio do coeficiente de determinação (Kutner et al., 2005).

Por fim, a nível de demonstração de resultado gráficos, com os Apêndices H e H são apresentados os gráficos de dispersão que revelam os valores de RMSE no algoritmo **com ML**, encontram-se mais concentrados em torno de valores baixos, **indicando maior precisão.** Em contrapartida, no algoritmo **sem ML**, à medida que o RMSE aumenta, o erro em distância também cresce de forma expressiva, demonstrando a **fragilidade do modelo sem ML** na previsão confiável das trajetórias. Esse padrão reforça a relevância do ML na mitigação de erros e na estabilização das previsões (Chai & Draxler, 2014; Dantas et al., 2023).

7.3. Análise da Entrevista

A análise da entrevista confirma que o ML tem um impacto significativo nos SAA da AAA, proporcionando **avanços na detecção, identificação e resposta** a ameaças aéreas. Os sistemas que integram ML, como os radares GM200 e GM400α, demonstram que a tecnologia está a ser **gradualmente adotada**, incluindo pelo Exército Português.

Contudo, **persistem desafios** como a dependência de **dados** de qualidade, a complexidade **computacional** e a necessidade de especialistas para a implementação e operação dos modelos. Além disso, as questões éticas e regulatórias evidenciam a necessidade de manter a **supervisão humana** na decisão final.

Dessa forma, o **ML não substituirá os militares** que operem diretamente os sistemas, mas servirá como ferramenta essencial para **potencializar a eficiência dos sistemas de defesa AA.** O impacto desta tecnologia será cada vez maior à medida que novos avanços forem integrados nos sistemas AAA, garantindo uma defesa mais robusta e adaptável às ameaças emergentes.

Com base nestes resultados, pode-se concluir que a **aplicação do ML na AAA** do Exército Português é **viável e benéfica**, desde que implementada com os devidos cuidados técnicos, éticos e operacionais.

7.4. Síntese

De forma a sintetizar e fundamentar este capítulo, vai-se dar resposta às HI supramencionadas. Desta forma, como explanado no enquadramento teórico por Agrawal et al. (2018) o **ML** pode ser entendido como uma máquina de previsão com diversas aplicações e no âmbito da defesa **apresenta a capacidade de filtrar falsos positivos** como referem Rudin (2019) e Yang et al. (2020) nas suas publicações, **é de extrema importância para aumentar a capacidade da classificação e identificação de alvos**. Aspeto que também é defendido pelo entrevistado. Como demonstrado pela análise estatística prévia, assumindo que os algoritmos (Apêndice B e C), estavam associados a SAA que referenciavam a integração de ML ou não, uma vez que através do *software Matlab and Simulink*, um *software* de simulação e, neste caso adequado à simulação radar (The MathWorks, 2022, 2024, 2025), pode-se afirmar que é verificada a HI1: “Algoritmos com base em modelos ML melhoram a precisão da identificação de alvos, reduzindo os falsos positivos e aumentando a percentagem de precisão na classificação de alvos”.

Já para verificar a HI2: “A implementação do ML nos SAA da AAA pode otimizar a tomada de decisão, proporcionando uma redução no tempo de resposta”, não nos podemos limitar à linearidade defendida por Molnar (2019) pois, o ML depreende diversas técnicas de aprendizagem, sendo que uma das maiores divergências cinge-se à componente da intervenção humana, ou seja, entre a aprendizagem supervisionada – onde há grande intervenção humana – e a aprendizagem não supervisionada – onde o sistema trabalha por si só (Han et al., 2011; Sarker & Kayes, 2020). É assim dada elevada ênfase a esta componente pelo entrevistado, que por um lado concorda com Oswald (2018) no entanto, não deixa de constatar que não poderá deixar de existir a intervenção humana no C2 e sistemas C4ISR (NATO Science and Technology Organization, 2022b), outrora seria uma imersão radical de influência pelos sistemas. Pode-se assim deduzir que, a **implementação do ML reduz o tempo de resposta entre a deteção, aquisição e resposta, otimizando a tomada de decisão, no entanto, não se deve deixar de contrapor as questões éticas e morais entre influências humanas e dos sistemas**.

Por fim, confirma-se a HI3: “A aplicação do ML enfrenta desafios operacionais significativos, incluindo um grande volume de informação e requisitos computacionais” uma vez que para Molnar (2019) **para treinar modelos ML são**

necessárias grandes quantidades de dados, onde a **qualidade e diversidade** dos mesmo é **fulcral** e que é realçado ainda por Goodfellow et al. (2016) que para além de reunir grandes quantidades de dados é que estes necessitam de representar diferentes tipos de cenários. **A escassez destas bases de dados** leva à informação enviesada que pode representar **falhas na deteção e classificação de alvos** (Meng et al., 2020). O entrevistado confirma e afirma o mesmo, realça que para além do volume de informação e que os dados sejam **fidedignos** são necessários **recursos humanos especialistas**, de forma a poder aumentar o nível de interpretação das informações.

Este aspeto relaciona-se diretamente com a necessidade computacional, sustentado por Choudhuri (2024) e Chouldechova & Roth (2020), que foi influência para a componente experimental, onde para a introdução do ML no algoritmo foi necessário reduzir o número de amostras, uma vez que existe uma exigência maior de processamento, elevando a complexidade computacional.

CONCLUSÕES

A presente dissertação procurou compreender o impacto do uso do ML nos SAA da AAA, com foco particular na realidade do Exército Português. Partindo da PP – “Qual o impacto do uso do ML nos Sistemas de Aquisição de Alvos da AAA na realidade do Exército Português?” – desenvolveu-se uma abordagem metodológica mista, alicerçada em análises teórica, experimental e quantitativa, que permitiu a obtenção de resultados consistentes e articulados com os objetivos e HI delineadas.

Os capítulos iniciais estabeleceram os fundamentos conceptuais do ML, os princípios de funcionamento dos SAA e a justificação técnica e estratégica para a integração desta tipologia de tecnologia no domínio da defesa antiaérea. A revisão da literatura demonstra que o ML tem sido progressivamente incorporado nos sistemas militares de última geração, contribuindo para a automatização de processos críticos como a deteção, seguimento e a classificação de ameaças aéreas.

A análise experimental evidencia empiricamente as vantagens da introdução do ML. Uma vez feita a comparação entre algoritmos com e sem ML revelam-se melhorias significativas em métricas quantitativas como a precisão da classificação de alvos e a previsão de trajetórias, destacando-se uma taxa de acerto de 99,3% nas execuções finais do modelo com ML. A análise estatística realizada através do *software Jamovi* confirmou, com base nos testes de hipóteses, que o algoritmo recorrente de ML apresenta um desempenho superior em termos de erro médio quadrático (RMSE) e erro métrico na distância de previsão.

A entrevista realizada com um especialista da Thales Edisoft Portugal enriqueceu a perspetiva prática da investigação, sendo que esta permitiu validar as conclusões teóricas e experimentais, trazendo à luz aspetos operacionais, técnicos e éticos da implementação do ML em contextos reais. Foram sublinhadas potencialidades como a melhoria na identificação de ameaças, a previsão de modos de ataque, a otimização da resposta e a redução da carga cognitiva dos operadores. Contudo, também foram identificadas limitações relevantes, como a necessidade de supervisão humana, a dependência de *datasets* robustos e os riscos de viés.

As três Perguntas Derivadas orientaram a análise crítica dos resultados: i) PD1 – As potencialidades e limitações do ML nos SAA da AAA foram claramente evidenciadas, reforçando a pertinência de um planeamento criterioso para a sua

adoção; ii) PD2 – A aplicabilidade do ML na AAA do Exército Português mostrou-se viável, com exemplos concretos como o radar GM200, que integra capacidades de ML, ainda que o processo se encontre numa fase embrionária; iii) PD3 – A importância estratégica do ML na AAA foi confirmada. A tecnologia apresenta-se como um vetor transformador para enfrentar ameaças contemporâneas — como os enxames de drones e mísseis hipersónicos —, enquanto suscita desafios ético-operacionais que não podem ser negligenciados.

Coadunando com o exposto, é possível tecer evidentes conclusões podendo elas ser centradas em sete pontos fulcrais. Em primeiro lugar, o ML é tecnologicamente viável e operacionalmente vantajoso para os SAA da AAA, permitindo ganhos efetivos na precisão da deteção e classificação de alvos, na rapidez de resposta e na capacidade de adaptação a ameaças aéreas emergentes. O uso de redes neurais e técnicas de DL possibilita não apenas a identificação precisa de objectos hostis, como também a antecipação de suas trajetórias e modos de ataque.

Em segundo lugar, a implementação prática do ML no contexto português já se encontra em curso. A aquisição, pelo Exército Português, do radar GM200 — equipado com capacidades de inteligência artificial — demonstra que a instituição reconhece a importância estratégica destas tecnologias e está a dar os primeiros passos no sentido da sua operacionalização. Ainda que embrionária, esta iniciativa reflete uma orientação para a modernização da AAA com base em soluções tecnológicas de nova geração.

De seguida, no que diz respeito à centralidade da supervisão humana. Apesar dos avanços e da crescente autonomia dos sistemas baseados no ML, permanece fundamental que decisões com impacto letal, como a neutralização de um alvo aéreo, estejam sujeitas ao juízo de valor e responsabilidade humana. A natureza preditiva e estatística dos modelos ML implica uma margem de incerteza que, em contexto militar, pode ter consequências críticas. Assim, o papel da supervisão e do comando humano deve continuar a ser assegurado.

Em quarto lugar, foi possível verificar que os sistemas modernos, como os desenvolvidos por empresas como a Thales, são concebidos de forma modular e interoperável, permitindo a integração de componentes baseados em ML mesmo em infraestruturas existentes. Esta característica garante que a evolução tecnológica possa

ser feita de forma faseada, sem comprometer a arquitetura dos sistemas em uso, facilitando a adoção gradual e sustentável das novas capacidades.

A quinta conclusão realça que a eficácia dos sistemas baseados em ML depende criticamente da qualidade e representatividade dos dados utilizados para treino dos modelos. A escassez de *datasets* militares viáveis, bem como a limitação na partilha de dados por questões de segurança e confidencialidade, constituem entraves significativos à implementação plena destas tecnologias. Para mitigar esta limitação, será necessário desenvolver políticas robustas de recolha, gestão e simulação de dados operacionais.

Em sexto lugar, a investigação evidenciou que existe margem para o desenvolvimento nacional nesta área. O investimento e a cooperação entre instituições militares, académicas e industriais, pode posicionar Portugal como um ator relevante na criação de soluções autónomas e customizadas para os seus sistemas de defesa aérea, reduzindo a dependência externa e promovendo a soberania tecnológica.

Por fim, o desenvolvimento e aplicação do ML na defesa deve respeitar princípios éticos e legais, acompanhando a evolução da regulamentação internacional sobre sistemas autónomos e inteligência artificial. A necessidade de garantir transparência, viabilidade e responsabilidade nas decisões tomadas por sistemas assistidos por ML impõe-se como um desafio transversal, exigindo que as Forças Armadas estejam preparadas para integrar estas preocupações desde a fase de conceção dos sistemas até à sua operação no terreno.

Conclui-se, assim, que a integração do ML nos SAA da AAA é não apenas tecnicamente viável, mas também operacionalmente desejável e estrategicamente necessária. Para que esta transformação seja bem-sucedida, será fundamental garantir investimentos em infraestrutura, formação de especialistas e definição de enquadramentos regulatórios e éticos robustos. A modernização da AAA exige, portanto, um esforço multidimensional que combine inovação tecnológica com doutrina, política e estratégia.

Esta dissertação contribui para esse esforço ao apresentar uma visão abrangente, fundamentada e aplicada sobre o papel do ML na defesa antiaérea, deixando ainda espaço para investigações futuras que explorem temas como o uso de ML em sistemas multi-agente, interoperabilidade em tempo real e arquitetura de redes neurais explicáveis em contextos de alto risco.

REFERÊNCIAS BIBLIOGRÁFICAS

- Abeynayake, C., Son, V., Shovon, M. H., & Yokohama, H. (2019). *Machine Learning based automatic target recognition algorithm applicable to ground penetrating radar data. Defense + Commercial Sensing*.
- Agrawal, A., Gans, J., & Goldfarb, A. (2018). *Prediction Machines: The Simple Economics of Artificial Intelligence*. Harvard Business Review Press.
- Alcántara Suárez, E. J., & Monzon Baeza, V. (2023). Evaluating the Role of Machine Learning in Defense Applications and Industry. *Machine Learning and Knowledge Extraction*, 5(4), 1557–1569. <https://doi.org/10.3390/make5040078>
- Alves, E., Leal, A., Lopes, M., & Fonseca, A. (2021). Performance Analysis Among Predictive Models of Lightning Occurrence Using Artificial Neural Networks and SMOTE. *IEEE Latin America Transactions*, 19(5), 755–762. <https://doi.org/10.1109/TLA.2021.9448309>
- Avelar, E. A., Da Silva, S. E., Boina, T. M., & Tormin, B. F. (2023). Emprego do algoritmo Naive Bayes para previsão do movimento dos preços de ativos em mercados de capitais. *Revista de Gestão e Secretariado (Management and Administrative Professional Review)*, 14(7), 12099–12115. <https://doi.org/10.7769/gesec.v14i7.2521>
- Benzaghta, M. A., Elwalda, A., Mousa, M., Erkan, I., & Rahman, M. (2021). SWOT analysis applications: An integrative literature review. *Journal of Global Business Insights*, 6(1), 55–73. <https://doi.org/10.5038/2640-6489.6.1.1148>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Brown, R., & Hwang, P. (2012). *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises* (4.^a ed.). Wiley.
- Bryman, A. (2016). *Social research methods* (5.^a ed.). Oxford University Press.
- Cao, L. (2018). Data Science. *ACM Computing Surveys*, 50(3), 1–42. <https://doi.org/10.1145/3076253>
- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250. <https://doi.org/10.5194/gmd-7-1247-2014>

- Choudhuri, S. S. (2024). THE IMPACT OF ARTIFICIAL INTELLIGENCE ON BUSINESS LEADERSHIP AND STRATEGY. *IJARCCCE*, 11(10).
<https://doi.org/10.17148/IJARCCCE.2022.111015>
- Chouldechova, A., & Roth, A. (2020). A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5), 82–89. <https://doi.org/10.1145/3376898>
- Couldry, N., & Mejias, U. A. (2020). *The Costs of Connection. How Data is Colonizing Human Life and Appropriating it for Capitalism*. Stanford University Press.
<https://doi.org/10.1515/9781503609754>
- Courtney, C. (2021). Operacionalização da Inteligência Artificial para a Guerra Algorítmica. *Military Review*, 76–88.
- Dantas, J., Geraldo, D., Medeiros, F., Maximo, M., & Yoneyama, T. (2023). *Real-Time Surface-to-Air Missile Engagement Zone Prediction Using Simulation and Machine Learning*.
- Davenport, T., Holweg, M., & Jeavons, D. (2023). *How AI Is Helping Companies Redesign Processes*.
- Davenport, T., & Mittal, N. (2023). *All-in On AI: How Smart Companies Win Big with Artificial Intelligence*. Harvard Business Press.
- EME. (2016). *PDE 3-37-00 - Tática de Artilharia Antiaérea*. Lisboa: Exército Português.
- Enders, P. (2019). Equation of Motion. Em *Classical Mechanics and Quantum Mechanics: An Historic-Axiomatic Approach* (pp. 151–158). BENTHAM SCIENCE PUBLISHERS. <https://doi.org/10.2174/9781681084497119010014>
- Esposito, E. (2022). *Artificial Communication: How Algorithms Produce Social Intelligence*. MIT Press.
- Feng, W., Hu, X., & He, X. (2024). Artificial Intelligence (AI)-Based Radar Signal Processing and Radar Imaging. *Electronics*, 13(21), 4251.
<https://doi.org/10.3390/electronics13214251>
- Fontana, S., & Di Lauro, F. (2022). An Overview of Sensors for Long Range Missile Defense. *Sensors*, 22(24), 9871. <https://doi.org/10.3390/s22249871>
- Fraley, J. B., & Cannady, J. (2017). The promise of machine learning in cybersecurity. *SoutheastCon 2017*, 1–6. <https://doi.org/10.1109/SECON.2017.7925283>

- Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1), 32–40. <https://doi.org/10.1109/TIT.1975.1055330>
- García-Orosa, B., Canavilhas, J., & Vázquez-Herrero, J. (2023). Algorithms and communication: A systematized literature review. *Comunicar*, 74.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Gunkel, D. (2020). *An introduction to communication and artificial intelligence*. Polity Books. <https://doi.org/10.1515/commun-2020-0039>
- Gürel, E., & Tat, M. (2017). SWOT ANALYSIS: A THEORETICAL REVIEW. *Journal of International Social Research*, 10(51), 994–1006. <https://doi.org/10.17719/jisr.2017.1832>
- Guzman, A. L., & Lewis, S. C. (2020). Artificial intelligence and communication: A Human–Machine Communication research agenda. *New Media & Society*, 22(1), 70–86. <https://doi.org/10.1177/1461444819858691>
- Haenlein, M., & Kaplan, A. (2019). A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence. *California Management Review*, 61(4), 5–14. <https://doi.org/10.1177/0008125619864925>
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3.^a ed.). Elsevier.
- Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, 29(2), 1–12. <https://doi.org/10.1145/335191.335372>
- Harari, Y. N. (2017). *Homo Deus: A Brief History of Tomorrow*. Penguin.
- Hebert, K. (2020). *A DECISÃO HUMANA ENTRE RACIOCÍNIO E INSTINTO EM FACE DA INTELIGÊNCIA ARTIFICIAL*. Escola de Guerra Naval.
- Helland, A. R., Willett, T. J., & Tisdale, G. E. (1981). *Application Of Image Understanding Automatic Tactical Target Acquisition*.
- Hermann, E. (2022). Artificial intelligence and mass personalization of communication content—An ethical and literacy perspective. *New Media & Society*, 24(5), 1258–1277. <https://doi.org/10.1177/14614448211022702>
- Jane’s Defense. (2025). *Radar and Electronic Warfare Systems: AN/MPQ-64 Sentinel*. Jane’s Information Group. .

- Jiang, W., Wang, Y., Li, Y., Lin, Y., & Shen, W. (2023). Radar Target Characterization and Deep Learning in Radar Automatic Target Recognition: A Review. *Remote Sensing*, 15(15), 3742. <https://doi.org/10.3390/rs15153742>
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4, 237–285. <https://doi.org/10.1613/jair.301>
- Kim, J., Lee, H., & Choi, S. (2018). Machine Learning based approach for demand forecasting anti-aircraft missiles. *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, 367–372.
- Kondaveeti, H. K., Kancharla, R., Kumar, K. R. S., & Vatsavayi, V. K. (2024). *Automatic Target Recognition from ISAR Images: A Review* (pp. 341–350). https://doi.org/10.1007/978-981-99-4795-9_32
- Kutner, M., Nachtsheim, C., Neter, J., & Li, W. (2005). *Applied Linear Statistical Models* (5.^a ed.). McGraw-Hill.
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
- Luenberger, D., & Ye, Y. (2008). *Linear and Nonlinear Programming* (F. Hillier, Ed.; 3.^a ed., Vol. 116). Springer.
- MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*. <https://api.semanticscholar.org/CorpusID:6278891>
- MATLAB. (2020, Janeiro 9). *Deep Learning with MATLAB Tutorial* [Broadcast]. <https://www.youtube.com/watch?v=xeh5xVC245U>
- Matlab for Engineers. (2017a, Novembro 9). *MATLAB skills, deep learning, sect 2: Making Predictions, Make a prediction* [Broadcast]. https://www.youtube.com/watch?v=d2ym_o31S9I&list=PL3jZiDXc4Fb-skWlRfG5IWvNQ_tu2eiky&index=2
- Matlab for Engineers. (2017b, Novembro 10). *MATLAB skills, deep learning, sect 3: Making Predictions, Obtaining pretrained networks* [Broadcast]. https://www.youtube.com/watch?v=GhSNfDgKJf0&list=PL3jZiDXc4Fb-skWlRfG5IWvNQ_tu2eiky&index=3

- Matlab for Engineers. (2017c, Novembro 11). *MATLAB skills, deep learning, sect 4: CNN Architecture, Layers in a CNN* [Broadcast].
https://www.youtube.com/watch?v=gjK70r0Rqzs&list=PL3jZiDXc4Fb-skWlRfG5IWvNQ_tu2eiky&index=4
- McCallum, A. (2005). Information Extraction. *Queue*, 3(9), 48–57.
<https://doi.org/10.1145/1105664.1105679>
- Meng, T., Jing, X., Yan, Z., & Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, 57, 115–129. <https://doi.org/10.1016/j.inffus.2019.12.001>
- Molnar, C. (2019). *A Guide for Making Black Box Models Explainable*. Lean Publishing.
- Montgomery, D. (2017). *Design and Analysis of Experiments* (9.^a ed.). Wiley.
- Morteza, S., & Mohammad-Reza, T. (2023). Analysis of Russian Missile Defense Systems in the Lense of Neo-Clasical Realism. *Journal of Central Eurasia Studies*, 1–7.
- Nachar, N. (2008). The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come from the Same Distribution. *Tutorials in Quantitative Methods for Psychology*, 4(1), 13–20. <https://doi.org/10.20982/tqmp.04.1.p013>
- Naeem, M., Rizvi, S. T. H., & Coronato, A. (2020). A Gentle Introduction to Reinforcement Learning and its Application in Different Fields. *IEEE Access*, 8, 209320–209344.
<https://doi.org/10.1109/ACCESS.2020.3038605>
- NATO Science and Technology Organization. (2022a). *2022 Highlights*.
- NATO Science and Technology Organization. (2022b). *AI and Big Data for Decision Making in NATO Command & Control (STO-TR-MSG-189)*.
- Opti-Num Solutions. (2022, Março 16). *Designing Multifunction Radars with MATLAB and Simulink* [Broadcast]. <https://www.youtube.com/watch?v=pdiFwEljds&t=3232s>
- Oswald, M. (2018). Algorithm-assisted decision-making in the public sector: framing the issues using administrative law rules governing discretionary power. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2128), 20170359. <https://doi.org/10.1098/rsta.2017.0359>
- Passbach, D. (2024). National and International Perspectives of the Hungarian Ground-Based Air Defence Forces (2). *Academic and Applied Research in Military and Public Management Science*, 23(2), 51–61. <https://doi.org/10.32565/aarms.2024.2.4>

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., & Louppe, G. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12.
- Pey, J. (2022). *STUDY ON THE USE OF DRONES IN PUBLIC SAFETY INTELLIGENCE OPERATIONS*. Universidade de Brasília.
- Ploumis, M. (2022). AI weapon systems in future war operations; strategy, operations and tactics. *Comparative Strategy*, 41(1), 1–18.
<https://doi.org/10.1080/01495933.2021.2017739>
- Provost, F., & Kohavi, R. (1998). Glossary of terms. Machine Learning—Special Issue on Applications of Machine Learning and the Knowledge Discovery Process. Em *Machine Learning* (4.^a ed., Vol. 2, pp. 121–135). <https://doi.org/10.1023/A:1017181826899>
- Puyt, R. W., Lie, F. B., & Wilderom, C. P. M. (2023). The origins of SWOT analysis. *Long Range Planning*, 56(3), 102304. <https://doi.org/10.1016/j.lrp.2023.102304>
- Rafael Advanced Defense Systems. (2025). *Iron Dome and EO/IR Systems*.
- Razali, N. M., & Wah, Y. B. (2011). *Power comparisons of Shapiro-Wilk , Kolmogorov-Smirnov , Lilliefors and Anderson-Darling tests*.
<https://api.semanticscholar.org/CorpusID:18639594>
- Rege, M., Blanch, R., & Mbah, K. (2018). *Machine Learning for Cyber Defense and Attack*.
<https://api.semanticscholar.org/CorpusID:232319054>
- Ribeiro, M. I. (2004). *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*. Instituto Superior Técnico.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215. <https://doi.org/10.1038/s42256-019-0048-x>
- Saab Group. (2023). *Giraffe AMB – Mobile Air Surveillance Radar*.
- Sankaran, J., & Fetter, S. (2020). Reexamining Homeland Missile Defense against North Korea. *The Washington Quarterly*, 43(3), 47–62.
<https://doi.org/10.1080/0163660X.2020.1813400>
- Sarker, I. H. (2019). A machine learning based robust prediction model for real-life mobile phone data. *Internet of Things*, 5, 180–193. <https://doi.org/10.1016/j.iot.2019.01.007>

- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>
- Sarker, I. H., Colman, A., Kabir, M. A., & Han, J. (2018). Individualized Time-Series Segmentation for Mining Mobile Phone User Behavior. *The Computer Journal*, 61(3), 349–368. <https://doi.org/10.1093/comjnl/bxx082>
- Sarker, I. H., Hoque, M. M., Uddin, Md. K., & Alsanoosy, T. (2021). Mobile Data Science and Intelligent Apps: Concepts, AI-Based Modeling and Research Directions. *Mobile Networks and Applications*, 26(1), 285–303. <https://doi.org/10.1007/s11036-020-01650-z>
- Sarker, I. H., & Kayes, A. S. M. (2020). ABC-RuleMiner: User behavioral rule-based machine learning method for context-aware intelligent services. *Journal of Network and Computer Applications*, 168, 102762. <https://doi.org/10.1016/j.jnca.2020.102762>
- Thales. (2021a). *Ground Master 400 Alpha, taking long-range air surveillance to the next level*.
- Thales. (2021b, Maio 19). *Augmented intelligence in critical systems*.
- Thales. (2024a). *Thales speeds up its development of AI for defence*.
- Thales. (2024b). *Trusted & Responsible AI in Thales Defence Systems*.
- The jamovi Project. (2022). *jamovi* (2.3).
- The MathWorks, I. (2022). *Statistics and Machine Learning Toolbox*.
- The MathWorks, I. (2024). *Radar Target Classification Using Machine Learning and Deep Learning*.
- The MathWorks, I. (2025). *MATLAB — Apps*.
- Tunçer, Ö. (2019). Application of Machine Learning Methods for Air Defense Target Classification. *2019 27th Signal Processing and Communications Applications Conference (SIU)*, 1–4.
- Upreti, A. (2022). *Convolutional Neural Network (CNN). A Comprehensive Overview*. <https://doi.org/10.20944/preprints202208.0313.v3>
- Verly, J. G., Delanoy, R. L., & Dudgeon, D. E. (1989). *Machine Intelligence Technology for Automatic Target Recognition*.

- Walker, W. E., Harremoës, P., Rotmans, J., van der Sluijs, J. P., van Asselt, M. B. A., Janssen, P., & Kreyer von Krauss, M. P. (2003). Defining Uncertainty: A Conceptual Basis for Uncertainty Management in Model-Based Decision Support. *Integrated Assessment*, 4(1), 5–17. <https://doi.org/10.1076/iaij.4.1.5.16466>
- Willmott, C., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30, 79–82. <https://doi.org/10.3354/cr030079>
- Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., & Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. *IEEE Access*, 6, 35365–35381. <https://doi.org/10.1109/ACCESS.2018.2836950>
- Xu, Y., Liu, X., Cao, X., Huang, C., Liu, E., Qian, S., Liu, X., Wu, Y., Dong, F., Qiu, C.-W., Qiu, J., Hua, K., Su, W., Wu, J., Xu, H., Han, Y., Fu, C., Yin, Z., Liu, M., ... Zhang, J. (2021). Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, 2(4), 100179. <https://doi.org/10.1016/j.xinn.2021.100179>
- Yang, Z., Tang, R., Bao, J., Lu, J., & Zhang, Z. (2020). A Real-Time Trajectory Prediction Method of Small-Scale Quadrotors Based on GPS Data and Neural Network. *Sensors*, 20(24), 7061. <https://doi.org/10.3390/s20247061>
- Zuboff, S. (2019). *The age of surveillance capitalism: the fight for a human future at the new frontier of power* (Vol. 6). Public Affairs. <https://doi.org/10.1007/s00146-020-01100-0>

ANEXOS E APÊNDICES

ANEXO A – Algoritmo Base (*MathWorks*)

```
outputDir = 'name';
if ~exist(name, 'dir')
    mkdir(name);
end
numRuns = 1;
logFile = fullfile(name, 'log.csv');
if isfile(logFile)
    % Read existing log data
    logData = readtable(logFile, 'PreserveVariableNames', true);
else
    columnNames = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'T'};
    logData = table([], [], [], [], [], [], [], [], [], 'VariableNames', columnNames);
    writetable(logData, logFile, 'WriteMode', 'overwrite', 'WriteRowNames', false);
end
for runNumber = 1:numRuns
    numSamples = 100;
    time = linspace(0, 20, numSamples);
    targetLabels = categorical(randi([1, 3], numSamples, 1), [1, 2, 3], {'Drone', 'Aircraft', 'Helicopter'});
    x = zeros(numSamples, 1);
    y = zeros(numSamples, 1);
    z = zeros(numSamples, 1);
    for i = 1:numSamples
        if targetLabels(i) == 'Drone'
            speedFactor = 0.8 + randn() * 0.1;
```

```
        x(i) = 100 + 50 * cos(speedFactor * time(i));
        y(i) = 200 + 50 * sin(speedFactor * time(i));
        z(i) = 50 + randn() * 5;
    elseif targetLabels(i) == 'Aircraft'
        x(i) = 50 + 0.2 * time(i);
        y(i) = 100 + 0.3 * time(i);
        z(i) = 100 + 0.05 * time(i) + randn() * 2;
    elseif targetLabels(i) == 'Helicopter'
        x(i) = 150 + 30 * sin(0.5 * time(i) + randn() * 0.1);
        y(i) = 150 + 30 * cos(0.3 * time(i) + randn() * 0.1);
        z(i) = 70 + 10 * sin(0.7 * time(i) + randn() * 0.2);
    end
end
noiseLevel = 5;
x = x + noiseLevel * randn(size(x));
y = y + noiseLevel * randn(size(y));
z = z + noiseLevel * randn(size(z));
vx = gradient(x, time);
vy = gradient(y, time);
vz = gradient(z, time);
numPredictions = 50;
predictedTrajectory = zeros(numPredictions, 3);
currentX = x(end);
currentY = y(end);
currentZ = z(end);
```

```

currentVx = vx(end);

currentVy = vy(end);

currentVz = vz(end);

for i = 1:numPredictions

    currentX = currentX + currentVx * (time(2)
- time(1));

    currentY = currentY + currentVy * (time(2)
- time(1));

    currentZ = currentZ + currentVz * (time(2) -
time(1));

    predictedTrajectory(i, :) = [currentX,
currentY, currentZ];

end

mseX = mean((x(end-numPredictions+1:end) -
predictedTrajectory(:, 1)).^2);

mseY = mean((y(end-numPredictions+1:end) -
predictedTrajectory(:, 2)).^2);

mseZ = mean((z(end-numPredictions+1:end) -
predictedTrajectory(:, 3)).^2);

rmseX = sqrt(mseX);

rmseY = sqrt(mseY);

rmseZ = sqrt(mseZ);

distanceError = sqrt((x(end)
- predictedTrajectory(end, 1))^2 + ...
(y(end) - predictedTrajectory(end,
2))^2 + ...
(z(end) - predictedTrajectory(end,
3))^2);

actualTrajectory = [x(end-
numPredictions+1:end), y(end-
numPredictions+1:end), z(end-
numPredictions+1:end)];

diffTrajectory = actualTrajectory -
predictedTrajectory;

trajectoryError = sqrt(sum(diffTrajectory.^2,
2));

trajectoryAccuracy = (1 -
mean(trajectoryError)
max(max(abs(actualTrajectory)))) * 100;

```

```

figure;

plot3(x, y, z, 'b-', 'LineWidth', 1.5); hold on;

plot3(predictedTrajectory(:, 1),
predictedTrajectory(:, 2), predictedTrajectory(:,
3), 'r--', 'LineWidth', 1.5);

legend('Actual Trajectory', 'Predicted
Trajectory');

xlabel('X Position'); ylabel('Y Position');
zlabel('Z Position');

title(['Trajectory Plot (Run ',
num2str(runNumber), ')']);

grid on;

saveas(gcf, fullfile(outputDir,
['TrajectoryPlot_Run', num2str(runNumber),
'.png']));

close(gcf);

logDataToAppend = table(runNumber, mseX,
mseY, mseZ, rmseX, rmseY, rmseZ,
trajectoryAccuracy, distanceError, ...
'VariableNames', {'RunNumber', 'MSE_X',
'MSE_Y', 'MSE_Z', 'RMSE_X', 'RMSE_Y',
'RMSE_Z', 'TrajectoryAccuracy',
'DistanceError'});

writetable(logDataToAppend, logFile,
'WriteMode', 'append', 'WriteRowNames', false);

metricsFile = fullfile(outputDir,
'MetricsLog.txt');

fid = fopen(metricsFile, 'a'); % Open file in
append mode

fprintf(fid, 'Run %d\n', runNumber);

fprintf(fid, 'Trajectory Accuracy: %.2f%%\n',
trajectoryAccuracy);

fprintf(fid, 'Distance Error: %.3f\n',
distanceError);

fprintf(fid, 'RMSE (X, Y, Z): %.4f, %.4f,
%.4f\n', rmseX, rmseY, rmseZ);

fprintf(fid, '=====\n');

fclose(fid);

end

```

```
disp(['All results saved to ', 'name']);
```

APÊNDICE A – Sistemas de Aquisição de Alvos

Os SAA da AAA são responsáveis pela detecção, seguimento e identificação de ameaças aéreas, fornecendo informações críticas para sistemas de armas e a tomada de decisão. Esses sistemas combinam radares, sensores EO/IR, SIGINT e redes de C2 para aumentar a eficiência operacional e minimizar o risco de fogo amigo.

A doutrina operacional da AAA enfatiza que a fase de detecção e alerta precoce é essencial para aumentar a capacidade de resposta contra ameaças aéreas. Segundo a NATO Science and Technology Organization (2022a), os SAA são a primeira linha de defesa, garantindo que aeronaves, SANT e mísseis sejam identificados e neutralizados antes de atingirem alvos críticos. Conforme descrito na doutrina operacional da AAA “O emprego dos radares visa providenciar, no mais curto espaço de tempo, o alerta, a identificação e o envio de dados sobre a ameaça aérea para as unidades táticas de defesa AA.” (EME, 2016, p. 64) e não servindo somente para alertar, detetar e identificar ameaças, os meios de aquisição desempenham uma missão fulcral na conexão entre as componentes de defesa aérea como conforme descrito igualmente pelo EME (2016) temos que

Os radares de AAA contribuem para a imagem operacional comum (*Common Operational Picture – COP*), proporcionando o aviso prévio aos sistemas de armas SHORAD e MRSAM/HIMAD sobre a aproximação de meios aéreos hostis, através do seu sistema de comando, controlo, comunicações, computadores e informações (*Command, Control, Communications, Computers, and Intelligence – C4I*) de AA. Os radares de AAA, além de contribuírem para a COP, também podem enviar os dados para o elemento de gestão do espaço aéreo, que por sua vez pode acionar os meios de Defesa Aérea para fazer face à ameaça (p. 64).

Assim sendo, os radares são a espinha dorsal dos SAA da AAA, projetados para detetar e fazer o *tracking* de alvos aéreos em diferentes distâncias e altitudes. Entre eles existem: i) **radares de aviso local** – usados para detetar alvos próximos, como SANT e

aeronaves de asa fixa e asa rotativa¹⁰; ii) **radares de vigilância** – utilizados para detectar e fazer o *tracking* aeronaves e mísseis balísticos (Yang et al., 2020; Jane’s Defense, 2025); iii) **radares de perseguição e conduta de tiro** – orientados para a ameaça aérea adquirida pelas radares previamente mencionados e os dados inerentes são “transmitidos por sistemas automáticos de C4ISR, tendo a capacidade de guiar automaticamente os sistemas de armas para intercepção dos alvos” (EME, 2016, p. 66).

Assim, os SAA da AAA para efeitos do presente estudo, agregamos em:

- **Radares de Vigilância e Aviso Local**

Os radares de vigilância e aviso local sendo o mecanismo fundamental dos sistemas de defesa antiaérea, projetados para detectar alvos aéreos em diferentes distâncias e altitudes. Estes radares são categorizados conforme seu alcance e finalidade: Radares de Aviso Local – Detetam ameaças próximas, como SANT de pequeno porte e aeronaves de asa rotativa em voos rasantes, garantindo proteção contra ataques de baixa altitude; Radares de Vigilância – Detetam e fazem o *tracking* de aeronaves de asa fixa, aeronaves de asa rotativa e mísseis balísticos a médias e longas distâncias, fornecendo dados para a coordenação de defesa aérea. É assim, possível numerar alguns exemplos desta tipologia de sistemas, tais como: AN/MPQ-64 Sentinel (EUA) – Radar de médio alcance projetado para defesa contra mísseis de cruzeiro e aeronaves de baixa assinatura radar (Saab Group, 2023); Giraffe AMB (SWE): Radar móvel 3D, com capacidade de detecção de aeronaves, SANT e mísseis, utilizado no sistema NASAMS (Jiang et al., 2023); 9S36 (RUS): Integrado ao sistema S-300, rastreia alvos em longas distâncias para fornecer informações de intercetação (Morteza & Mohammad-Reza, 2023).

- **Radares de Perseguição e Conduta de Tiro**

Os radares de conduta de tiro fornecem dados precisos para sistemas de armas antiaéreas, permitindo intercetações eficazes. Diferente dos radares de vigilância, estes são especializados no *tracking* contínuo de alvos e fornecem orientação precisa para mísseis terra-ar e sistemas de artilharia. E como supracitado pelo EME (2016) esta

¹⁰Aeronaves de asa fixa refere-se à tipologia de aeronave comumente denominada “avião”. Aeronaves de asa rotativa refere-se à tipologia de aeronave comumente denominada “helicóptero”. No entanto, de forma a simplificar a aplicação da terminologia na análise experimental do algoritmo, simplificou-se a denominação ‘Aircraft’ (avião), ‘Helicopter’ (helicóptero) e ‘Drone’ (drone).

tipologia radar é o cerne comunicacional entre os radares de aviso local e vigilância para com os sistemas de armas através de meios C4ISR. Elenca-se assim os seguintes sistemas: AN/MPQ-65 (EUA): Utilizado no sistema Patriot e oferece pesquisa de alvos até 170 km (Sankaran & Fetter, 2020); 1S91 "*Straight Flush*" (RUS): Radar de conduta de tiro do sistema SA-6 "*Gainful*", com alcance de 75 km (Passbach, 2024); High Power Illuminator (EUA): Utilizado no sistema RIM-7 Sea Sparrow, fornece iluminação de alvos para mísseis semi-ativos (Charles, 1991).

- **Sensores Eletro-Ópticos e Infravermelhos**

Remetendo a uma componente primordial dos SAA, os sensores EO/IR sendo uma tipologia complementar dos sistemas radar que oferece uma visão mais detalhada do alvo, especialmente útil em ambientes onde o radar pode ser menos eficaz, como em operações de baixa altitude ou áreas com interferência eletrônica (Fontana & Di Lauro, 2022; Yang et al., 2020). Tais como as câmaras térmicas – detetam o calor gerado pelos motores das aeronaves e mísseis e as câmaras ópticas de alta resolução – identificam visualmente os alvos. Temos assim o sistema *Forward Looking Infra-Red* (FLIR) da qual equipa os sistemas Pantsir-S1 (RUS) e os sistemas infravermelhos da *Rafael Advanced Defense Systems*, que equipam o *Iron Dome* (ISR) (Jiang et al., 2023; Morteza & Mohammad-Reza, 2023; Rafael Advanced Defense Systems, 2025).

- **Sistemas de Detecção Acústica**

Embora menos comum na AAA, os sistemas acústicos detetam a presença de alvos baseando-se nas assinaturas sonoras. São úteis especialmente contra *SANT* de baixa assinatura térmica e aeronaves *stealth* a operar em ambientes urbanos ou com elevada densidade florestal. Um simples exemplo de aplicação deste sistema é nas redes de microfones para triangulação de aeronaves de asa rotativa ou *SANT stealth* (The MathWorks, 2024; Yang et al., 2020).

- **Sistemas Integrados e Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance**

De forma a reunir toda a informação, sobre a alçada das linhas guia da NATO, para que o utilizador consiga perceber o alvo aéreo através de um simples monitor, é necessário que a informação seja toda complicada, clara e concisa de forma que em determinados ambientes operacionais onde haja necessidade de C2 estrito, a informação

necessita de ser imediata e precisa (NATO Science and Technology Organization, 2022a). Assim sendo, os sistemas integrados que combinam informações de várias fontes (radares, sensores EO/IR, SIGINT) para fornecer uma visão a unificação do campo de batalha. Assim sendo, e de acordo com Thales (2021b) e Thales (2024b), é de facto melhorada a identificação de alvos e a redução do risco de fogo amigo. Temos ainda, que os sistemas de C4ISR, ou seja, Comando, Controlo, Comunicações, Computadores, Informações, Vigilância e Reconhecimento, permitem a difusão em tempo real de informações sobre alvos entre diferentes plataformas, como: Baterias de AAA; Postos de Comando e os sistemas de defesa integrados (NATO Science and Technology Organization, 2022b; Thales, 2024). Assim, de forma a reunir a informação, através da Figura nº 26, são adaptados através do PDE 3-37-00 Tática de Artilharia Antiaérea (EME, 2016) e com a literatura supracitada, os componentes e sistemas da AAA.

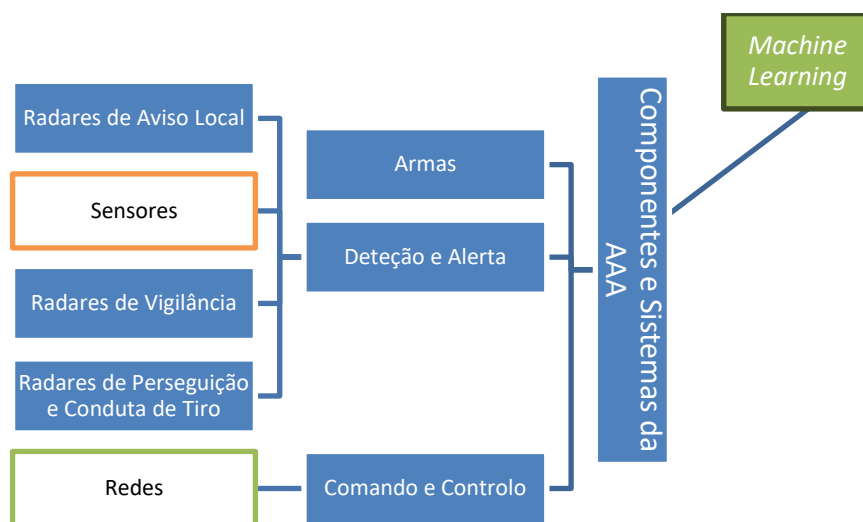


Figura nº 26 – Componentes e sistemas da AAA

Fonte: Adaptado de EME (2016)

APÊNDICE B – Algoritmo sem *Machine Learning*

```
% Definir o local final de dados
outputDir = 'Outputs_SemML';
if ~exist(outputDir, 'dir')
    mkdir(outputDir); % criar o diretório caso não exista
end

% Quantidade de vezes a executar
numRuns = 150;

% Criar documento .csv para dados
logFile = fullfile(outputDir, 'trajectoryLog.csv');

% Verificar se o documento .csv existe
if isfile(logFile)

% Lê dados existentes

logData = readtable(logFile, 'PreserveVariableNames', true);
else

% Cria o cabeçalho caso não existe o documento .csv
columnNames = {'RunNumber', 'MSE_X', 'MSE_Y', 'MSE_Z', 'RMSE_X', 'RMSE_Y', 'RMSE_Z', 'TrajectoryAccuracy', 'DistanceError'};

logData = table([], [], [], [], [], [], [], [], ...
'VariableNames', columnNames);

writetable(logData, logFile, 'WriteMode', 'overwrite', 'WriteRowNames', false);

end

% Loop
for runNumber = 1:numRuns

% N° de simulações radar
numSamples = 400;

time = linspace(0, 20, numSamples); % Time vector

% Determinar as labels dos alvos
targetLabels = categorical(randi([1, 3], numSamples, 1), [1, 2, 3], 'Drone', 'Aircraft', 'Helicopter');

% Criar trajetórias com base nas labels
x = zeros(numSamples, 1);
y = zeros(numSamples, 1);
z = zeros(numSamples, 1);

for i = 1:numSamples
    if targetLabels(i) == 'Drone'
        speedFactor = 0.8 + randn() * 0.1; % variação em velocidade aleatória
        x(i) = 100 + 50 * cos(speedFactor * time(i));
        y(i) = 200 + 50 * sin(speedFactor * time(i));
        z(i) = 50 + randn() * 5; % variar altitude aleatoriamente
    elseif targetLabels(i) == 'Aircraft'
        x(i) = 50 + 0.2 * time(i);
        y(i) = 100 + 0.3 * time(i);
        z(i) = 100 + 0.05 * time(i) + randn() * 2; % subida gradual
    elseif targetLabels(i) == 'Helicopter'
        x(i) = 150 + 30 * sin(0.5 * time(i) + randn() * 0.1);
        y(i) = 150 + 30 * cos(0.3 * time(i) + randn() * 0.1);
        z(i) = 70 + 10 * sin(0.7 * time(i) + randn() * 0.2);
    end
end

% Gaussian noise → realismo
noiseLevel = 5; % ajustável
x = x + noiseLevel * randn(size(x));
y = y + noiseLevel * randn(size(y));
z = z + noiseLevel * randn(size(z));

% cálculo de velocidade
vx = gradient(x, time);
```

```

vy = gradient(y, time);
vz = gradient(z, time);

% modelo de previsão simples

numPredictions = 50;
predictedTrajectory = zeros(numPredictions, 3);

currentX = x(end);
currentY = y(end);
currentZ = z(end);

currentVx = vx(end);
currentVy = vy(end);
currentVz = vz(end);

for i = 1:numPredictions

currentX = currentX + currentVx * (time(2) -
time(1));

currentY = currentY + currentVy * (time(2) -
time(1));

currentZ = currentZ + currentVz * (time(2) -
time(1));

predictedTrajectory(i, :) = [currentX, currentY,
currentZ];

end

% calcular Matriz de Confusão e Precisão

confusionMatrix = zeros(3);

for i = 1:numSamples

actualIdx = find(targetLabels(i) ==
categories(targetLabels));

predIdx = find(predicteDLabels(i) ==
categories(predicteDLabels));

confusionMatrix(actualIdx, predIdx) =
confusionMatrix(actualIdx, predIdx) + 1;

end

idAccuracy = sum(diag(confusionMatrix)) /
sum(confusionMatrix(:)) * 100;

% Erro → distância

```

```

mseX = mean((x(end-numPredictions+1:end) -
predictedTrajectory(:, 1)).^2);

mseY = mean((y(end-numPredictions+1:end) -
predictedTrajectory(:, 2)).^2);

mseZ = mean((z(end-numPredictions+1:end) -
predictedTrajectory(:, 3)).^2);

rmseX = sqrt(mseX);
rmseY = sqrt(mseY);
rmseZ = sqrt(mseZ);

distanceError = sqrt((x(end) -
predictedTrajectory(end, 1))^2 + ...
(y(end) - predictedTrajectory(end, 2))^2 + ...
(z(end) - predictedTrajectory(end, 3))^2);

% Calcular a precisão da trajetória

% Comparar a trajetória real vs previsão

actualTrajectory = [x(end-
numPredictions+1:end),
y(end-
numPredictions+1:end),
z(end-
numPredictions+1:end)];

diffTrajectory = actualTrajectory -
predictedTrajectory;

trajectoryError = sqrt(sum(diffTrajectory.^2, 2));
% Euclidean distance error for each prediction

trajectoryAccuracy = (1 - mean(trajectoryError) /
max(max(abs(actualTrajectory)))) * 100

% Guardar

figure;

c = confusionchart(confusionMatrix, {'Drone',
'Aircraft', 'Helicopter'});

c.Title = ['Confusion Matrix (Accuracy: ',
num2str(idAccuracy, '%.2f'), '%)'];

saveas(gcf, fullfile(outputDir,
['ConfusionMatrix_Run', num2str(runNumber),
'.png']));

close(gcf);

% Guardar

figure;

plot3(x, y, z, 'b-', 'LineWidth', 1.5); hold on;

```

```

plot3(predictedTrajectory(:, 1),
predictedTrajectory(:, 2), predictedTrajectory(:,
3), 'r--', 'LineWidth', 1.5);

legend('Actual Trajectory', 'Predicted
Trajectory');

xlabel('X Position'); ylabel('Y Position'); zlabel('Z
Position');

title(['Trajectory Plot (Run ',
num2str(runNumber), ')']);

grid on;

saveas(gcf, fullfile(outputDir,
['TrajectoryPlot_Run', num2str(runNumber),
'.png']));

close(gcf);

% Escreve métricas no documento .csv

logDataToAppend = table(runNumber, mseX,
mseY, mseZ, rmseX, rmseY, rmseZ,
trajectoryAccuracy, distanceError, ...

'VariableNames', {'RunNumber', 'MSE_X',
'MSE_Y', 'MSE_Z', 'RMSE_X', 'RMSE_Y',
'RMSE_Z', 'TrajectoryAccuracy',
'DistanceError'});

writetable(logDataToAppend, logFile,
'WriteMode', 'append', 'WriteRowNames', false);

metricsFile = fullfile(outputDir,
'MetricsLog.txt');

fid = fopen(metricsFile, 'a'); % Open file in
append mode

fprintf(fid, 'Run %d\n', runNumber);

fprintf(fid, 'Accuracy: %.2f%%\n',
trajectoryAccuracy);

fprintf(fid, 'Erro Distancia:
%.3f\n', distanceError);

fprintf(fid, 'Precisao de ID:
%.4f%%\n', idAccuracy);

%fprintf(fid, 'Confusion Matrix:\n');

fprintf(fid, '%d\t%d\t%d\n',
confusionMatrix);

fprintf(fid, 'RMSE (X, Y, Z): %.4f, %.4f,
%.4f\n', rmseX, rmseY, rmseZ);

```

```

fprintf(fid, '=====\n');

fclose(fid);

end

disp(['All results saved to ',outputDir]);

```

APÊNDICE C – Algoritmo com *Machine Learning*

```
for i = 1:l

    runNumber = i;

% Definir Diretório dos Outputs

    outputDir = 'Outputs_ComML'; % Set your
    desired folder name

    if ~exist(outputDir, 'dir')

        mkdir(outputDir); % Create the directory if it
        doesn't exist

    end

    numSamples = 1000; % Number of radar
    measurements

    time = linspace(0, 20, numSamples); % Time
    vector

% Determinar o nome das labels (Drone, Aircraft, Helicopter)

    targetLabels = categorical(randi([1, 3],
    numSamples, 1), [1, 2, 3], {'Drone', 'Aircraft',
    'Helicopter'});

% posições e velocidades iniciais

    x = zeros(numSamples, 1);

    y = zeros(numSamples, 1);

    z = zeros(numSamples, 1); % Add z-axis for 3D

    vx = zeros(numSamples, 1);

    vy = zeros(numSamples, 1);

    vz = zeros(numSamples, 1);

% Criar trajetórias aleatórias com base na label

    for i = 1:numSamples

        if targetLabels(i) == 'Drone'

% Movimento circular para simular um drone

% Variações aleatórias nas velocidades

            speedFactor = 0.8 + randn() * 0.1;

            x(i) = 100 + 50 * cos(speedFactor * time(i));

            y(i) = 200 + 50 * sin(speedFactor * time(i));
```

```
%Altitudes aleatórias z(i) = 50 + randn() * 5;

        elseif targetLabels(i) == 'Aircraft'

% Movimento linear para uma aeronave

            x(i) = 50 + 0.2 * time(i);

% Velocidade horizontal mais lenta

            y(i) = 100 + 0.3 * time(i);

% Aumento gradual na altura de forma a simular a ascensão de uma aeronave z(i) =
100 + 0.05 * time(i) + randn() * 2;

        elseif targetLabels(i) == 'Helicopter'

% Movimento oscilatório para um helicóptero

            x(i) = 150 + 30 * sin(0.5 * time(i) + randn()
            * 0.1); % Leve perturbação com alterações aleatórias

            y(i) = 150 + 30 * cos(0.3 * time(i) + randn()
            * 0.1);

            z(i) = 70 + 10 * sin(0.7 * time(i) + randn() *
            0.2);

        end

    end

% Ruído de Gauss para maior realismo nas leituras radar

    noiseLevel = 5;

    x = x + noiseLevel * randn(size(x));

    y = y + noiseLevel * randn(size(y));

    z = z + noiseLevel * randn(size(z));

    vx = gradient(x, time); % velocidade x

    vy = gradient(y, time); % Velocidade y

    vz = gradient(z, time); % Velocidade z

% Normalização numérica para simplicidade algébrica

    xNorm = (x - mean(x)) / std(x);

    yNorm = (y - mean(y)) / std(y);

    zNorm = (z - mean(z)) / std(z); % Normalize z-
    axis
```

<pre> vxNorm = (vx - mean(vx)) / std(vx); vyNorm = (vy - mean(vy)) / std(vy); vzNorm = (vz - mean(vz)) / std(vz); % Normalize vz % Features: [x, y, z, vx, vy, vz] featuresNorm = [xNorm, yNorm, zNorm, vxNorm, vyNorm, vzNorm]; labels = targetLabels; %% Treino, Validação e Treino (70-15-15) trainRatio = 0.7; valRatio = 0.15; testRatio = 0.15; numTrainSamples = floor(numSamples * trainRatio); numValSamples = floor(numSamples * valRatio); numTestSamples = numSamples - numTrainSamples - numValSamples; % Valores de treino trainFeatures = featuresNorm(1:numTrainSamples, :); trainLabels = labels(1:numTrainSamples); % Valores de validação valFeatures = featuresNorm(numTrainSamples+1:numTrainSa mples+numValSamples, :); valLabels = labels(numTrainSamples+1:numTrainSamples+n umValSamples); % Valores teste testFeatures = featuresNorm(numTrainSamples+numValSampl es+1:end, :); testLabels = labels(numTrainSamples+numValSamples+1:en d); % Treinar DT </pre>	<pre> decisionTreeModel = fitctree(trainFeatures, trainLabels); inputsNorm = [xNorm, yNorm, zNorm, vxNorm, vyNorm, vzNorm]'; outputsNorm = [xNorm(2:end), yNorm(2:end), zNorm(2:end)]'; trainInputs = con2seq(inputsNorm(:, 1:numTrainSamples-1)); % Sequence input for training trainOutputs = con2seq(outputsNorm(:, 1:numTrainSamples-1)); % Sequence output for training valInputs = con2seq(inputsNorm(:, 1:numTrainSamples+numValSamples+1:end- 1)); valOutputs = con2seq(outputsNorm(:, 1:numTrainSamples+numValSamples+1:end)); % Se existir um modelo ML será carregado o mesmo e re-treinado com novos dados senão é criado um novo if isfile('lstmNet3D.mat') load('lstmNet3D.mat', 'lstmNet'); disp('Loaded previous model. Fine-tuning the model...'); for i = 1:15 tic; % Definições do re-treino fineTuneOptions = trainingOptions('adam', ... 'InitialLearnRate', 1e-4, ... 'MaxEpochs', 10, ... % Fewer epochs for fine-tuning 'MiniBatchSize', 64, ... 'Shuffle', 'every-epoch', ... 'Plots', 'training-progress', ... 'Verbose', false); lstmNet = trainNetwork(trainInputs, trainOutputs, lstmNet.Layers, fineTuneOptions); trainingTime=toc; </pre>
--	--

```

save('lstmNet3D.mat', 'lstmNet');

disp(['Fine-tune iteration ', num2str(i), '
completed.']);

actualTrajectory =
[x(numTrainSamples+1:end),
y(numTrainSamples+1:end),
z(numTrainSamples+1:end)];

numTestSteps = size(actualTrajectory, 1);

predictedTrajectory = []; % Reset predicted
trajectory

currentStateNorm = [xNorm(end),
yNorm(end), zNorm(end), vxNorm(end),
vyNorm(end), vzNorm(end)];

while size(predictedTrajectory, 1) <
numTestSteps

    predictedOutput = predict(lstmNet,
currentStateNorm, 'MiniBatchSize', 1);

    predictedTrajectory = [predictedTrajectory;
predictedOutput'];

    currentStateNorm = [predictedOutput;
currentStateNorm(4:6)];

end

predictedTrajectory = predictedTrajectory .*
std([x, y, z]) + mean([x, y, z]);

%%

if size(predictedTrajectory, 1) >
size(actualTrajectory, 1)

    predictedTrajectory =
predictedTrajectory(end - size(actualTrajectory,
1) + 1:end, :);

elseif size(predictedTrajectory, 1) <
size(actualTrajectory, 1)

    predictedTrajectory = [padding;
predictedTrajectory];

end

if size(predictedTrajectory, 2) > 3

    predictedTrajectory = predictedTrajectory(:,
1:3); % Use only the first 3 columns (x, y, z)

elseif size(predictedTrajectory, 2) < 3

```

```

padding = zeros(size(predictedTrajectory,
1), 3 - size(predictedTrajectory, 2));

predictedTrajectory = [predictedTrajectory,
padding];

end

finalActual = actualTrajectory(end, :);

finalPredicted = predictedTrajectory(end, :);

actualTrajectory =
[x(numTrainSamples+1:end),
y(numTrainSamples+1:end),
z(numTrainSamples+1:end)];

if size(actualTrajectory, 1) >
size(predictedTrajectory, 1)

% fixar as trajetórias ao mesmo número de
previsões

    actualTrajectory =
actualTrajectory(1:size(predictedTrajectory, 1),
:);

elseif size(actualTrajectory, 1) <
size(predictedTrajectory, 1)

    padding = NaN(size(predictedTrajectory,
1) - size(actualTrajectory, 1),
size(actualTrajectory, 2));

    actualTrajectory = [actualTrajectory;
padding];

end

predictedTrajectory = [];

currentStateNorm = [xNorm(end),
yNorm(end), zNorm(end), vxNorm(end),
vyNorm(end), vzNorm(end)];

while size(predictedTrajectory, 1) <
size(actualTrajectory, 1)

    predictedOutput = predict(lstmNet,
currentStateNorm, 'MiniBatchSize', 1);

    predictedTrajectory = [predictedTrajectory;
predictedOutput'];

    disp('Predicted Output:');

    disp(predictedOutput');

```

```

        currentStateNorm = [predictedOutput;
        currentStateNorm(4:6)];

    end

% garantir as 3 colunas XYZ

    predictedTrajectory = predictedTrajectory(:,
    1:3);

% De-normalizar os dados

    predictedTrajectory = predictedTrajectory .*
    std([x, y, z]) + mean([x, y, z]);

% Desenho das trajetorias

    figure;

    hold on;

    plot3(actualTrajectory(:, 1), actualTrajectory(:,
    2), actualTrajectory(:, 3), ...

        'b-', 'DisplayName', 'Actual Trajectory');

    if ~isempty(predictedTrajectory)

        plot3(predictedTrajectory(:, 1),
        predictedTrajectory(:, 2), predictedTrajectory(:,
        3), ...

            'r--', 'DisplayName', 'Predicted Trajectory');

    else

        disp('Predicted Trajectory is empty!');

    end

    plot3(actualTrajectory(end, 1),
    actualTrajectory(end, 2), actualTrajectory(end,
    3), ...

        'bo', 'MarkerFaceColor', 'r', 'MarkerSize', 8,
        'DisplayName', 'Final Actual Position');

    plot3(predictedTrajectory(end, 1),
    predictedTrajectory(end, 2),
    predictedTrajectory(end, 3), ...

        'ro', 'MarkerFaceColor', 'g', 'MarkerSize', 8,
        'DisplayName', 'Final Predicted Position');

% Labels e títulos

    xlabel('X');

    ylabel('Y');

    zlabel('Z');

```

```

    title('Trajectory Prediction (Actual vs.
    Predicted)');

    legend;

    grid on;

    view(3);

    hold off;

    saveas(gcf, fullfile(outputDir,
    ['TrajectoryPlot_Run', num2str(runNumber),
    '.png']));

    close(gcf);

% MSE e RMSE

    mseX = mean((actualTrajectory(:, 1) -
    predictedTrajectory(:, 1)).^2);

    mseY = mean((actualTrajectory(:, 2) -
    predictedTrajectory(:, 2)).^2);

    mseZ = mean((actualTrajectory(:, 3) -
    predictedTrajectory(:, 3)).^2);

    rmseX = sqrt(mseX);

    rmseY = sqrt(mseY);

    rmseZ = sqrt(mseZ);

    disp(['MSE_X: ', num2str(mseX)]);

    disp(['MSE_Y: ', num2str(mseY)]);

    disp(['MSE_Z: ', num2str(mseZ)]);

    disp(['RMSE_X: ', num2str(rmseX)]);

    disp(['RMSE_Y: ', num2str(rmseY)]);

    disp(['RMSE_Z: ', num2str(rmseZ)]);

%% validação k-fold

    k = 10; % número de folds

    cv = cvpartition(trainLabels, 'Kfold', k);

    cvMSE = zeros(k, 1);

    for fold = 1:k

        trainIdx = training(cv, fold);

        valIdx = test(cv, fold);

        cvTrainFeatures = trainFeatures(trainIdx,
        :);

```

```

cvTrainLabels = trainLabels(trainIdx);

cvValFeatures = trainFeatures(valIdx, :);

cvValLabels = trainLabels(valIdx);

cvDecisionTree =
fitctree(cvTrainFeatures, cvTrainLabels);

cvPredictedDLLabels =
predict(cvDecisionTree, cvValFeatures);

cvAccuracy = sum(cvPredictedDLLabels ==
cvValLabels) / numel(cvValLabels) * 100;

cvPredictedNumeric =
double(cvPredictedDLLabels);

cvValNumeric = double(cvValLabels);

% MSE

cvMSE(fold) =
mean((cvPredictedNumeric - cvValNumeric).^2);

%cvMSE(fold) = mean((cvPredictedDLLabels -
double(cvValLabels)).^2);

disp(['Fold ', num2str(fold), ' Accuracy: ',
num2str(cvAccuracy), '%']);

end

% media MSE

avgMSE = mean(cvMSE);

disp(['Average CV MSE: ',
num2str(avgMSE)]);

%%

finalActual = actualTrajectory(end, :);

finalPredicted = predictedTrajectory(end, :);

disp(['Final Actual Position:']);

disp(['X: ', num2str(finalActual(1)), ', Y: ',
num2str(finalActual(2)), ', Z: ',
num2str(finalActual(3))]);

disp(['Final Predicted Position:']);

disp(['X: ', num2str(finalPredicted(1)), ', Y: ',
num2str(finalPredicted(2)), ', Z: ',
num2str(finalPredicted(3))]);

%Distância

```

```

finalPositionError = norm(finalPredicted -
finalActual);

% calcula a precisão

accuracy = (1 / (1 + finalPositionError));

% precisão em percentagem

accuracyPercentage = (1 / (1 +
finalPositionError)) * 100;

% Mostra o erro final de posição e a precisão da mesma

disp(['Final Position Error (Euclidean
distance in X, Y, Z): ',
num2str(finalPositionError)]);

disp(['Accuracy: ',
num2str(accuracyPercentage), '%']);

%%

%previsões com base na DT e dados validados

predictedClasses =
predict(decisionTreeModel, valFeatures);

% calcula a precisão da classificação

classificationAccuracy =
sum(predictedClasses == valLabels) /
numel(valLabels) * 100;

disp('Predicted Aircraft Type:');

disp(predictedClasses(1)); % Display the
predicted class for the first sample

% Matriz de confusão

figure;

confusionchart(valLabels,
predictedClasses); % Compare actual vs
predicted classes

title(['Decision Tree Classifier - Accuracy: ',
num2str(classificationAccuracy), '%']);

saveas(gcf, fullfile(outputDir,
['ConfusionMatrix_Run', num2str(runNumber),
'.png']));

close(gcf);

% Mostra a percentagem de precisão

```

```

        disp(['Classification Accuracy: ',
num2str(classificationAccuracy), '%']);
%%%"
%%"

        logFile = 'trainingLog.csv'; % Set the correct
path for the log file

        if isfile(logFile)

% Lê o ficheiro onde estão guardados os dados
das simulações

            logData = readtable(logFile,
'PreserveVariableNames', true);

        else

% Create the log file with headers if it
doesn't exist

            columnNames = {'RunNumber', 'MSE_X',
'MSE_Y', 'MSE_Z', 'RMSE_X', 'RMSE_Y',
'RMSE_Z', 'TrajectoryAccuracy',
'AircraftIDAccuracy', 'TrainingTime'};

            logData = table([], [], [], [], [], [], [], [],
[], [], ...

                'VariableNames', columnNames);

% Escreve no doc csv novas colunas e linhas ao
ficheiro

            writetable(logData, logFile, 'WriteMode',
'overwrite', 'WriteRowNames', false);

        end

        runNumber = height(logData) + 1;

        logDataToAppend = table(runNumber,
mseX, mseY, mseZ, rmseX, rmseY, rmseZ,
accuracyPercentage, classificationAccuracy,
trainingTime, ...

            'VariableNames', {'RunNumber',
'MSE_X', 'MSE_Y', 'MSE_Z', 'RMSE_X',
'RMSE_Y', 'RMSE_Z', 'TrajectoryAccuracy',
'AircraftIDAccuracy', 'TrainingTime'});

% Adiciona nova coluna ao ficheiro csv

        writetable(logDataToAppend, logFile,
'WriteMode', 'append', 'WriteRowNames', false);

        disp('Training metrics logged in CSV.');
```

```

        %end
    else

        lastTrainedModel = [];

%Todo o processo acima repete-se caso não
exista já criado e treinado um modelo de ML

        for i = 1:1

            disp('No existing model found. Training a
new one...');

            numSamples = 1000; % Number of radar
measurements

            time = linspace(0, 20, numSamples);

            targetLabels = categorical(randi([1, 3],
numSamples, 1), [1, 2, 3], {'Drone', 'Aircraft',
'Helicopter'});

            x = zeros(numSamples, 1);

            y = zeros(numSamples, 1);

            z = zeros(numSamples, 1); % Add z-axis for
3D

            vx = zeros(numSamples, 1);

            vy = zeros(numSamples, 1);

            vz = zeros(numSamples, 1); % Velocity in z

            for i = 1:numSamples

                if targetLabels(i) == 'Drone'

                    speedFactor = 0.8 + randn() * 0.1; %
Random variation in speed

                    x(i) = 100 + 50 * cos(speedFactor *
time(i));

                    y(i) = 200 + 50 * sin(speedFactor *
time(i));

                    z(i) = 50 + randn() * 5;

                    elseif targetLabels(i) == 'Aircraft'

                        x(i) = 50 + 0.2 * time(i); % Slower
horizontal speed

                        y(i) = 100 + 0.3 * time(i); % Gradual y-
direction motion

                        z(i) = 100 + 0.05 * time(i) + randn() *
2;
                end
            end
        end
    end
```

```

elseif targetLabels(i) == 'Helicopter'

    x(i) = 150 + 30 * sin(0.5 * time(i) +
randn() * 0.1);

    y(i) = 150 + 30 * cos(0.3 * time(i) +
randn() * 0.1);

    z(i) = 70 + 10 * sin(0.7 * time(i) +
randn() * 0.2);

    end

end

noiseLevel = 5;

x = x + noiseLevel * randn(size(x));

y = y + noiseLevel * randn(size(y));

z = z + noiseLevel * randn(size(z));

vx = gradient(x, time); % Velocity in x

vy = gradient(y, time); % Velocity in y

vz = gradient(z, time); % Velocity in z

xNorm = (x - mean(x)) / std(x);

yNorm = (y - mean(y)) / std(y);

zNorm = (z - mean(z)) / std(z); % Normalize
z-axis

vxNorm = (vx - mean(vx)) / std(vx);

vyNorm = (vy - mean(vy)) / std(vy);

vzNorm = (vz - mean(vz)) / std(vz); %
Normalize vz

% Features: [x, y, z, vx, vy, vz]

featuresNorm = [xNorm, yNorm, zNorm,
vxNorm, vyNorm, vzNorm];

labels = targetLabels;

trainRatio = 0.7;

valRatio = 0.15;

testRatio = 0.15;

numTrainSamples = floor(numSamples *
trainRatio);

numValSamples = floor(numSamples *
valRatio);

```

```

numTestSamples = numSamples -
numTrainSamples - numValSamples;

trainFeatures =
featuresNorm(1:numTrainSamples, :);

trainLabels = labels(1:numTrainSamples);

valFeatures =
featuresNorm(numTrainSamples+1:numTrainSa
mples+numValSamples, :);

valLabels =
labels(numTrainSamples+1:numTrainSamples+n
umValSamples);

testFeatures =
featuresNorm(numTrainSamples+numValSampl
es+1:end, :);

testLabels =
labels(numTrainSamples+numValSamples+1:en
d);

decisionTreeModel =
fitctree(trainFeatures, trainLabels);

inputsNorm = [xNorm, yNorm, zNorm,
vxNorm, vyNorm, vzNorm]'; % Input features as
sequences

outputsNorm = [xNorm(2:end),
yNorm(2:end), zNorm(2:end)]';

trainInputs = con2seq(inputsNorm(:,
numTrainSamples-1)); trainOutputs =
con2seq(outputsNorm(:,1:numTrainSamples-1));

valInputs = con2seq(inputsNorm(:,
1:numTrainSamples+numValSamples+1:end-
1));

valOutputs = con2seq(outputsNorm(:,
1:numTrainSamples+numValSamples+1:end));
% Validation output

tic;

% Definir a rede LSTM do novo modelo ML

inputSize = size(inputsNorm, 1); % 6
features [x, y, z, vx, vy, vz]

outputSize = 3; % Predict next
position [x_next, y_next, z_next]

numHiddenUnits = 20;

layers = [ sequenceInputLayer(inputSize)

```

```

lstmLayer(numHiddenUnits, 'OutputMode',
'sequence')

    dropoutLayer(0.2) % Dropout
layer for regularization

    lstmLayer(numHiddenUnits,
'OutputMode', 'sequence')

    dropoutLayer(0.2) % Another
Dropout layer

    fullyConnecteDLayer(100) % Fully
connected layer with 100 units

    dropoutLayer(0.2) % Dropout
layer to prevent overfitting

    fullyConnecteDLayer(outputSize)

    regressionLayer];

    bilstmLayer(100, 'OutputMode',
'sequence');

% Opções de treino do modelo

options = trainingOptions('adam', ...
'MaxEpochs', 100, ...
'InitialLearnRate', 0.001, ...
'LearnRateSchedule', 'piecewise', ...
'LearnRateDropFactor', 0.5, ...
'LearnRateDropPeriod', 50, ...
'GradientThreshold', 1, ...
'MiniBatchSize', 16, ...
'ValidationData', {valInputs, valOutputs},
'ValidationFrequency', 100, ...
'Verbose', false, ...
'Plots', 'training-progress');

% Treino do novo modelo

newModel = trainNetwork(trainInputs,
trainOutputs, layers, options);

trainingTime = toc;

% Apagar modelo antigo caso exista

if ~isempty(lastTrainedModel)

```

```

    clear lastTrainedModel;

end

% Update do último modelo

lastTrainedModel = newModel;

disp(['Training iteration ', num2str(i), '
completed.']);

% Após o ciclo, lastTrainedModel contem o
modelo ML mais atualizado

lstmNet = lastTrainedModel;

% A cada execução o modelo é guardado

save('lstmNet3D.mat', 'lstmNet');

actualTrajectory =
[x(numTrainSamples+1:end),
y(numTrainSamples+1:end),
z(numTrainSamples+1:end)];

numTestSteps = size(actualTrajectory, 1); %
Match the length of the actual trajectory

predictedTrajectory = [];

currentStateNorm = [xNorm(end), yNorm(end),
zNorm(end), vxNorm(end), vyNorm(end),
vzNorm(end)];

while size(predictedTrajectory, 1) <
numTestSteps

    predictedOutput = predict(lstmNet,
currentStateNorm, 'MiniBatchSize', 1);

    predictedTrajectory =
[predictedTrajectory; predictedOutput];

    currentStateNorm = [predictedOutput;
currentStateNorm(4:6)];

end

if size(predictedTrajectory, 1) >
size(actualTrajectory, 1)

    predictedTrajectory =
predictedTrajectory(end - size(actualTrajectory,
1) + 1:end, :);

elseif size(predictedTrajectory, 1) <
size(actualTrajectory, 1)

```

```

padding = NaN(size(actualTrajectory, 1) -
size(predictedTrajectory, 1),
size(predictedTrajectory, 2));

predictedTrajectory = [padding;
predictedTrajectory];

end

if size(predictedTrajectory, 2) > 3

predictedTrajectory =
predictedTrajectory(:, 1:3); % Use only the first
3 columns (x, y, z)

elseif size(predictedTrajectory, 2) < 3

padding = zeros(size(predictedTrajectory,
1), 3 - size(predictedTrajectory, 2));

predictedTrajectory =
[predictedTrajectory, padding];

end

finalActual = actualTrajectory(end, :);

finalPredicted = predictedTrajectory(end, :);

actualTrajectory =
[x(numTrainSamples+1:end),
y(numTrainSamples+1:end),
z(numTrainSamples+1:end)];

predictedTrajectory = predictedTrajectory .*
std([x, y, z]) + mean([x, y, z]);

if size(predictedTrajectory, 1) >
size(actualTrajectory, 1)

predictedTrajectory =
predictedTrajectory(1:size(actualTrajectory, 1),
:);

elseif size(predictedTrajectory, 1) <
size(actualTrajectory, 1)

padding = NaN(size(actualTrajectory, 1) -
size(predictedTrajectory, 1), 3);

predictedTrajectory =
[predictedTrajectory; padding];

end

predictedTrajectory = [];

```

```

currentStateNorm = [xNorm(end),
yNorm(end), zNorm(end), vxNorm(end),
vyNorm(end), vzNorm(end)];

while size(predictedTrajectory, 1) <
size(actualTrajectory, 1)

predictedOutput = predict(lstmNet,
currentStateNorm, 'MiniBatchSize', 1);

predictedTrajectory =
[predictedTrajectory; predictedOutput];

disp('Predicted Output:');

disp(predictedOutput);

currentStateNorm = [predictedOutput;
currentStateNorm(4:6)];

end

predictedTrajectory = predictedTrajectory(:,
1:3);

predictedTrajectory = predictedTrajectory .*
std([x, y, z]) + mean([x, y, z]);

figure;

hold on;

hold on;

plot3(actualTrajectory(:, 1),
actualTrajectory(:, 2), actualTrajectory(:, 3), ...
'b-', 'DisplayName', 'Actual Trajectory');

if ~isempty(predictedTrajectory)

plot3(predictedTrajectory(:, 1),
predictedTrajectory(:, 2), predictedTrajectory(:,
3), ...
'r--', 'DisplayName', 'Predicted
Trajectory');

else

disp('Predicted Trajectory is empty!');

end

plot3(actualTrajectory(end, 1),
actualTrajectory(end, 2), actualTrajectory(end,
3), ...
'bo', 'MarkerFaceColor', 'r', 'MarkerSize',
8, 'DisplayName', 'Final Actual Position');

```

```

    plot3(predictedTrajectory(end, 1),
    predictedTrajectory(end, 2),
    predictedTrajectory(end, 3), ...

        'ro', 'MarkerFaceColor', 'g', 'MarkerSize',
    8, 'DisplayName', 'Final Predicted Position');

    xlabel('X');

    ylabel('Y');

    zlabel('Z');

    title('Trajectory Prediction (Actual vs.
    Predicted)');

    legend;

    grid on;

    view(3);

    hold off;

    runNumber = height(logData) + 1;

    saveas(gcf, fullfile(outputDir,
    ['TrajectoryPlot_Run', num2str(runNumber),
    '.png']));

    close(gcf);

    %%

    mseX = mean((actualTrajectory(:, 1) -
    predictedTrajectory(:, 1)).^2);

    mseY = mean((actualTrajectory(:, 2) -
    predictedTrajectory(:, 2)).^2);

    mseZ = mean((actualTrajectory(:, 3) -
    predictedTrajectory(:, 3)).^2);

    rmseX = sqrt(mseX);

    rmseY = sqrt(mseY);

    rmseZ = sqrt(mseZ);

    disp(['MSE_X: ', num2str(mseX)]);

    disp(['MSE_Y: ', num2str(mseY)]);

    disp(['MSE_Z: ', num2str(mseZ)]);

    disp(['RMSE_X: ', num2str(rmseX)]);

    disp(['RMSE_Y: ', num2str(rmseY)]);

    disp(['RMSE_Z: ', num2str(rmseZ)]);

    k = 10;

```

```

    cv = cvpartition(trainLabels, 'KFold', k);

    cvMSE = zeros(k, 1);

    for fold = 1:k

        trainIdx = training(cv, fold);

        valIdx = test(cv, fold);

        cvTrainFeatures = trainFeatures(trainIdx,
    :);

        cvTrainLabels = trainLabels(trainIdx);

        cvValFeatures = trainFeatures(valIdx, :);

        cvValLabels = trainLabels(valIdx);

        cvDecisionTree =
    fitctree(cvTrainFeatures, cvTrainLabels);

        cvPredictedDLLabels =
    predict(cvDecisionTree, cvValFeatures);

        cvAccuracy = sum(cvPredictedDLLabels ==
    cvValLabels) / numel(cvValLabels) * 100;

        cvPredictedNumeric =
    double(cvPredictedDLLabels);

        cvValNumeric = double(cvValLabels);

        cvMSE(fold) =
    mean((cvPredictedNumeric - cvValNumeric).^2);

        cvMSE(fold) = mean((cvPredictedDLLabels
    - double(cvValLabels)).^2);

        disp(['Fold ', num2str(fold), ' Accuracy: ',
    num2str(cvAccuracy), '%']);

        end

        avgMSE = mean(cvMSE);

        disp(['Average CV MSE: ',
    num2str(avgMSE)]);

    %%

    finalActual = actualTrajectory(end, :);

    finalPredicted = predictedTrajectory(end, :);

    disp(['Final Actual Position:']);

    disp(['X: ', num2str(finalActual(1)), ', Y: ',
    num2str(finalActual(2)), ', Z: ',
    num2str(finalActual(3))]);

```

```

disp(['Final Predicted Position:']);

disp(['X: ', num2str(finalPredicted(1)), ', Y: ',
num2str(finalPredicted(2)), ', Z: ',
num2str(finalPredicted(3))]);

finalPositionError = norm(finalPredicted -
finalActual);

accuracy = (1 / (1 + finalPositionError));

accuracyPercentage = (1 / (1 +
finalPositionError)) * 100;

disp(['Final Position Error (Euclidean
distance in X, Y, Z): ',
num2str(finalPositionError)]);

disp(['Accuracy: ',
num2str(accuracyPercentage), '%']);

%%

predictedClasses =
predict(decisionTreeModel, valFeatures);

classificationAccuracy =
sum(predictedClasses == valLabels) /
numel(valLabels) * 100;

disp('Predicted Aircraft Type:');

disp(predictedClasses(1)); % Display the
predicted class for the first sample

figure;

confusionchart(valLabels,
predictedClasses);

title(['Decision Tree Classifier - Accuracy: ',
num2str(classificationAccuracy), '%']);

disp(['Classification Accuracy: ',
num2str(classificationAccuracy), '%']);

saveas(gcf, fullfile(outputDir,
['ConfusionMatrix_Run', num2str(runNumber),
'.png']));

close(gcf);

%%

%%

logFile = 'trainingLog.csv'; % Set the correct
path for the log file

if isfile(logFile)

```

```

logData = readtable(logFile,
'PreserveVariableNames', true);

else

columnNames = {'RunNumber', 'MSE_X',
'MSE_Y', 'MSE_Z', 'RMSE_X', 'RMSE_Y',
'RMSE_Z', 'TrajectoryAccuracy',
'AircraftIDAccuracy', 'TrainingTime'};

logData = table([], [], [], [], [], [], [], [],
[], ...

'VariableNames',
columnNames);

writetable(logData, logFile, 'WriteMode',
'overwrite', 'WriteRowNames', false);

end

runNumber = height(logData) + 1;

logDataToAppend = table(runNumber,
mseX, mseY, mseZ, rmseX, rmseY, rmseZ,
accuracyPercentage, classificationAccuracy,
trainingTime, ...

'VariableNames',
{'RunNumber', 'MSE_X', 'MSE_Y', 'MSE_Z',
'RMSE_X', 'RMSE_Y', 'RMSE_Z',
'TrajectoryAccuracy', 'AircraftIDAccuracy',
'TrainingTime'});

writetable(logDataToAppend, logFile,
'WriteMode', 'append', 'WriteRowNames', false);

disp('Training metrics logged in CSV.');
```

```

metricsFile = fullfile(outputDir,
'MetricsLog.txt');

fid = fopen(metricsFile, 'a'); % Open file in
append mode

fprintf(fid, 'Run %d\n', runNumber);

fprintf(fid, 'Accuracy: %.2f%%\n',
accuracyPercentage);

fprintf(fid, 'Erro Distancia:
%.3f\n', finalPositionError);

fprintf(fid, 'Precisao de ID:
%.4f%%\n', classificationAccuracy);

%fprintf(fid, 'Confusion Matrix:\n');
```

```

        %fprintf(fid, '%d\t%d\t%d\n',
confusionMatrix');

        fprintf(fid, 'RMSE (X, Y, Z): %.4f, %.4f,
%.4f\n', rmseX, rmseY, rmseZ);

        fprintf(fid, '=====\n');

        fclose(fid);

        end

end

metricsFile = fullfile(outputDir,
'MetricsLog.txt');

fid = fopen(metricsFile, 'a');

fprintf(fid, 'Run %d\n', runNumber);

```

```

        fprintf(fid, 'Accuracy: %.2f%%\n',
accuracyPercentage);

        fprintf(fid, 'Erro Distancia:
%.3f\n', finalPositionError);

        fprintf(fid, 'Precisao de ID:
%.4f%%\n', classificationAccuracy);

        %fprintf(fid, 'Confusion Matrix:\n');

        %fprintf(fid, '%d\t%d\t%d\n',
confusionMatrix');

        fprintf(fid, 'RMSE (X, Y, Z): %.4f, %.4f,
%.4f\n', rmseX, rmseY, rmseZ);

        fprintf(fid, '=====\n');

        fclose(fid);

end

```

APÊNDICE D – Resultados Análise Descritiva

Estatística Descritiva						
	Algoritmo	RMSE_X	RMSE_Y	RMSE_Z	%id	Erro (m)
N	Com ML	100	100	100	100	100
	Sem ML	100	100	100	100	100
Omisso	Com ML	0	0	0	0	0
	Sem ML	0	0	0	0	0
Média	Com ML	65.9	59.8	30.5	96.7	44.0
	Sem ML	1233	1539	743	33.4	3908
Erro-padrão da média	Com ML	1.60	1.62	0.818	0.338	2.43
	Sem ML	100	118	51.8	0.251	226
Mediana	Com ML	64.7	54.5	26.7	98.0	46.0
	Sem ML	1225	1735	694	33.4	4521
Desvio-padrão	Com ML	16.0	16.2	8.18	3.38	24.3
	Sem ML	1001	1182	518	2.51	2257
Mínimo	Com ML	10.0	11.3	21.3	84.0	3.00
	Sem ML	48.1	38.1	24.6	26.5	165
Máximo	Com ML	97.9	98.3	50.9	100	103
	Sem ML	2920	4650	1980	40.5	8876
Assimetria	Com ML	-0.510	-0.0239	0.650	-1.53	-0.00145
	Sem ML	0.178	0.474	0.547	-0.224	-0.231
Erro-padrão da Assimetria	Com ML	0.241	0.241	0.241	0.241	0.241
	Sem ML	0.241	0.241	0.241	0.241	0.241
Curtose	Com ML	1.11	0.209	-0.978	1.79	-1.04
	Sem ML	-1.63	-0.466	-0.548	0.645	-0.905
Erro-padrão da Curtose	Com ML	0.478	0.478	0.478	0.478	0.478
	Sem ML	0.478	0.478	0.478	0.478	0.478
W de Shapiro-Wilk	Com ML	0.904	0.935	0.869	0.805	0.957
	Sem ML	0.861	0.922	0.946	0.984	0.924
p Shapiro-Wilk	Com ML	< .001	< .001	< .001	< .001	0.002
	Sem ML	< .001	< .001	< .001	0.261	< .001

Figura nº 27 – Análise descritiva

APÊNDICE E – Resultados do teste Shapiro-Wilk

Pressupostos

Teste à Normalidade (Shapiro-Wilk)

	W	p
RMSE_X	0.891	< .001
RMSE_Y	0.860	< .001
RMSE_Z	0.847	< .001
%id	0.917	< .001
Erro (m)	0.869	< .001

Nota. Um p-value pequeno sugere a violação do pressuposto da normalidade

Figura nº 28 – Resultados dos pressupostos

APÊNDICE F – Resultados do teste *U* de Mann-Whitney

Teste t para amostras independentes

Teste t para amostras independentes

		Estatística	p
RMSE_X	U de Mann-Whitney	610	< .001
RMSE_Y	U de Mann-Whitney	461	< .001
RMSE_Z	U de Mann-Whitney	186	< .001
%id	U de Mann-Whitney	0	< .001
Erro (m)	U de Mann-Whitney	0	< .001

Nota. $H_a: \mu_{\text{Com ML}} \neq \mu_{\text{Sem ML}}$

Figura nº 29 – Resultados do teste para amostras independentes

APÊNDICE G – Resultados Regressão Linear

Regressão Linear

Medidas de Ajustamento do Modelo

Modelo	R	R ²
1	0.993	0.987

Coefficientes do Modelo - Erro (m)

Preditor	Estimativas	Erro-padrão	t	p
Intercepto ^a	-443.542	677.4975	-0.655	0.513
Algoritmo:				
Sem ML – Com ML	594.777	449.6773	1.323	0.188
RMSE_X	0.914	0.0312	29.304	< .001
RMSE_Y	1.272	0.0353	36.039	< .001
RMSE_Z	0.754	0.0773	9.763	< .001
%id	3.394	6.9979	0.485	0.628

^a Representa o nível de referência

Figura nº 30 – Regressão linear do erro em distância em função dos RMSE

APÊNDICE H – Gráficos Representativos

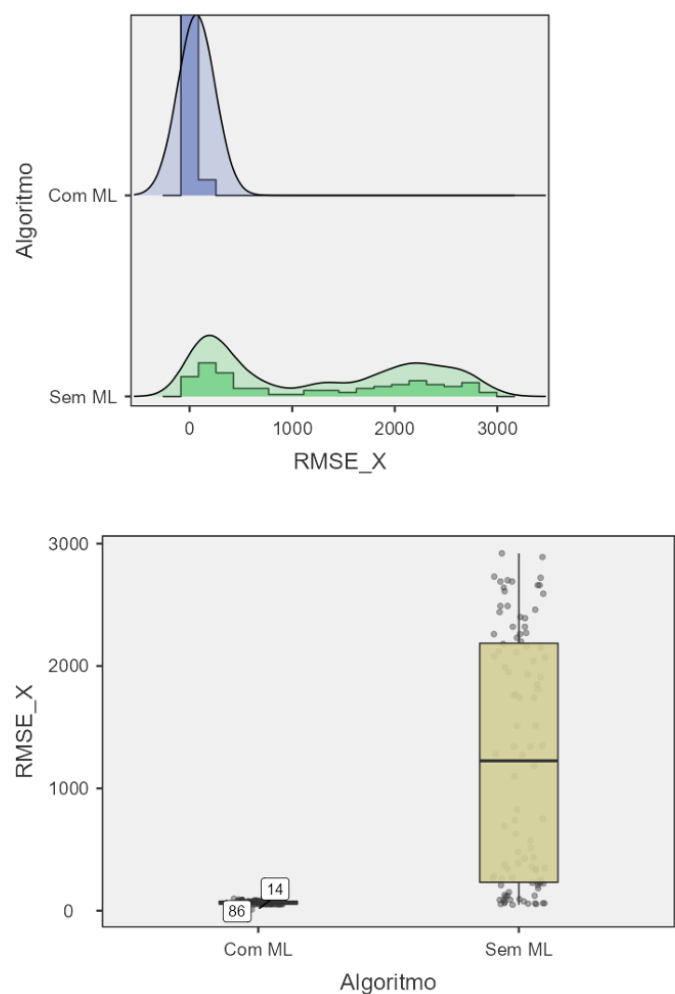


Figura nº 31 - Densidade e Boxplots do RMSE em x

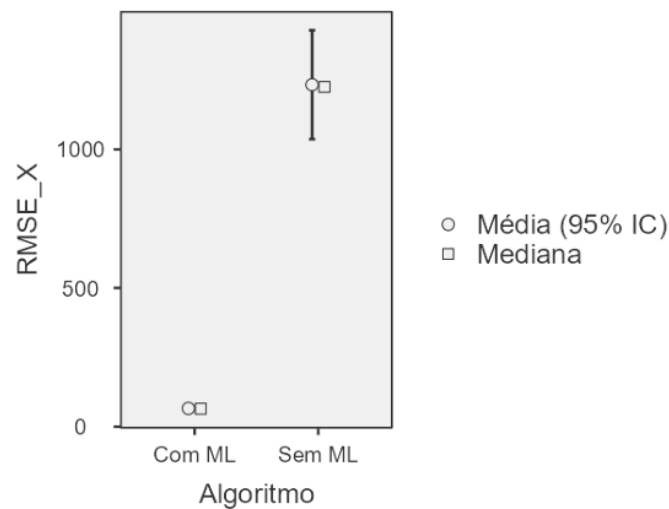


Figura nº 32 – Média e mediana do RMSE em x

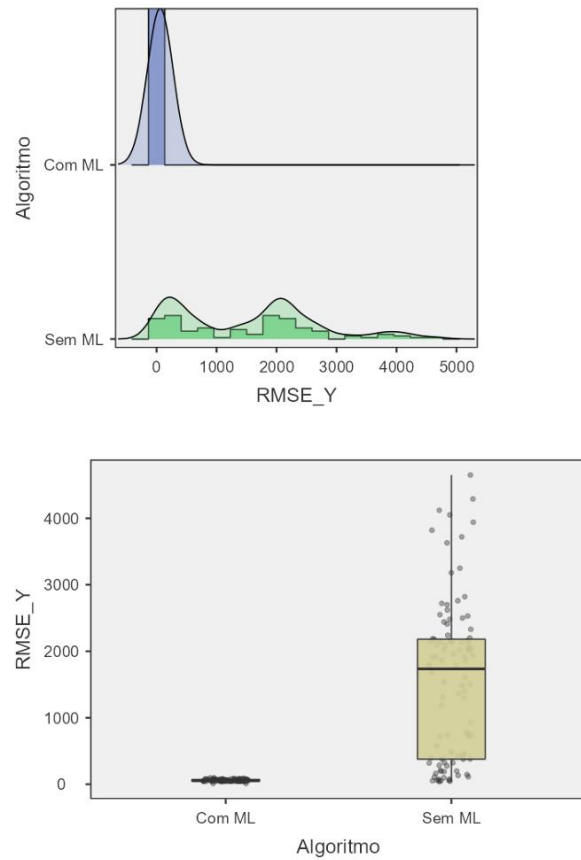


Figura nº 33 - Densidade e Boxplots do RMSE em y

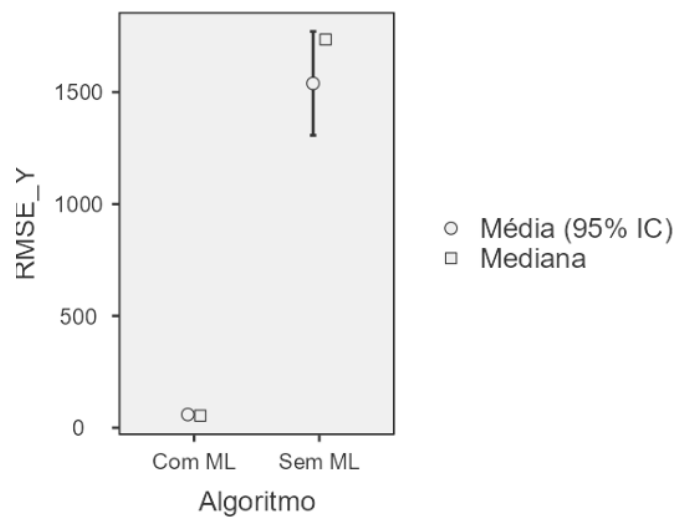


Figura nº 34 – Média e mediana do RMSE em y

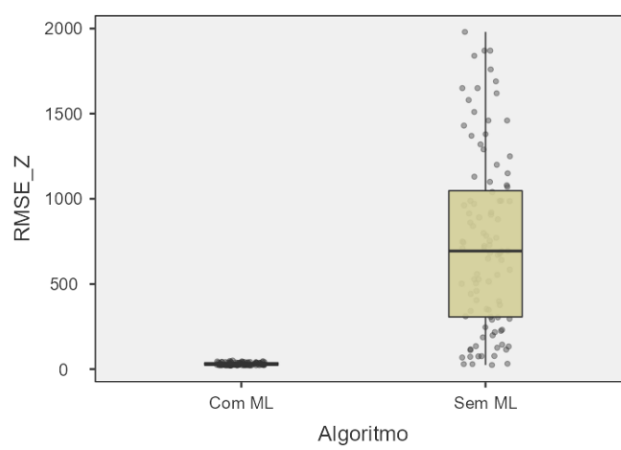
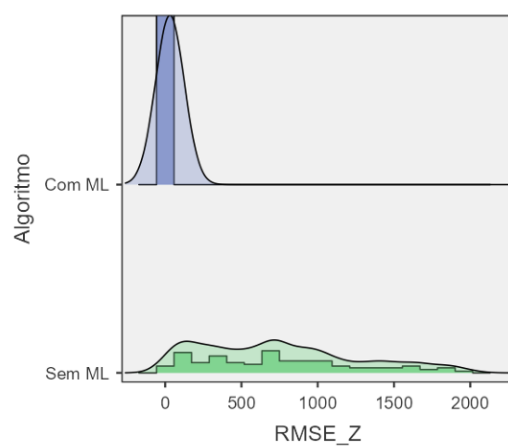


Figura nº 35 - Densidade e Boxplots do RMSE em z

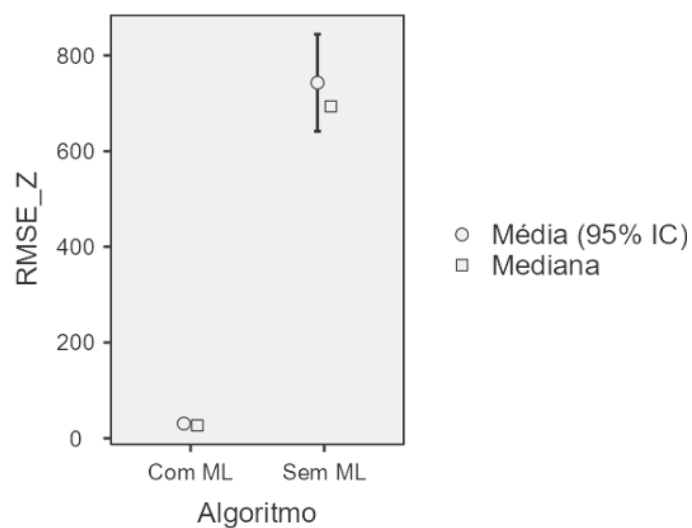


Figura nº 36 – Média e mediana do RMSE em z

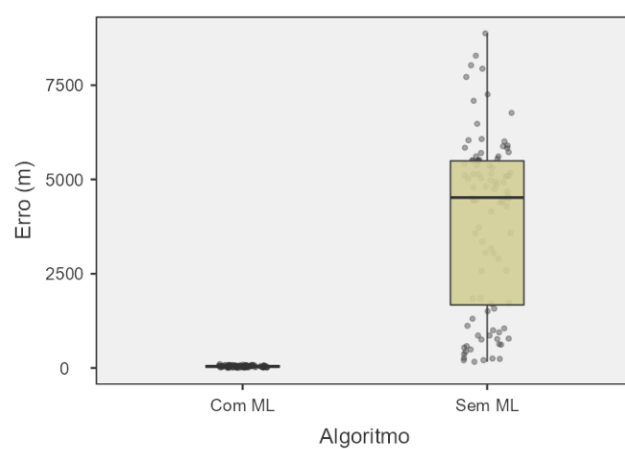
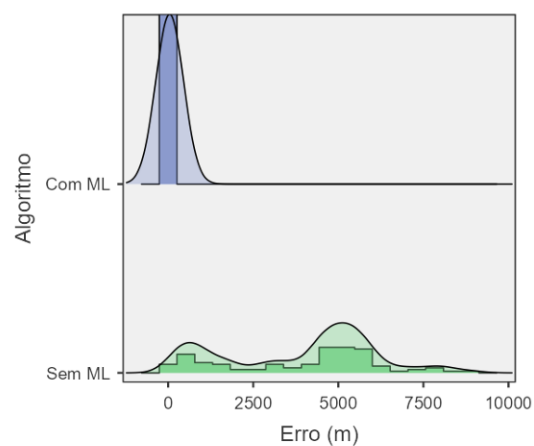


Figura nº 37 - Densidade e Boxplots dos erros em distância

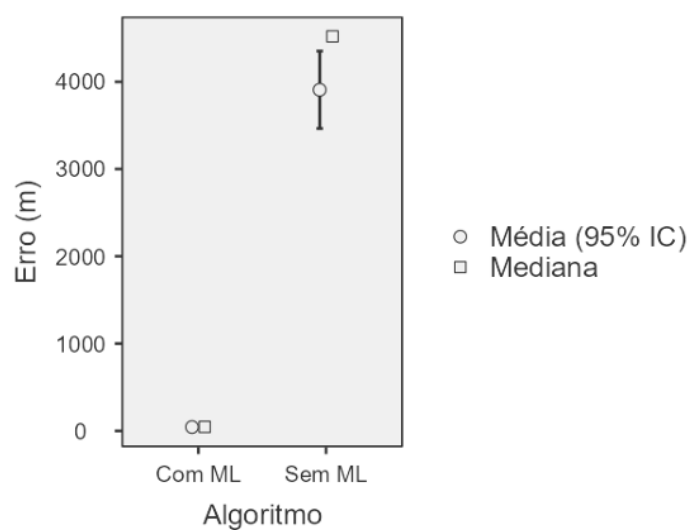


Figura nº 38 – Média e mediana dos erros em distância

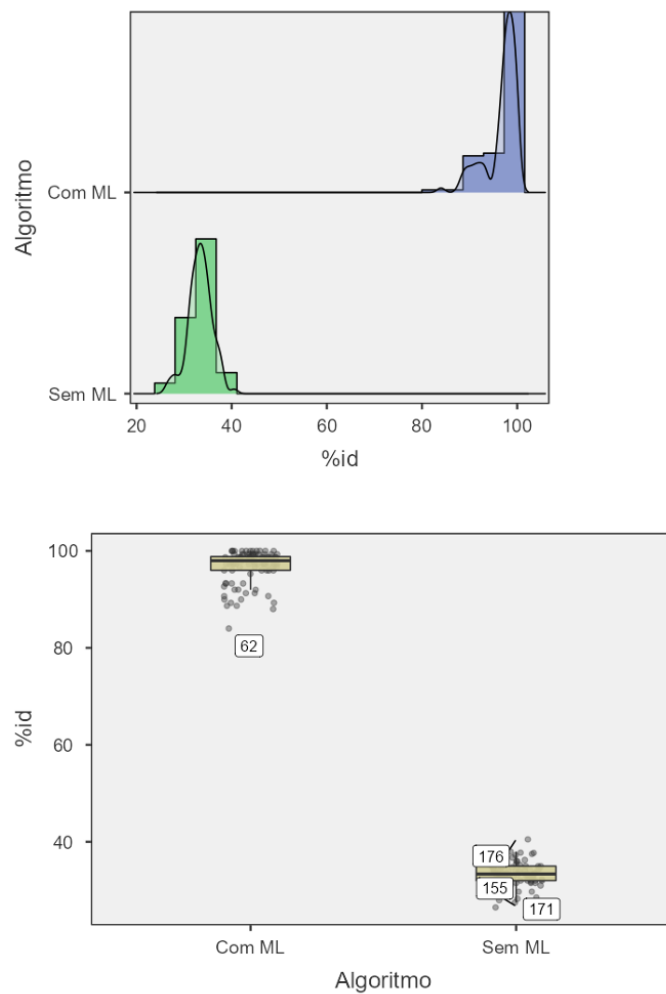


Figura nº 39 – Densidade e Boxplots da Identificação da ameaça

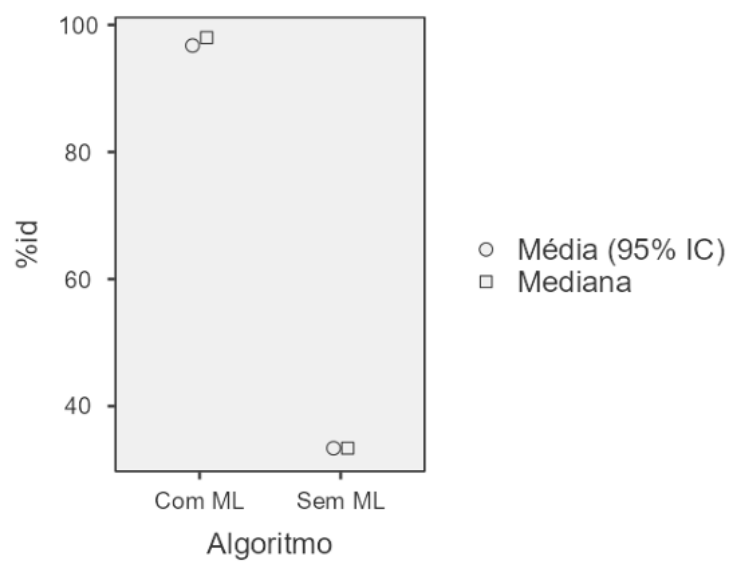


Figura nº 40 – Média e mediana da Identificação de ameaça

APÊNDICE I – Gráficos de Dispersão

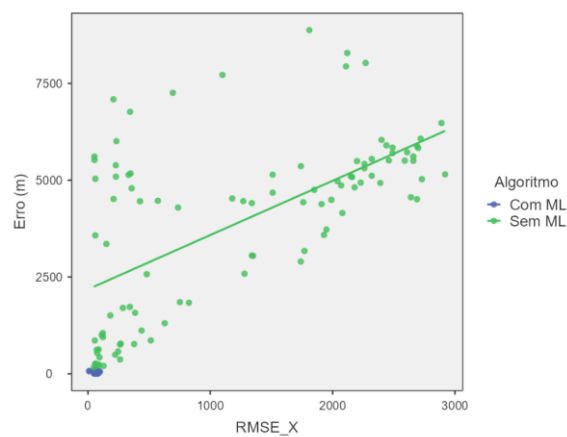


Figura nº 41 – RMSE dos valores em x

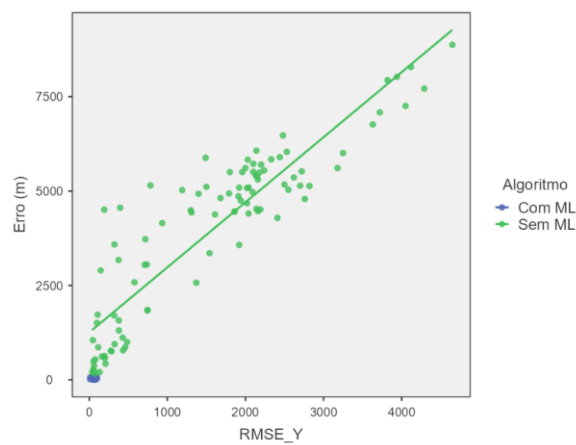


Figura nº 42 - RMSE dos valores em y

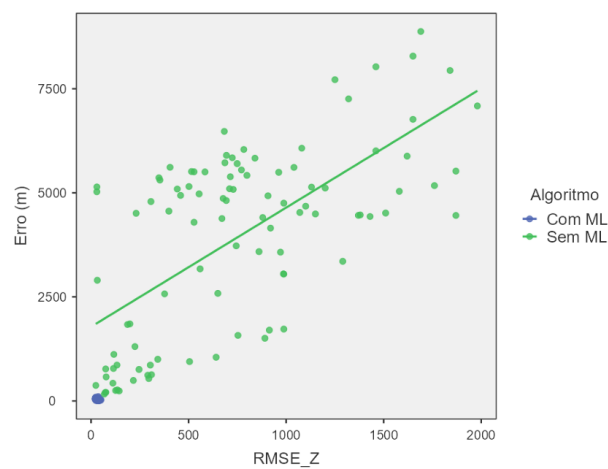


Figura nº 43 – RMSE dos valores em z

APÊNDICE J – Guião de Entrevista

Tabela nº 7 – Resumo enquadramento da entrevista e objectivos

Tópicos Entrevista	Relação com os Objectivos	Justificação
1.1.Sobre o entrevistado	Objectivo Geral	Introduz o entrevistado e sua experiência com ML na AAA
1.2.Sobre os sistemas em geral	Objectivo Específico 1	Identifica quais SAA utilizam ML
2.1. Implementação de ML	Objectivo Específico 1	Compreende como o ML está a ser aplicado nos sistemas da Thales
2.2. Desafios Técnicos	Objectivo Específico 2	Identifica dificuldades práticas e técnicas na implementação de ML
2.3. Interoperabilidade	Objectivo Específico 2	Avalia as limitações dos SAA com ML
3.1. Operações Militares	Objectivo Específico 3	Examina os benefícios do ML na AAA, especialmente no Exército Português
3.2. Futuro dos sistemas	Objectivo Específico 3	Avalia a necessidade de sistemas AAA que integrem ML e seu potencial futuro
4. Aspectos Éticos e Regulatórios	Objectivo Específico 2 e 3	Analisa riscos, regulamentações e impactos éticos do ML na defesa
5. Conclusão	Objectivo Geral	Permite um encerramento e insights adicionais sobre o impacto do ML

Tabela nº 8 – Questões entrevista

1.1.Sobre o entrevistado	1. "Pode partilhar a sua posição na Thales e a sua experiência com tecnologias de aquisição de alvos?" 2. "Qual é o papel da Thales no desenvolvimento de sistemas avançados para defesa aérea?" 3. "A Thales tem experiência em desenvolver SAA com ML especificamente para clientes militares portugueses?"
---------------------------------	---

1.2.Sobre os sistemas em geral	1. "Quais são os principais desafios enfrentados atualmente nos SAA?" 2. "Como os avanços na inteligência artificial estão a influenciar o setor da defesa aérea?" 3. "Quais são os principais SAA que atualmente utilizam ML?" 4. "Em Portugal, há iniciativas no Exército para incorporar ML nos sistemas AAA?"
2.1. Implementação de ML	1. "Como a Thales tem integrado o ML nos seus SAA?" 2. "Pode fornecer exemplos concretos de como ML melhora a precisão e eficácia no reconhecimento e classificação de alvos?" 3. "Existe alguma tecnologia baseada em ML da Thales que já esteja operacional em forças armadas da NATO?"
2.2. Desafios Técnicos	1. "Quais foram os maiores desafios técnicos ao implementar ML nesses sistemas?" 2. "Como lidar com dados incompletos ou de baixa qualidade no treino de modelos?" 3. "Quais são as limitações do ML na AAA em relação aos sistemas tradicionais?"
2.3. Interoperabilidade	1. "Os sistemas baseados em ML são compatíveis com infraestruturas de defesa já existentes?" 2. "Como é que os sistemas Thales garantem que a aprendizagem automática atenda aos requisitos de diferentes ambientes operacionais?" 3. "Existe risco de viés ou erro no reconhecimento de alvos ao utilizar ML?"
3.1. Operações Militares	1. "Pode partilhar exemplos concretos de como a aplicação de ML impactou diretamente operações ou exercícios militares?" 2. "Quais são os benefícios observados na resposta a ameaças aéreas, como drones e mísseis?" 3. "Houve algum caso documentado em que ML tenha melhorado a eficiência operacional em combate real?"

3.2. Futuro dos sistemas	1. "Considera que ML poderá substituir completamente a supervisão humana na aquisição de alvos no futuro próximo?" 2. "Qual é a visão da Thales sobre a integração de IA em sistemas totalmente autónomos?" 3. "O Exército Português demonstrou interesse em desenvolver sistemas AAA com maior autonomia usando ML?"
4. Aspectos Éticos e Regulatórios	1. "De que forma a Thales aborda questões éticas relacionadas ao uso de ML na defesa?" 2. "Há regulamentos internacionais que impactam o desenvolvimento ou uso de sistemas baseados em ML?" 3. "Como garantir que sistemas baseados em ML tomem decisões confiáveis e transparentes?"
5. Conclusão	1. "Gostaria de adicionar algo que ache relevante sobre o impacto do ML nos SAA?"

APÊNDICE K – Termo de Consentimento Informado

Termo de Consentimento Informado

No âmbito da Dissertação de Mestrado em Ciências Militares na especialidade de Artilharia, ministrado pela Academia Militar de Lisboa, sob orientação do Tenente-Coronel Artilharia Miguel Maldonado, sobre o **Impacto do *Machine Learning* nos Sistemas de Aquisição de Alvos da Artilharia Antiaérea**.

Com esta investigação pretende-se então efetuar uma caracterização do impacto do ML nos sistemas de aquisição de alvos, compreender quais as potencialidade e limitações face aos sistemas em vigor, e, por fim, avaliar qual a necessidade de empregar Sistemas de Aquisição de Alvos, na AAA, que façam uso do ML, por parte de militares que serão alvos de entrevistas. Através de uma revisão de literatura aprofundada foi possível compreender que o ML como ferramenta de análise de um grande conjunto de informação – big data analysis – e com capacidade de auto-aprendizagem através da análise experimental, é possível elevar a capacidade dos sistemas de aquisição de alvos da AAA de um modo em que o processo de deteção/aquisição de alvos e posteriormente, comando e controlo dos sistemas de armas da AAA, se torne num processo menos moroso, mais eficiente e preciso.

Para a realização deste estudo, é necessário a recolha de dados através de uma entrevista semiestruturada que será gravada por áudio.

Os dados deste estudo são anónimos e confidenciais.

Desde já agradeço a sua participação e disponibilidade em participar no presente estudo.

Desta forma, aceito participar neste estudo e permito a utilização dos dados que de forma voluntária forneço, confiando em que apenas serão utilizados para esta investigação e nas garantias de confidencialidade e anonimato que me são dadas pela investigadora.

Nome:

Assinado por: **Tiago Filipe Abreu Moura Guedes**
Num. de Identificação: 12336612
Data: 2025.02.21 11:33:16+00'00'

Assinatura:

Data:

Este documento é composto de 2 páginas e feito em duplicado: uma via para o investigador, outra para a pessoa que consente.

Gregory Carvalho

Asp Art

18303120

Assinado por: **GREGORY ANTÓNIO PORTILLO
DOURADO DE CARVALHO**
Num. de Identificação: 14846342
Data: 2025.02.21 09:22:53+00'00'

Figura nº 44 – Termo de Consentimento Informado