

Classificação de Projéteis Balísticos em Radares de Contrabateria com Aprendizado de Máquina

Eduardo Nepomuceno da Rocha, Daniel Miñan de Oliveira Crús, Julio Cesar Duarte e José Antonio Apolinário Jr.

Resumo—Sistemas de radar de contrabateria operam em ambientes dinâmicos e incertos, nos quais a rápida identificação de ameaças é importante para a tomada de decisão. Nesse contexto, torna-se fundamental a classificação confiável de projéteis balísticos sob severas restrições computacionais. Este trabalho compara algoritmos clássicos de aprendizado de máquina e redes neurais na classificação de projéteis de artilharia utilizando a base RAMP-RT. São avaliados o *Random Forest*, o *Gradient Boosting*, o perceptron multicamadas e a rede convolucional unidimensional, considerando acurácia, precisão, revocação, F1-score e custo computacional. Os resultados indicam acurácia entre 87% e 91% para modelos de aprendizado de máquina e 83% e 90% para redes neurais. A análise evidencia a complementaridade entre a eficiência computacional e a capacidade de extração de padrões, fornecendo diretrizes para uma seleção efetiva dos modelos.

Palavras-Chave—Aprendizado de máquina, Redes neurais, Classificação balística, Radar de contrabateria

Abstract—Counter-battery radar systems operate in dynamic and uncertain environments, where the rapid identification of threats is important for decision-making. In this context, reliable classification of ballistic projectiles under strict computational constraints becomes essential. This work compares classical machine learning algorithms and neural networks for the classification of artillery projectiles using the RAMP-RT dataset. The evaluated models include Random Forest, Gradient Boosting, multilayer perceptron, and one-dimensional convolutional neural networks, considering accuracy, precision, recall, F1-score, and computational cost. Results indicate accuracies ranging from 87% to 91% for machine learning models and 83% to 90% for neural networks. The analysis highlights the complementarity between computational efficiency and pattern extraction capability, providing guidelines for effective model selection.

Keywords—Machine learning, Neural networks, Ballistic classification, Counter-battery radar

I. INTRODUÇÃO

A classificação de projéteis balísticos em radares de contrabateria demanda a identificação rápida do tipo de munição para estimar os pontos de lançamento e de impacto. A variedade de calibres, regimes dinâmicos e ruídos nos sensores torna a tarefa complexa, exigindo métodos robustos, generalizáveis e computacionalmente eficientes.

Filtros não lineares, como variantes do Filtro de Kalman, estimam parâmetros cinemáticos e rastreiam trajetórias [1, 2]; contudo, não foram concebidos para classificação. Nesse contexto, técnicas de aprendizado de máquina podem ser aplicadas

para discriminar projéteis a partir de atributos cinemáticos e da seção reta radar (RCS, *Radar Cross Section*) [3].

O avanço das redes neurais convolucionais (CNNs, do inglês Convolutional Neural Networks) ampliou a capacidade de modelar representações complexas de dados [4, 5]. Entretanto, algoritmos clássicos de aprendizado seguem relevantes pela simplicidade de implementação e pela menor exigência de processamento [6, 2]. Assim, torna-se pertinente a comparação sistemática entre diferentes abordagens de aprendizado de máquina quanto ao custo computacional e ao desempenho.

Este estudo avalia abordagens de aprendizado de máquina na classificação de projéteis balísticos usando a base Rocket, Artillery, and Mortar Projectile Radar Targets (RAMP-RT) [1] e compara modelos clássicos e arquiteturas de aprendizado profundo quanto à acurácia, robustez a ruídos e custo computacional, com foco em aplicações para radares de contrabateria.

A contribuição deste trabalho consiste na classificação de projéteis a partir de atributos físicos das trajetórias, em contraste com abordagens frequentes que utilizam séries temporais brutas. Os resultados indicam que métodos baseados em árvores de decisão alcançam elevado desempenho com baixo custo computacional, enquanto redes neurais, como o Perceptron Multicamadas (MLP) e a Rede Neural Convolucional Unidimensional (CNN1D), apresentaram desempenho competitivo e alta capacidade discriminativa, embora com maior exigência de recursos. Além disso, o trabalho apresenta uma análise comparativa sistemática entre diferentes modelos de aprendizado de máquina, destacando o equilíbrio entre desempenho e custo computacional. As abordagens mostram-se complementares, oferecendo alternativas adequadas a diferentes restrições de processamento e requisitos de eficiência.

II. TRABALHOS RELACIONADOS

A literatura sobre classificação de projéteis contempla desde modelos de menor complexidade, como o SVM (*Support Vector Machine*) [7], até arquiteturas de redes neurais profundas. Estudos iniciais exploraram redes neurais artificiais e modelos ocultos de Markov, combinando parâmetros cinemáticos e RCS, com resultados promissores em cenários simulados [2, 8]. Posteriormente, modelos baseados em árvores de decisão, como o *Random Forest*, destacaram-se pelo baixo custo computacional e pela simplicidade de implementação em sistemas embarcados [6, 9]. Com a evolução das CNNs, passaram a predominar abordagens baseadas em sinais de radar e em trajetórias temporais, com alta acurácia e robustez diante de ruídos [10, 11].

Os estudos recentes sobre classificação de projéteis concentram-se predominantemente na utilização direta das

E. N. Rocha, PPGEE, Instituto Militar de Engenharia, Rio de Janeiro-RJ, e-mail: eduardo.nepomuceno@ime.eb.br; D. M. O. Crús, PPGED, Instituto Militar de Engenharia, Rio de Janeiro-RJ, e-mail: minan.daniel@ime.eb.br; J. A. Apolinário Jr., PPGEE, Instituto Militar de Engenharia, Rio de Janeiro-RJ, e-mail: apolin@ime.eb.br; J. C. Duarte, PPGSC, Instituto Militar de Engenharia, Rio de Janeiro-RJ, duarte@ime.eb.br

séries temporais de radar [11, 12, 13], favorecendo arquiteturas convolucionais que se beneficiam dos avanços em hardware de alto desempenho [14, 15]. Em contraste, tem sido explorada em menor escala a classificação a partir de atributos físicos estimados das trajetórias, que constitui uma alternativa metodológica que possibilita a implementação de algoritmos clássicos de classificação [2, 6]. Observa-se, contudo, a escassez de investigações comparativas que avaliem sistematicamente o desempenho e o custo computacional de modelos clássicos em relação a redes neurais, lacuna que fundamenta a pesquisa apresentada neste trabalho.

III. PROCEDIMENTO EXPERIMENTAL E BASE DE DADOS

Os vetores de características foram normalizados para reduzir as diferenças de escala entre os atributos. Devido ao desbalanceamento das classes, aplicaram-se técnicas de sobre-amostragem com replicação de amostras e geração artificial de dados, tais como o SMOTE (*Synthetic Minority Oversampling Technique*)[16], que combate o desequilíbrio por meio da geração sintética de amostras da classe minoritária.

Os modelos avaliados foram *Random Forest*, *Gradient Boosting*, MLP e CNN1D [17], escolhidos por sua capacidade de representação e pela ampla utilização no aprendizado de máquina. Adicionalmente, o KNN (*K-nearest Neighbors*)[18] foi incluído pela simplicidade e rapidez, já que não requer uma fase explícita de treinamento, enquanto o SVM[7] foi considerado pela eficácia na separação de classes e na flexibilidade no uso de diferentes kernels. O desempenho foi medido por acurácia, precisão, *recall* (também encontrado como “revocação” na literatura) e *F1-score*. Foram considerados cenários de validação cruzada *k-fold* com dados sintéticos, divisão 70% de treinamento e 30% de teste nas bases sintética e combinada (união das amostras sintéticas e reais), além de teste cego, em que os modelos são treinados com dados simulados e avaliados em dados reais.

A base RAMP-RT reúne medições sintéticas e reais de projéteis balísticos observados por radar, cujos estados são estimados pelo *Square-Root Cubature Kalman Filter* (SR-CKF) [1], sem conhecimento prévio do coeficiente balístico. A partir da filtragem, extraem-se parâmetros cinemáticos da trajetória e estatísticas de RCS, que compõem o conjunto de atributos utilizado nas etapas subsequentes. No total, a base contém 5.048 vetores de características derivados de 1.650 trajetórias simuladas e de 103 reais, obtidas em campanhas de tiro conduzidas pelo Centro Tecnológico do Exército e pelo Centro de Avaliação do Exército.

A base contempla nove classes de projéteis: morteiros (ML, MM, MH), artilharia (GL, GH), foguetes (RL, RM, RH) e indefinido (UN).¹ Cada amostra inclui os seguintes oito atributos: ângulo de tiro, velocidade inicial, alcance, altitude máxima, velocidade média, coeficiente balístico, média da RCS e Diferença Máxima de Espalhamento (MSD, do inglês *Maximum Scatter Difference*). As medições incorporam ruídos de alcance, ângulo e espalhamento, compatíveis com

as condições operacionais de Relação Sinal-Ruído (SNR, do inglês *Signal-to-Noise Ratio*). Na base real, há cinco instâncias classificadas como indefinidas (UN), correspondentes a casos em que o SR-CKF não convergiu ao estimar os estados das trajetórias.

TABELA I
QUANTIDADE DE DADOS POR CLASSE NAS BASES SINTÉTICA E REAL

Base	ML	MM	RM	MH	RL	GL	GH	RH	UN	Total
Sintética	961	876	700	655	633	621	283	282	37	5048
Real	-	57	-	41	-	-	-	-	5	103

Os dados sintéticos foram gerados por um simulador de trajetória de ponto-massa próprio, considerando a gravidade e o arrasto aerodinâmico, seguido da simulação de detecção radar com ruídos nas medições. A partir dessas detecções, aplica-se filtragem de Kalman para estimar parâmetros da trajetória e extrair atributos. As trajetórias reais foram obtidas pelo Sistema Transportável de Rastreo de Engenheiros em Voo, em disparos se afastando do radar. O conjunto real é limitado pelo alto custo e pela complexidade operacional das campanhas de tiro instrumentadas, resultando em poucas amostras e ausência de algumas classes, gerando um desbalanceamento natural considerado no procedimento experimental.²

IV. EXPERIMENTOS COM MODELOS NÃO NEURAI

Esta seção apresenta os resultados dos modelos clássicos de classificação aplicados à RAMP-RT. Inicialmente, são discutidos KNN e SVM como referências, seguidos dos métodos baseados em árvores de decisão, detalhados nas subseções.

Primeiramente, escolheu-se o algoritmo KNN como modelo de referência por sua simplicidade computacional e à ausência de fase explícita de treinamento. Inicialmente, foi avaliado por meio do treinamento na base sintética e por teste cego na base real, conforme descrito no procedimento experimental.

Observou-se degradação significativa no teste cego, indicando baixa generalização dos dados simulados para os dados reais. Testes com *oversampling* elevaram as métricas, mas exigem cautela, pois a ampliação artificial de classes pode distorcer a distribuição original. A divisão 70/30 nas bases sintética e combinada confirmou a sensibilidade do KNN à distribuição dos dados: o desempenho permaneceu elevado na base sintética, mas caiu quando dados reais foram incorporados ao treinamento.

Na sequência, o SVM foi treinado na base sintética com o kernel RBF (*Radial Basis Function*) e fator *Gamma* mantido no padrão *Scale*. Para os modelos KNN e SVM, os hiperparâmetros *k* e *C* foram determinados por busca em grade (*grid search*) com validação cruzada estratificada. Como a base real não inclui todas as classes da sintética, a avaliação considerou apenas as classes comuns a ambas, descartando as ausentes nos dados reais.

A aplicação de *oversampling* degradou o desempenho, possivelmente devido ao deslocamento do hiperplano de separação. O kernel linear também foi testado para verificar a

¹Na primeira letra da nomenclatura adotada, “M” indica morteiros, “G” projéteis de canhões e obuseiros, e “R” foguetes. A segunda letra corresponde a calibres pequenos (“L”), médios (“M”) e grandes (“H”).

²A base RAMP-RT está disponível em <https://github.com/DanielMinan/RAMP-RT>.

existência de separabilidade aproximadamente linear entre as classes. Para facilitar a interpretação, aplicou-se o PCA (*Principal Component Analysis*) para reduzir a dimensionalidade e permitir visualização em duas dimensões. A Figura 1 mostra o hiperplano obtido no espaço reduzido para os dados reais.

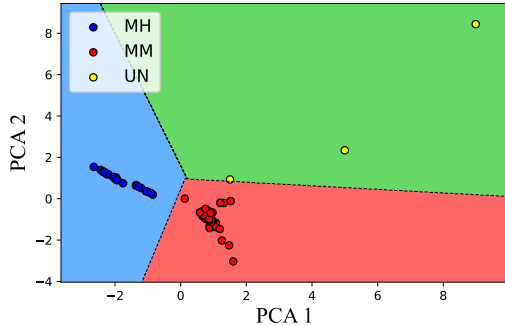


Fig. 1. Hiperplano do SVM para treino na base real com PCA.

Já o Random Forest mostrou alta capacidade de generalização no teste cego, treinado na base sintética com validação cruzada e busca aleatória de hiperparâmetros. As métricas computacionais indicam viabilidade para aplicações embarcadas, considerando tempo de treinamento, predição e uso de memória.

O uso de SMOTE nas classes minoritárias elevou a acurácia para 0,971 no teste cego, reforçando a robustez do método. Nos experimentos com divisão 70/30, o *Random Forest* manteve métricas superiores a 80% tanto na base sintética quanto na base combinada, evidenciando desempenho consistente entre os dois cenários.

Finalmente, a técnica de *Gradient Boosting* foi implementada utilizando o XGBoost, versão otimizada baseada em árvores de decisão treinadas sequencialmente para minimizar o erro residual. O desempenho foi comparável ao do *Random Forest*, confirmando a consistência das métricas em cenários equivalentes.

A Figura 2 apresenta a importância relativa dos atributos no *Gradient Boosting*, calculada pelo ganho médio de redução da função de perda nas divisões das árvores. Essa análise foi realizada apenas neste modelo, pois as árvores são construídas sequencialmente para reduzir o erro residual, fazendo com que as métricas reflitam diretamente a contribuição de cada variável. Em contraste, no *Random Forest* as árvores são geradas de forma independente e com elevado grau de aleatoriedade, de modo que as medidas tendem a refletir reduções médias de impureza, apresentando maior variabilidade entre execuções.

A aplicação do SMOTE teve impacto limitado, sugerindo que o modelo já captura os padrões das classes minoritárias. A interpretabilidade baseou-se na medida de importância fornecida pelo XGBoost. Embora técnicas como SHAP (*Shapley Additive Explanations*) [19] ofereçam maior detalhamento da contribuição individual de cada variável, sua aplicação exigiria cálculos repetidos para cada instância e atributo, resultando em um custo computacional elevado, sem benefícios proporcionais ao objetivo desta análise.

Os resultados das Tabelas II e III evidenciam diferenças claras entre os algoritmos. O KNN apresentou alta precisão,

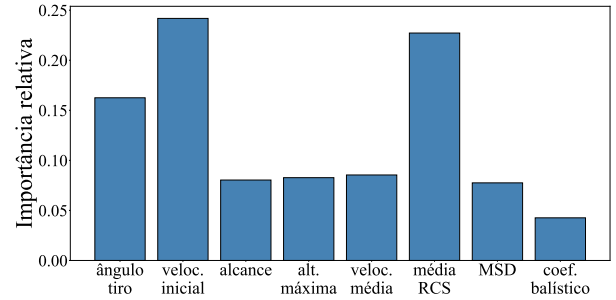


Fig. 2. Importância dos atributos obtida com a técnica *eXtreme Gradient Boosting* treinada na base sintética.

mas acurácia e *recall* reduzidos, indicando dificuldade ao se identificar a classe minoritária. O SVM foi ligeiramente superior, mas ainda limitado no *recall*. Entre os modelos baseados em árvores, o *Random Forest* e o XGBoost alcançaram métricas acima de 0,87, com maior equilíbrio entre acurácia, precisão e *recall*. O *Random Forest* obteve os melhores valores em todas as métricas, confirmando sua robustez, enquanto o XGBoost apresentou desempenho próximo.

Na comparação computacional, o *Random Forest* demandou menos tempo de treino, enquanto o XGBoost foi mais eficiente em termos de memória e do tempo de predição. Assim, ambos se mostram competitivos em acurácia, mas com diferenças relevantes em custo computacional que devem ser consideradas conforme o contexto de aplicação.

TABELA II
DESEMPENHO DOS ALGORITMOS DE APRENDIZADO DE MÁQUINA COM TREINAMENTO NA BASE SINTÉTICA E TESTE NA BASE REAL.

Modelo	Acurácia	Precisão	Recall	F1-score
KNN	0,641	0,981	0,641	0,735
SVM	0,680	0,964	0,680	0,793
Random Forest	0,913	1,000	0,913	0,948
XGBoost	0,874	0,981	0,874	0,919

TABELA III
MÉTRICAS COMPUTACIONAIS DOS MODELOS BASEADOS EM ÁRVORES

Modelo	Tempo de treino (s)	Memória (MB)	Tempo de predição (s)
Random Forest	49,67	59,68	0,22
XGBoost	56,31	51,10	0,01

V. EXPERIMENTOS COM REDES NEURAIS

Considerando a natureza unidimensional do vetor de atributos, optou-se pelo emprego de uma CNN1D, amplamente utilizada na literatura para extração automática de padrões locais em dados sequenciais e vetoriais. Inicialmente, foi utilizada uma CNN1D composta por três camadas convolucionais (64, 128 e 256 filtros), seguidas de normalização em lote, camada densa intermediária e saída com ativação softmax. Regularização L2 e dropout de 0.6 foram aplicados para reduzir sobreajuste. Todas as arquiteturas de CNNs foram treinadas em uma placa GPU T4, no ambiente Google Colab. Os resultados em diferentes cenários estão resumidos na Tabela IV.

TABELA IV

RESUMO DOS RESULTADOS DA CNN1D EM DIFERENTES CENÁRIOS

Cenário	Acurácia	Precisão	Recall	F1-score
Sintético → Real	0,835	1,000	0,835	0,901
Sintético + SMOTE → Real	0,922	1,000	0,922	0,953
Split 70/30 (Sintético)	0,798	0,796	0,798	0,794
Split 70/30 (Sintético+Real)	0,769	0,768	0,769	0,764

As métricas computacionais estão na Tabela V, comparando o treino sintético e o treino sintético com SMOTE, ambos avaliados na base real.

TABELA V

MÉTRICAS COMPUTACIONAIS DA CNN1D

Cenário	Treino (s)	Memória (MB)	Predição (s)
Sintético → Real	163,46	1716,50	0,094
Sintético + SMOTE → Real	192,64	1728,11	1,34

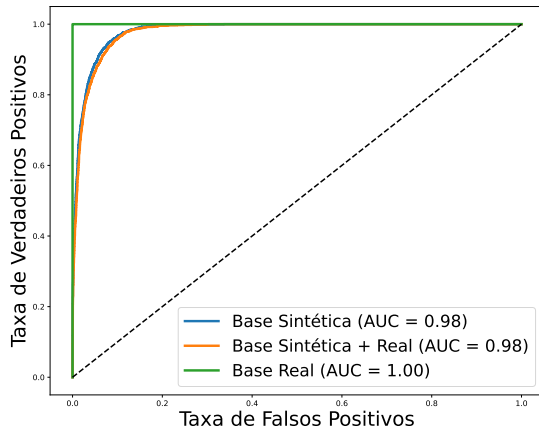


Fig. 3. Curvas ROC da CNN1D com valores os respectivos AUC para diferentes bases treinadas com divisão 70/30.

A Figura 3 apresenta as curvas ROC obtidas para a CNN1D, indicando, na legenda, os respectivos valores AUC, métrica que quantifica a capacidade de separação entre classes. Esta curva foi utilizada para evidenciar que, embora a CNN1D apresente acurácia inferior à dos modelos clássicos, mantém-se elevada capacidade de generalização dos padrões da base sintética para os dados reais. Não obstante, há um custo computacional superior referente aos modelos clássicos, reforçando a análise conjunta entre desempenho e infraestrutura. Embora o treino tenha sido realizado em GPU, não foram exploradas técnicas avançadas de paralelização, como o CUDA (*Compute Unified Device Architecture*) [14], plataforma da NVIDIA utilizada para este fim. Uma análise detalhada desse aspecto exigiria discutir primitivas e estratégias específicas de programação paralela, o que pertence ao domínio da otimização de hardware em baixo nível. Este trabalho não aborda tais questões, mas sim foca nos resultados dos algoritmos de classificação.

Como a CNN1D aplicada a dados tabulares se comporta de forma semelhante à das redes densas, foi implementado um MLP para comparação. O modelo foi composto por três

camadas densas (64, 32 e 16 unidades), seguidas de uma camada de saída *softmax*, sendo treinado com o otimizador Adam [20].

O treinamento foi feito com dados sintéticos e validação na base real. A Tabela VI apresenta os resultados para 120 e 200 épocas, definidos como limites superiores a partir da observação das curvas de perda, utilizadas para monitorar a evolução do aprendizado. Foi utilizado *early stopping*, interrompendo quando não havia melhoria significativa na função de perda, evitando sobreajuste do modelo.

TABELA VI

RESULTADOS DO MLP COM VALIDAÇÃO NA BASE REAL

Épocas	Acurácia	Precisão	Recall	F1-score
120	0,883	0,969	0,883	0,919
200	0,901	0,990	0,903	0,939

As métricas computacionais indicam que o MLP possui custo significativamente inferior ao da CNN, podendo ser executado em CPU com desempenho competitivo, o que o torna mais atrativo para aplicações embarcadas.

Os resultados com SMOTE no MLP não foram significativos, com redução de acurácia na validação real, sugerindo distorção na distribuição aprendida pela rede.

O treinamento de redes neurais profundas é estocástico, sobretudo devido à inicialização aleatória dos pesos. Para avaliar essa variabilidade, foram realizadas 50 execuções independentes da CNN1D apresentada, com dados, hiperparâmetros e épocas fixos. O objetivo foi analisar o comportamento da rede diante de diferentes inicializações.

Observou-se variação significativa nas métricas entre as execuções. Em uma execução específica, a acurácia atingiu 92% na base real e 94% na base sintética; contudo, esses valores não se repetiram de forma consistente, apresentando grande dispersão entre as execuções. A média das 50 execuções foi de 79,9% de acurácia na base real, com desvio padrão de 7,0%, evidenciando tanto o desempenho típico da CNN1D quanto sua sensibilidade às condições iniciais de treinamento.

Para garantir reprodutibilidade, os treinamentos usaram sementes fixas, permitindo a reprodução dos resultados nas mesmas condições. Nos modelos de árvores, o baixo custo computacional permitiu explorar hiperparâmetros por meio da validação cruzada, selecionando as configurações de melhor desempenho. Em redes neurais, entretanto, o treinamento é mais custoso, e a inicialização dos pesos influencia fortemente o resultado, dificultando essa exploração.

VI. CONCLUSÕES E TRABALHOS FUTUROS

Em cenários de defesa, a identificação e classificação de projéteis em radares de contrabateria constituem uma tarefa crítica para a tomada de decisão rápida e precisa. A complexidade dos sinais e as restrições de processamento em sistemas embarcados tornam esse problema particularmente desafiador. Neste contexto, este trabalho apresentou uma análise comparativa entre diversos algoritmos de aprendizado de máquina aplicados a essa tarefa, considerando também as métricas de desempenho e o custo computacional.

Os resultados indicam que as abordagens analisadas apresentam características complementares. Os algoritmos baseados em árvores, como *Random Forest* e *Gradient Boosting*, demonstraram elevada acurácia, robustez e eficiência computacional, com tempo de resposta e requisitos de memória compatíveis com aplicações embarcadas.

No caso do *Gradient Boosting*, conforme demonstrado, pode-se avaliar a contribuição relativa de cada atributo para a classificação. O MLP apresentou desempenho competitivo, com baixo custo computacional, enquanto a CNN1D apresentou maior capacidade discriminativa, embora com maior variabilidade devido à inicialização dos pesos. O estudo sugere que sistemas híbridos, combinando modelos clássicos e redes neurais, sejam promissores em aplicações de radar de contrabateria, equilibrando interpretabilidade, desempenho estatístico e custo computacional, com potencial para aprimorar soluções embarcadas já estabelecidas.

A continuidade desta pesquisa pode avançar em diferentes direções. Uma primeira linha envolve a otimização dos pesos das redes neurais, dada a variabilidade observada na inicialização aleatória, que resultou em desvios padrão da acurácia próximos a 7%. Técnicas evolutivas, como algoritmos genéticos, podem ser exploradas para reduzir essa variabilidade, realizando uma pré-calibragem dos pesos iniciais como etapa inicial do pipeline, que abrange desde o treinamento até a implementação em campo, garantindo que o treinamento se inicie a partir de valores mais adequados. Essa etapa deve ocorrer em ambiente computacional dedicado, anterior ao uso embarcado, devido ao elevado custo computacional.

Outra linha é integrar filtros de Kalman à classificação em sistemas híbridos. A combinação de dados tabulares e séries temporais pode fornecer informações complementares para refinar as estimativas de trajetória. Futuramente, implementações em CPU, GPU e FPGA podem equilibrar latência e viabilidade embarcada.

Os resultados obtidos indicam a viabilidade computacional da abordagem proposta: modelos treinados com dados sintéticos foram validados satisfatoriamente em base real, apesar das diferenças entre os conjuntos. Essa evidência confirma a viabilidade prática da abordagem e indica que soluções híbridas podem ser aplicadas com confiança em cenários reais de radar de contrabateria. O trabalho abre perspectivas para continuidade e estabelece uma base sólida para aplicações embarcadas, conciliando desempenho, interpretabilidade e eficiência.

AGRADECIMENTOS

Os autores expressam seus agradecimentos à FINEP, Projeto nº 01.24.0649.00, pelo financiamento parcial deste trabalho.

REFERÊNCIAS

- [1] D. M. O. Crús, “Abordagem integrada para estimação de trajetória e classificação de projéteis em radares de contrabateria usando filtros de kalman e aprendizado de máquina,” Master’s thesis, Instituto Militar de Engenharia, Rio de Janeiro, Brasil, 2026.
- [2] U. K. Singh, V. Padmanabhan, and A. Agarwal, “Dynamic classification of ballistic missiles using neural networks and hidden markov models,” *Applied Soft Computing*, vol. 19, pp. 280–289, 2014.
- [3] C. Abeynayake, V. Son, M. H. I. Shovon, and H. Yokohama, “Machine learning based automatic target recognition algorithm applicable to ground penetrating radar data,” in *Proc. SPIE Defense + Commercial Sensing, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV*, vol. 11012, p. 1101202, 2019.
- [4] E. A. Hadhrami, M. A. Mufti, B. Taha, and N. Werghi, “Classification of ground moving radar targets using convolutional neural network,” in *Proc. 22nd Int. Microwave and Radar Conf. (MIKON)*, pp. 127–130, 2018.
- [5] S. T. Westlake, T. N. Volonakis, J. Jackman, D. B. James, and A. Sherriff, “Deep learning for automatic target recognition with real and synthetic infrared maritime imagery,” in *Proc. SPIE Security + Defence, Artificial Intelligence and Machine Learning in Defense Applications II*, vol. 11543, p. 1154309, 2020.
- [6] T. D. Smithies, M. J. Campbell, N. Ramsbottom, and A. J. Toth, “A random forest approach to identify metrics that best predict match outcome and player ranking in the esports rocket league,” *Scientific Reports*, vol. 11, p. 19285, 2021.
- [7] M. Zhang, M. Treder, D. Marshall, and Y. Li, “Fast explanation of rbf-kernel svm models using activation patterns,” in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, pp. 1–8, 2024.
- [8] U. K. Singh and V. Padmanabhan, “Training by art-2 and classification of ballistic missiles using hidden markov model,” in *Pattern Recognition and Machine Intelligence (PRMI)*, vol. 8251 of LNCS, pp. 125–136, Springer, 2013.
- [9] D. Zhou and J. Yang, “Ballistic target recognition based on micro-motion characteristics using sequential hrps,” in *Proc. IEEE Int. Conf. Electronics Information and Emergency Communication (ICEIEC)*, pp. 165–168, 2013.
- [10] A. R. Kim, H. S. Kim, C. H. Kang, and S. Y. Kim, “The design of the 1d cnn-gru network based on the rcs for classification of multiclass missiles,” *Remote Sensing*, vol. 15, no. 3, p. 577, 2023.
- [11] R. Narasimhamurthy and O. I. Khalaf, “Deep learning network for classifying target of same shape using rcs time series,” in *Air Traffic Management and Control* (L. Li, ed.), ch. 3, London, UK: IntechOpen, 2021.
- [12] P. Uwizize, S. K. Rao, B. O. L. Jagan, H. M. Rai, S. Agarwal, and W. Pak, “Radar-based target tracking using deep learning approaches with unscented kalman filter,” *Applied Sciences*, vol. 14, no. 18, p. 8332, 2024.
- [13] Y. Kim, *Radar target classification using deep learning*, ch. 16, pp. 487–514. John Wiley & Sons, Ltd., 2023.
- [14] M. Rakhimov, S. Javliev, and R. Nasimov, “Parallel approaches in deep learning: use parallel computing,” in *Proc. 7th Int. Conf. Future Networks and Distributed Systems (ICFNDS)*, pp. 192–201, 2024.
- [15] J. Fix, C. Ren, A. C. Lopes, G. Morice, S. Kobayashi, I. Hinojosa, et al., “Deep learning for aircraft classification from vhf radar signatures,” *IET Radar, Sonar & Navigation*, vol. 15, pp. 697–707, 2021.
- [16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [17] A. Morehead, L. Ogden, G. Magee, R. Hosler, B. White, and G. Mohler, “Low cost gunshot detection using deep learning on the Raspberry Pi,” in *Proc. IEEE Int. Conf. Big Data*, pp. 3038–3044, Dec 2019.
- [18] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [19] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.