COMPARAÇÃO DE MÉTODOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE FALHAS DE LOCALIZAÇÃO EM ROBÔS MÓVEIS AUTÔNOMOS

COMPARISON OF MACHINE LEARNING METHODS FOR PREDICTING LOCALIZATION FAILURES IN AUTONOMOUS MOBILE ROBOTS

Clístenes Grizafis Bento Razer[1]
Anthom Nizer Rojas Montaño[2]

RESUMO

O presente trabalho compara quatro algoritmos de aprendizado de máquina para a predição de falhas de localização em robôs móveis autônomos. O método AMCL (*Adaptive Monte Carlo Localization*), amplamente utilizado com sensores LiDAR, pode apresentar falhas em ambientes com obstáculos dinâmicos, zonas com poucas características geométricas ou configurações simétricas. Com a finalidade de identificar qual algoritmo possui melhor desempenho na detecção antecipada dessas falhas, foram comparados os modelos *Random Forest*, *Support Vector Machine* (SVM), *MultiLayer Perceptron* (MLP) e XGBoost, utilizando o conjunto de dados sintético Robot Localization Failure Prediction Dataset, composto por 417.185 amostras rotuladas de 21 experimentos em 7 ambientes simulados no NVIDIA Isaac Sim. A partir dos dados brutos foram extraídas 22 características de leituras LiDAR e da nuvem de partículas do filtro AMCL. Os modelos foram avaliados com múltiplas métricas e validação cruzada estratificada com 5 *folds*. Os resultados indicam que os modelos baseados em árvores apresentam o melhor desempenho, com destaque para *Random Forest* (F1Score de 0,9604) e o XGBoost (F1 de 0,9516), sendo as características de dispersão das partículas as mais discriminativas. A análise por tipo de ambiente revela que mapas simétricos representam maior desafio para todos os modelos. O códigofonte completo para replicação do experimento é disponibilizado separadamente em repositório público.

Palavraschave: Robótica Móvel; Aprendizado de Máquina; AMCL; Filtro de Partículas; Localização.

ABSTRACT

[1] Graduando em Especialização em Inteligência Artificial Aplicada, Universidade Federal do Paraná (UFPR). E mail: clistenes@metabee.tec.br

[2] Doutor em Ciência da Computação, Universidade Federal do Paraná (UFPR). E mail: razer@ufpr.br

This work compares four machine learning algorithms for predicting localization failures in autonomous mobile robots. The AMCL (Adaptive Monte Carlo Localization) method, widely used with LiDAR sensors, can fail in environments with dynamic obstacles, featureless zones, or symmetric layouts. In order to identify which algorithm performs best in early detection of such failures, Random Forest, Support Vector Machine (SVM), MultiLayer Perceptron (MLP), and XGBoost models were compared using the synthetic dataset Robot Localization Failure Prediction Dataset, comprising 417,185 labeled samples from 21 experiments in 7 simulated environments in NVIDIA Isaac Sim. From the raw data, 22 statistical features were extracted from LiDAR readings and the AMCL particle cloud. Models were evaluated with multiple metrics and stratified 5fold crossvalidation. Results indicate that treebased models achieved the best performance, with Random Forest (F1Score of 0.9604) and XGBoost (F1 = 0.9516) standing out, and particle dispersion features being the most discriminative. Analysis by environment type reveals that symmetric maps pose the greatest challenge for all models. The complete source code for experiment replication is made available separately in a public repository.

Keywords: Mobile Robotics; Machine Learning; AMCL; Particle Filter; Localization.

1 INTRODUÇÃO

Robôs móveis autônomos (AMRs, do inglês *Autonomous Mobile Robots*) estão cada vez mais presentes em ambientes industriais, logísticos e de serviço, desempenhando tarefas como transporte de materiais, inspeção e vigilância. De acordo com Thrun, Burgard e Fox (2005), a navegação autônoma desses robôs depende fundamentalmente de uma estimativa precisa de sua posição e orientação no ambiente, processo conhecido como localização. Cadena *et al*. (2016) identificaram que desafios de robustez e escalabilidade na localização e mapeamento simultâneos (SLAM, do inglês *Simultaneous Localization and Mapping*) persistiam à época, e trabalhos recentes confirmam que esses desafios continuam relevantes, especialmente em ambientes dinâmicos e ambíguos (Knitt *et al*., 2025ª) (Eder; Reip; Steinbauer, 2022). No ecossistema ROS 2 (*Robot Operating System* 2), o pacote Nav2 fornece uma pilha completa de navegação que inclui o AMCL como método padrão de localização (Macenski *et al*., 2023).

O Método de Localização Adaptativa de Monte Carlo (AMCL, do inglês *Adaptive Monte Carlo Localization*), baseado em filtros de partículas, é uma das abordagens mais utilizadas para localização de robôs móveis equipados com sensores LiDAR (Thrun; Burgard; Fox, 2005) (Fox, 2003).

Falhas de localização em robôs autônomos podem ter consequências graves, desde colisões com obstáculos até interrupção completa de uma determinada operação. Portanto, a capacidade de predizer proativamente essas falhas, antes que elas ocorram ou se agravem,

éde grande valor prático. Essa predição pode permitir que o sistema de navegação haja de maneira corretiva, como solicitar relocalização, reduzir a velocidade ou alertar um operador humano.

Neste contexto, técnicas de aprendizado de máquina supervisionado oferecem abordagem promissora para monitoramento preditivo de localização. Trabalhos recentes demonstram a viabilidade de usar dados do filtro de partículas e sensores para detectar falhas de localização (Eder; Reip; Steinbauer, 2022) (Knitt *et al*., 2023) (Eder; Frering; SteinbauerWagner, 2023). Além disso, *benchmarks* recentes confirmam que modelos baseados em árvores de decisão como *Random Forest* (RF) e XGBoost, continuam superando abordagens de *deep learning* em dados tabulares (Grinsztajn; Oyallon; Varoquaux, 2022) (ShwartzZiv; Armon, 2022).

O presente trabalho teve como objetivo comparar quatro algoritmos de aprendizado de máquina, *Random Forest* (RF), *Support Vector Machine* (SVM), *MultiLayer Perceptron* (MLP) e XGBoost, para tarefa de predição binária de falhas de localização em robôs móveis. A comparação foi realizada utilizando um conjunto de dados sintético recente e abrangente, com análise detalhada por tipo de ambiente e configuração de obstáculos.

As principais contribuições deste trabalho são:

a) extração de características informativas a partir de dados brutos de LiDAR e da nuvem de partículas AMCL, criando uma representação compacta de 22 características adequada para algoritmos de aprendizado de máquina;
b) comparação sistemática de quatro algoritmos de classificação, avaliados com múltiplas métricas (acurácia, precisão, *recall*, F1Score e AUCROC) e validação cruzada;
c) análise segmentada do desempenho por tipo de ambiente (*Warehouse*, simétrico e assimétrico) e configuração de obstáculos (dinâmicos, estáticos e combinados);
d) disponibilização do códigofonte completo para replicação do experimento.

2 FUNDAMENTAÇÃO TEÓRICA

Com a finalidade de proporcionar melhor compreensão sobre os assuntos abordados, o presente capítulo está dividido em quatro partes, sendo elas robótica móvel e navegação autônoma, AMCL e filtro de partículas, falhas de localização e aprendizado de máquina para detecção de falhas.

2.1 ROBÓTICA MÓVEL E NAVEGAÇÃO AUTÔNOMA

Na robótica, robôs capazes de se mover ou navegar de um ponto a outro são chamados de robôs móveis e de acordo com Thrun, Burgard e Fox (2005), a navegação autônoma de robôs móveis envolve três problemas fundamentais: localização ("onde estou?"), mapeamento ("como é o ambiente?") e planejamento de caminho ("como chego ao destino?"). A localização é considerada o problema mais crítico, pois erros de localização comprometem diretamente o planejamento e execução de trajetórias. Cadena *et al*. (2016) apresentaram uma revisão abrangente do problema de SLAM, identificando desafios de robustez e escalabilidade que, conforme demonstrado por Knitt *et al*. (2025a) e Eder, Reip e Steinbauer (2022), permanecem atuais.

Em ambientes estruturados, como armazéns e fábricas, a abordagem mais comum é a localização baseada em mapa, onde o robô possui um mapa prévio do ambiente e utiliza sensores para estimar a posição dentro desse mapa. O sensor LiDAR *(Light Detection and Ranging*) é amplamente utilizado por fornecer medições de distância precisas em 360 graus, sendo robusto a variações de iluminação (Macenski *et al*., 2023). O ecossistema ROS 2 consolidouse como o padrão de facto para o desenvolvimento de software em robótica móvel. Macenski *et al*. (2023) descrevem em detalhes a arquitetura do pacote Nav2 (*Navigation* 2), que fornece uma pilha completa de navegação incluindo localização (AMCL), planejamento de caminho, controle e recuperação de falhas.

2.2 AMCL E FILTRO DE PARTÍCULAS

Segundo Fox (2003) *o Adaptive Monte Carlo Localization* (AMCL) é uma implementação do filtro de partículas para o problema de localização global e rastreamento de pose. Thrun, Burgard e Fox (2005) fornecem o tratamento matemático completo do filtro de partículas e suas variantes. O algoritmo funciona da seguinte forma: a) inicialização, um conjunto de N partículas é distribuído no espaço de estados (posição x, y e orientação), cada uma com um peso associado; b) predição, quando o robô se move, cada partícula é propagada de acordo com o modelo de movimento (odometria), adicionando ruído para representar a incerteza; c) atualização, ao receber uma leitura do LiDAR, o peso de cada partícula é atualizado com base na verossimilhança da leitura dada a pose da partícula e o mapa; d)

reamostragem, partículas com pesos baixos são eliminadas e substituídas por cópias de partículas com pesos altos, concentrando a distribuição nas regiões mais prováveis.

O AMCL adapta automaticamente o número de partículas usando KLD*sampling* (Fox, 2003), aumentando o número quando há incerteza e reduzindo quando a localização está convergida. Quando as partículas estão concentradas em uma região pequena, a localização é considerada confiável. Quando estão dispersas, indica incerteza ou possível falha.

2.3 FALHAS DE LOCALIZAÇÃO

Uma falha de localização ocorre quando a estimativa de pose do robô diverge significativamente da pose real. Knitt *et al*. (2025a) definem falha como a situação em que o erro de posição excede 0,4 metros ou o erro de orientação excede 0,4 radianos. Eder, Reip e Steinbauer (2022) adotam critérios semelhantes em seu trabalho sobre monitoramento de localização, confirmando que esses limiares são representativos de falhas operacionalmente relevantes. Dentre as principais causas de falhas destacamse: ambientes simétricos, cujos *layouts* repetitivos fazem com que múltiplas posições produzam leituras de LiDAR semelhantes, confundindo o filtro de partículas (Knitt *et al*., 2023) (Knitt *et al*., 2025a); obstáculos dinâmicos, como pessoas e outros robôs, que geram leituras inconsistentes com o mapa estático (Eder; Reip; Steinbauer, 2022) (Knitt *et al*., 2025a); *drift* de odometria, onde erros acumulados no modelo de movimento degradam a predição das partículas (Thrun; Burgard; Fox, 2005); e zonas sem features, como corredores longos e retos ou espaços abertos que oferecem pouca informação para distinguir posições (Knitt *et al*., 2023) (Eder; Reip; Steinbauer, 2022).

A detecção precoce dessas falhas é essencial para a operação segura de robôs autônomos. Abordagens tradicionais baseadas em limiares simples, como a dispersão das partículas acima de um valor fixo, são limitadas por não capturarem padrões complexos nos dados (Knitt *et al*., 2023) (Eder; Reip; Steinbauer, 2022).

2.4 APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE FALHAS

A detecção de falhas baseada em dados é uma área ativa de pesquisa em robótica e sistemas autônomos. No contexto de localização, a abordagem consiste em treinar modelos supervisionados com dados rotulados (nominal contra falha) para identificar padrões nos dados sensoriais que precedem ou acompanham falhas.

Eder, Reip e Steinbauer (2022) propuseram um monitor de localização baseado em aprendizado de máquina que utiliza características extraídas do filtro de partículas e de sensores para classificar o estado de localização do robô, demonstrando a viabilidade da abordagem com *Random Forest* e SVM. Em trabalho subsequente, Eder, Frering e SteinbauerWagner (2023) aplicaram técnicas de Inteligência Artificial Explicável (XAI, do inglês *Explainable Artificial Intelligence*) ao monitor de localização, revelando quais características são mais relevantes para a decisão dos modelos e confirmando a predominância das características de dispersão das partículas.

Knitt *et al*. (2023) propuseram um *framework* de *benchmarking* para localização *indoor* de AMRs em intralogística, identificando cenários críticos para falhas. Mais recentemente, Knitt *et al*. (2025a) disponibilizaram um conjunto de dados sintéticos abrangente especificamente projetado para o desenvolvimento de monitores de localização baseados em dados, utilizado neste trabalho.

Os quatro algoritmos comparados representam diferentes paradigmas: *Random Forest* (Breiman, 2001), conjunto de árvores treinadas em subconjuntos aleatórios dos dados; SVM (Cortes; Vapnik, 1995), que busca o hiperplano de separação com margem máxima, usando *kernel* RBF para fronteiras não lineares; MLP (Goodfellow; Bengio; Courville, 2016), rede neural com múltiplas camadas ocultas; e XGBoost (Chen; Guestrin, 2016), implementação otimizada de *gradient boosting*. A escolha desses algoritmos é respaldada por Grinsztajn, Oyallon e Varoquaux (2022) e ShwartzZiv e Armon (2022), que demonstraram que modelos baseados em árvores superam consistentemente abordagens de *deep learning* em dados tabulares.

3 MATERIAIS E MÉTODOS

O presente trabalho utiliza metodologia própria composta pelas etapas a seguir: a) obtenção e análise exploratória do conjunto de dados; b) extração de características estatísticas a partir dos dados brutos de LiDAR e partículas AMCL; c) préprocessamento dos dados (divisão treino/teste, normalização e balanceamento); d) treinamento e avaliação dos quatro modelos de aprendizado de máquina; e) validação cruzada estratificada com 5 *folds*; f) análise segmentada por tipo de ambiente e configuração de obstáculos.

3.1 DATASET

Para a realização dos experimentos foi utilizado o conjunto de dados Robot Localization Failure Prediction Dataset (Knitt *et al*., 2025b), descrito por Knitt *et al*. (2025a) e gerado utilizando o simulador NVIDIA Isaac Sim (NVIDIA, 2025). O conjunto de dados contém dados de um robô omnidirecional MoMo equipado com LiDAR Velodyne VLP16. Os dados são fornecidos em formato *Apache Parquet* e estão disponíveis publicamente.

O Quadro 1 apresenta a composição do conjunto de dados utilizado.

QUADRO 1 – COMPOSIÇÃO DO CONJUNTO DOS DADOS

Característica	**Valor**
Total De Amostras	417185
Amostras Nominais (Y=0)	320875 (76,91%)
Amostras De Falha (Y=1)	96310 (23,09%)
Número De Experimentos	21
Número De Ambientes	7
Leituras Lidar Por Amostra	3600 (360 graus, resolução 0,1grau)
Particulas Amcl Por Amostra	501

FONTE: Adaptado de Knitt *et al*. (2025b)

Os ambientes de simulação são compostos por: a) *Warehouse*, ambiente grande com corredores, prateleiras e equipamentos de manuseio; b) *Symmetric Maps* (13), três ambientes de pequena escala com *layouts* simétricos; c) *Unsymmetric Maps* (13), três ambientes de pequena escala com *layouts* assimétricos. Para cada ambiente foram utilizadas três configurações de obstáculos: apenas dinâmicos (25 obstáculos esféricos com trajetórias aleatórias), apenas estáticos (obstáculos cúbicos fixos não incluídos no mapa) e combinados (ambos os tipos simultaneamente).

O critério de rotulagem define que uma amostra é rotulada como falha (is_ delocalized = True) quando o erro de posição excede 0,4 metros ou o erro de orientação excede 0,4 radianos.

3.2 EXTRAÇÃO DE CARACTERÍSTICAS

Os dados brutos (3.600 leituras LiDAR + 501 partículas com posição e peso cada) foram transformados em um vetor compacto de 22 características numéricas, organizadas em três grupos. A extração foi realizada de forma incremental, processando um arquivo Parquet por vez para evitar acúmulo em memória. A escolha das características foi inspirada nos trabalhos de Eder, Reip e Steinbauer (2022) e Eder, Frering e SteinbauerWagner (2023). O Quadro 2 apresenta as 22 características extraídas.

QUADRO 2 – CARACTERÍSTICAS EXTRAÍDAS (22 NO TOTAL)

Grupo	Feature	Descrição
LiDAR (12)	Lidar_mean, lidar_std	Média e desvio padrão das 3.600 distâncias.
	lidar_min, lidar_max, lidar_median	Mínimo, máximo válido e mediana
	lidar_p10, lidar_p25, lidar_p75, lidar_p90	Percentis da distribuição
	lidar_max_ratio	Fração de leituras sem retorno
	lidar_entropy	Entropia de Shannon (50 *bins*)
	lidar_skewness	Assimetria da distribuição
Partículas AMCL (8)	particle_std_x, particle_std_y	Dispersão espacial das partículas
	particle_std_yaw	Dispersão angular (desvio padrão circular)
	particle_spread	Diâmetro da nuvem de partículas
	particle_weight_entropy	Entropia dos pesos
	particle_weight_max, particle_ weight_mean	Peso máximo e médio
	particle_weight_cv	Coeficiente de variação dos pesos
Pose AMCL (2)	amcl_x, amcl_y	Posição estimada pelo filtro

FONTE: O autor (2026)

As 12 características do LiDAR descrevem a geometria do ambiente ao redor do robô em cada instante. As 8 características de partículas medem a dispersão entre as próprias partículas de uma mesma amostra, não em relação à posição real (desconhecida durante a operação), sendo indicadores diretos da qualidade da localização: quando o filtro está confiante, as partículas convergem e os pesos se concentram; quando incerto, se espalham. As 2 características de pose permitem ao modelo associar regiões específicas do mapa a maior probabilidade de falha. Essa transformação reduz a dimensionalidade de milhares de valores brutos para 22, viabilizando o treinamento dentro das restrições de memória (seção 3.6).

3.3 PRÉPROCESSAMENTO

O préprocessamento dos dados seguiu as seguintes etapas: a) divisão treino/teste, 80% para treino (333.748 amostras) e 20% para teste (83.437 amostras), com estratificação para manter a proporção de classes; b) normalização, aplicação da escala padrão (média 0, desvio padrão 1) ajustado apenas nos dados de treino para evitar vazamento de dados; c) balanceamento de classes, subamostragem da classe majoritária (nominal) no conjunto de treino, resultando em 154.096 amostras balanceadas (77.048 por classe). O conjunto de teste foi mantido com a distribuição original para avaliação realista.

A estratégia de subamostragemfoi escolhida por sua simplicidade e eficácia em problemas com desbalanceamento moderado, conforme recomendado por Chawla *et al*. (2002) e FernándezDelgado *et al*. (2014).

3.4 MODELOS E HIPERPARÂMETROS

Os quatro modelos foram configurados com hiperparâmetros selecionados com base na literatura. O Quadro 3 apresenta os principais valores e suas justificativas.

QUADRO 3 – HIPERPARÂMETROS DO RANDOM FOREST

Modelo	Hiperparâmetro	Valor	Referência
Random Forest	n_estimators / max_depth	200 / 20	Probst, Wright e Boulesteix (2019)
	min_samples_split	5	Breiman (2001)
	class_weight	*balanced*	Breiman (2001)
SVM (RBF)	C / gamma	10,0 / *scale*	Hsu, Chang e Lin (2003)
	amostragem treino	50.000	Complexidade O(n²) (CORTES; VAPNIK, 1995)
MLP	hidden_layers	(128, 64, 32)	Goodfellow, Bengio e Courville (2016)
	solver / activation	Adam / ReLU	Kingma e Ba (2015)
	alpha / early_stopping	0,001 / True	Prechelt (2012)
XGBoost	n_estimators / max_depth	200 / 8	Chen e Guestrin (2016)
	learning_rate	0,1	Bentéjac, Csörgő e MartínezMuñoz (2021)
	subsample / colsample_bytree	0,8 / 0,8	Chen e Guestrin (2016)

FONTE: O autor (2026)

O *Random Forest* utiliza class_weight=balanced para compensar o desbalanceamento residual. O SVM foi treinado com subamostra de 50.000 instâncias devido à sua complexidade computacional. O MLP emprega arquitetura piramidal decrescente com *early stopping* para prevenir overfitting. O XGBoost utiliza profundidade menor que o RF, pois no *gradient boosting* cada árvore corrige os erros das anteriores.

3.5 MÉTRICAS DE AVALIAÇÃO

Os modelos foram avaliados com as seguintes métricas: a) acurácia, proporção de predições corretas sobre o total; b) precisão, proporção de verdadeiros positivos sobre o total de predições positivas (minimiza alarmes falsos); c) *recall* (sensibilidade), proporção de verdadeiros positivos sobre o total de positivos reais (minimiza falhas não detectadas); d) F1Score, média harmônica entre precisão e *recall* (métrica principal); e) AUCROC, área sob a curva ROC, que mede a capacidade discriminativa do modelo independente do limiar

dedecisão. Adicionalmente, foi realizada validação cruzada estratificada com 5 *folds* para estimar a robustez dos resultados.

3.6 AMBIENTE COMPUTACIONAL

Os experimentos foram executados no Google Colab com as seguintes especificações: CPU Intel Xeon, 12,7 GB de RAM, GPU NVIDIA T4 (utilizada apenas para aceleração do XGBoost). A limitação de memória RAM (12,7 GB) exigiu a adoção da estratégia de processamento incremental descrita na seção 3.2 para viabilizar a extração de características do conjunto de dados completo (~417 mil amostras). O códigofonte completo em Python está disponível em: <https://github.com/HobbiesProfBento/mllocalizationfailureprediction>.

4 EXPERIMENTOS E RESULTADOS

4.1 ANÁLISE EXPLORATÓRIA DOS DADOS

A análise exploratória revelou que em relação ao desbalanceamento de classes, 76,91% das amostras são nominais e 23,09% são falhas, confirmando a necessidade de balanceamento para o treinamento (CHAWLA *et al.*, 2002). Após a subamostragem, o conjunto de treino balanceado ficou com 154.096 amostras (77.048 por classe).

Quanto à variação entre ambientes, a taxa de falha varia significativamente entre os tipos de mapa: *warehouse* (18,86%), symmetric_exp_1 (16,77%), symmetric_exp_2 (27,60%), symmetric_exp_3 (27,16%), unsymmetric_exp_1 (23,60%) e unsymmetric_exp_2 (25,18%). A presença de apenas obstáculos dinâmicos gerou a maior taxa de falha (29,06%), enquanto apenas obstáculos estáticos apresentou a menor (17,45%).

No que diz respeito às características discriminativas, as características de dispersão das partículas apresentaram a maior correlação com o rótulo de falha: particle_weight_entropy ($r = 0,4532$), particle_weight_max ($r = -0,4487$), particle_std_yaw ($r = 0,4389$) e particle_weight_cv ($r = 0,4168$). As características do LiDAR apresentaram correlações mais baixas ($|r| < 0,12$), indicando que o estado do filtro de partículas é mais informativo que as leituras brutas do sensor. Esse resultado é consistente com os achados de Eder, Frering e SteinbauerWagner (2023), que identificaram as mesmas características como mais relevantes utilizando técnicas de XAI.

4.2 RESULTADOS COMPARATIVOS

A Tabela 1 apresenta a comparação de desempenho dos modelos no conjunto de teste (83.437 amostras não utilizadas no treinamento), utilizando as métricas definidas na seção 3.5, acrescidas do tempo de treino em segundos. Os valores em negrito indicam o melhor resultado em cada métrica.

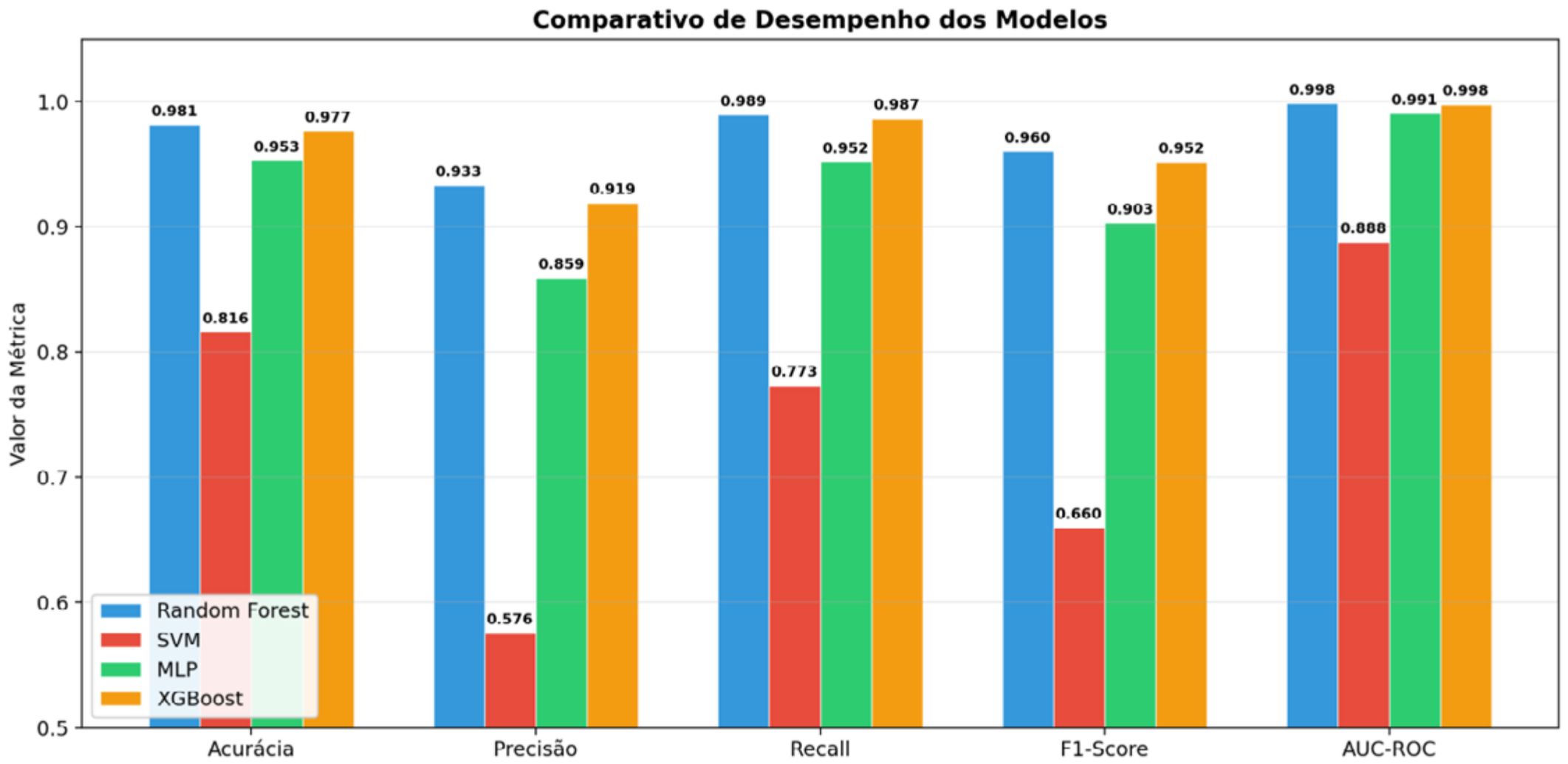
TABELA 1 – COMPARAÇÃO DE DESEMPENHO DOS MODELOS NO CONJUNTO DE TESTE

Modelo	Acurácia	Precisão	Recall	F1Score	AUCROC	Tempo Treino (S)
Random Forest	**0,9812**	**0,9332**	**0,9894**	**0,9604**	**0,9983**	19,43
SVM	0,8160	0,5756	0,7733	0,6599	0,8877	537,98
MLP	0,9528	0,8590	0,9520	0,9031	0,9905	232,64
XGBoost	0,9768	0,9191	0,9865	0,9516	0,9977	**2,35**

FONTE: O autor (2026)

A Figura 1 permite visualizar a comparação entre os modelos. No gráfico de barras agrupadas, observase que o *Random Forest* e o XGBoost dominam em todas as métricas, enquanto o SVM fica consistentemente abaixo dos demais.

FIGURA 1 – COMPARATIVO DE DESEMPENHO DOS MODELOS



FONTE: O autor (2026)

O *Random Forest* obteve o melhor desempenho em todas as métricas exceto tempo de treino, onde o XGBoost foi superior (2,35s contra 19,43s). A diferença entre os dois melhores modelos é pequena (F1: 0,9604 contra 0,9516), mas a diferença para o SVM é expressiva (F1: 0,6599), evidenciando a superioridade dos modelos baseados em árvores.

A Tabela 2 apresenta os resultados da validação cruzada estratificada com 5 *folds*. Os valores em negrito indicam o melhor resultado.

TABELA 2 – VALIDAÇÃO CRUZADA ESTRATIFICADA (5FOLD F1SCORE)

Modelo	Média	Desvio Padrão
Random Forest	0,8858	0,0125
SVM	0,6594	**0,0067**
MLP	0,7539	0,0275
XGBoost	**0,9012**	0,0099

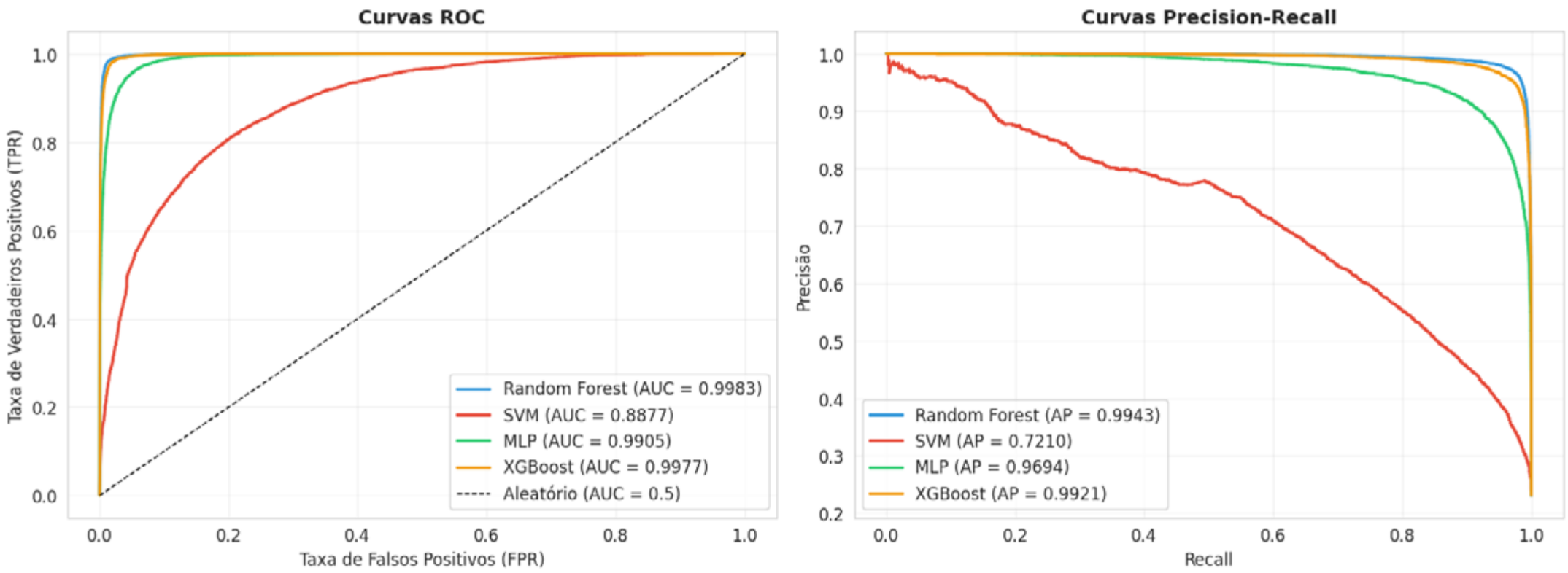
FONTE: O autor (2026)

Na validação cruzada, o XGBoost obteve a melhor média (0,9012), superando o *Random Forest* (0,8858). Isso sugere que o XGBoost generaliza melhor entre diferentes partições dos dados, enquanto o *Random Forest* obteve o melhor resultado na divisão fixa treino/teste. A variância do XGBoost (+/- 0,0099) também indica estabilidade, resultado consistente com as observações de Bentéjac, Csörgő e MartínezMuñoz (2021) sobre a capacidade de generalização do *gradient boosting*.

4.3 ANÁLISE DAS CURVAS ROC E *PRECISIONRECALL*

A Figura 2 apresenta dois gráficos complementares que avaliam a capacidade discriminativa dos modelos: as curvas ROC (à esquerda) e as curvas *PrecisionRecall* (à direita).

FIGURA 2 – CURVAS ROC E PRECISIONRECALL DOS QUATRO MODELOS



FONTE: O autor (2026)

Na curva ROC, que avalia o compromisso entre taxa de verdadeiros positivos (*recall*) e taxa de falsos positivos ao variar o limiar de decisão, as curvas do *Random Forest* (AUC = 0,9983) e do XGBoost (AUC = 0,9977) praticamente se sobrepõem próximas ao canto superior esquerdo, indicando excelente capacidade discriminativa com pouquíssimos alarmes falsos. A curva do MLP (AUC = 0,9905) também se aproxima do canto ideal, mas com curvatura ligeiramente mais suave. A curva do SVM (AUC = 0,8877) é visivelmente mais distante, indicando que, para alcançar *recall* alto, o SVM precisa aceitar taxa significativamente maior de alarmes falsos.

Na curva *PrecisionRecall*, especialmente informativa em problemas com classes desbalanceadas como este (23% de falhas contra 77% nominais), o *Random Forest* (AP = 0,9943) e o XGBoost (AP = 0,9921) mantêm precisão próxima de 1,0 mesmo com *recall* elevado. O MLP (AP = 0,9694) apresenta queda gradual na precisão à medida que o *recall* aumenta, enquanto o SVM (AP = 0,7210) mostra queda acentuada, comportamento consistente com seu baixo F1Score (0,6599).

A proximidade dos valores de AUCROC entre *Random Forest* e XGBoost (diferença de apenas 0,0006) indica que ambos os modelos são igualmente confiáveis para a tarefa, independentemente do limiar de decisão escolhido.

4.4 ANÁLISE DAS MATRIZES DE CONFUSÃO

A Tabela 3 apresenta as matrizes de confusão dos modelos, onde FN% indica a proporção de falhas reais não detectadas em relação ao total (19.262).

TABELA 3 – MATRIZES DE CONFUSÃO (CONJUNTO DE TESTE: 64.175 NOMINAIS, 19.262 FALHAS)

Modelo	VN	FP	FN	VP	FN%
Random Forest	62.810	1.365	205	19.057	1,06%
SVM	53.192	10.983	4.367	14.895	22,67%
MLP	61.164	3.011	925	18.337	4,80%
XGBoost	62.502	1.673	261	19.001	1,35%

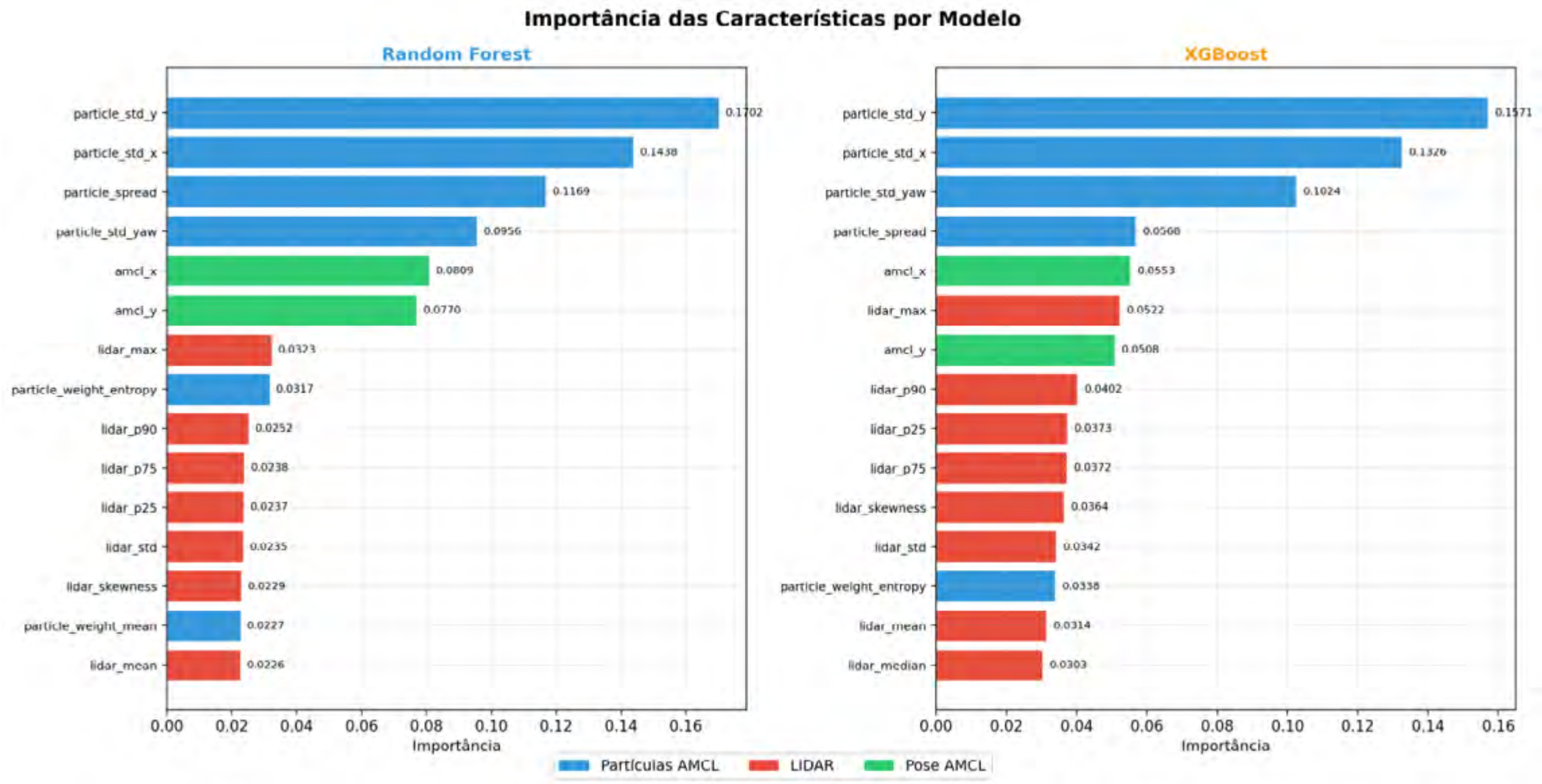
FONTE: O autor (2026)

O *Random Forest* apresentou o menor número de falsos negativos (205, apenas 1,06% das falhas reais), seguido pelo XGBoost (261, 1,35%). Isso significa que, de cada 100 falhas reais, o *Random Forest* deixou de detectar apenas 1. O SVM teve o pior desempenho neste aspecto, com 4.367 falhas não detectadas (22,67%), mais de uma em cada cinco falhas passou despercebida. Em aplicações de robótica, o *recall* é a métrica mais crítica, pois um falso negativo significa que o robô continua operando sem saber que está deslocalizado, podendo resultar em colisões ou comportamento errático (EDER; REIP; STEINBAUER, 2022; KNITT *et al*., 2023).

4.5 IMPORTÂNCIA DAS CARACTERÍSTICAS

A análise de importância das características foi realizada para *Random Forest* e XGBoost, que fornecem nativamente essa medida pela redução média de impureza nas árvores (Breiman, 2001) (Chen; Guestrin, 2016). O SVM e o MLP não oferecem esse recurso de forma direta sem técnicas adicionais como SHAP ou LIME (Eder; Frering; SteinbauerWagner, 2023). A Figura 3 apresenta a importância das 22 características para ambos os modelos.

FIGURA 3 – IMPORTÂNCIA DAS CARACTERÍSTICAS POR MODELO (RANDOM FOREST E XGBOOST)



FONTE: O autor (2026)

As quatro características mais importantes em ambos os modelos pertencem ao grupo de dispersão das partículas (particle_std_y, particle_std_x, particle_spread e particle_std_yaw), somando 52,65% no RF e 44,89% no XGBoost. Esse resultado confirma que a dispersão da nuvem de partículas é o principal indicador de falha, coerente com a interpretação física do filtro AMCL (Thrun; Burgard; Fox, 2005) e com as conclusões de Eder, Frering e SteinbauerWagner (2023). As características de pose (amcl_x, amcl_y) ocuparam a 5ª e 6ª posição no RF, indicando que certas regiões do mapa são mais propensas a falhas. As características do LiDAR contribuem de forma complementar (~20% do total), sendo que lidar_max_ratio e lidar_min apresentaram importância zero, sugerindo redundância.

4.6 ANÁLISE POR TIPO DE AMBIENTE

Para avaliar como o tipo de ambiente influencia o desempenho dos modelos, o conjunto de teste foi segmentado em três categorias: *warehouse* (ambiente grande com geometria rica), simétrico (três mapas de pequena escala com layouts repetitivos) e assimétrico (três mapas de pequena escala com layouts variados). A Tabela 4 apresenta o F1Score de cada modelo por categoria.

TABELA 4 – F1SCORE POR TIPO DE AMBIENTE

Modelo	Warehouse	Simétrico	Assimétrico
Random Forest	0,9772	0,9539	0,9824
SVM	0,6791	0,6541	0,6822
MLP	0,9428	0,8893	0,9484
XGBoost	0,9719	0,9460	0,9665

FONTE: O autor (2026)

Os mapas simétricos representaram o maior desafio para todos os modelos, com queda consistente no F1Score (RF: 0,9539 contra 0,9824 no assimétrico). A simetria cria ambiguidades nas leituras do LiDAR, dificultando tanto a localização quanto a detecção de falhas (Knitt *et al*., 2023) (Knitt *et al*., 2025a). Ainda assim, o *recall* permaneceu alto mesmo nos mapas simétricos (RF: 0,9886). Os ambientes assimétricos apresentaram o melhor desempenho (F1 até 0,9824 com RF), seguidos pelo *warehouse* (F1 até 0,9772), que possui features geométricas ricas.

4.7 ANÁLISE POR CONFIGURAÇÃO DE OBSTÁCULOS

Além do tipo de ambiente, a configuração de obstáculos também influencia a taxa de falha de localização. Cada experimento do conjunto de dados foi realizado com uma das três configurações: apenas obstáculos dinâmicos (25 esferas com trajetórias aleatórias), apenas obstáculos estáticos (cubos fixos não mapeados) ou ambos simultaneamente. A Tabela 5 apresenta a taxa de falha observada em cada configuração.

TABELA 5 – TAXA DE FALHA POR CONFIGURAÇÃO DE OBSTÁCULOS

Modelo	Amostras	Taxa De Falha
Apenas Dinâmico	117.554	29,06%
Dinâmico + Estático	121.145	23,44%
Sem Obstáculos	69.997	21,19%
Apenas Estático	108.489	17,45%

FONTE: O autor (2026)

Obstáculos dinâmicos geraram a maior taxa de falha (29,06%), pois as leituras do LiDAR se tornam inconsistentes com o mapa estático (Eder; Reip; Steinbauer, 2022) (Knitt *et al*., 2025a). A combinação de ambos os tipos resultou em taxa intermediária (23,44%), sugerindo que obstáculos estáticos podem fornecer referências adicionais que auxiliam parcialmente a localização. Obstáculos apenas estáticos apresentaram a menor taxa (17,45%), possivelmente porque são fixos e o filtro pode se adaptar parcialmente a eles..

5 DISCUSSÃO

Os destaque dos modelos baseados em árvores em dados tabulares é consistente com os *benchmarks* de Grinsztajn, Oyallon e Varoquaux (2022), ShwartzZiv e Armon (2022) e FernándezDelgado *et al*. (2014). A predominância das características de dispersão das partículas na importância dos modelos tem interpretação física clara: quando o filtro AMCL está incerto, as partículas se espalham no espaço de estados (Thrun; Burgard; Fox, 2005), conclusão corroborada por Eder, Frering e SteinbauerWagner (2023) com técnicas de XAI.

No que se refere ao risco de *overfitting*, os elevados valores de F1Score são respaldados pela validação cruzada com 5 *folds*, pela consistência entre diferentes tipos de ambiente e pela correlação fisicamente interpretável entre dispersão das partículas e estado de falha. Probst, Wright e Boulesteix (2019) e Chen e Guestrin (2016) destacam que RF e XGBoost possuem mecanismos inerentes de resistência a *overfitting*. Não obstante, o desempenho pode ser otimista em relação a cenários reais, onde ruído sensorial introduz complexidades não presentes nos dados sintéticos.

A variação de desempenho entre tipos de ambiente destaca a importância de treinar com dados diversificados (Knitt *et al*., 2023; Cadena *et al*., 2016). A metodologia é portável para qualquer robô que utilize AMCL com LiDAR, independente da versão do ROS, sendo necessária apenas adaptação na coleta dos tópicos e retreinamento com dados do ambiente alvo.

Quanto às limitações, os dados sintéticos, embora realistas, podem não capturar toda a complexidade de ambientes reais. A extração de características foi manual; abordagens de aprendizado profundo com dados brutos poderiam capturar padrões adicionais.

6 CONSIDERAÇÕES FINAIS

O presente trabalho comparou quatro algoritmos de aprendizado de máquina para a predição de falhas de localização em robôs móveis autônomos utilizando dados de LiDAR e filtro de partículas AMCL. Os principais achados são:

a) a predição de falhas de localização com aprendizado de máquina é viável e eficaz, com F1Score superior a 0,95 para os melhores modelos, utilizando características estatísticas extraídas dos dados do filtro de partículas e do LiDAR;

a) os modelos baseados em árvores apresentaram superioridade: *Random Forest* obteve o melhor F1Score (0,9604) na avaliação fixa, enquanto o XGBoost liderou na validação cruzada (F1 = 0,9012 +/- 0,0099) com o menor tempo de treino (2,35s). Ambos são recomendados para aplicações práticas, sendo o XGBoost preferível quando o tempo de inferência é crítico (0,26s contra 0,84s);
c) as quatro características mais importantes são todas de dispersão das partículas (particle_std_y, particle_std_x, particle_spread, particle_std_yaw), somando mais de 50% da importância total, confirmando que o estado do filtro de partículas é o principal indicador de falha de localização;
d) todos os modelos apresentaram desempenho inferior em ambientes simétricos (F1 de 0,9539 vs 0,9772 no *warehouse* para o RF), indicando a necessidade de estratégias específicas para esses cenários.

Trabalhos futuros incluem: avaliação com dados reais para validar a transferência simulaçãorealidade; uso de abordagens de aprendizado profundo (redes convolucionais ou recorrentes) diretamente nos dados brutos do LiDAR; integração do modelo preditivo no loop de navegação do robô para ações corretivas automáticas, como relocalização forçada, redução de velocidade ou aumento temporário do número de partículas; exploração de técnicas de aprendizado por transferência entre ambientes; e validação da portabilidade do modelo em robôs reais com diferentes configurações de hardware (LiDAR 2D/3D, diferentes resoluções) e software (ROS 1 Noetic, ROS 2 Humble/Jazzy), avaliando a necessidade de ajuste fino com dados do domínio alvo.

O códigofonte completo para replicação do experimento está disponível em: <https://github.com/HobbiesProfBento/mllocalizationfailureprediction>.

7 REFERÊNCIAS

BENTÉJAC, C.; CSÖRGŐ, A.; MARTÍNEZMUÑOZ, G. A comparative analysis of gradient boosting algorithms. **Artificial Intelligence Review**, v. 54, p. 19371967, 2021. Disponível em: https://doi.org/10.1007/s10462020098965. Acesso em: 09 fev. 2025.

BIAU, G.; SCORNET, E. A random forest guided tour. **TEST**, v. 25, n. 2, p. 197227, 2016. Disponível em: https://doi.org/10.1007/s1174901604817. Acesso em: 09 fev. 2025.

BREIMAN, L. Random Forests. **Machine Learning**, v. 45, n. 1, p. 532, 2001. Disponível em: https://doi.org/10.1023/A:1010933404324. Acesso em: 09 fev. 2025.

CADENA, C.; CARLONE, L.; CARRILLO, H.; LATIF, Y.; SCARAMUZZA, D.; NEIRA, J.; REID, I.; LEONARD, J. J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the RobustPerception Age. **IEEE Transactions on Robotics**, v. 32, n. 6, p. 13091332, 2016. Disponível em: https://doi.org/10.1109/TRO.2016.2624754. Acesso em: 09 fev. 2025.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. SMOTE: Synthetic Minority Oversampling Technique. **Journal of Artificial Intelligence Research**, v. 16, p. 321357, 2002. Disponível em: https://doi.org/10.1613/jair.953. Acesso em: 09 fev. 2025.

CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, p. 785794, 2016. Disponível em: https://doi.org/10.1145/2939672.2939785. Acesso em: 09 fev. 2025.

CORTES, C.; VAPNIK, V. Supportvector networks. **Machine Learning**, v. 20, n. 3, p. 273297, 1995. Disponível em: https://doi.org/10.1007/BF00994018. Acesso em: 09 fev. 2025.

EDER, M.; REIP, M.; STEINBAUER, G. Creating a robot localization monitor using particle filter and machine learning approaches. **Applied Intelligence**, v. 52, p. 69556969, 2022. Disponível em: https://doi.org/10.1007/s10489020021576. Acesso em: 09 fev. 2025.

EDER, M.; FRERING, L.; STEINBAUERWAGNER, G. Gaining Insights into a Robot Localization Monitor Using Explainable Artificial Intelligence. In: **Advances in Service and Industrial Robotics (RAAD 2023)**. Mechanisms and Machine Science, v. 135. Springer, 2023. Disponível em: https://doi.org/10.1007/9783031326066_20. Acesso em: 09 fev. 2025.

FERNÁNDEZDELGADO, M.; CERNADAS, E.; BARRO, S.; AMORIM, D. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? **Journal of Machine Learning Research**, v. 15, n. 90, p. 31333181, 2014. Disponível em: https://jmlr.org/papers/v15/delgado14a.html. Acesso em: 09 fev. 2025.

FOX, D. Adapting the Sample Size in Particle Filters Through KLDSampling. **The International Journal of Robotics Research**, v. 22, n. 12, p. 9851003, 2003. Disponível em: https://doi.org/10.1177/0278364903022012001. Acesso em: 09 fev. 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016. ISBN: 9780262035613. Disponível em: https://www.deeplearningbook.org/. Acesso em: 09 fev. 2025.

GRINSZTAJN, L.; OYALLON, E.; VAROQUAUX, G. Why do treebased models still outperform deep learning on tabular data? In: **Advances in Neural Information Processing Systems (NeurIPS)**, v. 35, 2022. Disponível em: https://arxiv.org/abs/2207.08815. Acesso em: 09 fev. 2025.

HSU, C.W.; CHANG, C.C.; LIN, C.J. A Practical Guide to Support Vector Classification. **Technical Report**, National Taiwan University, 2003. Disponível em: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf. Acesso em: 09 fev. 2025.

KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. In: **Proceedings of the 3rd International Conference on Learning Representations (ICLR)**, 2015. Disponível em: https://arxiv.org/abs/1412.6980. Acesso em: 09 fev. 2025.

KNITT, M.; MAROOFI, S.; ROSE, H.; BRAUN, P. Benchmarking for the Indoor Localization of Autonomous Mobile Robots in Intralogistics**. Logistics Journal: Proceedings**, 2023. DOI: 10.2195/lj_proc_knitt_de_202310_01. Disponível em: https://www.logisticsjournal.de/archive/proceedings/2023/5809/knitt_de_2023.pdf. Acesso em: 09 fev. 2025.

KNITT, M.; MAROOFI, S.; THAKKAR, M.; ROSE, H.; BRAUN, P. Synthetic Datasets for Data-Driven Localization Monitoring. **Logistics Journal: Proceedings**, 2025a. DOI: 10.2195/lj_proc_knitt_en_202503_01. Disponível em: https://proc.logisticsjournal.de/article/view/1186. Acesso em: 09 fev. 2025.

KNITT, M.; THAKKAR, M. B.; MAROOFI, S.; ROSE, H. W.; BRAUN, P. M**. Robot Localization Failure Prediction Dataset**. Hamburg:, 2025b. DOI: 10.15480/882.15836. Disponível em: https://tore.tuhh.de/entities/product/71944c77d6f34ecf9dce805e51b3a57f. Acesso em: 09 fev. 2025.

MACENSKI, S.; MOORE, T.; LU, D. V.; MERZLYAKOV, A.; FERGUSON, M. From the Desks of ROS Maintainers: A Survey of Modern & Capable Mobile Robotics Algorithms in the Robot Operating System 2. **Robotics and Autonomous Systems**, v. 168, 104493, 2023. Disponível em: https://doi.org/10.1016/j.robot.2023.104493. Acesso em: 09 fev. 2025.

NVIDIA. **NVIDIA Isaac Sim**. 2025. Disponível em: https://developer.nvidia.com/isaac/sim. Acesso em: 09 fev. 2025.

PRECHELT, L. Early Stopping – But When? In: MONTAVON, G.; ORR, G. B.; MÜLLER, K.R. (Eds.). **Neural Networks: Tricks of the Trade**. 2. ed. Lecture Notes in Computer Science, v. 7700. Springer, 2012. Disponível em: https://doi.org/10.1007/9783642352898_5. Acesso em: 09 fev. 2025.

PROBST, P.; WRIGHT, M. N.; BOULESTEIX, A.L. Hyperparameters and tuning strategies for random forest. **WIREs Data Mining and Knowledge Discovery**, v. 9, n. 3, e1301, 2019. Disponível em: https://doi.org/10.1002/widm.1301. Acesso em: 09 fev. 2025.

SHWARTZZIV, R.; ARMON, A. Tabular data: Deep learning is not all you need. **Information Fusion**, v. 81, p. 8490, 2022. Disponível em: https://doi.org/10.1016/j.inffus.2021.11.011. Acesso em: 09 fev. 2025.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. Cambridge: MIT Press, 2005. ISBN: 9780262201629.