

Avaliação de Desempenho de Modelos de ML para IoMT em uma Plataforma ARM de Borda

José Clebson¹, Marcos Vinícius², Rodrigo Cesar Lira³, Gustavo Callou⁴,
Verlayne Araújo², Diogo Lopes², Eduardo Tavares¹, Thiago Bezerra²

¹Centro de Informática,
Universidade Federal de Pernambuco, Recife, Brasil

²Instituto Federal de Educação, Ciências e Tecnologia de Pernambuco (IFPE)
Campus Palmares – PE – Brasil

³Instituto Federal de Educação, Ciências e Tecnologia de Pernambuco (IFPE)
Campus Paulista – PE – Brasil

⁴Departamento de Computação,
Universidade Federal Rural de Pernambuco, Recife, Brasil

{jcso2,eagt}@cin.ufpe.br, gustavo.callou@ufrpe.br

rodrigo.lira@paulista.ifpe.edu.br, mvvm@discente.ifpe.edu.br

{verlayne.araujo,diogo.lopes,thiago.bezerra}@palmares.ifpe.edu.br

Abstract. *Edge Artificial Intelligence (Edge AI) enables local data processing for latency-sensitive healthcare applications. In Ambient Assisted Living (AAL), tasks such as fall detection and facial recognition benefit from on-device inference, since cloud-only solutions may introduce delay. This paper presents an experimental study of Machine Learning (ML) models deployed on an ARM-based edge platform, the Orange Pi Plus, for fall detection and facial recognition. The analysis considers classification metrics, training time, and hardware measurements, including CPU usage, memory consumption, and operating temperature. The results show that tree-based models provide the best balance between predictive performance and computational cost for fall detection, while facial recognition attains high accuracy with different ML configurations.*

Resumo. *A Inteligência Artificial de Borda (Edge AI) permite o processamento local de dados para aplicações de saúde sensíveis à latência. Em ambientes de assistência à vida independente (AAL), tarefas como detecção de quedas e reconhecimento facial se beneficiam da inferência no próprio dispositivo, visto que soluções baseadas exclusivamente em nuvem podem introduzir atrasos. Este trabalho apresenta um estudo experimental de modelos de aprendizado de máquina (ML) implantados em uma plataforma de borda baseada em ARM, a Orange Pi Plus, para detecção de quedas e reconhecimento facial. A análise considera métricas de classificação, tempo de treinamento e medições de hardware, incluindo uso de CPU, consumo de memória e temperatura de operação. Os resultados mostram que modelos baseados em árvores oferecem o melhor equilíbrio entre desempenho preditivo e custo computacional na detecção de quedas, enquanto o reconhecimento facial atinge alta acurácia com diferentes configurações de ML.*

1. Introdução

A Internet das Coisas Médicas (IoMT) reúne dispositivos médicos, sensores e infraestruturas de comunicação voltados ao monitoramento contínuo de pacientes e ao suporte a serviços de saúde conectados [Valentim et al. 2024]. Aplicações relacionadas à segurança do paciente e ao apoio à vida independente têm recebido atenção em cenários de *Ambient Assisted Living* (AAL). Entre essas aplicações estão a detecção de quedas e o reconhecimento facial para controle de acesso e apoio à verificação de procedimentos, ambas dependentes de processamento adequado e de comunicação confiável entre sensores, dispositivos e nós computacionais [Rashidi and Mihailidis 2012, Bezerra et al. 2025].

Boa parte das soluções nessa área ainda adota uma organização centrada em nuvem, na qual os dados capturados são enviados a servidores remotos para processamento. Embora essa abordagem ofereça capacidade computacional elevada, ela pode introduzir atraso, ampliar o uso da rede e gerar preocupações relacionadas à privacidade dos dados [Aceto et al. 2018, Shi et al. 2016].

Com a adoção de dispositivos de borda em aplicações IoMT, a escolha do modelo de aprendizado de máquina passa a envolver mais do que o desempenho preditivo. Em plataformas com recursos limitados, o comportamento do modelo durante o treinamento e a inferência também influencia sua adequação ao cenário de uso, sobretudo em dispositivos com resfriamento passivo e operação contínua [Sze et al. 2017, Wang et al. 2020].

A maioria dos estudos em saúde na borda concentra-se em uma única tarefa, como detecção de quedas ou reconhecimento facial, e enfatiza métricas de classificação sem discutir as implicações dessas escolhas no dispositivo de execução [Ouloul et al. 2025]. Quando a análise de hardware é incluída, ela costuma tratar uma única natureza de entrada, o que limita a generalização das conclusões.

Em cenários AAL realistas, contudo, o mesmo nó de borda pode precisar executar tarefas que partem de representações de entrada distintas: *embeddings* produzidos por modelos pré-treinados, típicos do reconhecimento facial, e vetores de atributos derivados diretamente de sinais brutos, típicos da detecção de quedas a partir de aceleração ou silhuetas. Essa distinção é relevante porque diferentes representações induzem diferentes geometrias no espaço de atributos, o que pode favorecer classes específicas de algoritmos e produzir trade-offs de custo computacional não evidentes quando uma única tarefa é considerada isoladamente.

Este trabalho apresenta uma avaliação experimental de modelos clássicos de aprendizado de máquina (k-NN, SVM-RBF, Decision Tree e Random Forest) em uma plataforma ARM de borda (Orange Pi Plus), considerando duas tarefas de IoMT que representam regimes distintos de representação de entrada: reconhecimento facial baseado em *embeddings* de modelo pré-treinado e detecção de quedas a partir de vetores de atributos extraídos de sinais brutos. A avaliação combina métricas de classificação (acurácia, F1-score, AUC) com medições de hardware (uso de CPU, consumo de RAM e temperatura) e tempo de treinamento, em três tamanhos de conjunto de dados.

As principais contribuições do trabalho são as seguintes: (i) uma avaliação da interação entre representações de entrada e classificadores de aprendizado de máquina, considerando os trade-offs entre desempenho preditivo e custo computacional; (ii) uma caracterização conjunta do desempenho preditivo, tempo de treinamento e perfil térmico-

computacional para diferentes tarefas executadas em uma plataforma de borda com refrigeração passiva; e (iii) um conjunto de diretrizes para apoiar a seleção de modelos em aplicações IoMT de borda com restrições computacionais.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados e discute as principais abordagens encontradas na literatura. A Seção 3 descreve a metodologia adotada, incluindo os cenários avaliados, as configurações experimentais e os parâmetros considerados. A Seção 4 apresenta os resultados obtidos e suas respectivas discussões. Por fim, a Seção 5 apresenta as conclusões do trabalho e as perspectivas para pesquisas futuras.

2. Trabalhos relacionados

Esta seção apresenta trabalhos recentes relacionados ao uso de aprendizado de máquina em detecção de quedas e reconhecimento facial em plataformas de borda aplicadas à saúde. Yacchirema et al. [Yacchirema et al. 2019] propõem um sistema de detecção de quedas para idosos em ambientes internos, inserido em um contexto de IoT, no qual algoritmos de aprendizado de máquina são utilizados para melhorar a identificação de quedas.

Shu e Shu [Shu and Shu 2021] apresentam um sistema de detecção de quedas baseado em múltiplas câmeras, executado em uma plataforma local de baixo custo, no qual padrões de queda são analisados por técnicas de ML. Min et al. [Min et al. 2018] utilizam uma abordagem baseada em Support Vector Machine (SVM) para reconhecimento de quedas a partir de informações esqueléticas, explorando descritores derivados da ação humana para separar eventos de queda de movimentos normais.

Giri et al. [Giri et al. 2025] apresentam uma solução de detecção de quedas em tempo real baseada em YOLOv5 executada em Raspberry Pi, com foco em Edge AI para monitoramento em saúde. No reconhecimento facial, um sistema voltado para computação de borda combina técnicas de aprendizado com poucos exemplos e aumento de dados para construir uma solução de baixo custo [Jiménez Bravo et al. 2022].

Ouloul et al. [Ouloul et al. 2025] apresentam um sistema embarcado de reconhecimento facial baseado no descritor *overlapped local binary pattern*, considerando a relação entre acurácia e tempo de execução em uma arquitetura com processador ARM. Pecolt et al. [Pecolt et al. 2025] descrevem um sistema de identificação pessoal baseado em Raspberry Pi, com processamento local e ênfase em uma plataforma embarcada de baixo custo.

A Tabela 1 resume os estudos considerados nesta revisão. A maioria trata de uma única tarefa, detecção de quedas ou reconhecimento facial, e quando inclui análise da plataforma de execução, restringe-se a um único tipo de representação de entrada. Comparações diretas entre os valores reportados devem ser interpretadas com cautela, dado que datasets, divisões treino/teste e modelos diferem entre os trabalhos. A diferença deste trabalho está em avaliar, no mesmo dispositivo ARM de borda, duas tarefas que representam regimes distintos de representação de entrada (*embeddings* pré-treinados e vetores derivados de sinais brutos), reunindo métricas de classificação, tempo de treinamento e medições de hardware. Os resultados obtidos são compatíveis em ordem de grandeza com os reportados na literatura para tarefas equivalentes em dispositivos embarcados, e a análise comparativa de trade-offs é apresentada nas Seções 4.2 e 4.3.

Tabela 1. Trabalhos relacionados: escopo, plataforma e desempenho.

Trabalho	Face	Queda	Plataforma	Resultado reportado
[Yacchirema et al. 2019]	-	✓	Wearable + Gateway IoT	N/D
[Shu and Shu 2021]	-	✓	Android Box	Acur. 0,89
[Min et al. 2018]	-	✓	PC + Kinect v2.0	Acur. 0,921 (TST V2)
[Giri et al. 2025]	-	✓	Raspberry Pi 4B	F1 0,981
[Jiménez Bravo et al. 2022]	✓	-	Raspberry Pi 4	Acur. 0,910 (LFW)
[Ouloul et al. 2025]	✓	-	ARM	Acur. 0,989
[Pecolt et al. 2025]	✓	-	Raspberry Pi 3 B+	~10,5 s (tempo médio)
Este trabalho	✓	✓	Orange Pi Plus	Face: Acur. 0,982 / AUC 0,999 Queda: F1 0,864 / AUC 0,971

Valores reportados pelos próprios autores nos respectivos artigos. Dataset e protocolo experimental variam entre os trabalhos.

3. Metodologia

Esta seção descreve um fluxo metodológico composto por seis etapas para avaliar modelos de aprendizado de máquina em uma plataforma de borda com restrição de recursos, considerando os contextos de reconhecimento facial e detecção de quedas. A visão geral do processo é apresentada na Figura 1.

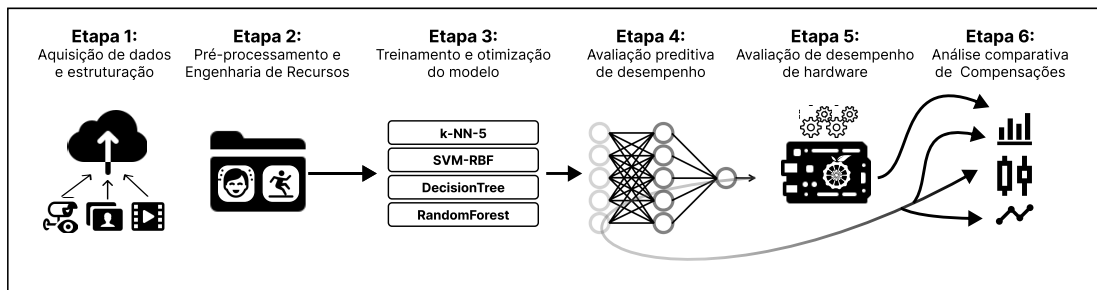


Figura 1. Metodologia

A Etapa 1 corresponde à aquisição e à estruturação dos dados. Nessa etapa, são selecionados conjuntos de dados públicos de modo a favorecer a reprodutibilidade e a comparação com outros estudos. Para o reconhecimento facial, utilizam-se repositórios de imagens, como o VGGFace2, enquanto a detecção de quedas é baseada em sequências de imagens organizadas em amostras rotuladas.

A Etapa 2 realiza o pré-processamento e a engenharia de atributos, convertendo os dados brutos em representações adequadas aos classificadores. No reconhecimento facial, são obtidos vetores de características que representam cada face e servem de entrada para os modelos. Na detecção de quedas, as sequências de imagens passam por pré-processamento e são transformadas em vetores de atributos compatíveis com os algoritmos considerados neste estudo. A Etapa 3 abrange o treinamento e a otimização dos modelos. Os algoritmos avaliados incluem abordagens clássicas de aprendizado de máquina, como k-NN, SVM, Decision Tree e Random Forest. Os modelos são treinados e testados com divisão estratificada dos dados, por exemplo, na proporção 70/30, e os hiperparâmetros são ajustados por meio de validação cruzada k -fold, adotando-se o F1-score como principal critério de seleção.

A Etapa 4 trata da avaliação de desempenho preditivo. Após o treinamento e o ajuste dos modelos, a análise é realizada sobre um conjunto de teste separado, não utilizado na etapa de otimização. Nessa fase, são calculadas métricas usuais de classificação, como F1-score e área sob a curva ROC (AUC) para problemas binários, além de acurácia e macro F1-score para problemas multiclasse. A Etapa 5 corresponde à avaliação de desempenho de hardware. Os modelos treinados são implantados na plataforma de borda alvo, uma placa Orange Pi, e, durante a inferência contínua, são monitorados o uso de CPU, o consumo de memória RAM e a temperatura do processador, com o objetivo de quantificar o custo computacional de cada configuração. Por fim, a Etapa 6 reúne os resultados de classificação e as medições de hardware para analisar a relação entre desempenho preditivo e uso de recursos, permitindo comparar as compensações observadas entre as abordagens avaliadas.

4. Estudo de caso

Esta seção apresenta o estudo de caso no qual modelos de ML são implantados em uma placa Orange Pi Plus para apoiar duas tarefas de saúde baseadas em imagens: reconhecimento facial e detecção de quedas. A placa executa Linux (aarch64) e possui processador ARM de 8 núcleos, com frequência máxima de 2,26 GHz, e 16 GB de RAM. A temperatura de operação é um indicador relevante nesse tipo de dispositivo com resfriamento passivo, pois cargas sustentadas podem elevar a temperatura, acionar *thermal throttling* e degradar o desempenho. As métricas de sistema (uso de CPU, consumo de RAM e temperatura) foram coletadas com as bibliotecas Python `psutil` e `gpiozero`, com amostragem de 1 segundo durante a execução dos scripts de inferência.

A avaliação utiliza conjuntos de dados públicos para ambas as tarefas: VGG-Face2 [Cao et al. 2018] para reconhecimento facial e SisFall [Sucerquia et al. 2017] para detecção de quedas. Para cada tarefa, foram definidos três conjuntos com tamanhos crescentes, denominados D1, D2 e D3. Essa estratégia permite avaliar a sensibilidade de cada algoritmo ao volume de dados disponível, aspecto relevante em cenários de borda nos quais a quantidade de amostras pode variar conforme o tempo de operação ou a disponibilidade de armazenamento local. Os conjuntos de reconhecimento facial contêm 2,997, 5,996 e 11,991 amostras, enquanto os de detecção de quedas contêm 5,000, 10,000 e 40,000 amostras. Todos os modelos são treinados e, em seguida, executados na Orange Pi Plus, de modo que tanto as métricas preditivas quanto o comportamento de hardware sejam medidos na plataforma de borda alvo.

O restante desta seção analisa três aspectos do estudo de caso: o desempenho preditivo dos modelos, o tempo de treinamento no dispositivo e o consumo de recursos de hardware durante a inferência.

4.1. Desempenho dos modelos de ML

A Tabela 2 apresenta o desempenho de classificação dos modelos de ML nos três tamanhos de conjunto de dados para cada tarefa. A avaliação considera acurácia, recall, F1-score e AUC, métricas que, em conjunto, descrevem a eficácia de cada modelo na discriminação entre classes sob as restrições do dispositivo de borda.

Na tarefa de reconhecimento facial, que utiliza embeddings como entrada, os modelos clássicos de ML alcançam bons níveis de acurácia na Orange Pi Plus. No maior

Tabela 2. Desempenho dos modelos para cada tarefa.

Conjunto	Modelo	Acur.	Recall	F1	AUC	Tarefa
D1	k-NN-5	0.980	–	0.980	0.993	Face
D1	SVM-RBF	0.980	–	0.980	0.999	Face
D1	DecisionTree	0.846	–	0.845	–	Face
D1	RandomForest	0.982	–	0.982	0.997	Face
D2	k-NN-5	0.970	–	0.970	0.992	Face
D2	SVM-RBF	0.977	–	0.977	0.999	Face
D2	DecisionTree	0.767	–	0.768	–	Face
D2	RandomForest	0.973	–	0.973	0.997	Face
D3	k-NN-5	0.976	–	0.975	0.993	Face
D3	SVM-RBF	0.982	–	0.982	0.999	Face
D3	DecisionTree	0.695	–	0.695	–	Face
D3	RandomForest	0.974	–	0.974	0.998	Face
<hr/>						
D1	k-NN-5	0.873	0.833	0.847	0.943	Fall
D1	SVM-RBF	0.781	0.526	0.669	0.897	Fall
D1	DecisionTree	0.832	0.776	0.796	0.825	Fall
D1	RandomForest	0.900	0.795	0.870	0.965	Fall
D2	k-NN-5	0.893	0.756	0.815	0.936	Fall
D2	SVM-RBF	0.850	0.587	0.709	0.887	Fall
D2	DecisionTree	0.871	0.788	0.792	0.849	Fall
D2	RandomForest	0.911	0.750	0.840	0.966	Fall
D3	k-NN-5	0.912	0.825	0.862	0.956	Fall
D3	SVM-RBF	0.845	0.607	0.723	0.907	Fall
D3	DecisionTree	0.854	0.785	0.782	0.837	Fall
D3	RandomForest	0.916	0.801	0.864	0.971	Fall

Acur. = Acurácia, Recall = Sensibilidade, F1 = F1-score, AUC = Área sob a curva ROC.

conjunto de dados (D3), SVM com kernel RBF e Random Forest obtêm acurácias de 0,982 e 0,974, respectivamente, seguidos por k-NN-5 com 0,976. O desempenho uniforme entre modelos distintos pode ser atribuído à qualidade dos embeddings, pois os vetores de características são gerados por um modelo pré-treinado (InceptionResnetV1) e já carregam informações discriminativas das faces, de modo que classificadores mais simples também conseguem separar bem as classes. A exceção é o DecisionTree, cuja acurácia cai de 0,846 em D1 para 0,695 em D3, o que indica sobreajuste a partições locais do espaço de atributos que se tornam menos representativas à medida que o número de identidades cresce.

A tarefa de detecção de quedas apresenta um comportamento distinto. Os modelos baseados em árvores apresentam os melhores resultados, com o Random Forest atingindo o maior F1-score (0,864) e a maior acurácia (0,916) em D3, enquanto a Decision Tree entrega resultados competitivos com menor complexidade. O k-NN-5 também apresenta bom desempenho (F1 de 0,862), ao passo que o SVM-RBF obtém F1-score inferior (0,723). Esse resultado pode ser explicado pela natureza dos atributos extraídos das imagens de queda: ao contrário dos embeddings faciais, esses vetores tendem a conter relações não lineares e interações entre variáveis que são melhor capturadas por partições hierárquicas (árvores) do que por uma superfície de decisão única (SVM).

Além disso, o desempenho do SVM-RBF sugere que o kernel RBF, embora seja adequado para embeddings faciais de alta dimensionalidade, não se adapta tão bem à distribuição dos atributos de queda. O Random Forest se apresenta como uma escolha adequada para detecção de quedas na borda, combinando recall de 0,801 e AUC de 0,971, aspecto relevante em aplicações de saúde nas quais uma queda não detectada pode trazer consequências ao monitoramento.

A Figura 2 complementa esses resultados com as curvas ROC no conjunto D3. Na Figura 2(a), referente ao reconhecimento facial, as curvas de SVM-RBF, Random Forest e k-NN-5 permanecem próximas da região superior esquerda e apresentam AUC de 0,999, 0,998 e 0,993, respectivamente, indicando elevada capacidade de discriminação. Na Figura 2(b), referente à detecção de quedas, as curvas aparecem mais dispersas. Nessa tarefa, o Random Forest apresenta a maior AUC (0,971), seguido de k-NN-5 (0,956), SVM-RBF (0,907) e Decision Tree (0,837). Esse comportamento indica maior dificuldade na separação entre as classes em comparação ao reconhecimento facial. Em particular, a Decision Tree apresenta a menor AUC, o que está de acordo com seu desempenho inferior nessa tarefa.

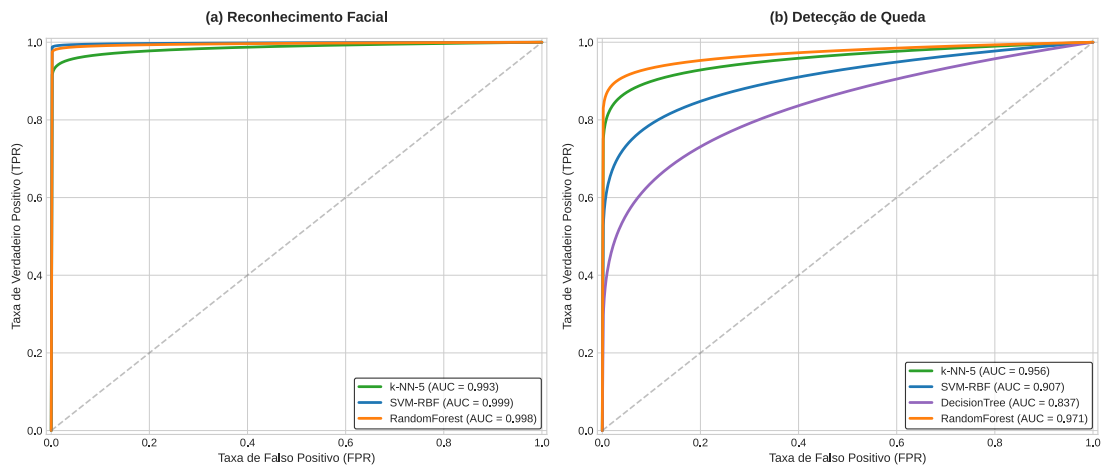


Figura 2. Curvas ROC

4.2. Tempo de treinamento no dispositivo de borda

Além do desempenho preditivo, o tempo de treinamento na Orange Pi Plus é um aspecto relevante em cenários que envolvem atualização local dos modelos. O tempo foi medido diretamente na placa para cada modelo e conjunto de dados. A Figura 3 apresenta esses resultados.

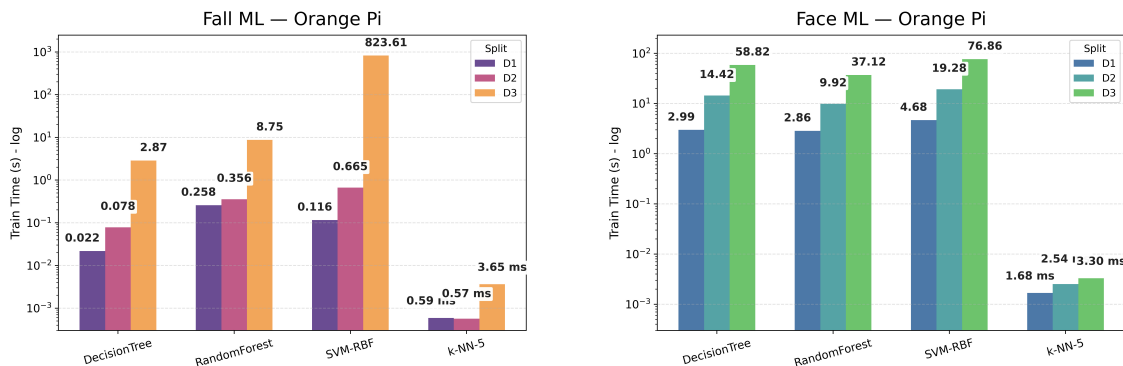


Figura 3. Tempo de treinamento dos modelos

No reconhecimento facial, o tempo de treinamento aumenta com o tamanho do conjunto de dados. No conjunto D3, o SVM-RBF apresentou o maior tempo, com 76,96 s,

seguido por Decision Tree, com 58,82 s, e Random Forest, com 37,12 s. O k-NN-5 manteve tempo reduzido, variando de 1,68 ms em D1 para 3,30 ms em D3, pois seu treinamento está associado principalmente ao armazenamento dos embeddings. Esses valores mostram que, nessa tarefa, modelos com desempenho preditivo semelhante podem demandar custos de atualização bastante diferentes na borda.

Na detecção de quedas, a diferença entre os modelos foi mais acentuada. No conjunto D3, o SVM-RBF atingiu 823,61 s, enquanto o Random Forest registrou 8,75 s e a Decision Tree 2,87 s. O k-NN-5 novamente apresentou o menor tempo, com valores entre 0,57 ms e 3,65 ms. Esse comportamento indica que, para essa tarefa, a escolha do modelo interfere de forma mais clara no custo de treinamento, especialmente quando o volume de dados cresce. Em cenários nos quais o sistema precise ser atualizado periodicamente no próprio dispositivo, essa diferença deve ser considerada em conjunto com as métricas preditivas.

4.3. Perfil de uso de hardware e trade-offs

A última parte do estudo de caso concentra-se no comportamento de hardware durante a inferência na Orange Pi Plus. Uso de CPU, alocação de memória e temperatura foram monitorados para cada tarefa. A Tabela 3 resume os resultados obtidos.

Tabela 3. Métricas de hardware

Tarefa	CPU (%)		Memória (MB)		Temperatura (°C)	
	Média	Máx.	Média	Máx.	Média	Máx.
Reconhecimento Facial	26,7	100,0	685,5	1145,3	56,4	60,0
Detecção de Quedas	14,2	70,8	617,8	697,4	49,9	52,0

Os resultados de uso de CPU mostram que a tarefa de reconhecimento facial impõe maior carga ao processador, com utilização média de 26,7% e picos de 100%. A detecção de quedas apresenta demanda inferior, com média de 14,2% e pico de 70,8%. Essa diferença está relacionada à natureza da entrada: o reconhecimento facial opera sobre embeddings gerados por um modelo pré-treinado, enquanto a detecção de quedas utiliza vetores de atributos mais compactos, resultando em menor custo por inferência. O consumo de memória segue padrão semelhante. O reconhecimento facial demanda maior uso médio de RAM (685,5 MB), com picos acima de 1 GB, refletindo o carregamento de modelos de extração de embeddings e do pipeline completo. A detecção de quedas opera com ocupação mais contida (617,8 MB em média), sem picos expressivos.

As medições de temperatura reforçam a análise. O reconhecimento facial apresenta a maior temperatura média (56,4 °C), coerente com o maior uso de CPU e memória. A detecção de quedas opera com temperatura média de 49,9 °C, compatível com operação contínua sem risco de *thermal throttling*. Esses valores indicam que ambas as tarefas são viáveis para execução sustentada no dispositivo, embora o reconhecimento facial exija mais atenção ao gerenciamento térmico. Essas métricas oferecem uma visão indireta do consumo de energia, já que o uso sustentado de CPU é a principal fonte de demanda de potência e dissipação térmica no dispositivo. Medições elétricas diretas exigiriam instrumentação específica e foram deixadas para trabalhos futuros.

4.4. Discussão

Os resultados das Seções 4.1 a 4.3 permitem uma análise integrada dos modelos avaliados nas duas tarefas. O SVM-RBF apresentou comportamento distinto entre os cenários analisados. No reconhecimento facial, em que a entrada corresponde a *embeddings* de 512 dimensões gerados pela InceptionResnetV1, o SVM-RBF alcança acurácia de 0,982 e AUC de 0,999 no conjunto D3, com tempo de treinamento de 76,96 s. Na detecção de quedas, em que a entrada consiste em atributos derivados diretamente de sinais brutos, o mesmo modelo apresenta F1-score de 0,723 e tempo de treinamento de 823,61 s, aproximadamente 94 vezes superior ao Random Forest (8,75 s) na mesma tarefa.

O kernel RBF apresentou resultados mais adequados quando aplicado a representações previamente estruturadas, como *embeddings* gerados por modelos pré-treinados. Em atributos derivados de sinais brutos, o modelo apresentou maior custo computacional e menor desempenho preditivo. Os modelos baseados em árvores mantiveram desempenho mais estável entre as tarefas, com tempos de treinamento variando entre 2,87 s e 37,12 s no conjunto D3. Dessa forma, a seleção do modelo em ambientes de borda deve considerar também o tipo de representação de entrada utilizado pela aplicação.

Para analisar conjuntamente desempenho preditivo e custo computacional, foi definido o índice de eficiência $E = M / (1 + \log_{10}(1 + t))$, em que M representa o F1-score na detecção de quedas ou a acurácia no reconhecimento facial, enquanto t corresponde ao tempo de treinamento em segundos. O termo logarítmico no denominador reduz o impacto de pequenas diferenças em tempos baixos e aumenta a penalização em modelos com tempos elevados. No conjunto D3 da detecção de quedas, os valores de E foram 0,86 para o k-NN-5, 0,49 para a Decision Tree, 0,43 para o Random Forest e 0,18 para o SVM-RBF.

No reconhecimento facial em D3, os resultados foram 0,98 para o k-NN-5, 0,38 para o Random Forest, 0,34 para o SVM-RBF e 0,25 para a Decision Tree. Embora o SVM-RBF apresente maior acurácia no reconhecimento facial, o custo computacional associado ao treinamento reduz sua eficiência em cenários com necessidade de atualização local do modelo. O índice proposto não substitui a análise multidimensional apresentada na Tabela 3, mas fornece um critério complementar para comparações entre desempenho e custo computacional em aplicações IoMT de borda.

Os resultados também permitem estabelecer algumas diretrizes para cenários IoMT semelhantes. Quando a entrada do classificador é composta por *embeddings* gerados por modelos pré-treinados, classificadores mais simples, como o k-NN, podem apresentar desempenho próximo ao de modelos mais complexos, com menor tempo de treinamento.

Em aplicações baseadas em atributos derivados de sinais brutos, modelos baseados em árvores, principalmente o Random Forest, apresentaram melhor relação entre desempenho preditivo e custo computacional quando comparados ao SVM-RBF. Além disso, a tarefa de reconhecimento facial apresentou maior demanda computacional, atingindo pico de 100% de uso de CPU e temperatura média de 56,4 °C. Esse comportamento deve ser considerado em plataformas com resfriamento passivo, devido às limitações térmicas durante execuções prolongadas.

A análise foi conduzida em uma única plataforma, a Orange Pi Plus, e utilizando

os datasets VGGFace2 e SisFall, o que limita a generalização dos resultados para outras plataformas ARM, como Raspberry Pi e Jetson Nano, bem como para outras aplicações IoMT. As medições de hardware consideraram métricas agregadas de CPU, memória e temperatura, sem instrumentação elétrica dedicada para análise de potência. Além disso, a análise estatística foi realizada a partir dos valores agregados das execuções, não incluindo intervalos de confiança e testes de significância estatística, aspectos que podem ser considerados em trabalhos futuros.

5. Conclusão

Este trabalho apresentou uma avaliação experimental de modelos clássicos de aprendizado de máquina em uma plataforma ARM de borda (Orange Pi Plus), considerando duas tarefas representativas de IoMT que partem de regimes distintos de representação de entrada: reconhecimento facial baseado em *embeddings* de modelo pré-treinado e detecção de quedas a partir de vetores derivados de sinais brutos. A análise considerou, de forma conjunta, métricas de classificação, tempo de treinamento e o comportamento computacional e térmico da plataforma.

O principal achado é que a interação entre a representação de entrada e o classificador escolhido produz trade-offs assimétricos entre desempenho preditivo e custo computacional. O SVM com kernel RBF, que se destaca em acurácia para o reconhecimento facial, é a pior escolha na detecção de quedas, com F1-score inferior e tempo de treinamento aproximadamente duas ordens de grandeza maior que o dos modelos baseados em árvores. Random Forest e k-NN, por sua vez, apresentam desempenho consistente em ambas as tarefas e oferecem melhor relação custo-benefício para cenários de borda com restrição de recursos. A partir desse achado, foram propostos um índice de eficiência (E) e diretrizes de seleção de modelos para aplicações IoMT de borda.

A avaliação restringe-se a uma única placa ARM e a dois datasets, sem medição elétrica direta de potência. Como trabalhos futuros, pretende-se ampliar a avaliação para outras plataformas embarcadas, como Raspberry Pi e Jetson Nano, incorporar fluxos contínuos de dados, realizar medições diretas de potência por meio de instrumentação dedicada e estender a análise estatística com múltiplas execuções, intervalos de confiança e testes de significância.

Referências

- Aceto, G., Persico, V., and Pescapé, A. (2018). The role of information and communication technologies in healthcare: taxonomies, perspectives, and challenges. *Journal of Network and Computer Applications*, 107:125–154.
- Bezerra, T., Vinicius, M., Ciane, A., Callou, G., França, C., and Tavares, E. (2025). An approach based on iot and machine learning for monitoring patients on healthcare centers. In *Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)*, pages 260–271. SBC.
- Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE.
- Giri, A., Hasib, A., Islam, M., Tazim, M. F., Rahman, M. S., Khadgi, M., and Akib, A. S. (2025). Real-time human fall detection using yolov5 on raspberry pi: An edge

- ai solution for smart healthcare and safety monitoring. In *International Conference on Data Analytics & Management*, pages 493–507. Springer.
- Jiménez Bravo, D. M., Lozano Murciego, A., Sales Mendes, A., Hernández de la Iglesia, D., Silva, L. A., et al. (2022). Edge face recognition system based on one-shot augmented learning. *International Journal of Interactive Multimedia and Artificial Intelligence*, 7(6):31–44.
- Min, W., Yao, L., Lin, Z., and Liu, L. (2018). Support vector machine approach to fall recognition based on simplified expression of human skeleton action and fast detection of start key frame using torso angle. *IET Computer Vision*, 12(8):1133–1140.
- Ouloul, M. I., Moutakki, Z., Amghar, A., and Afdel, K. (2025). Low-cost embedded facial recognition system based on overlapped local binary pattern. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 11:100924.
- Pecolt, S., Błazejewski, A., Królikowski, T., Maciejewski, I., Gierula, K., and Glowinski, S. (2025). Personal identification using embedded raspberry pi-based face recognition systems. *Applied Sciences*, 15(2):887.
- Rashidi, P. and Mihailidis, A. (2012). A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579–590.
- Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646.
- Shu, F. and Shu, J. (2021). An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box. *Scientific reports*, 11(1):2471.
- Sucerquia, A., López, J. D., and Vargas-Bonilla, J. F. (2017). Sisfall: A fall and movement dataset. *Sensors*, 17(1):198.
- Sze, V., Chen, Y.-H., Yang, T.-J., and Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329.
- Valentim, T., Callou, G., França, C., and Tavares, E. (2024). Availability and performance assessment of iomt systems: A stochastic modeling approach. *Journal of Network and Systems Management*, 32(4):95.
- Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., and Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. *IEEE communications surveys & tutorials*, 22(2):869–904.
- Yacchirema, D., de Puga, J. S., Palau, C., and Esteve, M. (2019). Fall detection system for elderly people using iot and ensemble machine learning algorithm. *Personal and Ubiquitous Computing*, 23(5):801–817.