PAULO HENRIQUE TRINDADE

ANÁLISE DE DESEMPENHO DE ALGORITMOS DE MACHINE LEARNING PARA A DETECÇÃO DE QUEDAS DE PESSOAS USANDO DADOS DO REPOSITÓRIO DO MUNDO REAL FARSEEING

Itaja´ı (SC), Fevereiro de 2026

UNIVERSIDADE DO VALE DO ITAJAÍ

PRÓ-REITORIA DE PÓS-GRADUAÇÃO, PESQUISA, EXTENSÃO E CULTURA

PROGRAMA DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA

ANÁLISE DE DESEMPENHO DE ALGORITMOS DE MACHINE LEARNING PARA A DETECÇÃO DE QUEDAS DE PESSOAS USANDO DADOS DO REPOSITÓRIO DO MUNDO REAL FARSEEING

por

Paulo Henrique Trindade

Dissertação apresentada como requisito parcial a` obtenção do grau de Mestre em Computação Aplicada.
Orientador(a): Alejandro Rafael Garcia Ramirez, Dr.

Itaja´ı (SC), Fevereiro de 2026

"Análise de desempenho de algoritmos de Machine Learning para a detecção de quedas de pessoas usando dados do repositório do mundo real FARSEEING"

Paulo Henrique Trindade

Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Computação Aplicada, Área de Concentração Computação Aplicada, e aprovada em sua forma final pelo Programa de Mestrado Acadêmico em Computação Aplicada da Universidade do Vale do Itajaí - UNIVALI.

Jordan Sausen Passinato Assinado de forma digital por Jordan Passinato Sausen 2026.03.14 16:31:07

Prof. Jordan Passinato Sausen, Dr.
Coordenador do Curso de Mestrado em Computação Aplicada

Apresentado perante a Banca Examinadora composta pelos Professores:

Documento assinado digitalmente
ALEJANDRO RAFAEL GARCIA RAMIREZ
Data: 02/03/2026 13:19:09-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Alejandro Rafael Garcia Ramirez
Orientador

Documento assinado digitalmente
gov.br
RAIMUNDO CELESTE GHIZONI TEIVE
Data: 02/03/2026 15:12:24-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Raimundo Celeste Ghizoni Teive (UNIVALI)
Membro

Documento assinado digitalmente
RODRIGO SANTANA
Data: 02/03/2026 13:37:53-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Rodrigo Sant' Ana (UNIVALI)
Membro

DE PAZ SANTANA JUAN FRANCISCO - 761257540
Firmado digitalmente por DE PAZ SANTANA JUAN FRANCISCO - 761257540
Fecha: 2026.03.02 17:25:18 +01'00'

Prof. Dr. Juan Francisco De Paz Santana (Universidad de Salamanca)
Membro externo

“A verdadeira ignorância não é a ausência de conhecimento, mas a recusa em adquiri-lo.”
. Karl Popper

AGRADECIMENTOS

Primeiramente, à minha família, que esteve ao meu lado durante todo o percurso deste trabalho, oferecendo apoio incondicional mesmo diante de inúmeros desafios. Entre eles, destacam-se momentos delicados como a gestação, as complicações do período pós-parto e a inevitável escassez de tempo que marcou esta etapa. Ainda assim, juntos, soubemos nos fortalecer, superar as dificuldades e transformar esse processo em uma conquista compartilhada. Sem esse suporte emocional e familiar, a conclusão deste trabalho não teria sido possível.

Ao Professor Dr. Alejandro Rafael Garcia Ramirez, por confiar em mim e por me propor este desafio acadêmico. Sua orientação foi fundamental para minha formação científica, conduzindo-me com rigor, ética e incentivo pelo caminho do conhecimento. Graças à sua orientação, pude compreender a profundidade e a gratificação da pesquisa científica, culminando no reconhecimento deste trabalho em um evento internacional.

Ao Professor Dr. Rodrigo Sant'ana, pela excelência na condução de sua disciplina, que ultrapassou o formato tradicional de apresentações didáticas e proporcionou uma imersão prática no universo da análise exploratória de dados. Sua abordagem aprofundada e aplicada foi determinante para o amadurecimento técnico deste estudo, tendo contribuição direta e significativa nos resultados alcançados e no reconhecimento acadêmico obtido.

A todos os participantes do projeto FARSEEING, incluindo cientistas, gestores de estudo/dados e equipes clínicas/administrativas, por possibilitarem este estudo.

Por fim, ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e ao Ministério da Ciência, Tecnologia e Inovações (MCTI) pelo financiamento deste trabalho e pelo apoio à participação presencial no evento internacional realizado no Peru. O recurso concedido viabilizou não apenas a apresentação dos resultados da pesquisa, mas também proporcionou uma experiência de grande enriquecimento cultural, acadêmico e de networking, ampliando de forma significativa minha visão sobre a pesquisa científica e sua inserção no contexto internacional.

ANÁLISE DE DESEMPENHO DE ALGORITMOS DE MACHINE LEARNING PARA A DETECÇÃO DE QUEDAS DE PESSOAS USANDO DADOS DO REPOSITÓRIO DO MUNDO REAL FARSEEING

Paulo Henrique Trindade

Fevereiro / 2026

Orientador: Alejandro Rafael Garcia Ramirez, Dr.

Área de Concentração: Computação Aplicada

Linha de Pesquisa: Tecnologias Assistivas

Palavras-chave: farseeing, quedas do mundo real, machine learning.

Número de páginas: 113

RESUMO

A detecção automática de quedas em pessoas idosas representa um desafio relevante no contexto da saúde assistiva, especialmente devido ao envelhecimento populacional e aos riscos associados a esse tipo de evento. Quedas estão entre as principais causas de hospitalização, perda de autonomia e mortalidade em idosos, tornando essencial o desenvolvimento de soluções capazes de identificá-las de forma confiável. Nesse contexto, este trabalho tem como objetivo avaliar o desempenho de algoritmos de aprendizagem de máquina na detecção de quedas a partir de dados reais coletados por sensores de acelerômetro triaxial, considerando as limitações e os desafios impostos por cenários do mundo real. A principal motivação do estudo reside no fato de que grande parte das pesquisas existentes utiliza dados simulados ou combina dados simulados e reais, os quais não refletem plenamente a complexidade, a variabilidade e o desbalanceamento característicos de eventos reais de queda.

A metodologia adotada envolveu a utilização de uma base composta por sinais de acelerômetros provenientes de eventos reais de queda, submetidos a etapas de pré-processamento, organização temporal, interpolação, padronização e janelamento dos sinais. Foram implementados e avaliados diferentes algoritmos de aprendizagem de máquina, incluindo *Decision Tree*, *Balanced Random Forest*, *Support Vector Machine*, *K-Nearest Neighbors*, redes neurais convolucionais unidimensionais (*1D-CNN*) e redes recorrentes do tipo *Long Short-Term Memory*. A avaliação do desempenho foi conduzida por meio de métricas adequadas a problemas com classes desbalanceadas, incluindo matriz de confusão, sensibilidade, especificidade, acurácia balanceada, valores preditivos positivo e negativo, coeficiente Kappa, taxa sem informação e testes estatísticos complementares. Além da divisão inicial entre treino e teste, foi aplicada validação repetida com diferentes partições dos dados, com o objetivo de avaliar a estabilidade dos modelos e reduzir a dependência de uma única separação amostral.

Os resultados indicaram desempenho elevado dos modelos baseados em árvores de decisão. Na avaliação por partição *holdout*, o *Balanced Random Forest* apresentou desempenho perfeito, com sensibilidade, especificidade, acurácia balanceada e coeficiente Kappa iguais a 1,0000, enquanto a *Decision Tree* alcançou sensibilidade de 100%, especificidade de 96,43%, acurácia balanceada de 98,21% e Kappa de 0.9153. Para avaliar a estabilidade do *Balanced Random Forest* em diferentes separações dos dados, o modelo foi submetido a uma validação repetida com dez partições independentes, preservando a estrutura de agrupamento dos eventos. Nessa configuração, o modelo manteve sensibilidade e valor preditivo negativo de 100%, sem qualquer falso negativo, porém com ocorrência de falsos positivos — isto é, atividades cotidianas classificadas incorretamente como quedas —, resultando em acurácia balanceada de 97,86%, especificidade de 95,71% e coeficiente Kappa de 0,8993, valores ligeiramente inferiores aos da partição única, o que é esperado dado o protocolo de validação mais rigoroso.

Dessa forma, a escolha do *Balanced Random Forest* como estratégia mais adequada não decorre exclusivamente do maior desempenho pontual, mas de sua maior aderência metodológica ao problema investigado, caracterizado por forte desbalanceamento entre classes, número reduzido de eventos reais de queda e necessidade de minimizar falsos negativos. A manutenção de 100% de sensibilidade ao longo da validação repetida reforça sua adequação para aplicações de saúde assistiva, nas quais a não detecção de uma queda representa o erro de maior impacto prático. Assim, os resultados demonstram que algoritmos de aprendizagem de máquina, especialmente aqueles capazes de lidar com desbalanceamento de classes, podem contribuir para o desenvolvimento de sistemas mais confiáveis de detecção automática de quedas a partir de dados reais.

ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR FALL DETECTION USING REAL-WORLD DATA FROM THE FARSEEING REPOSITORY

Paulo Henrique Trindade

February / 2026

Advisor: Alejandro Rafael Garcia Ramirez, Dr.

Area of Concentration: Applied Computer Science

Research Line: Assistive Technologies

Keywords: farseeing, real-world falls, machine learning.

Number of pages: 113

ABSTRACT

The automatic detection of falls in elderly individuals represents a significant challenge in the context of assistive healthcare, particularly due to population aging and the risks associated with such events. Falls are among the leading causes of hospitalization, loss of autonomy, and mortality in older adults, making the development of reliable detection solutions essential. In this context, this study aims to evaluate the performance of machine learning algorithms for fall detection using real-world data collected by triaxial accelerometer sensors, considering the limitations and challenges imposed by real-world scenarios. The primary motivation for this study lies in the fact that much of the existing research relies on simulated data or combines simulated and real data, which do not fully capture the complexity, variability, and class imbalance characteristics of actual fall events.

The methodology involved the use of a dataset comprising accelerometer signals from real fall events, subjected to preprocessing steps including temporal organization, interpolation, standardization, and signal windowing. Several machine learning algorithms were implemented and evaluated, including Decision Tree, Balanced Random Forest, Support Vector Machine, K-Nearest Neighbors, one-dimensional convolutional neural networks (1D-CNN), and Long Short-Term Memory recurrent networks. Model performance was assessed using metrics suited to imbalanced class problems, including confusion matrix, sensitivity, specificity, balanced accuracy, positive and negative predictive values, Cohen's Kappa coefficient, no-information rate, and complementary statistical tests. In addition to the initial train-test split, repeated validation across multiple data partitions was performed to assess model stability and reduce dependence on a single sample division.

The results indicated high performance among tree-based models. Under holdout evaluation, the Balanced Random Forest achieved perfect performance, with sensitivity, specificity, balanced accuracy, and Cohen's Kappa all equal to 1.0000, while the Decision Tree reached 100% sensitivity,

96.43% specificity, 98.21% balanced accuracy, and a Kappa of 0.9153. To assess the stability of the Balanced Random Forest across different data splits, the model was further evaluated through repeated validation with ten independent partitions, preserving the grouping structure of fall events. In this setting, the model maintained 100% sensitivity and negative predictive value, with no false negatives, but with the occurrence of false positives — that is, everyday activities incorrectly classified as falls — yielding a balanced accuracy of 97.86%, specificity of 95.71%, and a Kappa coefficient of 0.8993, values slightly lower than those obtained under the single holdout partition, as expected given the more rigorous validation protocol.

The selection of the Balanced Random Forest as the most appropriate strategy was therefore not based solely on peak performance, but on its stronger methodological alignment with the problem at hand, characterized by severe class imbalance, a limited number of real fall events, and the need to minimize false negatives. The consistent maintenance of 100% sensitivity throughout repeated validation reinforces its suitability for assistive health applications, in which failure to detect a fall represents the error with the greatest practical consequence. These findings demonstrate that machine learning algorithms — particularly those designed to handle class imbalance — can contribute to the development of more reliable systems for the automatic detection of falls from real-world data.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Acc	*Acurácia*
ADLs	*Activities of Daily Living*
CAPES	*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*
CE	*Critério de Exclusão*
CI	*Critério de Inclusão*
EBSCO	*Elton Bryson Stephens Company*
FARSEEING	*FAll Repository for the design of Smart and sElf-adaptive Environments prolonging INdependent liviNG*
FallAllD	*A Comprehensive Dataset of Human Falls and Activities of Daily Living*
FN	*False Negative*
FP	*False Positive*
HMMs	*Hidden Markov Models*
IEEE	*Institute of Electrical and Electronics Engineers*
KNN	*K-Nearest Neighbors*
LPV	*Valor de pico inferior antes do UPV*
LSTM	*Long Short-Term Memory*
MEMS	*Micro-Electromechanical Systems*
ML	*Machine Learning*
MLP	*Perceptron Multicamadas*
NPV	*Negative Predictive Value*
PPV	*Positive Predictive Value*
RF	*Random Forest*
RNNs	*Recurrent Neural Networks*
SMOTE	*Synthetic Minority Over-sampling Technique*
SVM	*Support Vector Machine*
TN	*True Negative*

TP	*True Positive*
UPV	*Valor de pico superior da aceleração*
WHO	*World Health Organization*

SUMÁRIO

1 INTRODUÇÃO

O corpo humano, com o passar dos anos, gradualmente vai se enfraquecendo, resultando em uma progressiva perda de sua saúde física. Essa evolução pode desencadear novas problemáticas e desafios para pessoas com baixa mobilidade, em especial os idosos, chegando a um estágio em que se faz necessária uma abordagem mais cautelosa. Indivíduos com mais de 60 anos, nesse contexto, tornam-se progressivamente mais suscetíveis a incidentes de queda (WHO, 2021).

National Institute for Health and Care Excellence (2024) destaca que a identificação precoce do risco de quedas, associada ao monitoramento contínuo e à implementação de tecnologias assistivas, pode contribuir significativamente para a redução da incidência de lesões graves em idosos. Essas diretrizes enfatizam a importância da integração entre avaliação clínica, análise de fatores ambientais e utilização de tecnologias digitais para monitoramento da mobilidade e detecção de eventos críticos, reforçando o potencial de sistemas baseados em sensores e algoritmos de inteligência artificial como ferramentas complementares no cuidado e na prevenção de quedas em indivíduos vulneráveis .

Quedas representam um risco para pessoas de todas as idades, gêneros e regiões geográficas. É observado, em diversos países, que os homens tendem a ter maior probabilidade de óbito em decorrência de quedas, enquanto as mulheres, por sua vez, sofrem mais quedas não fatais. Grupos particularmente vulneráveis incluem mulheres idosas e crianças pequenas, que estão mais sujeitas a quedas e a lesões de maior gravidade. Globalmente, os homens apresentam taxas mais elevadas de mortalidade e de anos de vida ajustados por lesões ocasionadas pelas quedas sofridas. Essa diferença pode ser explicada, em parte, pela maior exposição a comportamentos de risco e pela presença de fatores ocupacionais mais perigosos (WHO, 2021).

Segundo WHO (2021) as quedas são a segunda principal causas de mortes não intencionais no mundo. Estima-se que a cada ano 684 mil pessoas sofrem acidentes fatais, se concentrando 80% desses números em países de de baixa e média renda. No desafio relacionado à detecção de quedas, desdobram-se diversos aspectos, como variáveis fisiológicas, comportamentais e ambientais. As atuais soluções voltadas à detecção e antecipação desses eventos concentram-se majoritariamente em aspectos fisiológicos, tais como estilo de caminhada, acuidade da visão e cognição. Ressalta-se a carência de interfaces de usuário eficazes e mecanismos de feedback hábeis para a detecção de quedas (RAJAGOPALAN; LITVAN; JUNG, 2017).

Estudos recentes também reforçam a importância do uso de tecnologias baseadas em sensores e análise de dados para compreender e prevenir quedas em populações idosas. A utilização de sensores vestíveis e registros contínuos de movimento pode fornecer informações mais precisas sobre a dinâmica das quedas em contextos do mundo real, permitindo compreender melhor os mecanismos envolvidos nesses eventos. Além disso, a análise sistemática desses dados contribui para o desenvolvimento de estratégias mais eficazes de prevenção e para o aprimoramento de sistemas automatizados de detecção de quedas, possibilitando intervenções mais rápidas e potencialmente reduzindo o impacto clínico e social desses incidentes (HORSFALL et al., 2024).

Destaca-se os custos econômicos, pois o custo médio para o sistema de saúde por cada lesão sofrida por uma pessoa com 65 anos ou mais devido a uma queda é em torno de US$ 3.611 na Finlândia e em torno de US$ 1.049 na Austrália. Segundo dados do Canadá, estratégias preventivas eficazes podem reduzir a incidência de quedas entre crianças menores de 10 anos em 20%, com uma economia líquida de mais de US$ 120 milhões por ano (WHO, 2021).

Um outro aspecto determinante é a falta de dados relacionados ao evento de queda (KLENK; SCHWICKERT, 2016). Em sua grande maioria, as quedas acontecem e a própria pessoa não tem a capacidade de relatar com precisão como o fato ocorreu. Nesse intuito, tecnologias vestíveis podem ser um diferencial na coleta de informações, estando presentes nas pessoas mais vulneráveis aos eventos de queda (HALE; DELANEY; CABLE, 1993).

Diante do aumento da suscetibilidade a quedas em indivíduos com mais de 60 anos, este trabalho visa avaliar algoritmos de inteligência artificial, utilizando dados reais, para detectar quedas na população idosa. Por conta da dificuldade de se capturar um evento de queda, diversas bases de dados simuladas foram criadas, contudo estas não utilizam dados de quedas reais, nem de usuários reais, propensos às quedas.

Este trabalho utiliza a base de dados FARSEEING, que reúne informações fisiológicas, comportamentais e ambientais coletadas em tempo real de usuários propensos às quedas, e provenientes de cenários reais de vida cotidiana. A partir dessa base, busca-se implementar algoritmos de aprendizado de máquina, como a Árvore de Decisão (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Redes Convolucionais de uma dimensão (1D-CNN) e Long Short-Term Memory (LSTM), dentre outros, na detecção dos eventos de quedas, avaliando o desempenho destes através de métricas como acurácia balanceada, sensibilidade, especificidade valor preditivo positivo dentre outras. O estudo também considera o impacto do desbalanceamento de classes, comum em eventos raros como quedas, e propõe ajustes metodológicos para lidar com esse desafio. Futuramente,

os resultados poderão ser incorporados ao desenvolvimento de sistemas vestíveis de monitoramento em tempo real e estratégias preventivas, promovendo maior segurança, autonomia e qualidade de vida para populações vulneráveis, como idosos e pessoas com mobilidade reduzida.

1.1 PROBLEMA DE PESQUISA

Segundo a Organização Mundial da Saúde (WHO, 2021), pessoas idosas tornam-se progressivamente mais suscetíveis a quedas ao longo do envelhecimento. Evidências apontam que, após a primeira ocorrência, a probabilidade de um novo episódio aumenta em aproximadamente 30%, configurando um ciclo de risco crescente. As taxas de internação hospitalar decorrentes de quedas entre indivíduos com 60 anos ou mais variam entre 1,6 e 3,0 por 10.000 habitantes em países como Austrália, Canadá e Reino Unido da Grã-Bretanha e Irlanda do Norte. Em determinadas regiões, como a Austrália Ocidental e os Estados Unidos, esses valores são ainda mais elevados, situando-se entre 5,5 e 8,9 por 10.000 habitantes da população total, o que evidencia a gravidade do problema em escala global.

Além do impacto direto na saúde física, as quedas podem ocasionar lesões graves, hospitalizações prolongadas, perda de autonomia funcional e aumento significativo da mortalidade, configurando-se como um problema de saúde pública de caráter universal (WHO, 2021). Nesse contexto, estratégias tecnológicas capazes de identificar, classificar e compreender padrões associados a eventos de queda tornam-se fundamentais para subsidiar ações preventivas, otimizar respostas clínicas e apoiar o desenvolvimento de soluções assistivas.

Com o avanço dos dispositivos vestíveis e a disponibilidade de grandes repositórios de dados provenientes de sensores inerciais, surgem oportunidades para a aplicação de técnicas de aprendizado de máquina na detecção e classificação automática de quedas. Em especial, o repositório FARSEEING destaca-se por conter dados de quedas do mundo real, coletados em condições próximas às de uso cotidiano, o que representa um desafio adicional e, ao mesmo tempo, uma oportunidade relevante para validação de algoritmos em cenários realistas.

Diante desse contexto, este trabalho busca contribuir cientificamente para a área de detecção automática de quedas por meio da análise e aplicação de algoritmos de Machine Learning sobre dados reais. Assim, formula-se a seguinte questão de pesquisa:

1. Será possível classificar, com uma **sensibilidade** acima de 90%, as quedas relatadas no repositó-

rio de quedas do mundo real da FARSEEING usando algoritmos de Machine Learning?

1.1.1 Solução Proposta

A investigação compreendeu a avaliação do desempenho de algoritmos de *Machine Learning*. Para a validação dos algoritmos, foi utilizada a base de dados FARSEEING, que contém informações sobre quedas de usuários reais, conforme descrito por Klenk e Schwickert (2016). O objetivo principal consistiu em identificar e aplicar os algoritmos com melhor desempenho relatados na literatura para a análise de séries temporais, utilizando dados provenientes de sensores que monitoram um cenário realista do fenômeno de queda. Para alcançar esse objetivo, a base de dados foi submetida a uma série de testes por meio dos algoritmos previamente selecionados, conforme o estudo apresentado em (NUÑEZ, 2023).

Decision Tree (DT) – Segundo SCIKIT-LEARN (2025a), a árvore de decisão é um algoritmo de aprendizado supervisionado utilizado em tarefas de classificação e regressão, que realiza a segmentação recursiva do espaço de atributos por meio de testes lógicos do tipo $x_j \leq t$, onde x_j. O modelo constrói uma estrutura hierárquica de decisão, na qual cada nó interno representa uma condição sobre um atributo e cada folha corresponde a uma classe ou valor predito. As árvores de decisão destacam-se pela interpretabilidade e pela capacidade de modelar relações não lineares, sendo amplamente utilizadas como modelos base em métodos de aprendizado em conjunto, como o Random Forest.

Random Forest (RF) - Segundo Nielsen (2020) é um combinado de arvores de decisão, e em sua classificação ou regressão é resultado das médias dessas árvores considerando "sabedoria da multidão"que é formada por diversos modelos de menor complexidade que, juntos,possibilitam encontrar os melhores resultados.

K-Nearest Neighbors (KNN) - Tambem conhecido por "K Visinhos mais Próximos", é um algoritmo classificador de aprendizado supervisionado que pode ser aplicado para classificação e regressão. O valor de K define o número dos vizinhos mais próximos (IBM, 2025b).

Support Vector Machine (SVM) - Segundo SCIKIT-LEARN (2025) é um método de aprendizado de máquina utilizado em tarefas de classificação, regressão e detecção de *outliers*. O SVM é conhecido por sua eficácia em espaços de alta dimensionalidade e pela robustez ao lidar com dados não linearmente separáveis.(NOBLE, 2006)

Convolutional Neural Network unidimensional (1D-CNN) – As redes neurais convolucionais

unidimensionais são uma adaptação das CNNs tradicionais para o processamento de sinais temporais, nas quais as operações de convolução são aplicadas ao longo de uma única dimensão. Segundo Kiranyaz et al. (2021), as 1D-CNNs são particularmente adequadas para a análise de séries temporais de sensores, pois permitem a extração automática de características locais e invariantes no tempo, reduzindo a necessidade de engenharia manual de atributos. Em aplicações de detecção de quedas, esse tipo de arquitetura tem se mostrado eficaz na identificação de padrões dinâmicos associados a eventos abruptos, como impactos e mudanças rápidas de aceleração (ORDóñEZ; ROGGEN, 2016).

Long Short-Term Memory (LSTM) - Projetado para superar as limitações das RNNs convencionais. A LSTM é capaz de armazenar informações por períodos prolongados, sendo essa habilidade útil para prever e analisar séries temporais com intervalos desconhecidos entre eventos, onde métodos como *Hidden Markov model* HMMs e *Recurrent neural network* (RNNs) tradicionais apresentam dificuldades.

A avaliação dos algoritmos foi conduzida por meio do uso de métricas como: acurácia, intervalo de confiança de 95% (IC 95%), taxa sem informação (*No Information Rate — NIR*) e o respectivo p-valor para o teste *[Acc > NIR]*, coeficiente Kappa de Cohen, p-valor do teste de McNemar, sensibilidade (*recall*) e especificidade, além dos valores preditivos positivo (*PPV*) e negativo (*NPV*), prevalência, taxa de detecção, prevalência detectada e acurácia balanceada. Essas métricas permitiram uma comparação detalhada do desempenho de cada algoritmo na detecção de quedas, fornecendo uma base sólida para identificar aquele que apresentou melhor desempenho em um cenário real, bem como possibilitar a comparação dos resultados obtidos com aqueles reportados na literatura relacionada.

A implementação e os testes dos algoritmos, foram realizados utilizando as linguagens de programação *R* e *Python*, amplamente reconhecidas por sua extensa gama de bibliotecas e ferramentas voltadas ao *Machine Learning* e a métodos estatísticos. As principais bibliotecas empregadas no desenvolvimento e na avaliação dos modelos incluem:

Scikit-learn: Uma biblioteca robusta para *machine learning* em *Python* que fornece ferramentas simples e eficientes para análise de dados e modelagem preditiva. A *Scikit-learn* inclui implementações de diversos algoritmos de aprendizado supervisionado e não supervisionado, como regressão linear e logística, árvores de decisão, suporte vetorial (SVM), k-means, entre outros. (PEDREGOSA et al., 2011).

TensorFlow e Keras: *TensorFlow* é uma biblioteca de código aberto para computação numérica e Machine Learning. É amplamente utilizada para a construção e treinamento de modelos de

aprendizado profundo (*deep learning*) devido à sua capacidade de lidar com grandes volumes de dados e operações complexas. (ABADI ASHISH AGARWAL, 2015).

Pandas: Com pandas, é possível realizar operações complexas de limpeza, transformação e agregação de dados com poucas linhas de código. A biblioteca é especialmente útil para análises exploratórias de dados, preparação de dados para Machine Learning e integração com outras ferramentas analíticas (TEAM, 2024).

1.1.2 Hipótese

A partir destes elementos, foi concebida a hipótese de pesquisa deste estudo:

H1 - Hipótese.

Algoritmos de Machine Learning aplicados à base de dados de quedas reais FARSEEING, apresentam um desempenho similar, ou superior, quando comparados com similares algoritmos utilizados na literatura para este fim, inclusive, os que usam bases de dados simulados e reais de quedas, conforme identificado na revisão do Estado da Arte.

A hipótese será validada através de simulações, seguindo o método científico. O processo de avaliação incluirá a preparação dos dados, a implementação dos algoritmos, a execução dos testes e a análise dos resultados para determinar a viabilidade da hipótese proposta.

1.1.3 Delimitação de Escopo

Neste trabalho será considerada a base de dados de quedas reais FARSEEING (KLENK; SCHWICKERT, 2016) e os algoritmos RF, SVM e KNN previamente selecionados no trabalho do (NUÑEZ, 2023), adicionando o DT, 1D-CNN e LSTM. Serão utilizadas as linguagens de programação R e Python, junto a bibliotecas e ferramentas dedicadas ao *Machine Learning* e métodos estatísticos. Cabe salientar que não será realizada a implementação em hardware dos algoritmos, apenas será realizado o estudo computacional.

1.1.4 Justificativa

O estudo proposto é relevante devido à crescente necessidade de sistemas eficientes de detecção de quedas, especialmente para pessoas vulneráveis, como os idosos. A queda é uma das principais causas de ferimentos graves nesse público, levando a graves consequências como fraturas, perda de mobilidade e, em muitos casos, mortalidade.

- Qual é a relevância da solução da proposta?

  As quedas entre idosos são um problema de saúde pública. Segundo a WHO (2021), quedas são a segunda causa mundial de mortes acidentais ou não intencionais, especialmente em indivíduos com mais de 60 anos (WHO, 2021). Utilizar dados reais de quedas para desenvolver e avaliar algoritmos de inteligência artificial pode fornecer insights valiosos e precisos, contribuindo significativamente para a prevenção e registro de quedas.

- Qual é a complexidade da solução proposta?

  A solução proposta envolve a análise de dados reais de quedas, a aplicação de algoritmos de aprendizado de máquina e a validação dos modelos preditivos. Os dados das quedas reais são caracterizados por elevada variabilidade interindividual, presença de ruído, dados faltantes e forte desbalanceamento entre as classes, uma vez que eventos de queda são significativamente menos frequentes do que atividades da vida diária. Destaca-se que em um problema relacionado a eventos raros o foco é a classe minoritária, sendo a sensibilidade, acurácia balanceada e a especificidade métricas fundamentais para avaliar o impacto prático de falsos negativos e falsos positivos em sistemas de detecção de quedas.

- Qual é a aplicabilidade da solução?

  A solução encontraria aplicabilidade em ambientes residenciais, lares de idosos e hospitais. A análise de dados reais permitiria a criação de modelos preditivos mais precisos, que podem ser posteriormente desenhados para desenvolver sistemas de monitoramento e alerta que ajudem na detecção de quedas em diversos contextos, beneficiando tanto os idosos quanto os cuidadores.

- A solução é viável?

  A solução é viável devido ao avanço das ferramentas de aprendizado de máquina. Embora a implementação inicial possa exigir um esforço significativo, em termos de coleta e processamento de dados, os benefícios em termos de detecção de quedas e melhoria da qualidade de vida justificam o investimento neste estudo.

- Qual é o seu diferencial a outros similares?

  O diferencial desta proposta, em relação a soluções similares, consiste no uso exclusivo de dados reais de quedas, provenientes de monitoramento em condições não controladas, o que amplia a relevância clínica e a validade externa dos resultados obtidos. Além disso, destaca-se a ampliação da metodologia de investigação, com a aplicação de diferentes estratégias de janelamento centradas no evento, análise multivariada preservando os três eixos do acelerômetro e comparação sistemática entre múltiplos algoritmos de aprendizado supervisionado e arquiteturas de aprendizado profundo. A proposta também incorpora um protocolo de validação mais rigoroso, com controle de dependência por evento, tratamento explícito do desbalanceamento de classes e avaliação baseada em métricas complementares e testes estatísticos, proporcionando uma análise consistente do desempenho dos modelos.

- Qual é o problema que ele irá resolver?

  A solução proposta visa contribuir no Estado da Arte para o problema da alta incidência de quedas na população. Ao utilizar dados provenientes de um cenário real de quedas o estudo pretende fornecer uma base sólida para sistemas que detectem quedas de forma mais eficaz.

- Qual é a motivação para ele?

  A motivação para este trabalho é a carência de soluções eficazes para detectar quedas, um problema que já afeta milhões de pessoas em todo o mundo. A análise de dados provenientes de um cenário real de quedas oferece uma oportunidade singela para melhorar a precisão dos modelos preditivos e desenvolver estratégias mais eficazes de detecção de quedas. Isso não só melhoraria a qualidade de vida, mas também contribuiria a reduzir o impacto econômico e social associado a esses incidentes.

1.2 OBJETIVOS

Esta secção traz de maneira formal os objetivos deste trabalho, conforme descritos nas próximas subseções.

1.2.1 Objetivo Geral

Avaliar o desempenho de algoritmos de aprendizagem de máquina na detecção de quedas usando dados provenientes de um cenário real de quedas.

1.2.2 Objetivos Específicos

1. Implementar e testar os algoritmos de *machine learning* selecionados para a detecção de quedas.
2. Avaliar a acurácia balanceada, sensibilidade e especificidade de cada algoritmo utilizando a base de dados FARSEEING;
3. Comparar o desempenho desses algoritmos em relação com o Estado da Arte;
4. Documentar o processo e os resultados obtidos, destacando as limitações e possíveis melhorias futuras.

1.3 METODOLOGIA

Nesta seção, classifica-se a metodologia utilizada na pesquisa e apresenta-se uma síntese dos procedimentos metodológicos empregados para o desenvolvimento da dissertação.

1.3.1 Metodologia da Pesquisa

A fundamentação metodológica desta pesquisa está alinhada ao método hipotético-dedutivo, conforme discutido por (POPPER, 2004). Esse método caracteriza-se pela formulação de uma hipótese a partir de uma base teórica previamente estabelecida e por sua posterior verificação empírica por meio de experimentos. No contexto deste trabalho, formula-se a hipótese sobre o desempenho de algoritmos de *machine learning* aplicados à detecção de quedas, utilizando uma base de dados proveniente de usuários reais em situações próximas ao cotidiano.

Quanto à sua natureza, esta pesquisa caracteriza-se como aplicada, pois busca contribuir para a solução de um problema prático específico: a detecção automática de quedas. A aplicação de técnicas de *machine learning* sobre dados reais de sensores inerciais visa fornecer subsídios para o desenvolvimento futuro de sistemas de monitoramento e apoio à saúde assistiva, especialmente voltados a pessoas idosas ou com maior vulnerabilidade a eventos de queda.

Quanto à forma de abordagem do problema, a pesquisa é predominantemente quantitativa, uma vez que utiliza dados numéricos provenientes de sensores e avalia o desempenho dos algoritmos por meio de métricas estatísticas. No entanto, a investigação não se limita à mensuração quantitativa dos resultados, pois também envolve etapas de preparação dos dados, tratamento de sinais temporais,

segmentação em janelas, implementação computacional dos modelos, validação experimental e análise comparativa dos resultados obtidos.

Dessa forma, o estudo também apresenta caráter experimental-computacional, uma vez que diferentes algoritmos são implementados, treinados, testados e comparados sob condições controladas de pré-processamento e avaliação. A utilização de dados reais do repositório FARSEEING, associada à comparação entre modelos clássicos e arquiteturas de aprendizado profundo, permite analisar o comportamento dos algoritmos diante de um cenário caracterizado por variabilidade dos sinais, número reduzido de eventos de queda e desbalanceamento entre classes.

Sob o ponto de vista dos objetivos, a pesquisa é descritiva, explicativa e exploratória. É descritiva porque apresenta e organiza o desempenho dos algoritmos avaliados a partir de métricas, matrizes de confusão e resultados experimentais. É explicativa porque busca interpretar os motivos pelos quais determinados modelos apresentam melhor desempenho, especialmente em relação à detecção da classe minoritária, ao controle de falsos negativos e ao equilíbrio entre sensibilidade e especificidade. Também possui caráter exploratório, pois investiga diferentes estratégias de pré-processamento, janelamento, ajuste de hiperparâmetros e validação, buscando identificar configurações mais adequadas ao problema de detecção de quedas em dados reais.

Em síntese, esta pesquisa pode ser classificada da seguinte forma:

- **Quanto ao método:** hipotético-dedutiva, pois parte de uma hipótese formulada com base na literatura e a testa empiricamente por meio de experimentos computacionais.
- **Quanto à natureza:** aplicada, pois busca contribuir para uma solução prática relacionada à detecção automática de quedas em contextos de saúde assistiva.
- **Quanto à abordagem do problema:** predominantemente quantitativa, pois utiliza dados numéricos e métricas estatísticas para avaliar o desempenho dos modelos.
- **Quanto aos procedimentos:** experimental-computacional, pois envolve preparação dos dados, implementação, treinamento, validação e comparação de algoritmos de *machine learning*.
- **Quanto aos objetivos:** descritiva, explicativa e exploratória, pois descreve os resultados obtidos, interpreta o comportamento dos modelos e investiga diferentes estratégias metodológicas para a detecção de quedas.

Essa combinação metodológica permite realizar uma análise sistemática do desempenho dos algoritmos, compreender suas limitações e avaliar sua adequação ao contexto de aplicações em saúde assistiva, respeitando as características e restrições impostas por dados reais de quedas.

1.3.2 Procedimentos Metodológicos

Esta seção apresenta os procedimentos metodológicos executados nesta pesquisa. Eles têm por objetivo definir o caminho para os objetivos propostos neste trabalho.

- **Pesquisa bibliográfica**
  Realizar a identificação do estado da arte, na detecção de quedas utilizando *machine learning*. Examinar estudos relevantes que abordam o tema, utilizando dados de quedas reais e quedas simuladas.

- **Revisão da Bibliografia**
  Detalhar os fundamentos teóricos necessários para o desenvolvimento da pesquisa, com enfase a detecção de eventos raros em séries temporais, algoritmos de *machine learning* (DT, RF, KNN, SVM, 1D-CNN e LSTM) e Séries Temporais.

- **Preparação dos dados**
  Preparar os dados com o objetivo de viabilizar a modelagem de algoritmos de aprendizado de máquina voltados à detecção de quedas. Utilizar o *dataset* público FARSEEING, composto por sinais reais obtidos a partir de sensores inerciais vestíveis. Padronizar os arquivos de entrada mantendo apenas as variáveis de interesse. Filtrar os dados para foco no evento da queda. Identificar padrões temporais relevantes nos sinais de sensores durante os eventos de queda.

- **Treinamento e avaliação**
  Implementar modelos de classificação binária utilizando algoritmos, DT, RF, SVM, KNN, 1D-CNN, LSTM, com apoio das bibliotecas *scikit-learn*,(*Python*) e *Caret*, (R). Avaliar diferentes combinações de parâmetros e técnicas de pré-processamento.

- **Avaliar os resultados**
  Usar validação cruzada para assegurar consistência dos resultados. Comparar os modelos entre si como base nos estudos apresentados no estado da arte, com métricas como acurácia, sensibilidade. especificidade,taxas de Falsos Positivos (FP) e Falsos Negativos (FN).

- **Conclusão**

  Discutir os resultados da pesquisa, pontos de melhoria e trabalhos futuros.

1.4 ESTRUTURA DA DISSERTAÇÃO

Este trabalho esta estruturado em seis capítulos que descrevem, em sequencia, a fundamentação teórica, métodos empregados, os trabalhos relacionados, o projeto, e os resultados parciais alcançados. A seguir um resumo adotado. Capítulo 1, apresenta o contexto da pesquisa, a motivação, o problema investigado, os objetivos geral e específicos, bem como a justificativa e a delimitação do estudo. Capítulo 2, explora os principais conceitos relacionados à pesquisa, abordando os datasets utilizados, os aspectos biomédicos das quedas, os sensores inerciais com destaque para o acelerômetro, e os algoritmos de aprendizado de máquina aplicados à classificação de eventos. Capítulo 3, descreve os métodos empregados ao longo da pesquisa, detalhando a revisão sistemática e bibliográfica, a preparação dos dados, a implementação dos algoritmos, a avaliação dos modelos e a análise dos resultados. Capítulo 4, apresenta os resultados obtidos nas etapas de modelagem, avaliação e testes, discutindo o desempenho dos algoritmos com base nas métricas estabelecidas e comparando-os com trabalhos relacionados. Capítulo 5, expõe as conclusões da pesquisa, os principais aportes científicos e práticos. Capítulo 6, conclusões obtidas neste trabalho, contribuições realizadas e sugestões para trabalhos futuros.

1.5 CONSIDERAÇÕES

Este capítulo introduziu o contexto da pesquisa, tornando evidente a gravidade e complexidade do tema abordado. A partir da identificação do problema abordado, foi possível estabelecer o objetivo da pesquisa, formular a hipótese, delimitar o escopo, justificar a relevância social e científica da proposta. A solução apresentada busca aplicar algoritmos de aprendizado de maquina em um cenário realista, utilizando dados provenientes de quedas reais do repositório FARSEEING, com objetivo de promover avanços no campo de detecção de quedas. Acredita-se que os resultados obtidos poderão subsidiar o desenvolvimento de sistemas mais precisos e sensíveis, contribuindo para segurança, qualidade de vida e segurança de pessoas em situação de vulnerabilidade. A seguir, serão apresentados os fundamentos teóricos que sustentam essa proposta.

2 FUNDAMENTAÇÃO TEÓRICA

Para avançar na pesquisa, é fundamental compreender os elementos que a compõem, iniciando pela análise dos *datasets* utilizados, suas características e estruturação, com foco nos conjuntos de dados FARSEEING e FallAllD. Em seguida, é necessário aprofundar o entendimento sobre a dinâmica das quedas, incluindo os fatores que as provocam, sua representação em planos anatômicos, e a importância de reconhecer seus padrões no contexto do movimento humano.

Outro aspecto essencial é o estudo do sensor utilizado, o acelerômetro, que desempenha papel central na captação dos sinais associados às quedas. Por fim, este capítulo explora os principais algoritmos de aprendizado de máquina aplicados à detecção e classificação de quedas, abordando as técnicas de *Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), 1D-CNN e Long Short-Term Memory (LSTM)*. Esses elementos fornecem a base teórica necessária para o desenvolvimento e a análise do modelo proposto.

2.1 DATASETS

Nesta seção serão apresentados os *datasets* utilizados para desenvolvimento dessa pequisa. O objetivo é compreender as características de cada conjunto. Para isso serão abordados dos *datasets*: o *FARSEEING*, composto por dados de quedas reais; e o *FallAllD*, criado a partir de dados simulados.

2.1.1 FARSEEING

Klenk e Schwickert (2016), descrevem uma meta-base de dados de quedas reais, criada a partir de sensores corporais pelo consórcio FARSEEING *(FAll Repository for the design of Smart and Self-Adaptive Environments prolonging Independent livinG)* e parceiros associados. Entre janeiro de 2012 e dezembro de 2015, mais de 300 quedas reais foram registradas, representando a maior coleção de dados desse gênero até hoje. Para a verificação dos sinais e identificação dos eventos de queda, que resultaram em 208 quedas confirmadas, foram aplicados procedimentos como: diferenciação numérica, utilizada para estimar a aceleração angular; integração numérica, empregada na estimativa da velocidade e do deslocamento nos eixos horizontal e vertical; e o método de rotação baseado em quaternions (*strap-down integration*), utilizado para converter os sinais do sistema de coordenadas local do sensor para o sistema global de referência do corpo.

As quedas foram registradas em diferentes estudos e populações, com a maioria dos sinais de sensores incluindo acelerômetro, e também incluindo o giroscópio e magnetômetro. A base enfatiza a importância desses dados para melhorar a compreensão das quedas e possibilitar novas abordagens na avaliação de risco de queda, prevenção de quedas e detecção de quedas em idosos e pessoas com doenças. Além disso, são descritos os métodos de coleta de dados, incluindo uma revisão sistemática da literatura, a configuração dos sensores, a descrição dos sinais de queda e os procedimentos de processamento de sinal. A base de dados FARSEEING é aberta mediante solicitação.

2.1.2 FallAllD

O conjunto de dados apresentado por Saleh, Abbas e Jeannes (2021), *FallAllD: A Comprehensive Public Dataset for Human Fall Detection and Daily Living Activities*, apresenta uma pesquisa em detecção de quedas, diferenciando-as de atividades diárias (ADLs). Ela traz fornece configurações variadas em posicionamento dos sensores e uma variação nos modos de queda:

- Participantes: O FallAllD é um conjunto de dados abertos, composto por 26.420 arquivos coletados a partir de simulações realizadas por 15 participantes.
- Sensores e Localização: Para capturar as informações de movimento, foram utilizados três *"data-loggers"* (registradores de dados) posicionados estrategicamente em diferentes partes do corpo (cintura, pulso e pescoço):

  Os sensores embarcados nesses dispositivos incluem acelerômetros, giroscópios, magnetômetros e barômetros. A diversidade de localização e variação buscou maior quantidade que informações possível sobre as quedas.
- Diversidade de Atividades: O *dataset* engloba uma ampla gama de atividades, incluindo 19 tipos de ADLs e 15 tipos de quedas simuladas. Informações importantes para treinar modelos de aprendizado de máquina capazes de lidar com a complexidade e a variabilidade dos movimentos humanos no dia a dia.

  A inclusão de dados de participantes de diferentes faixas etárias (jovens adultos, adultos saudáveis com mais de 62 anos e um participante de 60 anos) contribuiram para a representatividade do conjunto.

O trabalho Saleh, Abbas e Jeannes (2021) não só criou o *dataset* como realizou a avaliação com diferentes algoritmos de aprendizado de máquina.

- Alta Performance na Detecção de Quedas: As avaliações de sistemas de detecção de quedas baseados em atividades físicas usando o FallAllD demonstraram altas taxas de sucesso em termos de acurácia, precisão e ganho. Por exemplo, um sistema avaliado no dataset alcançou 100% de sensibilidade na detecção de quedas quando o dispositivo era usado no pescoço, com uma taxa de apenas um falso positivo a cada 25 dias. Para o dispositivo no pulso, a sensibilidade também foi alta, com cerca de um falso positivo a cada 3 dias.
- Melhora na Acurácia: Estudos subsequentes utilizando o FallAllD, como a aplicação de um novo sistema de coordenadas chamado *"ground-face"* (que independe da orientação do dispositivo), demonstraram uma melhoria de aproximadamente 2% na acurácia da detecção de quedas. Isso sugere que o *dataset* permite o desenvolvimento e a validação de abordagens inovadoras.

O FallAllD oferece uma base de dados variada de quedas simuladas e também contribui para o desenvolvimento de soluções e para a segurança de indivíduos, especialmente idosos, ao considerar a complexidade dos movimentos e as variações no posicionamento dos sensores.

2.2 QUEDAS

A Definição de queda pode variar conforme o objetivo do estudo, se tratando de quedas em idosos vamos considerar a definição da Organização Mundial da Saúde (OMS). Uma queda é definida como um evento que resulta na queda inadvertida de uma pessoa no solo, no piso ou em outro nível inferior.

> Lesões relacionadas a quedas podem ser fatais ou não fatais, embora a maioria não seja fatal. Por exemplo, na República Popular da China, para cada morte de crianças devido a uma queda, há 4 casos de incapacidade permanente, 13 casos que requerem hospitalização por mais de 10 dias, 24 casos que requerem hospitalização por 1 a 9 dias e 690 casos que buscam atendimento médico ou faltam ao trabalho/escola. (WHO, 2021)

O envelhecimento da população global apresenta uma série de desafios, sendo a mobilidade um dos aspectos mais críticos para a qualidade de vida dos idosos. Com as estimativas das Nações Unidas apontando para um aumento significativo no número de pessoas com 60 anos ou mais até 2050, é imperativo que os sistemas de suporte e infraestrutura se adaptem a essa nova realidade. A mobilidade não é apenas uma questão de transporte; ela está intrinsecamente ligada ao bem-estar psicológico e à independência dos idosos (RUEDA, 2024).

Nesse contexto, as quedas emergem como um problema sério que afeta diretamente a capacidade de mobilidade e, consequentemente, a qualidade de vida dessa população. As quedas não só podem levar a ferimentos graves, como também impactam a confiança e a autonomia dos idosos, criando um ciclo vicioso que limita ainda mais sua mobilidade (WHO, 2021).

2.2.1 Planos Anatômicos

Os planos anatômicos são linhas imaginárias que servem como referência para dividir o corpo humano em diferentes regiões, Figura 1. Existem três planos, o plano frontal, também chamado de plano coronal, plano sagital e plano horizontal.(CEDUP, 2011).

O **plano coronal**, atravessa o corpo lateralmente, dividindo-o em partes anterior (frontal) e posterior (dorsal), (CAZETTA; OLIVEIRA; TAVARES, 2019). Os movimentos associados a esse plano são: Abdução (afastar um membro da linha média do corpo); Adução (aproximar um membro da linha média do corpo) e Inclinação lateral da cabeça ou tronco.

O **plano sagital**, por sua vez, divide o corpo em lados direito e esquerdo. Ele pode ser imaginado como uma parede vertical que se desloca paralelamente ao plano mediano do corpo. Os movimentos típicos nesse plano são a flexão e a extensão, como quando se dobra ou estica um membro (CAZETTA; OLIVEIRA; TAVARES, 2019).

Já o **plano horizontal**, corta o corpo horizontalmente, separando-o em partes superior e inferior. Nesse plano, o principal tipo de movimento observado é a rotação, comum em articulações como a do pescoço ou do tronco (CAZETTA; OLIVEIRA; TAVARES, 2019).

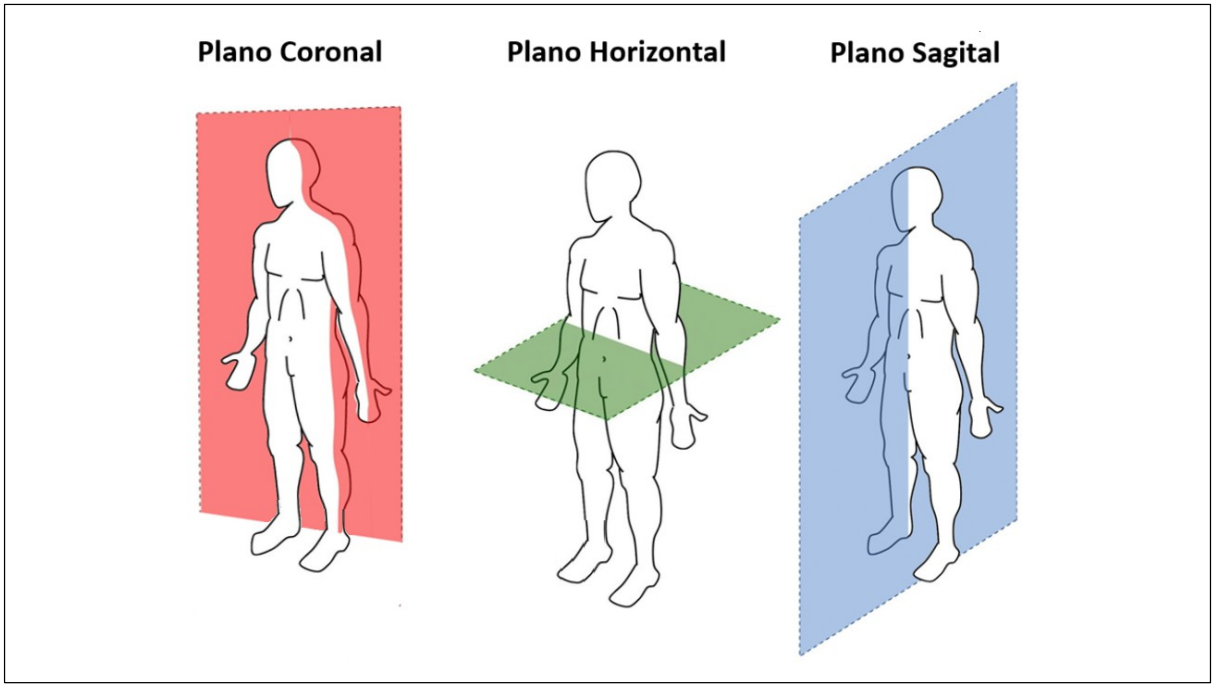

Figura 1. Representação dos principais planos anatômicos utilizados na análise do movimento humano
Fonte: Adaptado <https://www.odontoup.com.br/planos-anatomicos-odontologia/>.

É importante destacar que o movimento articular ocorre sempre em um plano específico e em torno de um eixo que lhe é perpendicular. A flexão e a extensão acontecem no plano sagital em torno do eixo frontal, enquanto os movimentos de abdução e adução ocorrem no plano frontal em torno do eixo sagital. Movimentos semelhantes, como o desvio radial e ulnar do punho, também ocorrem no plano frontal, girando ao redor do eixo sagital (CEDUP, 2011).

A Figura 2, traz uma representação dos tipos de queda: queda por tombamento, queda consciente e queda inconsciente.

A perda de equilíbrio ou tombamento, Figura 2a), demonstra o corpo do ponto de vista cinemático. Quando a linha vertical que passa pelo centro de gravidade está fora da base de apoio, o corpo começa a tombar. Se não houver reação a essa perda de equilíbrio, o corpo cairá no chão (ABBATE et al., 2010).

Ao considerar a queda de um corpo a partir de uma posição estática a uma altura h = H, a energia potencial Inicial mgh é convertida em energia cinética ao longo da queda, atingindo seu valor máximo imediatamente antes do impacto no solo $h = 0$. Durante o impacto, essa energia é completamente absorvida pelo corpo, de modo que, após o impacto, tanto a energia potencial quanto a

cinética se tornam nulas (ABBATE et al., 2010). Se a pessoa estiver consciente, parte desse impacto será absorvida pelo capacidade da pessoa de se proteger Figura 2b). Contudo, se a pessoa estiver inconsciente o acidente pode causar graves consequências, Figura 2c).

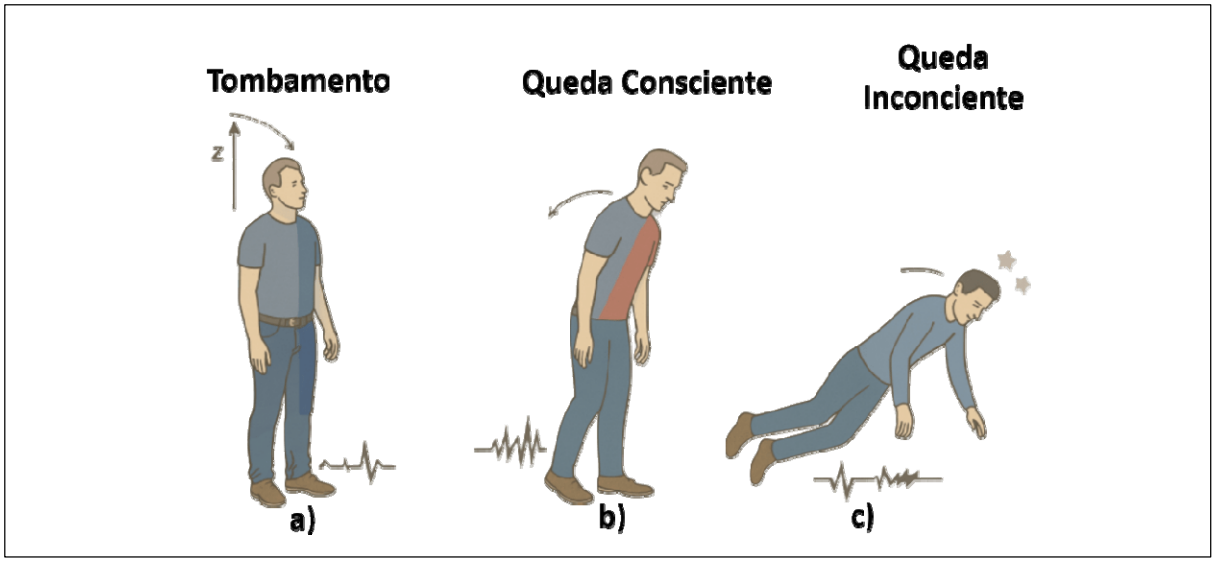

Figura 2. Principais tipos de quedas do corpo humano, representando diferentes direções e padrões de deslocamento corporal durante o evento de queda.

Fonte: Elaborado com auxílio do ChatGPT (2025).

As quedas geralmente ocorrem ao longo de dois planos anatômicos, sendo cruciais para a análise da biomecânica e movimento humano: o plano sagital, que divide o corpo em lados esquerdo e direito e está relacionado a quedas para frente ou para trás; e o plano coronal, que divide o corpo nas regiões anterior e posterior (frente e costas) e está associado a quedas laterais. Quando a pessoa está em posição estática (em pé, sem se mover), a queda tende a ocorrer na direção vertical (ABBATE et al., 2010). Fatores adicionais podem influenciar a direção da queda. Por exemplo, se a pessoa escorregar ou tropeçar, a direção da queda pode ser alterada para um ângulo, em vez de ser perfeitamente vertical.

A seguinte seção descreve o sensor mais empregado na problemática da detecção de quedas: o acelerômetro.

2.3 ACELERÔMETRO

Os acelerômetros são sensores que medem aceleração linear. Há diferentes tipos (piezoelétricos, capacitivos, resistivos), e seu funcionamento se baseia na segunda lei de Newton (F = ma). São baseados na transformação das forças em sinais elétricos, sendo bem comumente utilizados em equipamentos

eletrônicos, biomecânica, aplicações industriais e sistemas embarcados. Os acelerômetros podem ser piezoelétricos, que produzem cargas elétricas por compressão dos cristais; piezo resistivos, que ajustam sua resistência em resposta à aceleração; capacitivos, com variações de capacitância. Costumam ser encontrados na modalidade triaxial, possibilitando medir a aceleração em três eixos (X, Y, Z). Seus usos e aplicações estão presentes em: Sistemas embarcados; detecção de quedas, veículos autônomos, dentre outros usos (OMEGA, 2024).

Os acelerômetros piezoelétricos são ideais para medir vibrações e choques de alta frequência, sendo amplamente utilizados em testes estruturais e aeroespaciais. Já os piezorresistivos são eficazes em medições de baixa frequência e de longa duração, sendo aplicados em testes de colisão veicular, por exemplo. Os capacitivos, por sua vez, são adequados para medições de aceleração estática e de baixa frequência, sendo frequentemente integrados em sistemas embarcados e dispositivos MEMS (micro-electromechanical systems). O documento também enfatiza que a escolha adequada do tipo de acelerômetro depende das características da aplicação, como a faixa de frequência, o nível de aceleração, e o ambiente operacional (temperatura, umidade, etc.), influenciando diretamente a precisão e a durabilidade do sensor (ENDEVCO, 2012).

Para este trabalho, os acelerômetros triaxiais (Figura 3) constituem a principal fonte de sinal analisada ao longo do tempo. Essa escolha decorre das características do conjunto de dados utilizado, no qual todos os registros de quedas possuem medições de acelerômetro, enquanto menos de um terço dos eventos contam com dados de giroscópio e nenhum dos arquivos apresenta informações provenientes de magnetômetros. Dessa forma, a análise concentrou-se nos sinais de aceleração, especialmente nas janelas temporais que antecedem e sucedem o evento de queda. A compreensão do funcionamento e das características técnicas desse sensor torna-se, portanto, essencial para a construção de modelos preditivos capazes de identificar padrões relevantes associados a esses eventos.

A relevância do uso de acelerômetros em estudos científicos é evidenciada pelo crescente volume de publicações na área. Um levantamento realizado no Portal de Periódicos da CAPES indica que, no período entre 2024 e 8 de junho de 2025, foram identificadas mais de 3.200 publicações relacionadas ao uso de acelerômetros em diferentes contextos científicos, demonstrando o interesse crescente da comunidade acadêmica na aplicação desses sensores para análise de movimento e monitoramento de atividades humanas (VILLA et al., 2023).

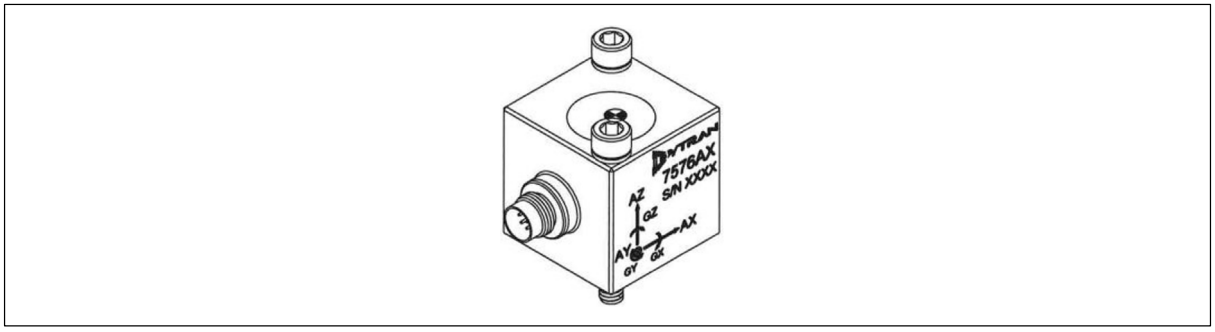

Figura 3. Representação de um acelerômetro triaxial, sensor inercial capaz de medir a aceleração nos eixos ortogonais X, Y e Z.

Fonte: Adaptado de <https://pdf.aeroexpo.online/pdf/dytran-instruments-inc/7576-series/176905-9555.html>.

O acelerômetro triaxial é capaz de medir acelerações em três direções (Figura 4). O eixo X (lateral - movimentos laterais), eixo Y (longitudinal - movimentos anteroposteriores) e eixo Z (movimento vertical; inclui aceleração gravitacional).

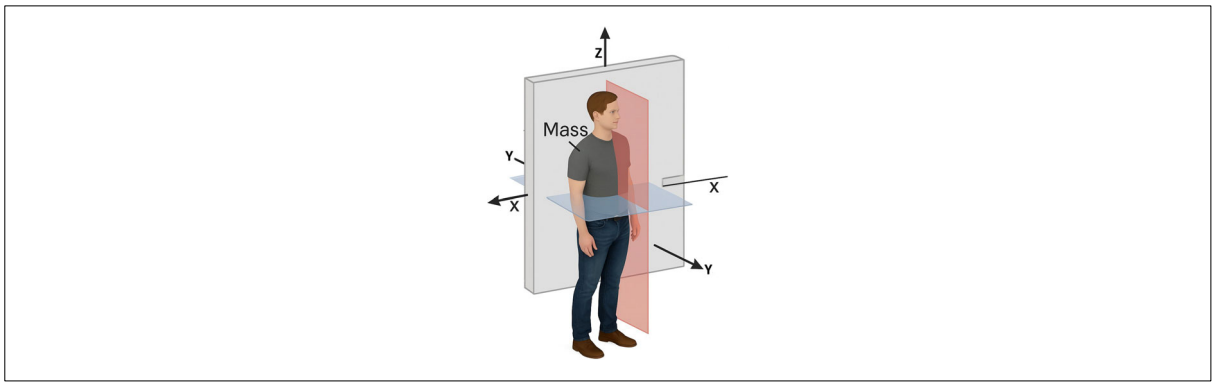

Figura 4. Representação dos eixos ortogonais (X, Y e Z) utilizados por um acelerômetro triaxial para medir a aceleração do corpo humano em três dimensões.

Fonte: Adaptado com auxílio do ChatGPT (2025).

Esses eixos correspondem às três dimensões do espaço cartesiano, permitindo ao sensor capturar o movimento do corpo em qualquer direção. O MEMS, detecta a variação de posição de uma massa interna suspensa, durante o movimento essa massa se desloca em função da aceleração e são convertidos em sinais elétricos pelo sensor. Essa capacidade de registro em 3 dimensões torna este componente essencial para este estudo.

2.4 MÉTRICAS DE DESEMPENHO

Nesta seção será detalhado as métricas utilizadas para apresentação dos resultados.

2.4.1 Matriz Confusão

A avaliação do desempenho dos modelos de aprendizado de máquina foi fundamentada na análise da **matriz de confusão**, a partir da qual são derivados os principais indicadores estatísticos de classificação.

Segundo Nielsen (2020) a matriz de confusão organiza as predições do modelo em termos de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, permitindo uma interpretação estruturada dos acertos e erros de classificação. Com base nessa decomposição, foi possível empregar um conjunto abrangente de métricas com o objetivo de quantificar não apenas a acurácia global do classificador, mas também sua capacidade de identificar corretamente eventos raros e críticos, como quedas, bem como de controlar a ocorrência de alarmes falsos. Essa abordagem é particularmente relevante em cenários de saúde assistiva, nos quais os custos associados aos diferentes tipos de erro são assimétricos.

Tabela 1. Modelo de matriz de confusão para as classes Positiva "Queda" e Negativa "Não Queda"

Real	**Predito**	
	Não Queda (0)	**Queda (1)**
Não Queda (0)	Verdadeiro Negativo (VN)	Falso Positivo (FP)
Queda (1)	Falso Negativo (FN)	Verdadeiro Positivo (VP)

VN: eventos de não queda corretamente classificados; FP: eventos de não queda classificados como queda (alarme falso); FN: quedas classificadas como não queda; VP: quedas corretamente identificadas.

2.4.2 Acurácia

A **acurácia** é definida como a proporção de predições corretas em relação ao total de observações, considerando conjuntamente verdadeiros positivos e verdadeiros negativos. Embora seja uma métrica intuitiva e amplamente utilizada, sua interpretação isolada pode ser inadequada em conjuntos de dados desbalanceados, nos quais altas taxas de acerto podem ser obtidas mesmo com desempenho insatisfatório na classe minoritária. Para complementar essa análise, o **intervalo de confiança de 95% (IC 95%)** da acurácia é empregado como uma medida estatística de incerteza, permitindo avaliar a

variabilidade associada à estimativa pontual (NIELSEN, 2020).

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.1}$$

Intervalo de confiança:

$$IC_{95\%} = \hat{p} \pm z_{0,975} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{2.2}$$

Em que $\hat{p}$ é a acurácia observada, n o número total de instâncias avaliadas e $z_{0,975}$ o quantil da distribuição normal padrão associado a um nível de confiança de 95

2.4.3 No Information Rate

A **No Information Rate (NIR)** corresponde à taxa de acerto obtida por um classificador ingênuo que sempre prediz a classe majoritária do conjunto de dados. Essa métrica estabelece uma linha de base mínima de desempenho, permitindo avaliar se um classificador apresenta ganho informacional em relação a uma estratégia trivial baseada apenas na distribuição das classes. A comparação entre a acurácia observada e a NIR pode ser formalizada por meio de um teste estatístico, cujo p-valor indica se a taxa de acerto do modelo é estatisticamente superior aquela obtida sem o uso de informação preditiva (KUHN, 2023).

$$\text{NIR} = \max\left(\frac{n_0}{n}, \frac{n_1}{n}\right) \tag{2.3}$$

Em que n_0 e n_1 correspondem às frequências das classes negativa e positiva, respectivamente, e n é o número total de observações.

2.4.4 Coeficiente Kappa de Cohen

O **coeficiente Kappa de Cohen** mede o grau de concordância entre as predições do modelo e os rótulos reais, ajustando o valor observado pelo acordo esperado ao acaso. Essa métrica é particularmente útil em cenários de desbalanceamento, nos quais a acurácia pode superestimar o desempenho real do classificador. Valores elevados de Kappa indicam concordância substancial ou quase perfeita. De forma complementar, o **teste de McNemar** é utilizado para avaliar a simetria entre erros de

classificação, permitindo identificar possíveis diferenças sistemáticas entre falsos positivos e falsos negativos (MARTINEZ, 2020).

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{2.4}$$

Em que p_o representa a proporção de concordância observada e p_e a concordância esperada ao acaso.

2.4.5 Sensibilidade

A **sensibilidade**, também denominada *recall* ou taxa de verdadeiros positivos, corresponde à proporção de instâncias da classe positiva corretamente identificadas. Essa métrica é essencial em aplicações nas quais a não detecção do evento de interesse implica consequências graves. Em contraposição, a **especificidade** mede a capacidade do classificador em reconhecer corretamente instâncias da classe negativa, estando diretamente relacionada ao controle de alarmes falsos (NIELSEN, 2020).

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \tag{2.5}$$

2.4.6 Valor Preditivo Positivo

O **valor preditivo positivo (PPV)** expressa a proporção de predições positivas que correspondem efetivamente à classe positiva real, enquanto o **valor preditivo negativo (NPV)** indica a confiabilidade das predições negativas. Diferentemente da sensibilidade e da especificidade, essas métricas dependem da prevalência da classe no conjunto de dados e refletem a confiabilidade prática das decisões do modelo (NIELSEN, 2020).

$$\text{PPV} = \frac{VP}{VP + FP} \tag{2.6}$$

Onde VP é o número de predições positivas corretas e FP o número de predições positivas incorretas.

2.4.7 Prevalência

A **prevalência** corresponde à proporção real de ocorrências da classe positiva no conjunto de dados e constitui um parâmetro fundamental para a interpretação de métricas preditivas. A **taxa de**

detecção indica a fração de eventos corretamente identificados, enquanto a **prevalência detectada** representa a proporção de instâncias classificadas como positivas pelo modelo, permitindo avaliar tendências de superestimação ou subestimação do evento (POWERS, 2020).

$$\text{Prevalência} = \frac{VP + FN}{VP + VN + FP + FN} \tag{2.7}$$

2.4.8 Acurácia Balanceada

A **acurácia balanceada** é definida como a média aritmética entre sensibilidade e especificidade, sendo amplamente recomendada para problemas de classificação binária com classes desbalanceadas. Essa métrica fornece uma avaliação mais equitativa do desempenho global do classificador ao atribuir pesos iguais às classes positiva e negativa (BRODERSEN et al., 2010).

$$\text{Acurácia Balanceada} = \frac{\text{Sensibilidade + Especificidade}}{2} \tag{2.8}$$

2.5 ALGORITMOS DE MACHINE LEARNING

Nesta seção serão abordados os algoritmos de inteligência artificial usados nesta dissertação.

2.5.1 Árvore de Decisão (Decision Tree)

Segundo a documentação oficial (SCIKIT-LEARN, 2025a), a classe DecisionTreeClassifier implementa um algoritmo de aprendizado supervisionado baseado em árvores de decisão, aplicável a problemas de classificação binária e multiclasse. O modelo segmenta recursivamente o espaço de atributos por meio de testes do tipo $x_j \leq t$, onde x_j representa um atributo e t um limiar de decisão, escolhidos de forma a maximizar a redução da impureza em cada nó.

O processo de indução da árvore ocorre até que um critério de parada seja satisfeito, como a pureza total dos nós, a profundidade máxima permitida ou a restrição no número mínimo de amostras para divisão. A predição final é determinada pela classe majoritária presente no nó folha alcançado, enquanto as probabilidades de classe são estimadas pela distribuição relativa das amostras nesse nó.

Embora apresentem alta interpretabilidade e baixo custo computacional na fase de inferência, árvores de decisão são modelos com elevada variância e forte tendência ao ajuste excessivo aos dados de treinamento, fenômeno conhecido como *overfitting*. Para mitigar esse efeito, o *Scikit-Learn* disponibiliza diversos hiperparâmetros de controle de complexidade, incluindo limites de profundidade, tamanho mínimo de nós e poda por complexidade de custo mínimo (SCIKIT-LEARN, 2025a).

Figura 5. Exemplo ilustrativo de uma Árvore de Decisão para classificação.

No *Scikit-Learn*, o DecisionTreeClassifier permite ajustes finos do processo de aprendizado por meio dos seguintes hiperparâmetros principais:

- **criterion**: define a métrica utilizada para avaliar a qualidade das divisões em cada nó.
  - "gini": impureza de Gini (padrão), computacionalmente eficiente.
  - "entropy": entropia da informação, baseada em ganho informacional.
  - "log_loss": minimiza a perda logarítmica, adequada para estimativas probabilísticas.
- **splitter**: estratégia utilizada para selecionar a divisão em cada nó.
  - "best": seleciona a melhor divisão possível.
  - "random": escolhe a melhor divisão dentre um subconjunto aleatório de candidatos.
- **max_depth**: profundidade máxima da árvore.
  - None: crescimento irrestrito até os critérios mínimos de parada.
  - Valores menores reduzem a variância e o risco de sobreajuste.
- **min_samples_split**: número mínimo de amostras exigidas para dividir um nó.
  - int: número absoluto de amostras.
  - float: fração do conjunto total de dados.
- **min_samples_leaf**: número mínimo de amostras em um nó folha.
  - Valores maiores produzem folhas mais estáveis e modelos mais generalizáveis.
- **max_features**: número de atributos considerados em cada divisão.
  - int: quantidade exata de variáveis.

  - float: fração do total de atributos.
  - "sqrt": raiz quadrada do número total de atributos.
  - "log2": logaritmo base 2 do número total de atributos.
  - None: considera todas as variáveis disponíveis.

- **class_weight**: define pesos associados às classes.
  - None: sem ponderação.
  - "balanced": pesos inversamente proporcionais à frequência das classes.
  - dict: pesos definidos manualmente.

- **ccp_alpha**: parâmetro de poda por complexidade de custo mínimo.
  - Valores maiores resultam em árvores mais simples e maior capacidade de generalização.

Em síntese, a Árvore de Figura 6, constitui um modelo fundamental em aprendizado de máquina, destacando-se pela interpretabilidade e simplicidade. Entretanto, sua aplicação prática requer controle explícito da complexidade estrutural, especialmente em bases ruidosas ou desbalanceadas, sendo comum seu uso como modelo base para métodos de aprendizado em conjunto, como o Random Forest.

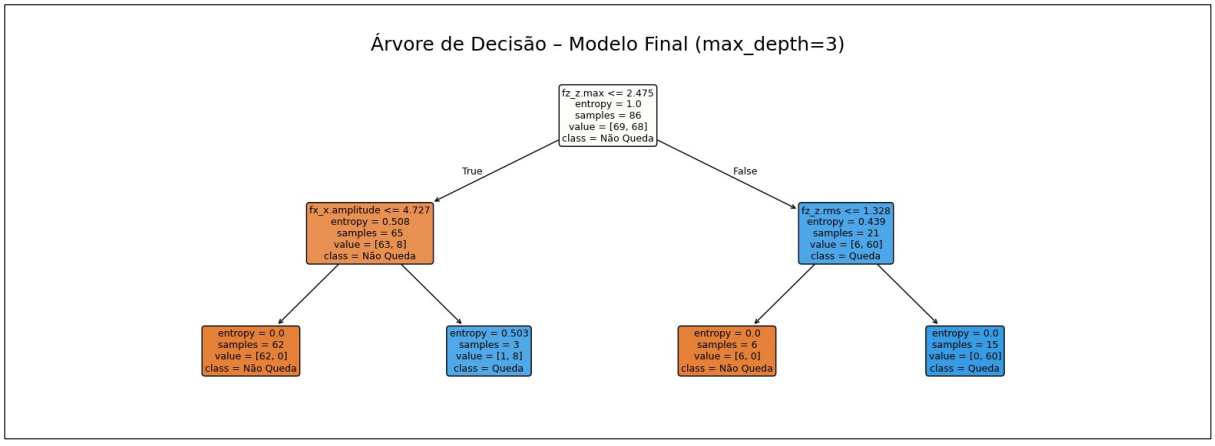

Figura 6. Representação do algoritmo Árvore de Decisão com profundidade máxima 3
Fonte: O Autor.

2.5.2 *Random Forest (RF)*

Segundo IBM (2025a) *Random Forest* é um modelo de de aprendizado em conjunto (*ensemble*) que faz uso de múltiplas árvores de decisão Figura 7. Esse algoritmos constrói diversas árvores a partir

de subconjuntos de dados aleatórios de dados (amostragem com reposição - *bootstrap*), e combina seus resultados por meio de votação no caso de classificação ou por média para regressão.

A árvore de decisão, base do RF, segmenta recursivamente os dados com base em perguntas como *"X < Y ?"*, até o momento que os critérios de parada sejam encontrados. Esses modelos têm alta variância e tendencia de adaptação, "decorar resultados"ou aprender o ruído ou informações sem relevância para os objetivos, anulando sua capacidade de generalização e realizar previsões, efeito conhecido tecnicamente como *overfitting*(SCIKIT-LEARN, 2025a).

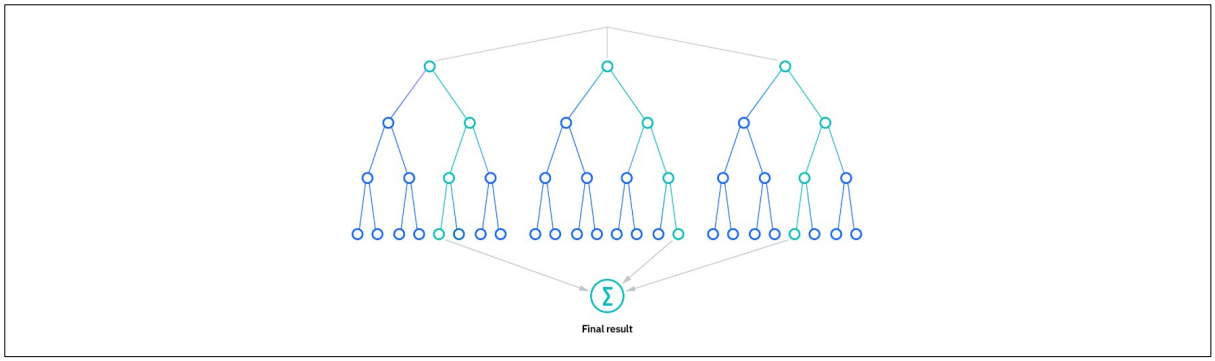

Figura 7. Representação do algoritmo Random Forest
Fonte: (IBM, 2025a).

A combinação de múltiplas árvores de decisão melhora a qualidade preditiva do modelo. No *Scikit-Learn*, existem variações tanto para tarefas de classificação quanto de regressão. A classe principal é dos classificadores e permite o uso de diversos hiperparametros como:

- ***n_estimators***: número de árvores na floresta (*default: 100*).
  - Quanto maior o número de árvores, mais estável e preciso tende a ser o modelo, reduzindo variância.
  - Entretanto, valores muito altos aumentam o custo computacional.
- ***criterion***:métrica utilizada para medir a qualidade das divisões em cada nó (*"gini"*[1], *"entropy"ou "log_loss"*).

  "gini", *"entropy"* ou *"log_loss"*.

[1] A **impureza de Gini** mede a probabilidade de classificar uma amostra de forma incorreta caso seja escolhida aleatoriamente segundo a distribuição de classes do nó. É dada por $Gini(t) = 1 - \sum_{i=1}^{C} p_i^2$, onde p_i é a proporção da classe i no nó t.

  - **gini**: mede a impureza de Gini; mais eficiente computacionalmente.
  - **entropy**: usa a entropia da informação; similar ao Gini mas tende a gerar árvores mais balanceadas.
  - **log_loss**: minimiza a perda logarítmica, útil em tarefas probabilísticas.

- ***max_features***: número de variáveis consideradas em cada divisão. Pode ser:
  - Um **inteiro**: número exato de features a considerar.
  - *"sqrt"*: usa $\sqrt{p}$, onde p é o número total de variáveis (padrão em classificação).
  - *"log2"*: usa $\log_2(p)$ features.
  - *None*: considera todas as variáveis disponíveis em cada divisão.

- **bootstrap**: (*True* ou *False*) indica se a amostragem dos dados para cada árvore será com reposição.
  - **True**: cada árvore é treinada com uma amostra bootstrap do conjunto de treino.
  - **False**: cada árvore usa diretamente o conjunto completo de dados.

- **oob_score**: (*True* ou *False*) habilita a estimação de erro via amostras *out-of-bag (OOB)*.
  - OOB usa os exemplos não selecionados no bootstrap para testar a árvore correspondente, funcionando como uma validação cruzada interna.

- **class_weight**: define como tratar classes desbalanceadas.
  - *"balanced"*: pondera automaticamente as classes de forma inversamente proporcional à sua frequência.
  - *"balanced_subsample"*: aplica o balanceamento em cada amostra de bootstrap separadamente.
  - Também pode ser um *dict* para pesos customizados por classe.

Componentes como *max_depth, min_samples_split, min_samples_leaf* e *ccp_alpha* permitem limitar o crescimento das árvores. Para cenários com classes altamente desbalanceadas, o pacote *imbalanced-learn* traz o ***BalancedRandomForestClassifier***, que executa *undersampling* da classe majoritária em cada amostra *bootstrap*, mantendo total controle sobre as proporções com *sampling_strategy, replacement*.

O Random Forest é um conjunto de árvores de decisão, cujas variações — desde balanceamento de dados até florestas extremamente aleatórias — permitem amplo uso em contextos de classificação e regressão. No Scikit-Learn, as implementações oferecem controles avançados que permitem modelar o desempenho para diferentes tipos e distribuições de dados, com recursos nativos de validação (OOB) e integração com pipelines.

2.5.3 *k-Nearest Neighbors (KNN)*

Segundo IBM (2025b) o knn, ou k-vizinhos mais próximos, é um algoritmo de aprendizado de máquina supervisionado usado para tarefas de classificação e regressão. Ele funciona encontrando os "k"pontos de dados mais próximos (vizinhos) de um novo ponto de dados não classificado e, em seguida, atribuindo a ele o rótulo de classe mais frequente (para classificação) ou o valor médio (para regressão) desses vizinhos. O KNN é conhecido por sua simplicidade e facilidade de compreensão, mas seu desempenho pode ser afetado pela escolha de "k"e pela métrica de distância utilizada. O KNN não constrói um modelo interno geral, mas simplesmente armazena instâncias dos dados de treinamento. A classificação é calculada por maioria simples dos votos dos vizinhos mais próximos de cada ponto: um ponto de consulta recebe a classe de dados que possui o maior número de representantes dentro dos vizinhos mais próximos do ponto (SCIKIT-LEARN, 2025b).

A Figura 8, possui duas classes, a classe A e a classe B, o ponto roxo é o elemento que queremos classificar como pertencente a classe A ou a classe B, se baseando aos vizinhos mais próximos nesse caso k = 7. Os vizinhos mais próximos são 3 da classe A e 4 da classe B, a distância euclidiana é calculada pelo algoritmo, classificando o ponto roxo como sendo da classe B.

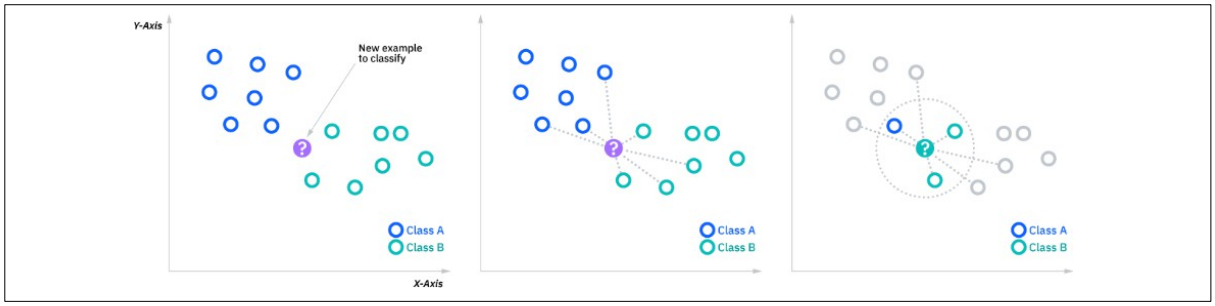

Figura 8. Representação do algoritmo KNN
Fonte: (IBM, 2025b).

2.5.4 *Suport Vector Machine (SVM)*

Conforme IBM (2023), a Support Vector Machine (SVM) é um algoritmo supervisionado utilizado para tarefas de classificação e regressão, notoriamente eficaz em espaços de alta dimensionalidade. A SVM busca encontrar um hiperplano ótimo que maximize a margem entre as classes, ou seja, a distância mínima entre o hiperplano e os vetores de suporte — os pontos de dados mais próximos ao limite de separação. O SVM inicialmente projeta os dados em um espaço de maior dimensão, quando não são linearmente separáveis, Figura 9, por meio do uso de funções kernel. Assim, torna possível a criação de um hiperplano de separação linear em um espaço transformado.

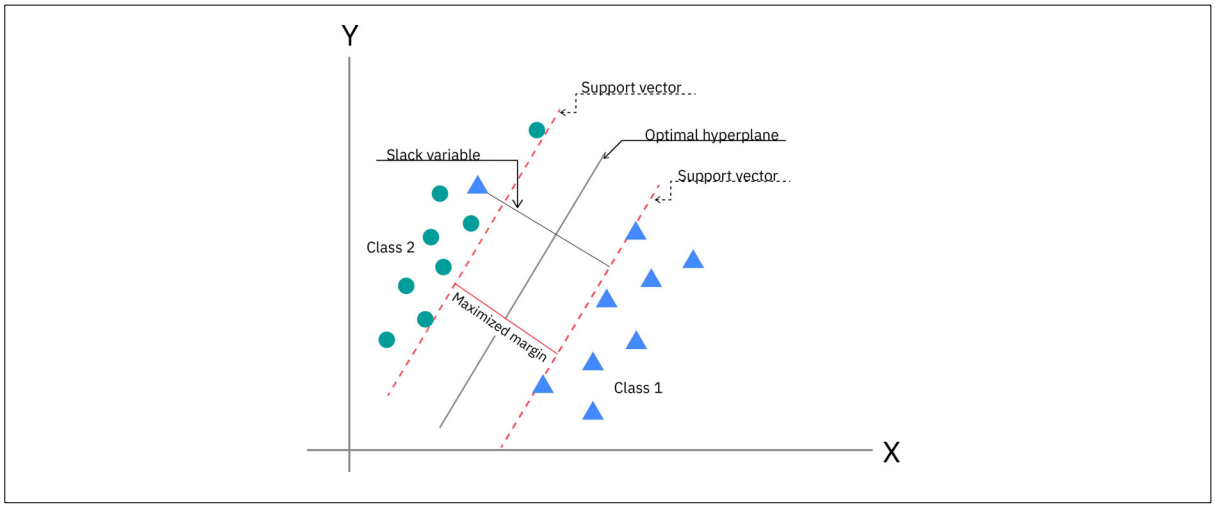

Figura 9. SVM
Fonte: (IBM, 2023).

A margem entre classes é maximizada, sendo ajustada pelo parâmetro de regularização C, que controla o *trade-off* entre maximizar a margem e permitir erros de classificação (*soft margin*).Os vetores que determinam essa margem são chamados de *support vectors* e possuem papel central na definição do modelo.Além da escolha do modelo e dos hiperparâmetros principais, a performance da SVM depende diretamente da função kernel utilizada. O kernel é responsável por transformar os dados de entrada em um espaço de maior dimensionalidade, possibilitando a separação linear de classes não linearmente separáveis no espaço original. O Scikit-learn oferece diversas opções de parametrizações, presente na Tabela 2:

Tabela 2. Principais Parâmetros do Modelo SVM (SVC – scikit-learn)

Parâmetro	Descrição
C	Parâmetro de regularização. Valores maiores tendem a overfitting, menores favorecem margens mais amplas.
kernel	Define o tipo de função de kernel: linear, poly, rbf, sigmoid ou precomputed.
degree	Grau do polinômio usado no kernel poly. Ignorado para os demais kernels.
gamma	Coeficiente do kernel para rbf, poly e sigmoid. Pode ser ′scale′, ′auto′ ou valor numérico.
coef0	Termo independente nos kernels poly e sigmoid.
shrinking	Aplica heurística de otimização para acelerar o treinamento (True por padrão).
probability	Ativa cálculo de probabilidades com calibração posterior. Reduz desempenho computacional.
tol	Tolerância para o critério de parada do otimizador.
cache_size	Tamanho do cache para cálculo do kernel (em MB).
class_weight	Ajusta pesos das classes. O valor ′balanced′ ajusta automaticamente conforme frequência das classes.
verbose	Habilita saída detalhada do processo de treinamento.
max_iter	Limite máximo de iterações (ou -1 para ilimitado).
decision_function_shape	Formato da decisão: ′ovr′ (um contra o resto) ou ′ovo′ (um contra um).
break_ties	Em problemas multiclasse, resolve empates considerando os valores de decisão.
random_state	Define semente aleatória para reprodutibilidade quando probability=True.

Fonte: Adaptado de *scikit-learn documentation*.

- **Kernel**
  - ***kernel='linear'***: aplica um kernel linear, sem transformação do espaço. É indicado quando os dados são aproximadamente separáveis por uma linha ou plano. Ideal para alta dimensionalidade e baixo número de amostras.
  - ***kernel='poly'***: utiliza um kernel polinomial, permitindo a criação de fronteiras de decisão mais complexas. O grau do polinômio é definido pelo parâmetro degree. Por exemplo, degree=3 implica em uma curva cúbica no espaço de decisão.
  - ***coef0***: termo constante que influencia o comportamento do kernel polinomial. É importante especialmente para kernels de grau baixo, ajustando a flexibilidade do modelo.
  - ***kernel='rbf'*** (Radial Basis Function, ou Gaussiano): é o mais utilizado. Mapeia os dados em um espaço de dimensão infinita. O parâmetro gamma define o alcance de influência de um ponto de dado.
    - ***gamma alto***: influência pequena, modelo mais complexo (risco de overfitting).
    - ***gamma baixo***: influência ampla, modelo mais suave (risco de underfitting).
  - ***kernel='sigmoid'***: simula o comportamento de um neurônio em redes neurais. Utiliza uma função sigmoide como separador. É raramente usado, mas pode ser útil em contextos específicos de classificação com limiares não lineares.

  - ***kernel='precomputed'***: utilizado quando a matriz de similaridade (kernel matrix) é previamente computada e fornecida ao modelo.

- ***C***: regula o grau de penalização por erro. Um C muito alto força o modelo a classificar todos os pontos corretamente (pode gerar overfitting), enquanto um valor baixo permite mais erros, priorizando margens maiores.
- ***degree***: Define o grau da função polinomial.
- ***shrinking***: booleano que define se o algoritmo de otimização deve aplicar heurísticas de eliminação para acelerar o cálculo. Em geral, manter como True melhora o desempenho computacional.
- ***probability***: se True, permite o cálculo da probabilidade das classes via calibração posterior, porém com custo computacional adicional.
- ***class_weight***: permite ajustar o peso das classes, útil para conjuntos de dados desbalanceados. O valor ′balanced′ ajusta os pesos de forma inversamente proporcional à frequência das classes.

2.5.5 Rede Neural Convolucional Unidimensional (1D-CNN)

Segundo a documentação oficial do *TensorFlow/Keras* (Keras Team, 2025; TensorFlow Team, 2025), as Redes Neurais Convolucionais Unidimensionais (1D-CNN) constituem uma classe de modelos de aprendizado profundo projetados para a extração automática de padrões locais em dados sequenciais, sendo amplamente empregadas em problemas de classificação e detecção de eventos em séries temporais, sinais biomédicos e dados provenientes de sensores.

Diferentemente das redes convolucionais bidimensionais, utilizadas principalmente em visão computacional, as 1D-CNN operam sobre sequências unidimensionais, aplicando filtros convolucionais ao longo do eixo temporal. Cada filtro realiza uma operação de convolução do tipo:

$$\hat{y}(t) = \sum_{i=0}^{k-1} w_i \cdot x(t+i) + b. \tag{2.9}$$

Onde $x(t)$ representa o sinal de entrada, w_i os pesos do filtro convolucional de tamanho k e b o termo de viés. Essa operação permite à rede aprender automaticamente padrões temporais discriminativos, como picos abruptos, oscilações e transições características de eventos específicos.

O processo de aprendizado ocorre por meio da composição de múltiplas camadas convolucionais, frequentemente intercaladas com camadas de normalização, funções de ativação não lineares e operações de *pooling*. As camadas de *pooling* reduzem a dimensionalidade do sinal, aumentando a invariância temporal e contribuindo para a redução do custo computacional. A predição final é realizada por camadas totalmente conectadas, que mapeiam as representações extraídas para as classes de interesse.

Embora apresentem elevada capacidade de modelagem e desempenho superior em tarefas de extração automática de características, as 1D-CNN demandam maior volume de dados e maior custo computacional durante o treinamento quando comparadas a modelos tradicionais. Adicionalmente, sua natureza de *caixa-preta* reduz a interpretabilidade do processo decisório, o que pode representar uma limitação em aplicações críticas (GOODFELLOW; BENGIO; COURVILLE, 2016).

No *Keras*, a camada Conv1D disponibiliza diversos hiperparâmetros para controle do processo de aprendizado, dentre os quais destacam-se:

- **filters**: número de filtros convolucionais utilizados na camada.
  - Valores maiores aumentam a capacidade de extração de padrões, ao custo de maior complexidade computacional.
- **kernel_size**: tamanho do filtro convolucional.
  - Define a extensão temporal do padrão aprendido.
  - Valores menores capturam padrões locais; valores maiores capturam dependências temporais mais longas.
- **strides**: passo de deslocamento do filtro ao longo da sequência.
  - Valores maiores reduzem a resolução temporal da saída.
- **padding**: estratégia de preenchimento da sequência.
  - onsiste: sem preenchimento.
  - "same": preserva o tamanho temporal da entrada.
- **activation**: função de ativação aplicada à saída convolucional.
  - ReLU: amplamente utilizada devido à eficiência computacional e mitigação do problema do gradiente desvanecente.

- **pool_size**: tamanho da janela de *pooling* em camadas MaxPooling1D.
  - Controla o grau de redução temporal do sinal.
- **dropout**: taxa de desligamento aleatório de neurônios durante o treinamento.
  - Atua como técnica de regularização, reduzindo o risco de sobreajuste.

Em síntese, as Redes Neurais Convolucionais Unidimensionais configuram-se como uma abordagem robusta para o processamento de séries temporais, destacando-se pela capacidade de aprendizado automático de características discriminativas. Todavia, sua aplicação prática exige cuidados relacionados à quantidade de dados disponíveis, à escolha adequada de hiperparâmetros e ao controle do sobreajuste, sendo frequentemente utilizadas em conjunto com técnicas de regularização e validação rigorosa.

2.5.6 *Long Short-Term Memory (LSTM)*

A LSTM incorpora uma estrutura composta por células de memória que retêm informações relevantes ao longo de longos intervalos temporais, reguladas por três mecanismos principais: porta de entrada, que decide quais informações são armazenadas; porta de esquecimento, que define o que deve ser descartado; e porta de saída, que controla o fluxo de informação para as etapas subsequentes. Essa configuração permite ao modelo filtrar seletivamente os dados que devem ser memorizados ou ignorados, garantindo maior estabilidade no treinamento (HOCHREITER; SCHMIDHUBER, 1997).

O artigo clássico Hochreiter e Schmidhuber (1997), introduz a arquitetura Long Short-Term Memory (LSTM) como uma solução eficiente para os problemas de desvanecimento e explosão do gradiente observados em Redes Neurais Recorrentes (RNNs) tradicionais. A proposta visa superar as limitações que impedem o aprendizado de dependências de longo prazo em tarefas sequenciais. O artigo apresenta uma série de experimentos comparativos entre LSTM e RNNs convencionais, utilizando tarefas sintéticas como os problemas de cópia e adição, ambas projetadas para avaliar a capacidade de memorização em sequências de grande extensão. Os resultados indicam que a LSTM consegue aprender e generalizar com eficácia, enquanto as RNNs comuns falham sistematicamente em contextos que envolvem memória de longo prazo.

A Tabela 3, demostra alguns parâmetros disponíveis função.

Tabela 3. Principais Parâmetros da Camada LSTM (Keras)

Parâmetro	Descrição
units	Número de neurônios da camada LSTM. Define a dimensionalidade do espaço de saída.
activation	Função de ativação aplicada à saída (padrão: tanh).
recurrent_activation	Função de ativação usada nas transições internas (padrão: sigmoid).
dropout	Fração de unidades descartadas na transformação de entrada (valor entre 0 e 1).
recurrent_dropout	Fração de unidades descartadas no estado recorrente.
return_sequences	Retorna sequência completa ou apenas a última saída. Importante para empilhar camadas.
return_state	Retorna, além da saída, os estados ocultos e de memória. Útil para arquiteturas complexas.
go_backwards	Se True, processa a sequência de entrada de trás para frente.
stateful	Se True, mantém o estado entre os batches, útil para séries temporais contínuas.
kernel_initializer	Inicializador para os pesos da entrada (padrão: glorot_uniform).
recurrent_initializer	Inicializador para os pesos recorrentes (padrão: orthogonal).
unit_forget_bias	Se True, adiciona 1 ao viés do "forget gate"no início do treinamento.
unroll	Se True, desenrola a rede para sequências curtas, melhorando desempenho com maior uso de memória.
use_bias	Se True, adiciona vetor de viés à saída (ativado por padrão).
bias_initializer	Inicializador dos vieses (padrão: zeros).
bias_regularizer	Função de regularização aplicada ao viés.
bias_constraint	Restrições aplicadas ao vetor de viés.
kernel_regularizer	Regularizador aplicado aos pesos da entrada (kernel).
recurrent_regularizer	Regularizador aplicado aos pesos recorrentes.
activity_regularizer	Regularizador aplicado à saída da camada.
kernel_constraint	Restrição aplicada aos pesos da entrada (kernel).
recurrent_constraint	Restrição aplicada aos pesos recorrentes.
mask	Máscara binária indicando quais timesteps devem ser ignorados (aplicada durante treinamento).
initial_state	Permite fornecer os estados iniciais manualmente.
training	Define se a camada está em modo de treino ou inferência (relevante para dropout).
seed	Semente para inicialização aleatória dos dropouts.
use_cudnn	Controla se o backend com suporte CUDA será utilizado (padrão: auto).
input	Tensor de entrada (útil para inspeção e debug de arquiteturas).

Fonte: Adaptado de TensorFlow (2024).

A arquitetura LSTM não apenas resolve as falhas estruturais das RNNs, mas também oferece uma solução robusta e escalável, com aplicações potenciais em áreas como processamento de linguagem natural, reconhecimento de fala, modelagem de séries temporais e outros problemas envolvendo dados sequenciais complexos (HOCHREITER; SCHMIDHUBER, 1997).

2.6 CONSIDERAÇÕES

Este capitulo apresentou os principais fundamentos teóricos necessários para a compreensão e desenvolvimento desta pesquisa. Foram discutidos os *datasets* utilizados, destacando suas particularidades e relevância para a modelagem do problema; aspectos biomecânicos das quedas, incluindo apresentação dos planos anatômicos e tipos de impacto; funcionamento e a importância do , os algoritmos de aprendizado de máquina selecionados nesta pesquisa com base na revisão do estado da

arte. Estes conceitos formam a base que sustenta as decisões metodológicas adotadas nos próximos capítulos, especialmente no que se refere à preparação dos dados, construção dos modelos e analise dos resultados.

3 TRABALHOS RELACIONADOS

Na revisão sistemática foram pesquisados trabalhos que têm como objetivo o uso de dados reais ou simulados de quedas de pessoas. Além disso, foram investigados trabalhos cujo propósito era avaliar o desempenho desses algoritmos na detecção de quedas. Para conduzir esta pesquisa, o Quadro 1 ilustra os repositórios selecionados como fontes de referência.

Tabela 4. Repositórios eletrônicos

Repositório	Endereços de Acesso
CAPES	https://www.periodicos.capes.gov.br//
EBSCO	https://www.univali.br/biblioteca/Paginas/default.aspx
Google Scholar	https://scholar.google.com/
IEEE Xplore	https://ieeexplore.ieee.org/
ScienceDirect	https://www.sciencedirect.com/

Fonte: O autor.

Neste capítulo, serão apresentados trabalhos que têm como foco compreender e detectar o processo de queda. Inicialmente, é apresentado um resumo do protocolo de busca utilizado para localizar os principais estudos. Por fim, os trabalhos são analisados e comparados com os resultados esperados nesta dissertação.

Esta revisão sistemática da literatura (RSL) foi conduzida com base nas diretrizes do protocolo PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*), amplamente utilizado para garantir transparência, rastreabilidade e reprodutibilidade em revisões sistemáticas. O protocolo PRISMA orienta as etapas de identificação, triagem, elegibilidade e inclusão dos estudos, permitindo documentar de forma estruturada o processo de seleção dos artigos analisados (PAGE et al., 2021).

A partir dessa abordagem metodológica, foi definido um protocolo de busca com o objetivo de identificar estudos relacionados à detecção de quedas utilizando dispositivos vestíveis capazes de coletar sinais provenientes do corpo humano, bem como analisar eventos de queda e os padrões que antecedem esses eventos.

3.1 PROTOCOLO DE BUSCA

A condução desta revisão seguiu a abordagem RSL, com base nas diretrizes propostas por Kitchenham (2004) e complementadas com protocolo PRISMA. A metodologia foi organizada em três etapas principais: **(i)** planejamento da revisão, contemplando a definição das perguntas de pesquisa, da *string* de busca e dos critérios de inclusão e exclusão; **(ii)** condução da busca, com a aplicação da *string* nos repositórios científicos selecionados; e **(iii)** análise e seleção dos estudos, realizada a partir da leitura dos títulos, resumos e textos completos, conforme os critérios previamente estabelecidos.

O protocolo de busca teve como objetivo identificar estudos relacionados à detecção de quedas com o uso de dispositivos vestíveis capazes de coletar sinais provenientes do corpo humano, bem como investigar eventos de queda e os padrões que os antecedem, buscando assegurar rigor metodológico na identificação, seleção e síntese dos trabalhos analisados.

Para a recuperação dos estudos na literatura científica, foi definida a seguinte *string* de busca:

("FARSEEING"OR "real-world falls") AND "machine learning"

A expressão de busca foi estruturada com o objetivo de recuperar estudos relacionados à detecção de quedas por meio de técnicas de aprendizado de máquina, contemplando dois contextos principais. O primeiro refere-se a pesquisas que utilizam explicitamente o repositório FARSEEING como fonte de dados experimentais. O segundo abrange estudos que investigam quedas reais (*real-world falls*) coletadas em ambientes não controlados, desde que associadas ao uso de métodos de aprendizado de máquina.

A utilização do operador lógico "OR"permite ampliar a recuperação de trabalhos que atendam a qualquer um desses contextos, enquanto o operador "AND"garante que os estudos selecionados estejam necessariamente relacionados à aplicação de técnicas de aprendizado de máquina. Dessa forma, a *string* de busca foi definida de modo a equilibrar abrangência e especificidade, assegurando a identificação de estudos relevantes para os objetivos desta revisão.

A partir dessa estratégia de busca, foram definidas duas Perguntas de Pesquisa (PP), que orientaram a identificação e a seleção dos estudos relevantes para esta revisão:

PP1: Quais estudos utilizam o repositório FARSEEING em métodos de detecção de quedas baseados em técnicas de *machine learning*?

PP2: Quais estudos utilizam dados de quedas reais (*real-world falls*) para o treinamento ou avaliação de algoritmos de *machine learning* aplicados à detecção de quedas?

Essas perguntas permitiram delimitar o escopo da revisão, direcionando a análise para trabalhos que utilizam dados provenientes de sensores vestíveis e aplicam técnicas de aprendizado de máquina na identificação automática de eventos de queda.

Tabela 5. Repositórios consultados e resultados da triagem

Repositório	**URL**	**Resultado da busca**	**Selecionado**
CAPES	<https://www.periodicos.capes.gov.br>	357	1
IEEE Xplore	<https://ieeexplore.ieee.org>	13	3
ScienceDirect	<https://www.sciencedirect.com>	8	0
EBSCO	<https://www.ebsco.com>	387	0
Google Scholar	<https://scholar.google.com.br>	135	1
Inclusão Manual	<https://biblioteca.univali.br/>	1	1
Total	–	**901**	**6**

Fonte: O autor.

Após a delimitação do escopo, foram estabelecidos critérios de inclusão e exclusão com o objetivo de selecionar os estudos mais relevantes para o tema investigado. O critério **CI01** determinou a inclusão apenas de artigos publicados entre 01/01/2015 e 30/06/2025, de modo a contemplar produções recentes e representativas do estado da arte na área. Já o critério **CI02** restringiu a seleção a estudos redigidos em língua inglesa, por se tratar do idioma predominante na comunicação científica internacional.

Tabela 6. Critérios de inclusão e exclusão.

Inclusão	**Exclusão**
CI01: Data de publicação entre 01/01/2015 e 30/06/2025.	**CE01:** Artigos que não tenham como foco principal a detecção de quedas.
	CE02: A análise do título não demonstrou aderência ao tema.
	CE03: A análise do *abstract* não demonstrou aderência ao tema.
	CE04: Artigos com restrição de acesso.
CI02: Texto escrito em inglês.	**CE05:** Artigos duplicados.
	CE06: Artigos que não utilizam sensores inerciais como fonte de dados.
	CE07: Artigos que não apresentam estudo experimental.

Fonte: O autor.

Após a aplicação dos critérios de inclusão, obteve-se um conjunto inicial de trabalhos potencialmente relevantes. Em seguida, foram aplicados os critérios de exclusão com o objetivo de refinar a amostra e manter apenas os estudos diretamente alinhados ao escopo desta pesquisa. O critério **CE01** foi utilizado para eliminar artigos cujo foco principal não fosse a detecção de quedas, ainda que mencionassem o tema de forma secundária. Os critérios **CE02** e **CE03** foram empregados em uma etapa de triagem preliminar, com base na análise do título e do resumo (*abstract*), respectivamente.

Além disso, foram excluídos os artigos com restrição de acesso (**CE04**), os trabalhos duplicados (**CE05**) e os estudos que não utilizaram sensores inerciais como fonte de dados (**CE06**), como aqueles fundamentados exclusivamente em visão computacional. Também foram descartados os artigos que não apresentaram estudo experimental (**CE07**). Uma vez que o escopo metodológico desta pesquisa está centrado na comparação com algoritmos clássicos de aprendizado de máquina, considerando uma possível aplicação futura em dispositivos vestíveis.

A *string* de busca definida foi aplicada em cinco repositórios científicos: CAPES, *IEEE Xplore*, ScienceDirect, EBSCO e Google Scholar. Ao todo, foram recuperados 901 trabalhos. As maiores quantidades de resultados foram obtidas nas bases EBSCO, com 387 registros, e CAPES, com 357 registros, seguidas por Google Scholar, com 135, IEEE Xplore, com 13, e ScienceDirect, com 8. Adicionalmente, um estudo foi incluído manualmente por sua relevância para o tema.

Após a remoção dos trabalhos duplicados e a aplicação dos critérios de exclusão, 872 estudos foram descartados, resultando em 29 trabalhos pré-selecionados para leitura na íntegra. Ao final do processo de triagem e elegibilidade, 6 artigos compuseram a amostra final desta revisão.

3.2 TRABALHOS SELECIONADOS

3.2.1 Nuñes (2023)

Nuñez (2023), neste trabalho discute o problema das quedas, que afetam pessoas, com uma prevalência maior entre pessoas com mais de 60 anos. Diversos fatores, como baixa massa muscular e uso de medicamentos, aumentam o risco de quedas. A prevenção é desafiadora devido à variedade de fatores predisponentes.

O trabalho analisa a utilização de algoritmos de *machine learning (ML)* e um sistema de notificação para detectar quedas. Os algoritmos são treinados com dados de acelerômetros e giroscópios do *dataset* público *FallAllD*, que fornece leituras de sensores posicionados na cintura, na região lombar

e no pulso dos usuários que realizaram as quedas simuladas. Na análise computacional realizada, o algoritmo *Random Forest* apresentou o melhor desempenho em termos de acurácia (97,22%).

A pesquisa de Nuñez (2023) propôs uma solução para detecção de quedas usando o algoritmo SVM embarcado em uma *Raspberry Pi Zero W*. A solução envia a notificação da queda via *WiFi* O modelo embarcado manteve a performance com tempo médio de classificação de 4,97 ms.

3.2.2 Maray et al. (2023)

Maray et al. (2023) investiga a aplicação de técnicas de *Transfer Learning* para a detecção automática de quedas utilizando redes neurais recorrentes do tipo *Long Short-Term Memory* (LSTM) e dados de acelerômetro provenientes de dispositivos vestíveis. O estudo parte do pressuposto de que a escassez de dados rotulados de quedas, aliada à heterogeneidade entre dispositivos e usuários, limita o desempenho e a generalização de modelos treinados do zero. Nesse contexto, os autores propõem a reutilização de conhecimento aprendido a partir de um conjunto de dados maior, visando melhorar o desempenho da detecção de quedas em cenários com conjuntos de dados pequenos e heterogêneos.

Para a avaliação da abordagem proposta, foram utilizados três conjuntos de dados coletados por dispositivos vestíveis posicionados no pulso dos participantes: *Microsoft Band (MSBAND), Huawei Watch* e um meta-sensor vestível. Os sinais de acelerômetro, compostos pelos eixos X, Y e Z, foram segmentados em janelas temporais e utilizados diretamente como entrada do modelo, sem a extração manual de características estatísticas. As quedas consideradas no estudo são predominantemente simuladas, realizadas por voluntários em ambiente controlado, enquanto as atividades da vida diária (ADLs) correspondem a movimentos reais executados pelos participantes.

O modelo LSTM foi inicialmente treinado utilizando o conjunto de dados *MSBAND*, caracterizado por maior volume de amostras, e posteriormente adaptado para os demais conjuntos de dados por meio de *Transfer Learning*. Nesse processo, as camadas recorrentes do modelo foram congeladas, preservando os padrões temporais aprendidos, enquanto as camadas densas finais foram ajustadas com uma quantidade reduzida de dados do dispositivo-alvo. O desempenho do modelo foi avaliado em diferentes cenários experimentais, incluindo validação treino/teste (70/30), validação cruzada *leave-one-person-out* e testes em tempo real, considerando tanto a transferência entre dispositivos distintos quanto entre diferentes posições do sensor. Os resultados apresentam na Tabela 7.

Tabela 7. Resultados obtidos por Maray et al. (2023) para detecção de quedas utilizando LSTM com e sem *Transfer Learning*

Rank	Cenário Experimental	Validação	F1-score	AUC-PR
1	MSBAND → Meta-sensor (*Transfer Learning*)	Train/Test (70/30)	**0,93**	0,85
2	Ensemble (Pulso esquerdo + direito, *Transfer Learning*)	Leave-One-Person-Out	**0,85**	–
3	MSBAND → Huawei Watch (*Transfer Learning*)	Train/Test (70/30)	**0,82**	0,80
4	MSBAND → Meta-sensor (*Transfer Learning*)	Leave-One-Person-Out	**0,79**	–
5	MSBAND → Huawei Watch (*Transfer Learning*)	Leave-One-Person-Out	**0,75**	–
6	Meta-sensor (Pulso esquerdo → direito, *Transfer Learning*)	Leave-One-Person-Out	**0,73**	–
7	MSBAND → Meta-sensor (sem *Transfer Learning*)	Train/Test (70/30)	0,81	0,81
8	MSBAND → Huawei Watch (sem *Transfer Learning*)	Train/Test (70/30)	0,68	0,73
9	MSBAND → Meta-sensor (sem *Transfer Learning*)	Leave-One-Person-Out	0,65	–
10	MSBAND → Huawei Watch (sem *Transfer Learning*)	Leave-One-Person-Out	0,64	–
11	Meta-sensor (Pulso esquerdo → direito, sem *Transfer Learning*)	Leave-One-Person-Out	0,63	–

Fonte: Adaptado de Maray et al. (2023).

Os resultados demonstraram que a aplicação de *Transfer Learning* melhora consistentemente o desempenho da detecção de quedas em todos os cenários avaliados, com ganhos expressivos no F1-score e na área sob a curva Precision–Recall, além de redução do número de falsos positivos quando comparado ao treinamento do modelo a partir do zero. O melhor desempenho foi observado na abordagem em conjunto, combinando dados dos pulsos esquerdo e direito. Os autores concluem que o *Transfer Learning* aplicado a redes LSTM constitui uma estratégia eficaz e escalável para a detecção de quedas em dispositivos vestíveis, especialmente em contextos com limitações de dados, destacando como trabalhos futuros a avaliação com dados de quedas reais envolvendo idosos e a incorporação de múltiplos sensores.

3.2.3 Palmerini et al. (2020)

No estudo realizado, foi avaliado a eficácia de algoritmos de aprendizado de máquina aplicados à detecção de quedas, utilizando exclusivamente dados reais provenientes do repositório FARSEEING. A motivação do estudo partiu da limitação recorrente em pesquisas anteriores, que frequentemente

utilizaram dados simulados, comprometendo a aplicabilidade prática dos modelos em cenários reais. Para contornar essa limitação, os autores propuseram uma abordagem baseada em características extraídas de um modelo temporal de múltiplas fases da queda, com a intenção de representar com maior precisão os padrões de sinais captados por sensores inerciais.

Foram analisadas 143 quedas reais, registradas por acelerômetros triaxiais posicionados na região lombar dos participantes. O estudo apresenta um pipeline de pré-processamento orientado à extração de atributos, no qual são definidos dois conjuntos principais de características: (i) um conjunto de 90 variáveis estatísticas convencionais, incluindo medidas como média, desvio padrão, valores máximos e mínimos, amplitude e outras estatísticas descritivas dos sinais de aceleração; e (ii) um conjunto de 65 atributos baseados em um modelo multifásico da queda, estruturado nas fases de pré-impacto, impacto e pós-impacto, permitindo capturar padrões específicos de cada etapa do evento.

Os sinais brutos são inicialmente segmentados em janelas móveis de 9 segundos, com sobreposição de 50%, estratégia adotada para garantir que o evento completo da queda — incluindo suas fases anteriores e posteriores — esteja contido em ao menos uma janela analisada. A partir dessas janelas, os dados são transformados em representações derivadas por meio de engenharia de características, na qual os sinais temporais são convertidos em vetores de atributos estáticos que sintetizam o comportamento do sinal ao longo do tempo.

Esse processo envolve a identificação das diferentes fases do evento dentro de cada janela, seguida da extração de variáveis estatísticas globais e específicas por fase, tais como intensidade do impacto, variações abruptas no sinal e padrões característicos antes e após o evento. Como consequência, ocorre uma redução da dimensionalidade temporal dos dados, uma vez que a série temporal original é substituída por um conjunto fixo de atributos, o que, conforme discutido no estudo.

Além disso, a utilização de janelas longas com sobreposição e a aplicação de subamostragem da classe majoritária — mantendo aproximadamente 10% das atividades de vida diária (ADLs) no treinamento — evidenciam uma estratégia voltada à mitigação do desbalanceamento entre classes.

A avaliação comparativa incluiu cinco algoritmos supervisionados: Naïve Bayes (NB), Regressão Logística (LR), Random Forest (RF), Support Vector Machines (SVM) e k-Nearest Neighbors (k-NN). A modelagem foi conduzida por meio de validação cruzada estratificada em cinco, com separação por sujeito, de modo a evitar dependência entre os conjuntos de treino e teste. Para o classificador SVM, foi realizada ainda uma validação interna adicional com busca em grade para ajuste dos hiperparâmetros. A avaliação contemplou métricas clássicas de desempenho, como sensibilidade,

especificidade e F1-score, além da taxa de falsos alarmes por hora (FA/h), considerada particularmente relevante para aplicações em cenário real.

Tabela 8. Resultados obtidos por Palmerini et al. (2020)

Classifier	Features	AUC	Sensitivity (%)	Specificity (%)	FA/h	PPV (%)	F1 (%)
Naïve Bayes	MultiPhase	0,996	88,1	99,1	1,09	39,0	54,1
Logistic Regression	MultiPhase	0,996	83,2	99,3	0,76	46,6	59,8
KNN	MultiPhase	0,958	83,9	99,2	0,92	42,1	56,1
Support Vector Machines	**MultiPhase**	**0,993**	**81,1**	**99,5**	**0,56**	**53,7**	**64,6**
Random Forests	MultiPhase	0,989	83,2	98,9	1,32	33,3	47,6
Naïve Bayes	Conventional	0,977	95,1	95,5	5,23	12,6	22,3
Logistic Regression	Conventional	0,987	84,6	98,5	1,70	28,3	42,5
KNN	Conventional	0,959	85,3	98,8	1,42	32,4	46,9
Support Vector Machines	Conventional	0,986	83,9	98,6	1,61	29,3	43,4
Random Forests	Conventional	0,985	88,8	98,6	1,57	31,0	45,9
Threshold-based (Kangas et al.)	–	–	30,1	99,3	0,82	22,9	26,3

Fonte: Adaptado de Palmerini et al. (2020).

Os melhores resultados foram obtidos pelo SVM treinado com as características derivadas do modelo multifásico de queda, alcançando sensibilidade de 81,1%, especificidade de 99,5%, F1-score de 64,6% e uma taxa de 0,56 falsos alarmes por hora. Embora a área sob a curva ROC (AUC = 0,993) indique elevada capacidade discriminativa global, essa métrica não captura integralmente o impacto do desbalanceamento, evidenciado pelo valor moderado de F1-score e pela baixa precisão. O valor moderado de F1-score evidencia o impacto do desbalanceamento entre quedas e atividades de vida diária, refletido principalmente na precisão (53,7%). Em comparação, os modelos baseados em características estatísticas convencionais apresentaram desempenho inferior, com reduções consistentes no F1-score e aumento da taxa de falsos alarmes, demonstrando que a estruturação temporal da extração de características contribuiu de forma relevante para a melhoria do desempenho.

O Random Forest apresentou sensibilidade semelhante, porém com maior taxa de falsos alarmes e menor equilíbrio geral. Já os métodos baseados exclusivamente em limiar e o Naïve Bayes com características convencionais obtiveram os desempenhos mais modestos, especialmente em termos de F1-score.

O estudo demonstrou que a utilização de dados reais de quedas possibilita alcançar desempenho relevante, embora ainda limitado em termos de sensibilidade, desde que acompanhada por estratégias

adequadas de extração de características e validação por sujeito. Os autores enfatizam a importância de métricas que reflitam o uso em tempo real, como a taxa de falsos alarmes por hora, e destacam o SVM como uma alternativa promissora para implementação embarcada, devido ao seu equilíbrio entre desempenho e custo computacional reduzido.

Apesar dos resultados promissores, observa-se um equilíbrio relevante entre sensibilidade e taxa de falsos alarmes, característico de problemas com forte desbalanceamento entre classes. Embora o modelo SVM com características multifásicas tenha apresentado a menor taxa de falsos alarmes por hora (0,56 FA/h), sua sensibilidade (81,1%) indica que uma parcela relevante de eventos de queda ainda não é detectada. Esse aspecto é particularmente crítico em aplicações reais, nas quais a não detecção de uma queda (falso negativo) pode representar riscos severos à integridade do indivíduo. Assim, o estudo evidencia a dificuldade de se alcançar simultaneamente alta sensibilidade e baixo número de falsos alarmes em cenários reais, reforçando a complexidade inerente ao problema.

No que se refere ao tratamento do desbalanceamento entre classes, os autores optaram por uma estratégia de subamostragem da classe majoritária durante o treinamento, mantendo aproximadamente 10% das atividades de vida diária (ADLs). Essa decisão visa reduzir o viés do classificador em relação à classe predominante, ainda que implique na perda de parte da variabilidade dos dados normais. Tal abordagem reflete um compromisso entre equilíbrio das classes e preservação de características relevantes do comportamento cotidiano dos usuários.

Como principal contribuição, o trabalho demonstra que a incorporação de conhecimento sobre a dinâmica temporal da queda, por meio de engenharia de atributos orientada a fases, pode melhorar significativamente o desempenho de algoritmos clássicos de aprendizado de máquina em dados reais. Além disso, o estudo reforça a importância de protocolos de validação baseados em separação por sujeito e da utilização de métricas alinhadas ao contexto de aplicação, como a taxa de falsos alarmes por hora. Dessa forma, Palmerini et al. estabelecem uma referência relevante na literatura ao evidenciar que, mesmo com dados reais e altamente desbalanceados, é possível obter desempenho competitivo por meio de escolhas metodológicas adequadas no pré-processamento, extração de características e avaliação dos modelos.

3.2.4 Caya et al. (2018)

Caya et al. (2018), investiga o uso de aprendizado de máquina supervisionado para detectar quedas por meio de um dispositivo vestível baseado em acelerômetro. A pesquisa busca solucionar a

escassez de dados reais de quedas, um fator que impacta diretamente a confiabilidade dos sistemas de detecção. Para isso, os autores desenvolveram um modelo que combina dados simulados e reais do FARSEEING. A proposta central do estudo é que a inclusão de dados reais melhora significativamente a precisão dos algoritmos na classificação de quedas.

A coleta de dados foi realizada com 15 voluntários, que executaram 10 tipos de quedas e 10 atividades diárias. No total, foram processadas 900 amostras, a partir das quais foram extraídas diversas características estatísticas, incluindo média, desvio padrão, variância, curtose e análise de componentes principais (*Principal Component Analysis – PCA*), Tabela 9.

O modelo foi treinado utilizando diferentes técnicas de aprendizado supervisionado, como Support Vector Machines (SVM), Decision Trees e K-Nearest Neighbors (KNN). Dois modelos foram comparados: um treinado exclusivamente com quedas simuladas e não quedas e outro que incluía quedas simuladas e não quedas, mais quedas reais. O segundo modelo apresentou um desempenho superior, alcançando 100% na detecção das quedas reais.

Tabela 9. Modelos de classificação com melhor desempenho e diferentes combinações de atributos

Combinação de Atributos	**Modelo de Classificação**	**Acurácia (%)**
Mean, std, var, kurt, skew, pca	Quadratic SVM	95.4
Mean, std, skew, pca	Cubic SVM	95.3
std, var, kurt, skew, pca	Cubic SVM	95.2
Mean, std, , kurt, , pca	Cubic SVM	95.1
Mean, std, , kurt, , pca	Fine KNN	95.1
Mean, , var, kurt, skew, pca	Quadratic SVM	95.1
Mean, std, var, skew	Cubic SVM	95.1
Mean, std, var, kurt, skew	Cubic SVM	95.0
Mean, std, kurt, skew, pca	Cubic SVM	95.0

Fonte: Adaptado Caya et al. (2018)

As medidas de desempenho utilizadas foram a acurácia; taxa de verdadeiros positivos e taxa de verdadeiros negativos. Os resultados da matriz confusão gerada com dados simulados consideram dois modelos. O modelo 1 usa dados de quedas e não quedas simuladas. O modelo 2 usa dados de quedas e não quedas simuladas e dados de quedas reais. O Modelo 1, treinado apenas com dados simulados, apresentou 257 verdadeiros positivos e 13 falsos negativos ao ser testado com dados simulados, com uma acurácia de 94,81%. Já o Modelo 2, que incorporou dados reais de queda ao treinamento, melhorou seu desempenho com 258 verdadeiros positivos e apenas 12 falsos negativos, alcançando acurácia de 95,56%. Na validação com quedas reais, o Modelo 1 obteve apenas 10% de taxa de verdadeiros positivos (3 TP e 27 FN), enquanto o Modelo 2 obteve 100% de acerto (30 TP e 0 FN), demonstrando a importância da inclusão de dados reais no treinamento. Já na validação com atividades simuladas, o

Modelo 2 também superou o Modelo 1, atingindo 96,67% de TPR contra 93,33%. Esses resultados evidenciam que algoritmos treinados com dados reais são mais robustos e confiáveis para detectar quedas em ambientes não controlados.

Os resultados evidenciam que modelos treinados apenas com dados simulados não conseguem generalizar bem para quedas reais. No entanto, a combinação de dados simulados e reais melhora consideravelmente a capacidade de detecção dos algoritmos. Para trabalhos futuros, os autores sugerem o uso de sensores mais precisos, a ampliação da base de dados e a aplicação de Transfer Learning para otimizar ainda mais a eficácia dos modelos.

3.2.5 Silva, Sousa e Cardoso (2018)

O trabalho de Silva, Sousa e Cardoso (2018), apresenta uma abordagem de aprendizado por transferência, para obter maior precisão na detecção de quedas em ambientes reais do repositório FARSEEING. O trabalho combina dados de quedas reais com dados de quedas simuladas. Foram utilizados 650 quedas e 410 não quedas coletadas com smartfones a 100 Hz e 22 arquivos com 23 quedas reais e 877 não quedas. Utilizando de classificadores.

ina, quando adequadamente configuradas e avalsfer learning para detecção de quedas, utilizando dados simulados de voluntários jovens combinados com dados reais do projeto (KLENK; SCHWICKERT, 2016), coletados em ambientes hospitalares com idosos. Os autores aplicaram algoritmos supervisionados — como ***Random Forest, Perceptron Multicamadas (MLP) e AdaBoost*** — em um pipeline que envolve extração de 16 atributos no domínio do tempo, padronização das features e técnicas de balanceamento como SMOTE, Balance Cascade e modelos de ranking. O objetivo foi avaliar o desempenho de classificadores treinados com diferentes combinações de dados reais e simulados. Os resultados mostram que o uso de dados simulados para treinamento pode melhorar a generalização quando testado com quedas reais, sendo a combinação de dados reais e simulados mais eficaz do que o uso exclusivo de um dos dois tipos. O método Balance Cascade destacou-se por apresentar menor taxa de classificações incorretas. A acurácia superou 95% em todos os cenários com conjuntos combinados, validando a eficácia do uso de transfer learning e aprendizado com dados desbalanceados.

Para lidar com as diferenças estruturais entre os dados reais e simulados, os autores adotaram um método de segmentação temporal padronizado, utilizando janelas de 7,5 segundos sem sobreposição para isolar eventos de queda e não queda. No caso do dataset FARSEEING, os eventos de queda

foram previamente anotados, e o trecho de 7,5 segundos centrado na queda foi considerado como amostra positiva. Já os trechos anteriores à queda, com no mínimo 10 segundos de antecedência, foram utilizados para compor as janelas negativas. A mesma lógica de segmentação foi aplicada aos dados simulados, garantindo uniformidade no processo. Além disso, os sinais foram resampleados para 100Hz, padronizando a frequência de amostragem entre dispositivos distintos. Essa abordagem permitiu não apenas uma comparação justa entre os dois tipos de dados, mas também a utilização de técnicas de aprendizado de máquina em um pipeline coeso, com extração de características consistentes e avaliação controlada de desempenho em cenários mistos de validação.

Tabela 10. Resultados dos modelos conforme combinações de treino e teste

#	Treinamento	Teste	Acc (%)	AUC (%)	Prec (%)	Rec (%)	FP	FN
1	Misto	Real	96	97	99	96	21	0
2	Real	Misto	97	98	99	97	20	0
3	Misto	Misto	99	67	99	99	1	4
4	Simulado	Real	72	61	98	92	144	3
5	Real	Simulado	71	66	71	71	73	20
6	Real	Real	99	99	100	99	3	0

Fonte: Silva, Sousa e Cardoso (2018).

Os resultados apresentados pelos autores demonstram que a inclusão de dados simulados de não quedas no treinamento (caso 1) não altera a taxa de falsos negativos (quedas reais preditas como não quedas), mas aumenta os falsos positivos, devido à maior variabilidade do conjunto de treinamento. Além disso, comparando o caso 1 (treinamento com quedas reais e simuladas) com o caso 4 (apenas quedas simuladas), observa-se uma melhora no desempenho ao combinar os tipos de dados. Já nos casos 2 e 3, quando o teste é realizado com um conjunto misto (quedas reais e não quedas simuladas), o modelo treinado com dados mistos (caso 3) apresenta menos falsos positivos do que aquele treinado apenas com dados reais (caso 2), ainda que o risco de falsos negativos seja maior. Esses achados reforçam a importância de equilibrar a composição dos dados no treinamento, de modo a maximizar a generalização e minimizar erros críticos na aplicação real.

3.2.6 Bourke et al. (2016)

Este artigo apresenta um estudo que utiliza *Machine Learning* para detectar quedas em idosos, por meio de sensores inerciais posicionados na quinta vértebra lombar (L5). O trabalho faz parte do projeto FARSEEING, que busca coletar dados reais de quedas para desenvolver algoritmos mais precisos e eficazes.

Para isso, os pesquisadores empregaram acelerômetros e giroscópios para registrar tanto quedas quanto atividades diárias normais (ADLs – *Activities of Daily Living*), com o objetivo de criar um modelo capaz de distinguir essas situações. A principal motivação do estudo é que a maioria dos algoritmos anteriores foi treinada com dados simulados, o que pode comprometer sua precisão quando aplicados no mundo real. Bourke et al. (2016), reforça a importância do uso de dados reais no desenvolvimento de algoritmos para detecção de quedas, evitando as limitações comuns de modelos baseados em simulações.

Este estudo utilizou um conjunto de dados contendo 100 quedas reais, das quais 89 quedas foram selecionadas por possuírem registros completos tanto do acelerômetro tri-axial (TA) quanto do giroscópio (TG).

As análises foram realizadas com base em séries temporais cinesiológicas, ou seja, dados que descrevem o movimento do corpo ao longo do tempo, extraídos dos sensores. Esses sinais foram processados e analisados no software *Matlab*, com o objetivo de identificar padrões que diferenciem quedas reais de ADLs.

Bourke et al. (2016), na construção do seu modelo usou 12 variáveis, Tabela 11, estatísticas e cinemáticas para realizar o treinamento, realizando diversas configurações entre as variáveis de entrada. Árvore de decisão C4.5, implementado no software Weka 3.6, treinamento com validação cruzada de 10 folds ajustado com número mínimo de instâncias por 5 folhas e utilizado técnica SMOTE (Synthetic Minority Over-sampling Technique) aplicado para balancear a base de dados, aumentando o número de amostras da classe minoritária (quedas).

Tabela 11. Variáveis utilizadas no modelo de detecção de quedas

Variável	**Descrição**
UPV	Valor de pico superior da aceleração (RSS)
LPV	Valor de pico inferior antes do UPV
Mudança angular máxima	Maior variação de ângulo de postura
Tempo do ângulo máximo	Tempo relativo em que o ângulo máximo ocorre
Velocidade vertical mínima	Menor valor de velocidade no eixo vertical
Velocidade horizontal máxima	Maior valor de velocidade horizontal (RSS)
Velocidade horizontal máxima antes do UPV	Valor anterior ao pico na horizontal
Velocidade vertical mínima antes do UPV	Valor anterior ao pico na vertical
Deslocamento vertical antes do UPV	Distância vertical percorrida antes do pico
Ângulo (TG)	Ângulo obtido via giroscópio
Velocidade angular (TG)	Velocidade de rotação medida pelo giroscópio
Aceleração angular (TG)	Aceleração rotacional medida pelo giroscópio

Fonte: O Autor.

O estudo destaca que, a inclusão de todas as doze variáveis disponíveis não resultou em uma melhoria estatisticamente significativa do desempenho, o que indica que a escolha criteriosa de variáveis é de alta relevância. Esse modelo segundo demonstrado na Tabela 12, em seus melhores resultados, alcançou 87,3 % de acurácia, com sensibilidade de 88% e especificidade de 87% considerando as atividades da vida diária (ADLs) com picos de aceleração acima de 1,5g. Quando aplicado ao conjunto completo de ADLs (n=3466), a especificidade aumentou para 99%, a capacidade do modelo em evitar falsos positivos em contextos não associados a quedas. A área sob a curva ROC foi de 88%, o que demonstra capacidade discriminativa do modelo (BOURKE et al., 2016).

Tabela 12. Composição dos sensores e desempenho dos modelos

Tri-axial acc Upper Peak Value (UPV)	Tri-axial acc Lower peak value (LPV)	Maximum angle Change	Maximum angle time	Vertical velocity minimum	Horizontal velocity maximum	Horizontal velocity max before UPV	Vertical velocity min before UPV	Vertical displacement before the UPV	Gyroscope Angle	Gyroscope Angular Acceleration	Gyroscope Angular Velocity	Correctly classified %	Sensitivity (n=367)	Specificity (n=368)	Accuracy	ROC area	Specificity (n=3466)
X	X	X	X			X	X	X	X	X	X	**87.3**	**0.88**	0.87	**0.87**	**0.88**	0.99
X	X	X	X	X	X	X	X	X	X	X	X	87.2	**0.88**	0.87	**0.87**	**0.88**	0.99
X	X	X	X						X	X	X	85.4	0.86	0.85	0.85	0.86	0.98
X	X					X	X	X				85.2	0.86	0.85	0.85	0.86	0.98
X						X	X	X				85.4	0.86	0.85	0.85	0.86	0.98
X									X	X	X	85.9	0.87	0.85	0.86	0.87	0.99
X	X								X	X	X	86.1	0.87	0.86	0.86	0.87	0.99
X	X	X	X			X	X	X				84.6	0.81	0.88	0.85	0.82	0.99
X	X											85.4	0.86	0.85	0.85	0.86	0.99
X	X	X	X									85.2	0.83	0.88	0.85	0.83	0.99
X	X			X	X							85.4	0.83	0.88	0.85	0.83	0.99
X	X	X	X	X	X							85.3	0.81	**0.91**	0.86	0.86	0.99

Fonte: Bourke et al. (2016).

Além disso, o trabalho reforça a importância de utilizar dados de quedas reais para validação dos algoritmos, algo ainda pouco explorado na literatura. A abordagem baseada em sensores lombares mostrou-se eficaz ao capturar padrões de movimento vertical e angular associados ao evento de queda. Tais achados consolidam o modelo como uma referência para sistemas de detecção em ambientes do cotidiano. O uso de técnicas de aprendizado de máquina aplicadas a sinais biomecânicos se mostra promissor para soluções assistivas voltadas à população idosa.

3.3 ANÁLISE COMPARATIVA

Esta subseção apresenta uma análise comparativa entre diversos estudos no campo da detecção de quedas, as base de dados, e o uso de sensores vestíveis, cada um abordando diferentes aspectos relacionados aos mecanismos de quedas e os que as antecedem. Os estudos selecionados exploram a coleta de dados através de sensores corporais e o uso de algoritmos de aprendizado de máquina para a detecção de quedas, assim como o monitoramento contínuo de parâmetros físicos.

A Tabela 13 resume as características principais de cada estudo, incluindo o tipo de *dataset* utilizado, os objetivos específicos do estudo, os algoritmos de processamento de dados implementados, os sensores empregados, as aplicações práticas e os principais resultados alcançados. Esta análise comparativa visa destacar as contribuições individuais de cada estudo para o avanço da tecnologia de sensores vestíveis na saúde e na qualidade de vida.

Tabela 13. Comparativo entre trabalhos selecionados para detecção de quedas

Trabalho	Descrição do dataset	Objetivo / estudo	Algoritmos	Sensores	Resultados
3.2.1 Nuñes (2023)	Dataset público **com apenas quedas simuladas**.	Implementar algoritmo de ML e estudar o desempenho em microcontrolador.	*Random Forest*, *SVM* e *k-NN*.	Acelerômetro e giroscópio.	**RF**: Acurácia [97,22%]; Especificidade [98,42%]; Sensibilidade [94,33%]. **SVM**: Acurácia [93,25%]; Especificidade [97,59%]; Sensibilidade [72,72%].
3.2.2 Maray et al. (2023)	Quedas simuladas, quedas reais e ADLs reais.	Aplicar *Transfer Learning* para melhorar a generalização da detecção de quedas em cenários com conjuntos de dados pequenos e dispositivos heterogêneos.	Rede neural recorrente do tipo *LSTM* com *Transfer Learning*.	Acelerômetro (pulso).	F1-score [93%]; AUC-PR [85%].
3.2.3 Palmerini et al. (2020)	Dados do FARSEEING **exclusivamente** (143) **quedas reais**.	Criar modelo com apenas dados reais; avaliar impacto no modelo.	*SVM, Random Forest, KNN, Naive Bayes, Logistic Regression*.	Acelerômetro.	Sensibilidade [95,1%]; Especificidade [95,5%]; PPV [12,6%].
3.2.4 Caya et al. (2018)	**Quedas reais** do FARSEEING + **quedas simuladas** com 15 voluntários.	Criar modelo com dados reais e simulados; avaliar impacto da fusão.	*SVM, Decision Tree, KNN; PCA*.	Acelerômetro.	Modelo 2 (com dados reais): Acurácia [95,56%]; 100% de detecção de quedas reais.
3.2.5 Silva, Sousa e Cardoso (2018)	**Quedas reais** do FARSEEING + **quedas simuladas** (coletadas com *smartphones*).	Aplicar *Transfer Learning* para melhorar a generalização com dados reais.	*Random Forest*, *MLP*, *AdaBoost*; *SMOTE* e *Balance Cascade*.	Acelerômetro em *smartphone* e sensores corporais.	Sensibilidade [96%]; Precisão [99%]; Acurácia [96%]; AUC [97%].
3.2.6 Bourke et al. (2016)	Dados do FARSEEING **exclusivamente** (100 quedas **reais**).	Avaliar modelos com dados reais usando sensor inercial.	*Decision Tree* – Regressão com seleção de variáveis estatísticas.	Acelerômetro e giroscópio.	Acurácia [87%]; Sensibilidade [88%]; Especificidade [87%] (1,5g); AUC [88%].
Este Trabalho	Dados do exclusivos do FARSEEING **exclusivamente** (23 quedas **reais**).	Avaliar modelos com dados reais com análise estatística aprofundada.	*DT, BRF, KNN, SVM, 1D-CNN e LSTM* – Classificação de evetos.	Acelerômetro.	Acurácia; Acurácia Balanceada; Sensibilidade; Especificidade; NIR; Kappa; VPP; VPN; Prevalência; p-valor; intervalo de confiança.

Fonte: O autor.

3.4 CONSIDERAÇÕES

Os trabalhos selecionados nesta seção serviram de base para a compreensão dos desafios e estratégias atuais na detecção de quedas por meio de sensores inerciais. As abordagens apresentadas contribuíram para a definição dos métodos adotados nesta pesquisa, oferecendo subsídios teóricos e práticos que orientaram tanto a preparação dos dados quanto a escolha e avaliação dos algoritmos de aprendizado de máquina empregados.

4 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento da solução proposta para o problema de detecção de quedas utilizando dados do repositório FARSEEING. O foco está na classificação do evento de queda, dada sua natureza rara e relevância. A proposta visa contribuir com estratégias de pré-processamento e modelagem de séries temporais de sensores inerciais (acelerômetros), de modo a maximizar o aprendizado dos algoritmos sobre o evento de interesse.

4.1 BASE DE DADOS

A base de dados utilizada neste trabalho, Seção 2.1.1,é composta por arquivos organizados em quatro níveis de verificação da ocorrência de queda, identificados na Figura 11 e são:

1. Possível Queda (Verificação Fraca);
2. Queda Provável (Verificação Moderada);
3. Queda Quase Certa (Verificação Alta);
4. Queda Confirmada (Verificação Completa).

Para fins deste trabalho, foi utilizada a classificação binaria para identificar o evento de queda, que para essa aplicação sendo "0"para não queda e "1"para queda.

A Figura 10, demostra um trecho de 4 segundos de arquivo de quedas disponibilizado. O objetivo é identificar os trechos com maior variabilidade de de sinais.

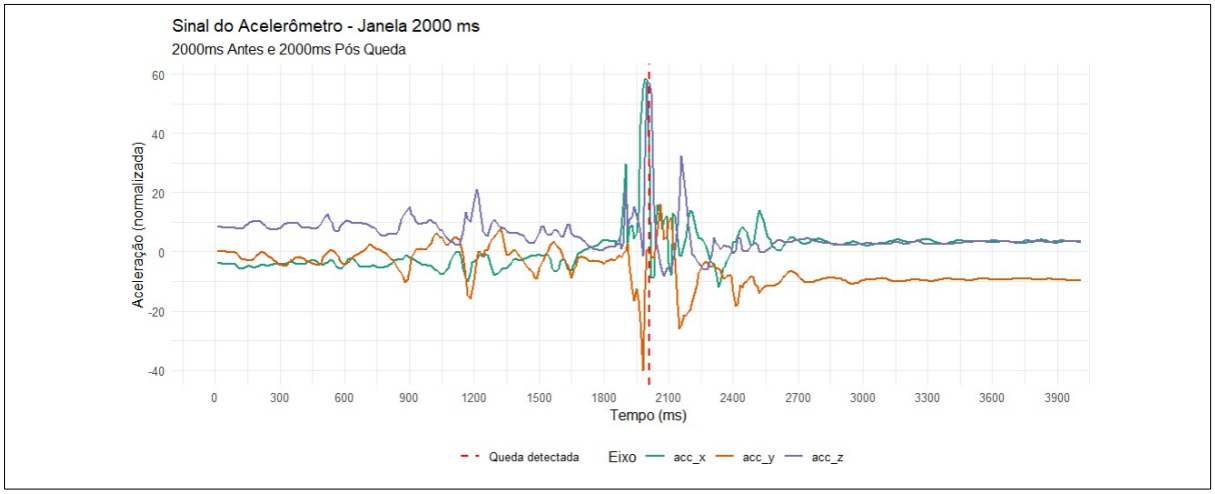

Figura 10. Trecho de queda.
Fonte: O autor.

4.1.1 Sensores Utilizados

A base de dados FARSEEING emprega os seguintes sensores vestíveis:

- *ActivPAL3* (acelerômetro) – 20 Hz, posicionado na coxa;
- *MiniMod* (acelerômetro) – 100 Hz, posicionado na região lombar (L5);
- *Hybrid* (acelerômetro e giroscópio) – 100 Hz, também posicionado na região lombar (L5).

Nesta dissertação serão utilizados apenas os dados provenientes dos acelerômetros. Essa decisão foi motivada pela própria estrutura da base de dados disponibilizada, na qual nem todos os arquivos contêm informações completas de todos os sensores, conforme ilustrado na Figura 11. Observa-se que apenas uma parcela dos registros possui dados combinados de acelerômetro e giroscópio e sem dados de magnetrômetro, enquanto a todos os arquivos contém dados de aceleração, além de apresentarem diferentes frequências de amostragem.

Dessa forma, a utilização conjunta de múltiplos sensores implicaria na necessidade de descarte de parte significativa dos dados ou na aplicação de técnicas de imputação e reconstrução de sinais, o que poderia introduzir viés, inconsistências e perda de qualidade nas análises. Em cenários com dados reais e já limitados em quantidade de eventos de queda, a presença de dados faltantes compromete a os modelos e a confiabilidade dos resultados obtidos.

Adicionalmente, destaca-se que o acelerômetro triaxial é reconhecido na literatura como suficiente para a detecção de quedas, uma vez que captura de forma direta as variações abruptas de movimento, os picos de impacto e os períodos de baixa aceleração associados ao evento. Assim, a adoção exclusiva desse sensor permite manter a consistência do conjunto de dados, maximizar o aproveitamento dos registros disponíveis e garantir maior aplicabilidade prática da solução proposta, sem prejuízo na capacidade de representação do fenômeno analisado.

4.1.2 Resumo dos Arquivos

A base de dados utilizada neste trabalho é composta por 22 arquivos contendo registros de sensores inerciais utilizados para a detecção de quedas. Esses arquivos foram organizados segundo três critérios principais: tipo de sensor, frequência de amostragem e nível de verificação de queda. A Figura 11 apresenta a distribuição detalhada dos arquivos segundo esses critérios.

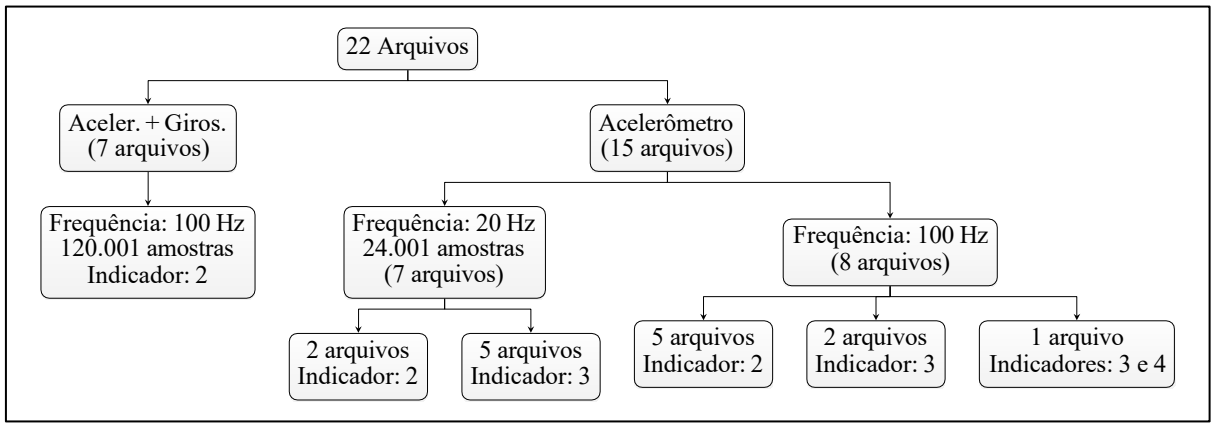

Figura 11. Visão geral da estrutura da base de dados disponibilizada: quantidade, sensores, frequência e tipo de falha.
Fonte: O Autor.

Inicialmente, os arquivos se dividem entre duas categorias principais: aqueles que possuem sensores de acelerômetro e giroscópio combinados (7 arquivos) e aqueles com apenas acelerômetro (15 arquivos). Todos os arquivos da primeira categoria foram registrados com frequência de 100 Hz, contendo 120.001 amostras por arquivo e são identificados com indicador de falha nível 2.

A segunda categoria, composta exclusivamente por sensores de acelerômetro, subdivide-se conforme a frequência de aquisição. São 7 arquivos com frequência de 20 Hz e 8 arquivos com 100 Hz. Entre os arquivos com 20 Hz, dois possuem indicador de falha 2 e os outros cinco possuem indicador de falha 3. Já entre os arquivos com 100 Hz, cinco estão marcados com indicador de falha 2, dois com

indicador 3, e um arquivo específico apresenta múltiplos eventos com indicadores 3 e 4.

Tabela 14. Composição da base de dados disponibilizado pelo repositório FARSEEING

Arquivo	Categoria de sensores	# Amostras	Duração (s)	Hz	Indicador de Falha
F_04835861-01-2013-10-16-13-19-35.csv	Acelerômetro	24 001	1 200	20	2
F_17744725-01-2009-02-24-16-26-44.csv	Acelerômetro	120 001	1 200	100	2
F_20989385-01-2013-05-16-16-16-00.csv	Acelerômetro	24 001	1 200	20	2
F_23112025-01-2013-05-21-06-21-54.csv	Acelerômetro	24 001	1 200	20	3
F_36551836-01-2013-05-21-17-39-57.csv	Acelerômetro	24 001	1 200	20	3
F_38243026-05-2009-04-29-08-53-16.csv	Acelerômetro	120 001	1 200	100	3
F_42990421-01-2011-02-19-15-59-57.csv	Acel.+Girosc.	120 001	1 200	100	2
F_42990421-02-2011-02-19-22-58-03.csv	Acel.+Girosc.	120 001	1 200	100	2
F_42990421-03-2011-03-23-16-50-02.csv	Acel.+Girosc.	120 001	1 200	100	2
F_47451392-01-2014-01-22-22-42-34.csv	Acelerômetro	24 001	1 200	20	3
F_63414187-01-2013-05-30-22-48-27.csv	Acelerômetro	24 001	1 200	20	3
F_67458491-01-2013-11-07-12-36-22.csv	Acelerômetro	24 001	1 200	20	3
F_72858619-01-2008-06-26-07-27-49.csv	Acelerômetro	120 001	1 200	100	2
F_72858619-02-2008-06-26-11-29-16.csv	Acelerômetro	120 001	1 200	100	2
F_74827807-04-2009-02-09-22-39-54.csv	Acelerômetro	120 001	1 200	100	3, 4
F_74827807-07-2009-02-16-19-10-44.csv	Acelerômetro	120 001	1 200	100	2
F_79761947-03-2012-06-24-02-09-34.csv	Acel.+Girosc.	120 001	1 200	100	2
F_91943076-01-2009-02-26-12-42-02.csv	Acelerômetro	120 001	1 200	100	2
F_91943076-02-2009-02-26-22-15-13.csv	Acelerômetro	120 001	1 200	100	2
F_96201346-01-2011-05-18-18-57-16.csv	Acel.+Girosc.	120 001	1 200	100	2
F_96201346-03-2011-05-21-06-19-42.csv	Acel.+Girosc.	120 001	1 200	100	2
F_96201346-05-2011-06-03-22-36-07.csv	Acel.+Girosc.	120 001	1 200	100	2

Fonte: O Autor.

4.1.3 Processamento dos Arquivos

Os arquivos originais foram fornecidos no formato .mat (Matlab). Para o processamento, optou-se pela linguagem R por apresentar vantagens com manipulação dos dados nessa etapa. A biblioteca utilizada para a leitura dos dados foi a *rhdf5*. Após a importação dos arquivos, foram atribuídos nomes apropriados às colunas para facilitar a análise e estruturação dos dados.

4.2 TRANSFORMAÇÃO DA BASE DE DADOS

Após a leitura dos arquivos originais no formato .mat, foi necessário realizar uma série de transformações para que os dados pudessem ser utilizados na etapa de pré-processamento e modelagem. O objetivo principal dessa fase foi padronizar as estruturas dos arquivos, adaptar frequências de

amostragem e gerar um conjunto de dados unificado e tratável por algoritmos de aprendizado de máquina.

4.2.1 Pré-Processamento

Inicialmente, os arquivos foram convertidos de .mat para .csv, mantendo-se apenas as colunas relacionadas aos eixos dos acelerômetros e ao marcador de evento (indicador de queda). Foram eliminados metadados todas as demais colunas, tento em vista que as informações dos giroscópios são insuficientes e não seria significativas para este trabalho. Durante essa etapa, também foram realizados os seguintes procedimentos:

- Uniformização dos nomes das colunas para garantir consistência entre os arquivos;
- Conversão das frequências para 100 Hz nos casos de arquivos originalmente gravados a 20 Hz, por meio de interpolação linear;
- Aplicação de filtros para suavizar ruídos de leitura.

4.2.1.1 Interpolação Linear

A **interpolação linear** preenche valores desconhecidos supondo variação em linha reta entre dois pontos amostrados consecutivos. Dados $(t_i,\ y_i)$ e $(t_{i+1},\ y_{i+1})$, o valor estimado para qualquer $t \in [t_i,\ t_{i+1}]$ é:

$$\hat{y}(t) = y_i + \frac{t - t_i}{t_{i+1} - t_i}\left(y_{i+1} - y_i\right). \tag{4.1}$$

No presente trabalho, cada série foi reamostrada de 20 Hz ($\Delta t = 0,05$ s) para 100 Hz ($\Delta t = 0,01$ s), ou seja, inseriram-se *quatro* novos pontos entre duas amostras originais. Para o eixo x (demais eixos análogos), obtêm-se os valores:

$$\hat{x}_{i,k} = x_i + \frac{k}{5}\left(x_{i+1} - x_i\right), \qquad k = 0, 1, 2, 3, 4, 5. \tag{4.2}$$

Assim, $k = 0$ e $k = 5$ reproduzem exatamente as amostras originais, enquanto $k = 1$ a $k = 4$ geram as quatro amostras intermediárias equidistantes ($0,01$ s) necessárias para elevar a série a 100 Hz.

4.2.1.2 Padronização com Z-Score

Foi aplicada a padronização via *z-score*, uma técnica amplamente utilizada para transformar distribuições de dados em escalas com média zero e desvio padrão unitário. A transformação é descrita pela Equação 4.3:

$$z(t) = \frac{acc(t) - \bar{a}cc}{\sigma} \tag{4.3}$$

Onde $\bar{a}cc$ representa a média dos valores do eixo e σ é o desvio padrão. Essa padronização permite destacar oscilações abruptas nos sinais, que são características esperadas nos eventos de queda, mesmo quando comparadas entre indivíduos com diferentes padrões de movimento ou intensidade de passos.

4.2.1.3 Aplicação e Comparação Visual

As abordagens foram aplicadas sobre janelas temporais com 150 ms, previamente recortadas ao redor dos eventos de interesse. Os resultados foram analisados visualmente por meio de gráficos que destacam os valores normalizados e os pontos de ocorrência da queda, permitindo observar a eficácia das técnicas em destacar a perturbação gerada pelo evento. A Figura 12 ilustra esse comportamento antes e depois da aplicação da *normalização* e *z-score*, demonstrando a clareza com que a técnica evidencia o evento.

Figura 12. Na parte superior da figura é aplicada a normalização por media dos eixo e, na parte inferior, é aplicado o z-score.

Fonte: O autor.

4.2.2 Segmentação em Janelas Temporais

Para adequar os dados ao formato de entrada dos algoritmos, os sinais contínuos foram segmentados em janelas fixas de **3000 milesegundos(ms)**, o que corresponde a **300 amostras**, tendo como centro o evendo de queda para este caso de uma queda. A divisão do range a utiliza a seguinte regra. Range = range - queda ou range + queda.

A Figura 13, ilustra uma janela com 1 quedas, recortada em 150ms antes da queda e 150ms após a queda. As janelas foram rotuladas de acordo com os marcadores de queda presentes nos arquivos, sendo:

- **Janela queda:** Contém o evento de queda dentro da faixa temporal;
- **Janela não queda:** Corresponde a trechos anteriores à queda ou atividades diárias normais.

4.2.3 Balanceamento

A segmentação dos sinais contínuos foi realizada por meio de janelas temporais fixas. Nesse processo, o sinal bruto é dividido em trechos de curta duração sem sobreposição, permitindo que os algoritmos de aprendizado de máquina operem sobre blocos de dados contextualizados.

Optou-se por janelas de **3000 ms**, Figura 13, estes centrados no evento de queda, sendo **1500 ms** antes da queda e **1500 ms** pós queda, note que na imagem temos ***W1,W2,W4,W5***, trechos de **600 ms** classificados como não queda e ***W3*** central o trecho de queda. O tamanho da população o que equivale a 300 amostras para sensores a 100 Hz. Esse intervalo foi definido com base na variância do sinal no instante da queda, que demonstram que a maior parte da atividade cinemática associada à queda está concentrada em curtos períodos ao redor do impacto. Silva, Sousa e Cardoso (2018), empregaram janelas padronizadas de 7,5 segundos (com 750 amostras a 100 Hz) para isolar quedas e não quedas, centradas no ponto de impacto, de forma a garantir consistência na entrada dos classificadores e facilitar o treinamento supervisionado.

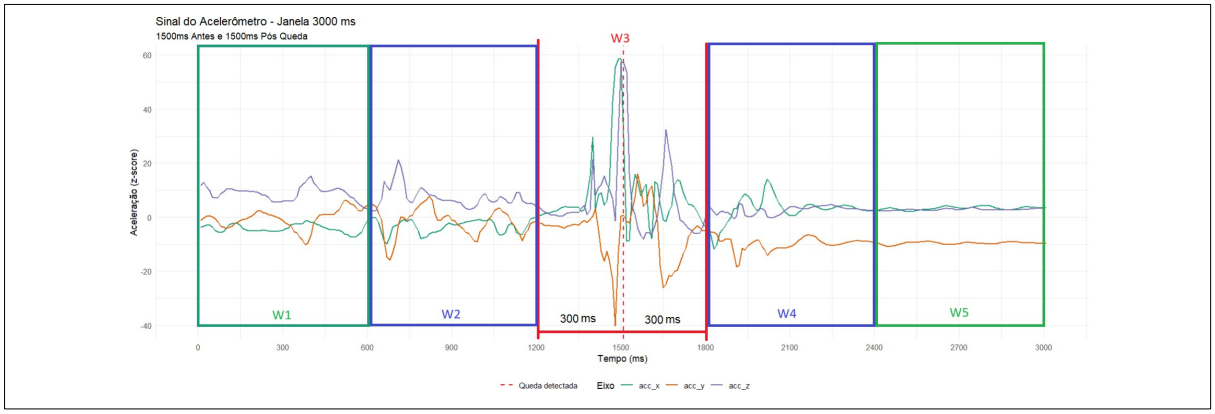

Figura 13. Janela ADLs e Queda
Fonte: O autor.

Queda 600ms = 60 linhas 30 antes e 30 depois da queda Não queda 4x 600ms = 240 linhas totais 2 janelas antes, 2 depois (sem sobrepor queda), em um total de 92 não quedas e 23 quedas Figura 14.

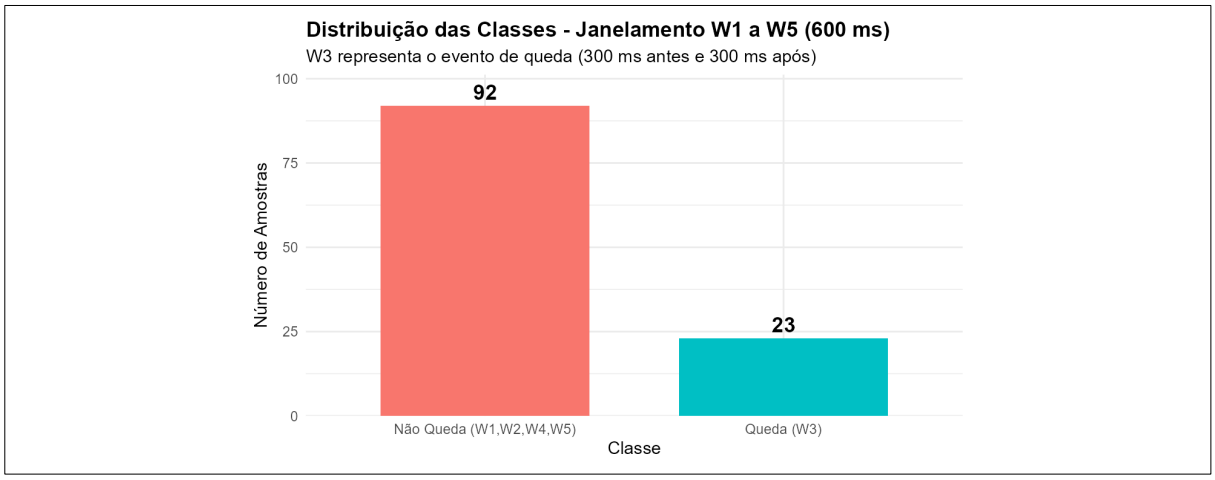

Figura 14. Distribuição das classes queda e não queda
Fonte: O autor.

O uso de janelas fixas traz vantagens como a padronização do vetor de entrada, redução da variabilidade interamostral e controle da dimensionalidade dos dados. Além disso, facilita a aplicação de técnicas de extração de características, como média, desvio padrão, energia e curtose, que são comumente utilizadas em estudos de detecção de quedas (CAYA et al., 2018).

Considerando o desbalanceamento natural entre quedas e não-quedas, foi aplicado o método *under-sampling* para equilibrar as classes nas análises iniciais. As janelas finais foram exportadas em

Tabela 15. Estatísticas descritivas das 10 variáveis padronizadas (Z-score) dos sinais dos acelerômetros.

Variável	Min	Q1	Mediana	Média	Q3	Máx	Desvio
acc_x_1	−2.830	−0.913	−0.569	−0.406	0.065	1.104	0.725
acc_x_10	−1.899	−0.901	−0.548	−0.413	0.080	1.054	0.699
acc_x_11	−1.718	−0.571	0.056	0.022	0.732	1.823	0.854
acc_x_12	−3.135	0.139	0.476	0.387	0.920	2.553	0.849
acc_x_13	−1.783	−0.833	−0.419	−0.378	0.016	1.216	0.678
acc_x_14	−1.917	−0.646	0.007	−0.019	0.609	2.054	0.880
acc_x_15	−2.494	0.113	0.492	0.380	0.844	2.426	0.795
acc_x_16	−1.363	−0.871	−0.529	−0.373	−0.009	1.378	0.666
acc_x_17	−2.693	−0.796	0.005	−0.062	0.572	2.378	0.967
acc_x_18	−2.034	0.040	0.474	0.373	0.843	2.782	0.804

Fonte: O autor.

formato tabular e unificadas em um único arquivo .csv, pronto para ser utilizado nas etapas de treino e validação dos modelos preditivos.

4.3 ALGORITMOS DE ML

Após a etapa de pré-processamento descrita e a construção do conjunto de atributos a partir das janelas temporais do acelerômetro, foram conduzidos experimentos de classificação binária com diferentes algoritmos de aprendizado de máquina, abrangendo modelos clássicos e arquiteturas de aprendizado profundo. Os modelos avaliados incluem *Decision Tree* (DT), *Balanced Random Forest* (BRF), *Support Vector Machine* (SVM), *k-Nearest Neighbors* (KNN), *1D-CNN* e *LSTM*.

Os experimentos apresentados nesta seção correspondem aos resultados obtidos no conjunto de teste, com separação estratificada e, quando aplicável, controle de vazamento de informação por agrupamento de evento. A análise prioriza métricas sensíveis ao desbalanceamento, com ênfase na classe minoritária (queda), especialmente sensibilidade (*recall*), acurácia balanceada e valores preditivos, além da interpretação das matrizes de confusão.

4.3.1 Árvore de Decisão - Hiperparâmetros

A Árvore de Decisão foi configurada a partir de experimentos sucessivos com diferentes combinações de hiperparâmetros. O objetivo foi obter um modelo interpretável e alinhado à dinâmica biomecânica das quedas em idosos, sem incorrer em sobreajuste decorrente da limitada quantidade de dados disponíveis. Após a análise de inúmeras configurações, o modelo final foi definido com profun-

didade máxima 2 (max_depth=2), seleção de atributos via logaritmo base 2 (max_features='log2') e pesos ajustados para enfatizar a classe minoritária (class_weight=0:1, 1:4).

Justificativa dos Hiperparâmetros

- **criterion = 'entropy'.** A entropia foi selecionada como medida de impureza para privilegiar divisões mais informativas, sobretudo em cenários desbalanceados. Em datasets pequenos, a entropia costuma produzir splits mais sensíveis a variações relevantes entre as classes, contribuindo para capturar nuances que diferenciam quedas reais de atividades intensas que não resultam em queda.
- **max_depth = 2.** A profundidade máxima de dois níveis foi deliberadamente escolhida após observar que árvores mais profundas apresentavam maior risco de sobreajuste, memorizando flutuações específicas dos sinais de acelerômetros. A limitação da profundidade força o modelo a concentrar suas decisões nos atributos de maior importância fisiológica — tipicamente picos de aceleração e métricas de energia vertical — resultando em uma árvore mais generalizável e altamente interpretável. Essa escolha também se alinha à literatura, que recomenda modelos simples em conjuntos de dados com poucas amostras.
- **min_samples_leaf = 3.** A exigência de pelo menos três amostras por folha atua como uma forma de regularização, impedindo a criação de folhas suportadas por eventos isolados. Considerando o tamanho reduzido do dataset, essa restrição reduz a probabilidade de decisões instáveis e reforça consistência estatística nas regiões de decisão.
- **max_features = 'log2'.** A seleção de atributos por log2(p) — sendo (p) o número de features — diminui a chance de que uma única variável dominante seja usada repetidamente nos splits. Esse critério reduz correlações artificiais entre variáveis derivadas do próprio sensor e força o modelo a explorar combinações complementares. Nos experimentos realizados, esse ajuste demonstrou desempenho superior ao uso de 'sqrt' ou ao uso irrestrito de todas as features.
- **class_weight = 0:1, 1:4.** A atribuição de peso quatro vezes maior para a classe "queda" aumenta a penalidade dos falsos negativos durante a escolha dos splits. Isso é crucial em aplicações onde deixar de detectar uma queda é considerado muito mais problemático do que gerar falsos alarmes. Esse ajuste influenciou diretamente a sensibilidade do modelo, garantindo maior ênfase na classe minoritária sem comprometer a interpretabilidade.

- **ccp_alpha = 0.0.** Optou-se por não aplicar poda baseada em complexidade, pois os demais hiperparâmetros já impunham forte regularização estrutural. A poda adicional não apresentou benefícios durante os experimentos e, em alguns casos, reduziu a separabilidade entre quedas e não quedas.

Resultados

A matriz de confusão na Tabela 16 apresenta os resultados do classificador *Decision Tree Classifier*, considerando uma divisão de dados em 70% para treino e 30% para teste.

Tabela 16. Matriz de Confusão – Decision Tree

Real	**Predito**	
	0	**1**
0	27	1
1	0	7

Verdadeiros Negativos (27): Representam eventos da classe "Não Queda" corretamente classificados como "Não Queda". Este valor contribui diretamente para a métrica de especificidade do modelo.

Falsos Positivos (1): Um evento de "Não Queda" foi incorretamente classificado como "Queda". Esse tipo de erro gera um alarme falso, podendo causar desconforto ou preocupação ao indicar uma situação de risco inexistente.

Falsos Negativos (0): Nenhuma instância da classe "Queda" foi incorretamente classificada como "Não Queda". Este é considerado o erro mais crítico em sistemas de detecção de quedas, pois poderia impedir que um evento real recebesse assistência.

Verdadeiros Positivos (7): Sete eventos de "Queda" foram corretamente identificados pelo modelo. Esse resultado contribui diretamente para a sensibilidade do sistema, indicando sua capacidade de detectar corretamente os eventos de queda.

A Tabela 17 apresenta o desempenho da ***Árvore de Decisão*** configurada com profundidade máxima igual a 2 e limiar de decisão $t = 0{,}40$, valor selecionado por maximizar o equilíbrio entre sensibilidade e especificidade, mantendo controle sobre falsos positivos.

Tabela 17. Estatísticas de desempenho do modelo Árvore de Decisão (threshold = 0,40)

Métrica	**Valor**
Sensibilidade (Recall)	1.0000
Acurácia Balanceada	0.9821
Intervalo de Confiança 95 %	(0.9162, 1.0000)
Taxa Sem Informação (NIR)	0.8000
p-valor [Acc > NIR]	0.00396
Kappa de Cohen	0.9153
p-valor do teste de McNemar	1.0000
Acurácia	0.9714
Especificidade	0.9643
Valor Preditivo Positivo (PPV)	0.8750
Valor Preditivo Negativo (NPV)	1.0000
Prevalência	0.2000
Taxa de Detecção	0.2000
Prevalência Detectada	0.2286

Fonte: O autor.

4.3.2 Balance Random Forest - Hiperparâmetros

O modelo Random Forest foi treinado em Python, apresentando resultados promissores no treinamento com as variáveis estatísticas extraídas e utilizando divisão 70% Treinamento e 30% para teste. A seguir são detalhadas as configurações utilizadas e apresentados os resultados obtidos.

Parâmetros do modelo.

- **n_estimators = 600** – grande numero de árvores reduz variância e melhora a cobertura da classe rara.
- **max_depth = 5** = tentar equilibrar detalhamento × generalização.
- **criterion = entropy** – favorece splits que separam a minoria mesmo com poucos exemplos.
- **min_samples_leaf = 1** – evita folhas com ruído de 1 exemplo, mas permite *outliers*.
- **sampling_strategy = auto** – cada árvore vê toda a minoria + #maj = #min, amplificando o sinal das quedas.
- **bootstrap = true & oob_score = true** – habilitou validação fora-da-bolsa para monitorar AP sem hold-out extra.

Resultados

A matriz de confusão na Tabela 18 apresenta os resultados do classificador Balanced Random Forest (BRF).

Tabela 18. Matriz de Confusão – BRF

Real	**Predito**	
	0	**1**
0	28	0
1	0	7

Fonte: O autor.

Verdadeiros Negativos (28): Representam eventos da classe "Não Queda" corretamente classificados como "Não Queda". Este valor contribui diretamente para a especificidade do modelo.

Falsos Positivos (0): Nenhum evento de "Não Queda" foi incorretamente classificado como "Queda". Dessa forma, não foram gerados alarmes falsos que poderiam indicar uma situação de risco inexistente.

Falsos Negativos (0): Nenhuma instância da classe "Queda" foi incorretamente classificada como "Não Queda". Este é considerado o erro mais crítico em sistemas de detecção de quedas, pois poderia impedir que um evento real recebesse assistência.

Verdadeiros Positivos (7): Sete eventos de "Queda" foram corretamente identificados pelo modelo, refletindo elevada capacidade de detecção dos eventos de queda.

A Tabela 19, apresenta as métricas obtidas com o *Balanced Random Forest* após o ajuste do limiar que prioriza o *recall* da classe positiva (quedas). O limiar foi escolhido pela maximização do F_2, penalizando duas vezes mais os falsos negativos do que os falsos positivos.

Tabela 19. Estatísticas de desempenho do modelo BRF

Métrica	**Valor**
Sensibilidade (Recall)	1.0000
Acurácia Balanceada	1.0000
Intervalo de Confiança 95 %	(1.0000, 1.0000)
Taxa Sem Informação (NIR)	0.8000
p-valor [Acc > NIR]	0.0004
Kappa de Cohen	1.0000
p-valor do teste de McNemar	1.0000
Acurácia	1.0000
Especificidade	1.0000
Valor Preditivo Positivo (PPV)	1.0000
Valor Preditivo Negativo (NPV)	1.0000
Prevalência	0.2000
Taxa de Detecção	0.2000
Prevalência Detectada	0.2000

Fonte: O autor.

Os resultados iniciais obtidos no conjunto de teste *holdout* apresentaram acurácia balanceada de 100 %, sensibilidade de 100 % e especificidade de 100 %, indicando separação perfeita entre as classes para a divisão específica dos dados utilizada. Entretanto, resultados perfeitos em conjuntos de teste únicos podem levantar questionamentos quanto à robustez e à capacidade de generalização do modelo, especialmente em bases de dados relativamente pequenas.

Dessa forma, com o objetivo de avaliar a estabilidade do desempenho do modelo e reduzir a possibilidade de resultados otimistas decorrentes de uma única divisão dos dados, foi realizada uma validação adicional baseada em múltiplas repetições do procedimento de separação entre treino e teste, com preservação dos grupos de origem dos eventos. Essa estratégia é associada à validação de Monte Carlo com agrupamento (*Group Monte Carlo Cross-Validation*), por combinar sucessivas partições independentes da base com controle da dependência entre amostras pertencentes ao mesmo grupo.

Como descrito anteriormente, a base de dados utilizada neste estudo é composta por 115 eventos, sendo 23 eventos de queda e 92 eventos de não queda. Em cada repetição do processo de validação, foi realizada uma divisão do tipo *holdout*, mantendo aproximadamente 70 % dos dados para treinamento e 30 % para teste. Assim, cada conjunto de teste continha aproximadamente 35 eventos, sendo cerca de 28 eventos de não queda e 7 eventos de queda.

Para aumentar a robustez da avaliação, esse procedimento foi repetido dez vezes, utilizando diferentes sementes para a geração das divisões dos dados. As sementes utilizadas foram: 7, 13, 21, 29,

37, 42, 84, 105, 202 e 999.

A adoção de múltiplas partições independentes permitiu avaliar a consistência do desempenho do modelo sob diferentes configurações de treinamento e teste. Além disso, a preservação dos grupos de origem contribuiu para reduzir o risco de vazamento de informação entre os conjuntos, tornando a análise menos dependente de uma única partição e mais adequada às características da base utilizada.

Ao longo das dez repetições, cada divisão avaliou aproximadamente 35 eventos, totalizando cerca de 350 avaliações ao final do processo. Considerando a distribuição média das classes nos conjuntos de teste, foram avaliados aproximadamente 280 eventos de não queda e 70 eventos de queda ao longo de todas as repetições.

A matriz de confusão agregada obtida a partir dessas avaliações é apresentada na Tabela 20.

Tabela 20. Matriz de Confusão – BRF (Validação Repetida)

Real	**Predito**	
	0	**1**
0	268	12
1	0	70

Fonte: O autor.

A matriz de confusão agregada permite observar o comportamento do modelo ao longo de todas as repetições realizadas.

Verdadeiros Negativos (268): instâncias da classe "Não Queda" corretamente classificadas como "Não Queda". Esse valor contribui diretamente para o cálculo da especificidade do modelo.

Falsos Positivos (12): instâncias da classe "Não Queda" classificadas incorretamente como "Queda". Esses casos representam alarmes falsos, que podem gerar desconforto ou acionamentos desnecessários em aplicações reais.

Falsos Negativos (0): nenhuma instância da classe "Queda" foi incorretamente classificada como "Não Queda" (**FN**=0). Esse tipo de erro é considerado o mais crítico em sistemas de detecção de quedas, pois pode impedir que um evento real receba assistência adequada.

Verdadeiros Positivos (70): instâncias da classe "Queda" corretamente identificadas pelo modelo. Esse resultado evidencia a elevada capacidade do algoritmo em detectar eventos de queda.

A Tabela 21 apresenta as métricas de desempenho obtidas com o modelo ***Balanced Random***

Forest considerando a validação repetida. O limiar de decisão foi ajustado priorizando o *recall* da classe positiva (quedas), sendo determinado pela maximização da métrica F_2, que penaliza duas vezes mais os falsos negativos do que os falsos positivos.

Tabela 21. Estatísticas de desempenho do modelo BRF (Validação Repetida)

Métrica	**Valor**
Sensibilidade (Recall)	1.0000
Acurácia Balanceada	0.9786
Intervalo de Confiança 95 %	(0.9467, 0.9848)
Taxa Sem Informação (NIR)	0.8000
p-valor [Acc > NIR]	0.00000
Kappa de Cohen	0.8993
p-valor do teste de McNemar	0.00049
Acurácia	0.9657
Especificidade	0.9571
Valor Preditivo Positivo (PPV)	0.8537
Valor Preditivo Negativo (NPV)	1.0000
Prevalência	0.2000
Taxa de Detecção	0.2000
Prevalência Detectada	0.2343

Fonte: O autor.

Os resultados obtidos indicam desempenho elevado do modelo, com sensibilidade de 100 % e especificidade de 95,71 %, resultando em acurácia balanceada de 97,86 %. Esses resultados demonstram que o modelo apresenta elevada capacidade de detectar eventos de queda sem comprometer significativamente a classificação correta de eventos de não queda. A ausência de falsos negativos observada nas repetições reforça a adequação do modelo para aplicações de detecção de quedas, nas quais a identificação correta do evento é considerada prioritária em relação à ocorrência eventual de alarmes falsos.

t

4.3.3 Algoritmo SVM - Hiperarâmetros

O conjunto de dados foi novamente separado em 70% para treino e 30% para teste utilizando a função *train_test_split* com estratificação *stratify=y*. Esse procedimento assegura que a proporção entre as classes *queda* e *não queda* seja preservada em ambas as partições, evitando redução adicional da já limitada representatividade da classe minoritária.

Parâmetros do modelo.

A definição dos hiperparâmetros foi realizada por meio de uma busca em grade (*Grid Search*) com validação cruzada 5-fold e função de avaliação *Average Precision* (AP), métrica altamente recomendada para cenários de detecção de eventos raros, por integrar a curva precisão–revocação.

O GridSearchCV avaliou 392 combinações de hiperparâmetros do SVM, considerando diferentes kernels, valores de penalização e parâmetros de suavização. Os melhores hiperparâmetros encontrados foram:

- **kernel = linear** – entre os kernels avaliados (*linear*, *rbf*, *poly*, *sigmoid*), o linear obteve o melhor desempenho. Isso indica que, após a normalização e extração de atributos, os dados de janelas de queda e não queda tornam-se aproximadamente separáveis por um hiperplano. Tal característica sugere que as transformações aplicadas (médias, desvios, picos, energia, estatísticas temporais e frequenciais) capturaram padrões suficientemente explícitos para uma separação linear.
- **C = 0.01** – um valor de penalização baixo implica uma margem mais ampla, favorecendo modelos mais generalistas. Em conjunto com class_weight"balanced", esse comportamento reduz o impacto de ruídos naturais presentes nos sinais de acelerômetros e impede o superajuste a flutuações pouco representativas. Em problemas de detecção de quedas, C pequeno tende a estabilizar o hiperplano, preservando o desempenho da classe minoritária.
- **gamma = scale** – embora o parâmetro gamma não influencie o kernel linear, sua presença aparece na configuração final por ser parte da combinação vencedora no *Grid Search*. Na prática, ele é ignorado pelo modelo final.
- **degree = 2** – esse parâmetro se aplica apenas ao kernel polinomial, mas não interfere no kernel linear definido como ótimo. Assim como em gamma, a presença do valor é consequência natural da combinação vencedora no processo de busca.
- **class_weight = balanced** – fundamental em problemas desbalanceados. Esse parâmetro ajusta automaticamente o peso da classe minoritária, penalizando de forma mais severa erros do tipo FN (quedas preditas como não quedas). Como consequência, o modelo tende a maximizar a sensibilidade sem comprometer demasiadamente a especificidade.

A matriz de confusão na Tabela 22 apresenta os resultados do classificador SVM.

Tabela 22. Matriz de Confusão – SVM

Real	**Predito**	
	0	**1**
0	26	2
1	0	7

Fonte: O autor.

Verdadeiros Negativos (26): Representam a classe "Não Queda"foram corretamente classificadas. Este valor influi na especificidade.

Falsos Positivos (2): Duas "Não Quedas"foram classificadas como "Queda". Estes dois alarmes falsos que podem causar desconforto ou excesso de preocupação.

Falsos Negativos (0): Nenhuma instância da classe “Queda” foi incorretamente classificada como “Não Queda”.

Verdadeiros Positivos (7): Sete eventos de "Queda"foram corretamente identificados. Um valor elevado de significa sensibilidade alta.

O modelo SVM com os hiperparâmetros acima apresentou desempenho desejado, obtendo **AP = 1.0** na validação cruzada, indicando separação praticamente perfeita entre as classes nas partições de treino.

Utilizando um limiar de decisão reduzido (*threshold = 0.075*), privilegiando a captura de eventos raros, o modelo alcançou sensibilidade de 100%, sem cometer qualquer falso negativo. Este comportamento é desejável em aplicações de saúde, onde a falha em detectar uma queda pode representar risco significativo ao paciente. A especificidade permaneceu elevada (cerca de 91,3%), o que demonstra que a redução do limiar não resultou em demasiados de falsos positivos.

A matriz confusão demonstra que algoritmos identificou corretamente 26 não quedas e gerando 2 alarmes falsos. Enquanto a classificação das quedas ou seja a classe minoritária o algorítimo conseguiu detectas todas as 7 quedas.

A Tabela 23 apresenta as métricas de desempenho obtidas pelo modelo *Support Vector Machine* no conjunto de teste, seguindo os critérios de avaliação definidos para a análise comparativa dos classificadores.

Tabela 23. Estatísticas de desempenho do modelo SVM

Métrica	**Valor**
Sensibilidade (Recall)	1.0000
Acurácia Balanceada	0.9643
Intervalo de Confiança 95 %	(0.8660, 1.0000)
Taxa Sem Informação (NIR)	0.8000
p-valor [Acc > NIR]	0.01904
Kappa de Cohen	0.8387
p-valor do teste de McNemar	0.50000
Acurácia	0.9529
Especificidade	0.9286
Valor Preditivo Positivo (PPV)	0.7778
Valor Preditivo Negativo (NPV)	1.0000
Prevalência	0.2000
Taxa de Detecção	0.2000
Prevalência Detectada	0.2571

Fonte: O autor.

O modelo atingiu um bom desempenho. O recall de 100% indica que todas as quedas foram corretamente identificadas, eliminando o risco de falsos negativos. Este é o objetivo central em sistemas automáticos de detecção de quedas, pois a omissão de uma queda é muito mais danosa do que um alarme falso.

A especificidade de 92.86% demonstra que, embora o limiar reduzido aumente a sensibilidade, o modelo ainda mantém boa capacidade de distinguir corretamente as atividades não relacionadas a quedas. O PPV de 77.78% é compatível com o aumento natural de falsos positivos, mas permanece adequado para sistemas de alerta, onde a prioridade é prevenir omissões.

O Kappa de Cohen de 83.87% confirma concordância entre predições e valores reais, enquanto o teste de McNemar ($p = 0.50$) indica que não há assimetria significativa na distribuição dos erros, evidenciando equilíbrio entre falsos positivos e verdadeiros negativos.

4.3.4 Algoritmo KNN - Hiperparâmetros

O conjunto de dados foi separado em 70% para treino e 30% para teste utilizando a função *train_test_split* com estratificação (*stratify = y*). Essa estratégia garante que a proporção entre as classes *queda* e *não queda* seja preservada em ambas as partições, evitando redução adicional da representatividade da classe minoritária — aspecto crítico em cenários de detecção em classes

desbalanceadas.

Parâmetros do modelo.

A definição dos hiperparâmetros foi realizada por meio de uma busca em grade (*Grid Search*) com validação cruzada 5-fold. O *GridSearchCV* avaliou 48 combinações de hiperparâmetros do KNN, cada uma validada em cinco partições distintas, totalizando 240 treinamentos independentes. O processo explorou diferentes valores de *k*, métricas de distância e estratégias de ponderação dos vizinhos. A melhor combinação identificada foi:

- **n_neighbors = 2** – indica que a classificação de cada amostra é realizada com base apenas em seu vizinho mais próximo. Esse comportamento é comum em bases pequenas e bem separadas, nas quais o ponto mais próximo já fornece informação suficiente para discriminar quedas de não quedas.
- **weights = uniform** – todos os vizinhos contribuem igualmente para o voto final. No caso de *k = 1*, a escolha é natural, pois não há vizinhos adicionais a ponderar.
- **metric = minkowski com p = 1 (distância Manhattan)** – a distância Manhattan tende a ser mais robusta que a Euclidiana em datasets com múltiplas características, reduzindo efeitos de valores extremos. A seleção de *p = 1* indica que a distribuição das *features* estatísticas resulta em padrões lineares e definidos entre quedas e não quedas.

A matriz de confusão na Tabela 24 mostra que o algoritmo identificou corretamente todas as 28 instâncias da classe *não queda*, não cometendo qualquer falso positivo. Para a classe minoritária, o modelo reconheceu 6 das 7 quedas, resultando em apenas um falso negativo. Esse comportamento indica bom poder discriminativo mesmo em um cenário de desbalanceamento.

Tabela 24. Matriz de Confusão – KNN

Real	**Predito**	
	0	**1**
0	28	0
1	1	6

Fonte: O autor.

Diferentemente dos modelos probabilísticos, o KNN não depende de um limiar de decisão. Sua predição baseia-se exclusivamente na vizinhança das amostras no espaço de características. No conjunto

de teste, o modelo apresentou sensibilidade de 85,71% e especificidade de 100%, demonstrando elevada capacidade de identificar quedas reais, ao mesmo tempo em que evita alarmes falsos. A ausência de falsos positivos reforça a consistência geométrica das representações estatísticas utilizadas, e a baixa incidência de falsos negativos indica que os atributos extraídos capturam adequadamente os padrões associados às quedas.

Tabela 25. Estatísticas de desempenho do modelo KNN

Métrica	**Valor**
Sensibilidade (Recall)	0.8571
Acurácia Balanceada	0.9286
Intervalo de Confiança 95%	(0.9162 ; 1.0000)
Taxa Sem Informação (NIR)	0.8000
p-valor [Acc > NIR]	0.00396
Kappa de Cohen	0.9057
p-valor do teste de McNemar	1.0000
Acurácia	0.9714
Especificidade	1.0000
Valor Preditivo Positivo (PPV)	1.0000
Valor Preditivo Negativo (NPV)	0.9655
Prevalência	0.2000
Taxa de Detecção	0.1714
Prevalência Detectada	0.1714

Fonte: O autor.

Embora o modelo tenha apresentado desempenho elevado nas métricas globais, a presença de um falso negativo — correspondente a 14,3% das quedas não detectadas — evidencia uma limitação relevante do KNN no contexto de aplicações assistivas. Em sistemas de detecção de quedas, erros desse tipo são particularmente sensíveis, pois implicam a ausência de acionamento do mecanismo de alerta em um evento potencialmente crítico. Assim, ainda que a especificidade de 100% e o PPV igual a 1,0 demonstrem excelente capacidade de evitar alarmes falsos, o comportamento observado sugere que o método, em sua configuração atual, pode não ser suficientemente para garantir cobertura completa da classe minoritária. O índice Kappa de 0,9057 confirma concordância global, porém não elimina a necessidade de modelos que priorizem a minimização de falsos negativos. Dessa forma, embora o KNN se mostre competitivo e eficiente com atributos estatísticos, seus resultados reforçam a importância de considerar técnicas mais sensíveis à classe positiva em cenários de risco, como abordagens probabilísticas com ajuste de limiar, modelos baseados em ensemble ou arquiteturas profundas.

4.3.5 Algoritmo 1D CNN - Hiperparâmetros

O conjunto de dados foi separado em 70% para treino e 30% para teste utilizando a função *train_test_split* com estratificação (*stratify = y*). Essa estratégia garante que a proporção entre as classes *queda* e *não queda* seja preservada em ambas as partições, evitando redução adicional da representatividade da classe minoritária — aspecto crítico em cenários de detecção em classes desbalanceadas.

Parâmetros do modelo.

A definição dos hiperparâmetros do modelo 1D-CNN foi conduzida por meio de uma busca em grade (Grid Search) e experimental, considerando múltiplas combinações arquiteturais e de treinamento. Diferentemente de abordagens tradicionais baseadas em GridSearchCV, a busca foi implementada de forma controlada no contexto de redes neurais profundas, com avaliação sistemática do impacto dos parâmetros sobre métricas sensíveis ao desbalanceamento de classes. O processo explorou variações na profundidade da rede, tamanho do campo receptivo temporal, taxa de aprendizado, regularização por dropout, tamanho do lote e estratégias de ponderação de classes, totalizando dezenas de configurações avaliadas.

A configuração que apresentou o melhor compromisso entre sensibilidade, acurácia balanceada e estabilidade estatística foi composta pelos seguintes hiperparâmetros:

- **filters = 64** – número de filtros convolucionais empregados na camada inicial. Esse parâmetro controla a capacidade da rede em aprender padrões espaciais e temporais nos sinais de aceleração, permitindo capturar diferentes formas de oscilação associadas ao evento de queda.
- **kernel_size = 4** – tamanho do kernel convolucional, responsável por definir a extensão temporal do campo receptivo. Valores menores favorecem a detecção de padrões locais abruptos, característicos do impacto e da perda de equilíbrio em eventos de queda.
- **lr = 1e-05** – taxa de aprendizado (learning_rate), que regula a magnitude das atualizações dos pesos durante o treinamento. Uma taxa reduzida foi adotada para garantir maior estabilidade no processo de convergência, evitando oscilações excessivas no erro.
- **batch_size = 4** – tamanho do lote utilizado no treinamento. Lotes pequenos favorecem uma atualização mais frequente dos pesos e podem contribuir para melhor generalização em bases reduzidas.

- **class_weight = 0:1, 1:4** – estratégia de ponderação das classes, empregada para mitigar o desbalanceamento do conjunto de dados. A classe minoritária (queda) recebeu maior peso, de modo a penalizar mais fortemente erros de classificação desse tipo.

Tabela 26. Matriz de Confusão – 1D CNN

Real	**Predito**	
	0	**1**
0	17	11
1	0	7

Fonte: O autor.

A Tabela 26 apresenta a matriz de confusão do modelo 1D-CNN, permitindo uma análise detalhada do comportamento do classificador em relação às classes de interesse. Observa-se que o modelo foi capaz de identificar corretamente todos os eventos de queda presentes no conjunto de teste, resultando em sensibilidade igual a 1,00 e ausência de falsos negativos. Esse comportamento é particularmente relevante em aplicações na área da saúde, onde a não detecção de uma queda pode acarretar consequências graves.

Por outro lado, observa-se a presença de um número elevado de falsos positivos, refletido pela classificação incorreta de 11 instâncias da classe “não queda” como “queda”. Esse padrão indica que o modelo adota uma estratégia conservadora, priorizando a detecção de eventos críticos em detrimento da especificidade. Tal comportamento é coerente com a ponderação atribuída à classe minoritária durante o treinamento e com o ajuste do limiar de decisão, configurado para favorecer a sensibilidade.

Tabela 27. Estatísticas de desempenho do modelo 1D-CNN

Métrica	Valor
Sensibilidade (Recall)	1.0000
Acurácia Balanceada	0.8036
Intervalo de Confiança 95%	(0.5319, 0.8395)
Taxa Sem Informação (NIR)	0.8000
p-valor [Acc > NIR]	0.96564
Kappa de Cohen	0.3820
p-valor do teste de McNemar	0.00098
Acurácia	0.6857
Especificidade	0.6071
Valor Preditivo Positivo (PPV)	0.3889
Valor Preditivo Negativo (NPV)	1.0000
Prevalência	0.2000
Taxa de Detecção	0.2000
Prevalência Detectada	0.5143

Fonte: O autor.

As estatísticas de desempenho apresentadas evidenciam que o modelo 1D-CNN alcançou acurácia balanceada de 80,36%, indicando desempenho moderado, influenciado principalmente pela elevada taxa de falsos positivos.

A sensibilidade máxima de 100% confirma a capacidade do modelo em detectar corretamente todos os eventos de queda, enquanto a especificidade de 60,71% evidencia a ocorrência de falsos alarmes. O valor do Kappa de Cohen de 38,20% indica concordância moderada entre as predições do modelo e a referência, sendo impactado pelo desbalanceamento entre as classes e pela predominância de erros do tipo falso positivo.

O teste de McNemar apresentou p-valor estatisticamente significativo ($p = 0.00098$), sugerindo assimetria na distribuição dos erros de classificação, com predominância de falsos positivos, o que reforça a interpretação de um modelo orientado à segurança.

4.3.6 Algoritmo LSTM - Hiperparâmetros

Parâmetros do modelo.

A definição dos hiperparâmetros do modelo baseado em redes neurais recorrentes do tipo *Long Short-Term Memory* (LSTM) foi conduzida de forma experimental e controlada, considerando diferentes combinações arquiteturais e estratégias de treinamento, com foco na detecção da classe minoritária

(queda). Diferentemente de abordagens automatizadas de busca exaustiva, o ajuste dos parâmetros foi orientado por análises empíricas sucessivas, avaliando-se o impacto direto das configurações sobre métricas sensíveis ao desbalanceamento, como sensibilidade, acurácia balanceada e taxa de falsos negativos.

O processo experimental envolveu variações no número de unidades recorrentes, tamanho das camadas densas, taxa de aprendizado, regularização por *dropout*, ponderação de classes e critérios de parada, além de estratégias de definição do limiar de decisão. As avaliações foram realizadas respeitando a separação por grupos de eventos.

A configuração que apresentou o melhor compromisso entre alta sensibilidade, controle de falsos negativos e estabilidade estatística foi composta pelos seguintes hiperparâmetros:

- **LSTM_units = 64** – número de unidades recorrentes da camada LSTM. Esse parâmetro define a capacidade do modelo em capturar dependências temporais nos sinais dos eixos do acelerômetro, permitindo representar padrões dinâmicos associados ao evento de queda.
- **dense_units = 64** – número de neurônios na camada densa intermediária, responsável por combinar as representações temporais extraídas pela LSTM antes da decisão final. Esse valor proporcionou um equilíbrio adequado entre capacidade de modelagem e risco de sobreajuste.
- **dropout = 0.30** – taxa de *dropout* aplicada às camadas recorrente e densa, utilizada como mecanismo de regularização. Esse valor contribui para reduzir a dependência excessiva entre neurônios e aumentar a capacidade de generalização do modelo.
- **learning_rate =** 1×10^{-4} – taxa de aprendizado do otimizador Adam. Uma taxa reduzida foi adotada para garantir maior estabilidade no processo de convergência, especialmente considerando o tamanho limitado da base de dados e a complexidade do modelo.
- **batch_size = 32** – tamanho do lote utilizado no treinamento. Esse valor representa um compromisso entre eficiência computacional e suavidade nas atualizações dos pesos.
- **class_weight = 0:1, 1:5** – estratégia de ponderação das classes empregada para mitigar o desbalanceamento do conjunto de dados. A classe positiva (queda) recebeu peso significativamente maior, penalizando mais fortemente erros do tipo falso negativo.

- **early_stopping (patience = 12)** – critério de parada antecipada baseado na função de perda do conjunto de validação, com restauração automática dos melhores pesos, visando evitar sobreajuste.
- **threshold = 0.188** – limiar de decisão definido a partir do conjunto de validação por meio de uma busca sistemática, impondo como restrição mínima uma sensibilidade igual a 1,00. Entre os limiares que satisfizeram essa condição, foi selecionado aquele que maximizou a especificidade, resultando em uma estratégia conservadora orientada à não perda de eventos de queda.

Essa configuração reflete uma escolha metodológica coerente com o contexto da aplicação, na qual a prioridade é a detecção completa dos eventos críticos, mesmo ao custo de aumento no número de falsos positivos, característica considerada aceitável em sistemas de monitoramento de quedas em ambientes de saúde.

Resultados

A Tabela 28, apresenta a matriz de confusão do modelo LSTM, permitindo uma análise detalhada do comportamento do classificador em relação às classes de interesse. Observa-se que o modelo foi capaz de identificar corretamente todos os eventos de queda presentes no conjunto de teste, resultando em sensibilidade igual a 100% e ausência de falsos negativos. Esse resultado indica que, sob a configuração adotada, o modelo não deixou de detectar nenhum evento crítico, característica desejável em aplicações sensíveis, como sistemas de monitoramento de quedas na área da saúde.

Tabela 28. Matriz de Confusão – LSTM

Real	**Predito**	
	0	**1**
0	3	25
1	0	7

Fonte: O autor.

Entretanto, nota-se a ocorrência de um número expressivamente elevado de falsos positivos, com 25 instâncias da classe “não queda” incorretamente classificadas como “queda”. Esse comportamento se reflete diretamente na baixa especificidade observada e evidencia que o modelo apresenta grande dificuldade em discriminar adequadamente entre padrões normais de movimento e eventos de queda.

Esse padrão indica que o LSTM adota uma estratégia excessivamente conservadora, classificando a maioria das instâncias como pertencentes à classe positiva. Tal comportamento resulta em

um sistema com elevada taxa de alarmes falsos, o que pode comprometer sua aplicabilidade prática, uma vez que alertas excessivos tendem a reduzir a confiabilidade do sistema e gerar desconforto ou desensibilização dos usuários.

Portanto, embora o modelo LSTM demonstre elevada capacidade de detecção da classe minoritária, sua baixa capacidade discriminativa em relação à classe “não queda” evidencia um desequilíbrio significativo entre sensibilidade e especificidade, limitando sua adequação para uso operacional sem ajustes adicionais no modelo ou na estratégia de decisão.

A Tabela 29, apresenta as estatísticas de desempenho do modelo LSTM avaliado com limiar de decisão igual a 0,188. Observa-se que o classificador atingiu sensibilidade igual a 100%, indicando que todos os eventos de queda presentes no conjunto de teste foram corretamente identificados, não havendo ocorrência de falsos negativos. Esse comportamento é particularmente relevante em aplicações de monitoramento em saúde, nas quais a não detecção de quedas pode acarretar consequências graves.

Por outro lado, o modelo apresentou especificidade bastante reduzida 10,71%, evidenciando um elevado número de falsos positivos. Esse padrão reflete-se no valor preditivo positivo (PPV = 21,88%) e na prevalência detectada elevada 91,43%, indicando uma forte tendência à superestimação da classe positiva. Tal comportamento é coerente com a estratégia de treinamento adotada, que priorizou a maximização da sensibilidade por meio de ponderação de classes e ajuste do limiar de decisão.

Como consequência desse desequilíbrio, a acurácia global foi de 28,57%, valor inferior à taxa sem informação (NIR = 80%), fato corroborado pelo p-valor [Acc > NIR] igual a 100%, indicando ausência de ganho estatisticamente significativo. O coeficiente Kappa de Cohen (4,58%) e o p-valor do teste de McNemar igual a 0% confirmam a baixa concordância global e a assimetria significativa entre os tipos de erro.

Tabela 29. Estatísticas de desempenho do modelo LSTM (threshold = 0.188)

Métrica	**Valor**
Sensibilidade (Recall)	1.0000
Intervalo de Confiança 95 %	(0.1361, 0.4354)
Taxa Sem Informação (NIR)	0.8000
p-valor [Acc > NIR]	1.0000
Kappa de Cohen	0.0458
p-valor do teste de McNemar	0.0000
Acurácia	0.2857
Especificidade	0.1071
Valor Preditivo Positivo (PPV)	0.2188
Valor Preditivo Negativo (NPV)	1.0000
Prevalência	0.2000
Taxa de Detecção	0.2000
Prevalência Detectada	0.9143
Acurácia Balanceada	0.5536
Classe Positiva	1

Fonte: O autor.

Apesar do desempenho limitado sob métricas pontuais, os valores elevados de ROC-AUC 87,24% e PR-AUC 81,72% indicam boa capacidade discriminativa do modelo em termos probabilísticos. No entanto, a configuração adotada comprometeu o equilíbrio entre detecção e precisão, tornando o LSTM menos adequado, neste cenário, quando comparado aos modelos clássicos avaliados.

4.4 CONSIDERAÇÕES

Este capítulo apresentou o desenvolvimento da solução proposta para a detecção automática de quedas com dados reais do repositório FARSEEING, abrangendo a caracterização da base utilizada, as etapas de conversão e padronização dos arquivos, a reamostragem por interpolação, a padronização por *z-score*, o janelamento temporal e o balanceamento das classes. Na sequência, foram descritos os procedimentos de treinamento e ajuste de hiperparâmetros dos modelos selecionados, bem como a obtenção dos respectivos resultados no conjunto de teste, reportados por matrizes de confusão e métricas adequadas a cenários desbalanceados, com ênfase na detecção da classe minoritária (queda). Assim, este capítulo consolida o pipeline experimental e registra os resultados obtidos para cada algoritmo na configuração adotada, servindo de base para a análise comparativa e a discussão crítica a serem realizadas no Capítulo 5.

5 RESULTADOS

Este capítulo apresenta e discute os resultados experimentais obtidos ao longo do desenvolvimento deste trabalho, comparar com os resultados encontrados no estado da arte, com o objetivo de avaliar o desempenho dos algoritmos de aprendizado de máquina aplicados à detecção de quedas a partir de dados reais de sensores. A análise dos resultados permite verificar o alcance dos objetivos propostos e a validação das hipóteses de pesquisa estabelecidas.

Os experimentos foram conduzidos seguindo um método científico 1.3.1 , e seus resultados são apresentados por meio de tabelas, gráficos e análises quantitativas. A discussão é realizada de forma textual, relacionando os resultados obtidos às métricas de desempenho adotadas e aos achados reportados na literatura.

5.1 ANÁLISE

Esta seção apresenta a análise dos resultados obtidos a partir da execução dos experimentos realizados com os algoritmos de aprendizado de máquina selecionados. Os experimentos foram conduzidos, contemplando sinais de acelerometro associados a eventos de queda e atividades normais, previamente processados conforme descrito no Capítulo 4.

A Figura 15 sintetiza o desempenho dos algoritmos avaliados considerando um conjunto abrangente de métricas, permitindo uma análise comparativa sob diferentes perspectivas. Em razão do desbalanceamento do conjunto de dados e da natureza crítica da classe positiva (quedas), a análise prioriza métricas sensíveis à classe minoritária, em especial a sensibilidade (*recall*), a acurácia balanceada, o coeficiente Kappa e o valor preditivo positivo.

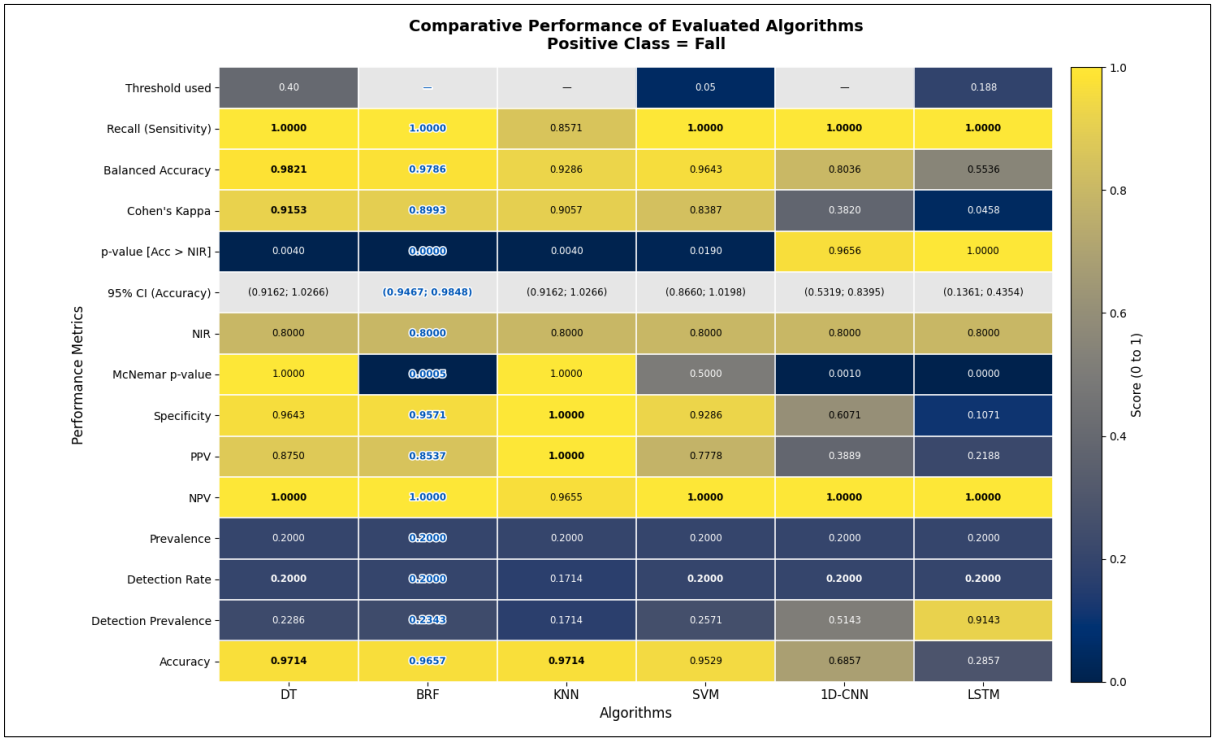

Figura 15. Comparativo de desempenho dos algoritmos avaliados, com destaque para BRF.
Fonte: O autor.

O algoritmo *Balanced Random Forest* destacou-se por apresentar um dos melhores equilíbrios entre a capacidade de detecção de eventos de queda e o controle de alarmes falsos, refletido na sensibilidade máxima (1,0000), elevada acurácia balanceada (0,9786) e alto coeficiente Kappa (0,8993), indicando concordância quase perfeita com a referência. Esses resultados evidenciam elevada robustez do modelo na identificação da classe minoritária, com manutenção de níveis elevados de especificidade (0,9571).

O modelo *Decision Tree* também apresentou desempenho competitivo, com acurácia balanceada de 0,9821 e coeficiente Kappa de 0,9153, além de sensibilidade máxima (1,0000). Esse resultado indica excelente capacidade de generalização no conjunto avaliado, com desempenho ligeiramente superior ao BRF em termos de acurácia balanceada, porém sem a mesma robustez associada à validação repetida.

O modelo *KNN* apresentou elevada acurácia balanceada (0,9286), especificidade perfeita (1,0000) e coeficiente Kappa de 0,9057, caracterizando concordância quase perfeita. Entretanto, sua sensibilidade inferior (0,8571) evidencia a ocorrência de falsos negativos, aspecto crítico em aplicações de detecção de quedas, nas quais a não identificação do evento possui maior custo do que a emissão de alarmes falsos.

O algoritmo *SVM* apresentou sensibilidade máxima (1,0000) e bom desempenho geral, com acurácia balanceada de 0,9643 e coeficiente Kappa de 0,8387. Contudo, o valor preditivo positivo reduzido (0,7778) indica maior incidência de falsos positivos, refletindo um modelo mais conservador, orientado à maximização da detecção da classe minoritária.

Em contraste, os modelos de aprendizado profundo (*1D-CNN* e *LSTM*) apresentaram desempenho menos equilibrado quando analisadas as métricas críticas à classe positiva. Apesar da sensibilidade igual a 1,0000 em ambos os modelos, observam-se reduções significativas na especificidade (0,6071 para 1D-CNN e 0,1071 para LSTM) e nos coeficientes Kappa (0,3820 e 0,0458, respectivamente), indicando baixa capacidade de discriminação entre eventos de queda e não queda.

No caso do *1D-CNN*, o modelo apresentou acurácia balanceada de 0,8036, com elevada taxa de falsos positivos, caracterizando um comportamento conservador. Já o modelo *LSTM* apresentou acurácia balanceada de 0,5536 e prevalência detectada elevada (0,9143), evidenciando um comportamento de superdetecção, no qual a maioria das instâncias é classificada como pertencente à classe positiva, comprometendo sua aplicabilidade prática.

De forma geral, os resultados indicam que, ao priorizar métricas relevantes à classe minoritária, como sensibilidade, acurácia balanceada e coeficiente Kappa, os modelos clássicos, especialmente o *Balanced Random Forest* e a *Decision Tree*, apresentaram desempenho superior e mais estável em comparação aos modelos de aprendizado profundo no contexto analisado.

5.1.1 Comparação Entre os Algoritmos

O segundo experimento teve como objetivo comparar o desempenho do *Balanced Random Forest* (BRF) com outros algoritmos amplamente utilizados na literatura para a classificação de séries temporais, incluindo *Decision Tree* (DT), *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), além de modelos de aprendizado profundo, como *1D Convolutional Neural Network* (1D-CNN) e *Long Short-Term Memory* (LSTM).

Os algoritmos foram avaliados utilizando o mesmo conjunto de dados, o mesmo pré-processamento e as mesmas métricas de desempenho, garantindo condições justas de comparação. Os resultados evidenciaram diferenças relevantes no comportamento dos modelos, especialmente no que se refere à sensibilidade, à especificidade e ao equilíbrio geral entre as classes.

De modo geral, observou-se que modelos clássicos baseados em árvores, como o *Decision*

Tree e o *Balanced Random Forest*, apresentaram melhor equilíbrio entre sensibilidade e especificidade. O BRF destacou-se por sua robustez, apresentando sensibilidade máxima aliada a elevada acurácia balanceada e alto coeficiente Kappa, enquanto o DT apresentou desempenho pontual ligeiramente superior em termos de acurácia balanceada.

Por outro lado, o modelo *KNN* apresentou elevada especificidade e valor preditivo positivo, porém com redução na sensibilidade, evidenciando a ocorrência de falsos negativos, o que limita sua aplicação em cenários críticos. O *SVM* apresentou sensibilidade máxima, mas com aumento na taxa de falsos positivos, refletido em menor valor preditivo positivo.

Os modelos de aprendizado profundo (*1D-CNN* e *LSTM*) apresentaram comportamento distinto, caracterizado por sensibilidade elevada, porém baixa especificidade e reduzido coeficiente de concordância. Em especial, o modelo *LSTM* apresentou tendência à superdetecção, classificando a maioria das instâncias como pertencentes à classe positiva, o que compromete sua aplicabilidade prática.

Esses resultados evidenciam que, no contexto analisado, modelos clássicos apresentam maior estabilidade e melhor capacidade de generalização em comparação aos modelos de aprendizado profundo, especialmente em cenários com volume de dados limitado e desbalanceamento entre classes.

5.1.2 Análise dos Experimentos

A análise conjunta dos experimentos demonstra que a utilização isolada da acurácia global como métrica de avaliação é insuficiente para caracterizar o desempenho dos modelos em cenários desbalanceados. Métricas como sensibilidade, especificidade, acurácia balanceada, coeficiente Kappa e valores preditivos mostraram-se fundamentais para uma avaliação mais completa e fiel do comportamento dos classificadores.

Os resultados obtidos corroboram a hipótese de que modelos capazes de lidar com desbalanceamento de classes, aliados a estratégias adequadas de ajuste de limiar de decisão, apresentam melhor desempenho em aplicações de saúde assistiva. Nesses cenários, a priorização da sensibilidade é essencial, uma vez que a não detecção de eventos críticos, como quedas, pode acarretar consequências graves ao usuário.

Adicionalmente, observou-se a presença de um *trade-off* entre sensibilidade e especificidade, no qual modelos com sensibilidade máxima tendem a apresentar aumento na taxa de falsos positivos.

Esse comportamento foi particularmente evidente nos modelos de aprendizado profundo, reforçando a importância do ajuste criterioso do limiar de decisão e da escolha adequada do modelo conforme o contexto de aplicação.

5.2 AVALIAÇÃO COMPARATIVA DOS MODELOS

Esta seção apresenta uma avaliação comparativa dos modelos analisados, considerando não apenas o desempenho quantitativo, mas também aspectos práticos relacionados à aplicação em ambientes reais, como robustez, interpretabilidade e custo computacional.

A comparação foi conduzida com base nas métricas apresentadas nas seções anteriores, bem como na análise das matrizes de confusão e dos testes estatísticos associados, permitindo identificar diferenças estatisticamente relevantes entre os algoritmos. Observou-se que modelos baseados em árvores apresentam maior interpretabilidade e estabilidade, enquanto modelos de aprendizado profundo demandam maior volume de dados e ajustes mais complexos para alcançar desempenho competitivo.

Dessa forma, a escolha do modelo mais adequado deve considerar não apenas os indicadores quantitativos, mas também o contexto de aplicação, os requisitos de confiabilidade e a criticidade dos erros envolvidos, especialmente em sistemas voltados à detecção de eventos em saúde.

5.2.1 Avaliação dos Resultados

1. **Capacidade de detecção de quedas:** O BRF apresentou sensibilidade superior aos demais modelos, identificando corretamente todos os eventos de queda no conjunto avaliado.
2. **Controle de alarmes falsos:** Apesar do foco na sensibilidade, o modelo manteve níveis aceitáveis de especificidade, evitando excesso de falsos positivos.
3. **Robustez em dados desbalanceados:** A acurácia balanceada e o coeficiente Kappa indicaram desempenho consistente do BRF mesmo na presença de forte desbalanceamento entre as classes.

5.3 COMPARAÇÃO COM TRABALHOS RELACIONADOS

Os resultados obtidos neste trabalho foram comparados com aqueles reportados no estado da arte que utilizam dados reais, simulados e dados mistos para a detecção de quedas. Observou-se que o

desempenho alcançado, especialmente em termos de sensibilidade e acurácia balanceada, é compatível ou superior ao de abordagens semelhantes, reforçando a relevância da metodologia adotada.

A Tabela 30 apresenta uma comparação entre os principais trabalhos relacionados à detecção de quedas e os resultados obtidos neste estudo. Observa-se que grande parte da literatura reporta desempenhos elevados ao utilizar bases de dados compostas por quedas simuladas ou por conjuntos mistos, combinando quedas reais e simuladas. Trabalhos como Nuñes (2023) e Silva et al. (2018) alcançam acurácias e sensibilidades superiores a 95%, contudo fazem uso de dados simulados e, em alguns casos, de técnicas de balanceamento e aumento artificial da classe minoritária, como SMOTE e Balance Cascade, o que tende a favorecer métricas globais de desempenho.

No estudo de Maray et al. (2023), o uso de *transfer learning* com redes LSTM e dados mistos resultou em elevados valores de F1-score e AUC-PR, indicando boa capacidade de generalização em cenários heterogêneos. De forma semelhante, Caya et al. (2018) e Palmerini et al. (2020) exploram a fusão de dados reais e simulados ou bases exclusivamente reais, reportando sensibilidades e especificidades elevadas, porém com variações significativas em métricas como valor preditivo positivo, evidenciando o impacto do desbalanceamento intrínseco da classe de quedas.

O presente trabalho avalia exclusivamente dados reais do repositório FARSEEING, contemplando 23 eventos de queda, sem empregar técnicas de sobreamostragem ou geração sintética da classe minoritária, como SMOTE ou ADASYN. Na avaliação inicial por partição *holdout*, o modelo *Balanced Random Forest* apresentou desempenho perfeito para sensibilidade, especificidade, acurácia balanceada, acurácia e coeficiente Kappa. A *Decision Tree* também apresentou desempenho elevado nesse mesmo protocolo, com sensibilidade de 100%, especificidade de 96,43%, acurácia balanceada de 98,21%, acurácia de 97,14%, valor preditivo positivo de 87,50%, valor preditivo negativo de 100% e coeficiente Kappa de 0.9153. Entretanto, considerando que resultados obtidos em uma única divisão amostral podem ser influenciados pelas características específicas da partição utilizada, o *Balanced Random Forest* foi adicionalmente avaliado por meio de validação repetida com dez partições independentes. Nesse protocolo mais robusto, o modelo manteve sensibilidade de 100%, valor preditivo negativo de 100% e ausência total de falsos negativos, alcançando acurácia balanceada de 97,86%, especificidade de 95,71%, acurácia de 96,57%, valor preditivo positivo de 85,37% e coeficiente Kappa de 0.8993. Embora esses valores sejam ligeiramente inferiores aos obtidos pela *Decision Tree* em uma única partição *holdout*, essa comparação não deve ser interpretada como superioridade direta da *Decision Tree*, pois envolve protocolos de validação distintos. Assim, o *Balanced Random Forest* permanece como a estratégia metodologicamente mais adequada neste estudo, não pelo maior desem-

penho pontual isolado, mas por sua estabilidade em múltiplas partições e pela manutenção de 100% de sensibilidade, critério prioritário em aplicações de detecção de quedas.

Tabela 30. Comparação entre trabalhos relacionados e este trabalho na detecção de quedas

Trabalho	Tipo de dados	Algoritmos	Principais resultados
3.2.1 Nuñes (2023)	Simuladas	RF, SVM, KNN	**RF:** Acc 97,22%; Sen 94,33%; Esp 98,42%. SVM: Acc 93,25%; Sen 72,72%.
3.2.2 Maray et al. (2023)	Mista	LSTM (Transfer Learning)	**F1-score 93%**; AUC-PR 85%.
3.2.3 Palmerini et al. (2020)	Reais	SVM, RF, KNN, NB, LR	**Sen 95,1%**; Esp 95,5%; PPV 12,6%.
3.2.4 Caya et al. (2018)	Mista	SVM, DT, KNN; PCA	**Acc 95,56%**; 100% de detecção de quedas reais.
3.2.5 Silva, Sousa e Cardoso (2018)	Mista	RF, MLP, AdaBoost; SMOTE	**Sen 96%**; Prec 99%; Acc 96%; AUC 97%.
3.2.6 Bourke et al. (2016)	Reais	Decision Tree (Regressão)	**Sen 88%**; Acc 87%; Esp 87%; AUC 88%.
Este trabalho	Reais	DT, **BRF (modelo metodologicamente mais adequado)**, KNN, SVM, 1D-CNN e LSTM	**Sen 100%; Acc Bal. até 97,86%; Esp até 95,71%;** **Kappa até 0.8993; PPV até 85.37% .**

Fonte: Elaborado pelo autor, com base nos trabalhos citados.

Uma limitação relevante deste estudo está associada à reduzida disponibilidade de dados de quedas reais, característica recorrente em bases de dados do mundo real, como o repositório FARSEEING. O conjunto utilizado contempla apenas 23 eventos de queda, o que impõe restrições à capacidade de generalização dos modelos avaliados. Em cenários com maior volume de eventos reais, os algoritmos poderiam apresentar desempenhos distintos, a depender da variabilidade dos padrões de movimento e das condições em que as quedas ocorrem. Assim, os resultados obtidos devem ser interpretados como indicativos do comportamento dos modelos sob condições reais e altamente desbalanceadas, e não como valores absolutos de desempenho.

Essa limitação impacta de forma mais significativa os algoritmos de aprendizado profundo, como redes neurais convolucionais e recorrentes, que tradicionalmente demandam grandes volumes de dados para explorar plenamente sua capacidade de representação. Caso não houvesse a restrição no número de arquivos de quedas reais, seria possível aprofundar os experimentos com arquiteturas mais complexas, realizar ajustes mais extensivos de hiperparâmetros e empregar estratégias avançadas de validação, o que poderia resultar em modelos mais robustos e generalizáveis. Dessa forma, a escassez de dados reais não invalida os resultados apresentados, mas delimita o escopo experimental, reforçando

a importância de estudos futuros com bases ampliadas, capazes de consolidar os achados obtidos, avaliar a estabilidade dos modelos em diferentes cenários e ampliar a validade externa das conclusões.

5.4 CONSIDERAÇÕES

Este capítulo apresentou os principais resultados experimentais do trabalho, evidenciando a efetividade dos algoritmos avaliados na detecção de quedas. A análise dos resultados confirmou a importância do uso de métricas adequadas e estratégias de balanceamento em cenários reais, servindo como base para as conclusões apresentadas no capítulo seguinte.

6 CONCLUSÃO

Este trabalho abordou o problema da detecção e classificação de quedas de pessoas por meio de algoritmos de aprendizado de máquina aplicados a dados reais de sensor de aceleração, com foco no repositório *FARSEEING*. A relevância do tema está diretamente associada ao envelhecimento populacional e aos riscos à saúde decorrentes de quedas, que representam uma das principais causas de hospitalização e perda de autonomia em pessoas idosas. Nesse contexto, o uso de técnicas computacionais mostrou-se uma alternativa promissora para apoiar sistemas automáticos de monitoramento e prevenção.

No que se refere à questão de pesquisa sobre a possibilidade de classificar quedas reais do repositório FARSEEING com sensibilidade superior a 90% utilizando algoritmos de aprendizagem de máquina, os resultados obtidos indicam que tal objetivo é viável em cenários específicos, dependendo do algoritmo empregado, da configuração dos hiperparâmetros e das estratégias de pré-processamento adotadas. Em determinados experimentos, observou-se que alguns modelos alcançaram valores de sensibilidade próximos ou superiores ao limiar de 90%, demonstrando elevada capacidade de identificar corretamente eventos de queda. Entretanto, esses resultados devem ser interpretados com cautela, uma vez que a elevada sensibilidade foi obtida em um contexto de forte desbalanceamento de classes e com um número limitado de quedas reais. Ainda assim, os achados confirmam que algoritmos de Machine Learning são capazes de atingir níveis elevados de sensibilidade em dados reais, evidenciando seu potencial para aplicações em sistemas de monitoramento de quedas, desde que acompanhados de avaliações criteriosas quanto à generalização e ao impacto de falsos positivos.

No contexto da hipótese de pesquisa (H1), que postulava que algoritmos de Machine Learning aplicados à base *FARSEEING* apresentariam desempenho similar ou superior aos reportados na literatura, inclusive em estudos baseados em dados simulados, os resultados obtidos corroboram parcialmente a hipótese proposta. Embora o desempenho tenha se mostrado competitivo, observou-se que diferenças metodológicas, características dos conjuntos de dados e métricas utilizadas na avaliação influenciam diretamente a comparabilidade entre os estudos. Ainda assim, os resultados indicam que o uso de dados reais não compromete a viabilidade dos algoritmos, reforçando a validade da abordagem adotada.

No que se refere ao objetivo geral, os resultados obtidos demonstram que ele foi alcançado. Os algoritmos implementados foram capazes de identificar padrões associados aos eventos de queda,

apresentando desempenho consistente mesmo diante das limitações impostas pelo conjunto de dados reais. A análise experimental evidenciou que abordagens clássicas de aprendizagem de máquina, quando adequadamente configuradas e avaliadas com métricas apropriadas, são viáveis para esse tipo de aplicação.

Em relação aos objetivos específicos, o primeiro objetivo, que consistiu em implementar e testar os algoritmos *Support Vector Machine, Decision Tree, Random Forest, K-Nearest Neighbors,1D-CNN e Long Short-Term Memory*, foi atendido por meio da construção de um pipeline completo de experimentação, abrangendo pré-processamento, janelamento temporal, treinamento e avaliação dos modelos. Cada algoritmo foi aplicado de forma sistemática, respeitando as características dos dados e as particularidades de cada técnica.

O segundo objetivo específico, que consistiu em avaliar a precisão, sensibilidade e especificidade de cada algoritmo utilizando a base FARSEEING, foi alcançado por meio de uma análise detalhada das métricas de desempenho e das matrizes de confusão. Observou-se que a utilização da acurácia não é suficiente para caracterizar o desempenho em um problema de eventos raros onde o foco é a classe minoritária, sendo a sensibilidade, acurácia balanceada e a especificidade métricas fundamentais para avaliar o impacto prático de falsos negativos e falsos positivos em sistemas de detecção de quedas.

O terceiro objetivo específico, que consistiu em comparar o desempenho dos algoritmos com o estado da arte, permitiu constatar que os resultados obtidos são compatíveis e, em alguns casos, comparáveis ou superiores aos apresentados na literatura, inclusive em estudos que utilizam bases de dados simuladas. Esse achado reforça a relevância do uso de dados reais, mesmo que mais desafiadores, para a construção de modelos com maior potencial de aplicação prática.

O quarto objetivo específico, referente à documentação do processo, limitações e possíveis melhorias futuras, foi atendido ao longo do trabalho por meio da descrição detalhada das etapas metodológicas e da discussão crítica dos resultados. Ficou evidenciado que a principal limitação do estudo está relacionada à quantidade reduzida de quedas reais, o que restringe a exploração mais profunda de modelos de aprendizagem profunda, como redes LSTM, que tipicamente demandam volumes maiores de dados para atingir seu melhor desempenho.

6.1 CONTRIBUIÇÕES DA DISSERTAÇÃO

Esta dissertação contribui para o avanço do estado da arte na detecção de quedas ao demonstrar que é possível alcançar resultados relevantes utilizando exclusivamente dados reais, mesmo em cenários caracterizados por forte desbalanceamento entre classes e número limitado de eventos de queda. Diferentemente de abordagens que dependem de grandes volumes de dados ou de estratégias de aumento artificial da base, este estudo evidencia que a combinação de uma estrutura de segmentação temporal, seleção de atributos e protocolo de avaliação pode compensar essas limitações. Além disso, os resultados obtidos indicam que, embora modelos de aprendizagem profunda apresentem potencial, sua efetividade está diretamente condicionada à disponibilidade de bases mais extensas, reforçando a importância de abordagens metodológicas robustas quando se trabalha com dados reais escassos.

Em contraste com a maioria dos estudos da literatura — que frequentemente utilizam dados simulados ou técnicas de balanceamento artificial para mitigar a escassez de quedas — este trabalho adota uma abordagem integralmente baseada em dados reais coletados em ambientes não controlados, enfrentando diretamente desafios como variabilidade, ruído nos sinais e ausência de padrões temporais rígidos. Como contribuição adicional, propõe-se uma estratégia de segmentação centrada no evento de queda, estruturando os dados em janelas fixas de 3000 ms subdivididas em cinco segmentos (W1–W5), o que permite representar explicitamente o contexto temporal do evento crítico. Essa modelagem difere das abordagens convencionais baseadas em janelas longas e deslizantes, e proporciona uma representação mais consistente do comportamento dinâmico da queda, favorecendo a aprendizagem dos modelos mesmo em bases reduzidas.

Especificamente em relação ao trabalho de Palmerini et al. (2020), que também utiliza dados reais do repositório FARSEEING, esta dissertação apresenta contribuições adicionais relevantes. Enquanto Palmerini et al. (2020) realiza a detecção de quedas a partir de janelas temporais extensas e deslizantes, nas quais diferentes fases do evento (pré-impacto, impacto e pós-impacto) são analisadas de forma agregada por meio de atributos projetados manualmente, o presente estudo propõe uma segmentação temporal explícita e centrada no evento de queda, estruturando os dados em janelas fixas de 3000 ms subdivididas em cinco segmentos (W1–W5), nos quais o instante crítico é isolado e representado diretamente., permitindo tratar o problema como uma tarefa de classificação supervisionada de segmentos diretamente associados à queda. Além disso, amplia-se o escopo experimental com a inclusão de algoritmos de aprendizado profundo, como 1D-CNN e LSTM, inexistentes no estudo de referência, bem como a aplicação de modelos robustos ao desbalanceamento, como o Balanced Random Forest. Outro diferencial importante é a condução dos experimentos sem o uso de

técnicas artificiais de balanceamento, demonstrando que é possível alcançar resultados expressivos — incluindo sensibilidade de 100% e elevada acurácia balanceada — mesmo em um cenário restritivo com apenas 23 eventos de queda. Por fim, a incorporação de um conjunto mais abrangente de métricas estatísticas, como coeficiente Kappa, taxa sem informação (NIR), intervalos de confiança e valor-p, proporciona uma avaliação mais rigorosa e confiável do desempenho dos modelos. Em conjunto, esses elementos caracterizam uma contribuição metodológica e experimental que vai além do estado da arte estabelecido por Palmerini, reforçando a originalidade e a relevância científica desta dissertação.

6.2 TRABALHOS FUTUROS

Como trabalhos futuros, sugere-se, primeiramente, a ampliação da base de dados com novos registros de quedas reais, de modo a reduzir o desbalanceamento entre classes e permitir uma exploração mais aprofundada de modelos de aprendizagem profunda. A disponibilidade de um volume maior de eventos reais possibilitaria o treinamento de arquiteturas mais complexas, bem como a aplicação de técnicas avançadas de regularização e validação cruzada com maior robustez estatística.

Adicionalmente, em consonância com tendências observadas em trabalhos recentes da literatura, recomenda-se a investigação de abordagens mistas que combinem dados reais e dados simulados de quedas, além de dados de ADLs com atividades de maior intensidade. Esse tipo de estratégia pode contribuir para enriquecer o conjunto de treinamento, preservar características relevantes dos dados reais e, ao mesmo tempo, explorar cenários específicos ou raros por meio de simulações controladas. A adoção dessa abordagem híbrida pode representar um caminho promissor para mitigar a escassez de dados reais, mantendo a validade dos modelos em contextos práticos.

Outro passo relevante consiste na avaliação dos algoritmos em ambientes embarcados ou sistemas de monitoramento em tempo real, considerando restrições computacionais, consumo energético e latência. Essa etapa permitiria analisar a viabilidade de implementação dos modelos em dispositivos vestíveis ou sistemas de assistência à saúde, aproximando os resultados acadêmicos de aplicações reais. Em conjunto, essas extensões podem contribuir para a consolidação de soluções mais robustas, escaláveis e aplicáveis no contexto da prevenção e monitoramento automático de quedas.

6.3 CONSIDERAÇÕES

Em síntese, os resultados obtidos confirmam a relevância científica e prática do uso de algoritmos de aprendizagem de máquina na detecção de quedas com dados reais, evidenciando tanto o

potencial dessas técnicas quanto os desafios inerentes ao domínio, especialmente no que se refere à disponibilidade de dados e à generalização dos modelos. Como desdobramento direto desta pesquisa, os principais resultados deram origem a um artigo científico publicado Trindade, Venturi e Ramirez (2025), no CIMPS (14th International Conference on Software Process Improvement), realizado na *Pontificia Universidad Católica del Perú*, no período de 15 a 17 de outubro de 2025, em Lima, Peru (Apêndice A), tendo sido premiado com o *The Best Article in IEEE* (Apêndice B). Tal reconhecimento ocorreu no contexto de um evento científico internacional com revisão por pares da IEEE, inserido no campo da engenharia de software e melhoria de processos, e reforça a consistência metodológica, a originalidade da abordagem e a contribuição efetiva deste trabalho para o avanço do estado da arte na detecção automática de quedas.

REFERÊNCIAS

ABADI ASHISH AGARWAL, e. a. M. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Software available from tensorflow.org. Disponível em: <https://www.tensorflow.org/>.

ABBATE, S.; AVVENUTI, M.; CORSINI, P.; LIGHT, J.; VECCHIO, A. Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: a survey. In: MERRETT, G. V.; TAN, Y. K. (Ed.). **Wireless Sensor Networks**. Rijeka: IntechOpen, 2010. cap. 9. Disponível em: <https://doi.org/10.5772/13802>.

BOURKE, A. K.; KLENK, J.; SCHWICKERT, L.; AMINIAN, K.; IHLEN, E. A.; MELLONE, S.; HELBOSTAD, J. L.; CHIARI, L.; BECKER, C. Fall detection algorithms for real-world falls harvested from lumbar sensors in the elderly population: a machine learning approach. **Annual International Conference of the IEEE Engineering in Medicine and Biology Society**, v. 2016, p. 3712–3715, Aug 2016. ISSN 2694-0604. Disponível em: <https://doi.org/10.1109/EMBC.2016.7591534>.

BRODERSEN, K. H.; ONG, C. S.; STEPHAN, K. E.; BUHMANN, J. M. The balanced accuracy and its posterior distribution. In: **2010 20th International Conference on Pattern Recognition**. [S.l.: s.n.], 2010. p. 3121–3124.

CAYA, M. V. C.; MAGWILI, G. V.; AGULTO, D. L.; LARANANG, R. J.; PALOMO, L. K. G. Supervised machine learning-based fall detection. **2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)**, p. 1–6, 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:77385641>.

CAZETTA, V.; OLIVEIRA, R. C.; TAVARES, J. M. Os atlas anatômicos como pedagogia cultural e o pós-vida das imagens. **Educação & Realidade**, Universidade Federal do Rio Grande do Sul - Faculdade de Educação, v. 44, n. 3, p. e89165, 2019. Recebido em: 22 dez. 2018; aprovado em: 22 mar. 2019. Acesso em: 7 jul. 2025. Disponível em: <https://www.scielo.br/j/edreal/a/GL38YxDnY7tD4KkCrTpndcG/>.

CEDUP, S. da Educação do Estado do C. **Massoterapia, biomecânica e cinesiologia**. 2011. <https://www.seduc.ce.gov.br/wp-content/uploads/sites/37/2011/01/massoterapia_biomecanica_cinesiologia.pdf>. Acesso em: 7 jul. 2025.

ENDEVCO, C. **Accelerometer Types and Applications – Technical Paper 327**. [S.l.], 2012. Acesso em: 26 maio 2025. Disponível em: <https://web.archive.org/web/20171214072240/https://endevco.com/news/newsletters/2012_07/tp327.pdf>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

HALE, W. A.; DELANEY, M. J.; CABLE, T. Accuracy of patient recall and chart documentation of falls. **The Journal of the American Board of Family Medicine**, The Journal of the American Board of Family Medicine, v. 6, n. 3, p. 239–242, 1993. ISSN 1557-2625. Disponível em: <https://www.jabfm.org/content/6/3/239>.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, p. 1735–1780, 1997. Disponível em: <https://api.semanticscholar.org/CorpusID:1915014>.

HORSFALL, I. et al. Understanding real-world falls using wearable sensor data: implications for fall prevention and detection. **Injury Prevention**, v. 30, n. 4, p. 272–278, 2024. Disponível em: <https://injuryprevention.bmj.com/content/30/4/272>.

IBM. **O que é uma máquina de vetor de suporte (SVM)? — IBM Think**. 2023. Atualizado em: 27 dez. 2023. Acesso em: 16 jul. 2025. Disponível em: <https://www.ibm.com/br-pt/think/topics/support-vector-machine>.

____. **What is random forest?** 2025. <https://www.ibm.com/think/topics/random-forest>. Acesso em: 14 jul. 2025.

____. **What is the k-nearest neighbors (KNN) algorithm?** 2025. <https://www.ibm.com/think/topics/knn>. Acesso em: 14 jul. 2025.

Keras Team. **Conv1D layer**. 2025. Acesso em: 2025. Disponível em: <https://keras.io/api/layers/convolution_layers/convolution1d/>.

KIRANYAZ, S.; AVCI, O.; ABDELJABER, O.; INCE, T.; GABBOUJ, M.; INMAN, D. J. 1d convolutional neural networks and applications: A survey. **Mechanical Systems and Signal Processing**, v. 151, p. 107398, 2021. ISSN 0888-3270. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0888327020307846>.

KITCHENHAM, B. **Procedures for Performing Systematic Reviews**. Keele, Staffs, UK, 2004. Joint Technical Report. Software Engineering Group, Department of Computer Science, Keele University and Empirical Software Engineering, National ICT Australia Ltd. (NICTA Technical Report 0400011T.1). ISSN: 1353-7776.

KLENK, J.; SCHWICKERT, L. e. a. The farseeing real-world fall repository: a large-scale collaborative database to collect and share sensor signals from real-world falls. **European Review of Aging and Physical Activity**, v. 13, n. 8, 2016. Disponível em: <https://eurapa.biomedcentral.com/articles/10.1186/s11556-016-0168-9>.

KUHN, M. **caret: Classification and Regression Training**. [S.l.], 2023. Documentação oficial e vignettes do pacote. Disponível em: <https://cran.r-project.org/package=caret>.

MARAY, N.; NGU, A. H.; NI, J.; DEBNATH, M.; WANG, L. Transfer learning on small datasets for improved fall detection. **Sensors**, v. 23, n. 3, 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/3/1105>.

MARTINEZ, E. Z. **Métricas de Avaliação de Classificadores**. 2020. Vídeo aula. Disponível em: <https://eaulas.usp.br/portal/video.action?idItem=9668>. Acesso em: jan. 2026. Disponível em: <https://eaulas.usp.br/portal/video.action?idItem=9668>.

National Institute for Health and Care Excellence. **Falls: assessment and prevention in older people and in people 50 and over at higher risk**. 2024. NICE Guideline NG249. Disponível em: <https://www.nice.org.uk/guidance/ng249>.

NIELSEN, A. **Análise Prática de Séries Temporais: Predição com Estatística e Aprendizado de Máquina**. [S.l.]: Alta Books, 2020. 257 p. ISBN 978-85-5081-562-6.

NOBLE, W. S. What is a support vector machine? **Nature Biotechnology**, v. 24, n. 12, p. 1565–1567, December 2006. ISSN 1546-1696. Disponível em: <https://doi.org/10.1038/nbt1206-1565>.

NUÑEZ, T. O. **Avaliação do Desempenho de Algoritmos de Machine Learning Aplicados ao Problema de Detecção de Quedas de Pessoas**. Dissertação (Mestrado), 2023.

OMEGA, D. **What is an Accelerometer? Working Principles, Types, and Applications**. 2024. Acesso em: 26 maio 2025. Disponível em: <https://www.dwyeromega.com/en-us/resources/accelerometers>.

ORDóñEZ, F. J.; ROGGEN, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. **Sensors**, v. 16, n. 1, 2016. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/16/1/115>.

PAGE, M. J.; MCKENZIE, J. E.; BOSSUYT, P. M.; BOUTRON, I.; HOFFMANN, T. C.; MULROW, C. D. et al. The prisma 2020 statement: an updated guideline for reporting systematic reviews. **BMJ**, v. 372, p. n71, 2021.

PALMERINI, L.; KLENK, J.; BECKER, C.; CHIARI, L. Accelerometer-based fall detection using machine learning: Training and testing on real-world falls. **Sensors**, v. 20, n. 22, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/22/6479>.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

POPPER, K. R. **A lógica da pesquisa científica**. [S.l.]: Editora Cultrix, 2004.

POWERS, D. M. W. Evaluation: From precision and recall to informedness, markedness and correlation. **Journal of Machine Learning Technologies**, v. 2, n. 1, p. 37–63, 2020. Disponível em: <https://arxiv.org/abs/2010.16061>.

RAJAGOPALAN, R.; LITVAN, I.; JUNG, T.-P. Fall prediction and prevention systems: Recent trends, challenges, and future research directions. **Sensors**, v. 17, n. 11, 2017. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/17/11/2509>.

RUEDA, C. F. C. **Aexperiência de mobilidade urbana no contexto do envelhecimento sob uma aobrdagem do Desing Inclusivo**. Dissertação (Mestrado), 2024.

SALEH, M.; ABBAS, M.; JEANNES, R. L. FallAllD: An Open Dataset of Human Falls and Activities of Daily Living for Classical and Deep Learning Applications. **IEEE Sensors Journal**, Institute of Electrical and Electronics Engineers, v. 21, n. 2, p. 1849–1858, jan. 2021. Disponível em: <https://hal.science/hal-03102536>.

SCIKIT-LEARN. **1.4. Support Vector Machines — scikit-learn 1.7.0 documentation**. 2025. <https://scikit-learn.org/stable/modules/svm.html>. Acessado em: 5 de julho de 2025.

SCIKIT-LEARN. **Decision Trees**. 2025. <https://scikit-learn.org/stable/modules/tree.html>. Versão 1.7.0. Acesso em: 14 jul. 2025.

____. **sklearn.neighbors: Nearest Neighbors — User Guide**. 2025. <https://scikit-learn.org/stable/modules/neighbors.html>. Versão 1.7.0, janeiro 2025. Acesso em: 14 jul. 2025.

SILVA, J.; SOUSA, I.; CARDOSO, J. Transfer learning approach for fall detection with the farseeing real-world dataset and simulated falls. In: **2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**. [S.l.: s.n.], 2018. p. 3509–3512.

TEAM, T. pandas development. **pandas-dev/pandas: Pandas**. Zenodo, 2024. Disponível em: <https://doi.org/10.5281/zenodo.10957263>.

TENSORFLOW, D. **tf.keras.layers.LSTM — TensorFlow v2.15**. 2024. <https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM>. Acesso em: 16 jul. 2025.

TensorFlow Team. **Convolutional Neural Networks**. 2025. Acesso em: 2025. Disponível em: <https://www.tensorflow.org/tutorials/images/cnn>.

TRINDADE, P. H.; VENTURI, R. B.; RAMIREZ, A. R. G. Analysis of machine learning algorithms for detecting falls in individuals using data from the farseeing repository. In: **2025 14th International Conference On Software Process Improvement (CIMPS)**. [S.l.: s.n.], 2025. p. 173–177.

VILLA, S. García-de; CASILLAS-PéREZ, D.; JIMéNEZ-MARTíN, A.; GARCíA-DOMíNGUEZ, J. J. Inertial sensors for human motion analysis: A comprehensive review. **IEEE Transactions on Instrumentation and Measurement**, v. 72, p. 1–39, 2023.

WHO, W. H. O. **FALLS**. 2021. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/falls>. Acesso em: 04 jul. 2025.