

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

VÍTOR SORPILE GERALDO

COMPARAÇÃO DE MÉTODOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO
DE QUEDA BASEADA EM UNIDADES DE MEDIÇÃO INERCIAL

Joinville
2024

VÍTOR SORPILE GERALDO

COMPARAÇÃO DE MÉTODOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO
DE QUEDA BASEADA EM UNIDADES DE MEDIÇÃO INERCIAL

Trabalho apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica, no Curso de Engenharia Mecatrônica, do Centro Tecnológico de Joinville, da Universidade Federal de Santa Catarina.

Orientador: Dr. Pablo Andretta Jaskowiak

Joinville
2024

Dedico este trabalho à minha família.

AGRADECIMENTOS

Agradeço à minha mãe, Sandra, e à toda a minha família, pelo apoio e incentivo em todas as etapas que me trouxeram até aqui e ao longo dessa jornada.

Agradeço à minha namorada, Laura, pelo amor e companheirismo durante essa fase.

Agradeço ao professor e orientador Dr. Pablo Andretta Jaskowiak pelo auxílio e orientação tanto no desenvolvimento deste trabalho quanto nos semestres de monitoria.

Agradeço aos meus amigos e colegas pelo momentos de descontração que tornaram esse caminhada mais leve.

RESUMO

As quedas trazem sérios riscos aos indivíduos, especialmente aos idosos, que tendem a ter redução nas capacidades de movimentação. Tais riscos são agravados quando a pessoa não consegue se levantar sozinha e não há alguém perto que possa auxiliá-la. Em função disso, os sistemas de detecção de queda aparecem como uma alternativa para reduzir as sequelas causadas por esses acidentes. Tais dispositivos processam dados obtidos por sensores para identificar quando ocorreu uma queda e alertar familiares e serviços de emergência. Um elemento importante desses sistemas é o algoritmo que efetivamente realiza a detecção das quedas. Nesse contexto, este trabalho busca comparar o desempenho de métodos de aprendizado de máquina (AM) na identificação de quedas a partir de dados de acelerômetros e giroscópios posicionados em diferentes locais do corpo. Para isso, foi utilizada uma base de dados que contém características extraídas dos dados brutos dos sensores e técnicas de treinamento e avaliação de modelos comuns na literatura de AM. Os métodos avaliados nesse estudo são K-Nearest Neighbors (KNN), Random Forest (RF), Multi-Layer Perceptron (MLP) e Support Vector Machine (SVM). Os melhores resultados, visando a usabilidade e confiabilidade de um sistema de detecção de quedas, foram obtidos pelo RF a partir dos dados dos sensores localizados no tornozelo.

Palavras-chave: inteligência artificial; saúde; acelerômetro; giroscópio.

ABSTRACT

Falls pose serious risks to individuals, especially to the elderly, who tend to experience reduced mobility. These risks are exacerbated when the person is unable to get up on their own and there is no one nearby to assist. As a result, fall detection systems have emerged as an alternative to mitigate the consequences of such accidents. These devices process data obtained from sensors to identify when a fall has occurred and alert family members and emergency services. An important component of these systems is the algorithm that effectively performs the fall detection. In this context, this study aims to compare the performance of machine learning algorithms in identifying falls based on data from accelerometers and gyroscopes positioned on different locations of the body. For this purpose, a dataset containing features extracted from raw sensor data was used, along with training and evaluation techniques commonly found in the machine learning literature. The methods evaluated in this study are K-Nearest Neighbors (KNN), Random Forest (RF), Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM). The best results, considering the usability and reliability of a fall detection system, were achieved by the RF method using data from sensors positioned on the ankle.

Keywords: artificial intelligence; healthcare; accelerometer; gyroscope.

LISTA DE FIGURAS

Figura 1 – Sentidos de medição de giroscópios e acelerômetros de 3 eixos . . .	13
Figura 2 – Diagrama de etapas de um sistema de detecção de quedas	14
Figura 3 – Exemplos de KNN	17
Figura 4 – Exemplo de topologia de um MLP	18
Figura 5 – Exemplos de kernel para SVM	20
Figura 6 – Exemplo de validação cruzada K-Fold com K=5	21
Figura 7 – Comparação entre validação cruzada K-Fold e estratificada K-Fold .	22
Figura 8 – Matriz de confusão genérica	23
Figura 9 – Divisão da base de dados inicial	28
Figura 10 – Recall por posicionamento e características	32
Figura 11 – Precisão por posicionamento e características	33
Figura 12 – f1-score por posicionamento e características	34
Figura 13 – Acurácia por posicionamento e características	35

LISTA DE QUADROS

Quadro 1 – Características disponíveis na base de dados	27
Quadro 2 – Espaço de busca dos hiperparâmetros	30
Quadro 3 – Hiperparâmetros mais frequentes	31
Quadro 4 – Lista de atividades	37

LISTA DE TABELAS

Tabela 1 – Melhores resultados	36
Tabela 2 – Porcentagem média de erros de identificação do KNN	38
Tabela 3 – Porcentagem média de erros de identificação do RF	39
Tabela 4 – Porcentagem média de erros de identificação do MLP	39
Tabela 5 – Porcentagem média de erros de identificação do SVM	40

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
FN	False Negative
FP	False Positive
IA	Inteligência Artificial
KNN	K-Nearest Neighbors
MEMS	Sistemas Microeletromecânicos
MLP	Multi-Layer Perceptron
OMS	Organização Mundial de Saúde
RF	Random Forest
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	ACELERÔMETROS E GIROSCÓPIOS	13
2.2	SISTEMAS DE DETECÇÃO DE QUEDA	14
2.3	APRENDIZADO DE MÁQUINA	16
2.3.1	K-Nearest Neighbors	16
2.3.2	Random Forest	17
2.3.3	Multi-Layer Perceptron	18
2.3.4	Support Vector Machine	19
2.4	AVALIAÇÃO DE DESEMPENHO DE MODELOS DE CLASSIFICAÇÃO	20
2.4.1	Stratified K-Fold Cross Validation	20
2.4.2	Grid Search Cross Validation	21
2.4.3	Métricas de Avaliação para Classificação Binária	22
2.5	TRABALHOS RELACIONADOS	24
3	MATERIAIS E MÉTODOS	26
3.1	BASE DE DADOS	26
3.1.1	Preparação dos Dados	28
3.2	MÉTODOS ESCOLHIDOS	28
3.2.1	Treinamento dos Modelos e Otimização de Hiperparâmetros	29
4	RESULTADOS	31
5	CONCLUSÕES	41
	REFERÊNCIAS	43

1 INTRODUÇÃO

Sendo a segunda maior causa de mortes acidentais no planeta, ultrapassando o número de 684 mil fatalidades por ano, e com cerca de 172 milhões de acidentes causando sequelas de curto ou longo prazo no mesmo período, as quedas são um grave problema para os sistemas de saúde (Organização Mundial de Saúde - OMS, 2021b). Entre 2000 e 2020, foram registradas 1.746.097 internações hospitalares no Brasil devido a quedas de idosos, pessoas de 60 anos ou mais, gerando custos de mais de dois bilhões de reais ao Sistema Único de Saúde do Brasil (Lima *et al.*, 2022).

Uma queda pode ser definida como um evento de deslocamento não intencional de uma pessoa ao chão ou nível inferior, usualmente de forma inesperada. Apesar de todas as pessoas estarem sujeitas a quedas e lesões decorrentes, a idade é um fator relevante no tipo e severidade da lesão observada. Nesse sentido, os idosos são um dos grupos de alto risco, visto que com o envelhecimento há um declínio nas capacidades física, sensorial e cognitiva (OMS, 2021a)

Segundo Igual, Medrano e Plaza (2013), o tempo que um idoso fica no chão após cair é um dos pontos cruciais para determinar a severidade da queda, visto que muitos podem não conseguir se levantar sozinhos. Tal fato, pode levar a complicações como desidratação e hipotermia, sendo essa situação crítica quando a pessoa perde a consciência ou mora sozinha. Além disso, é possível que quedas estejam atreladas a um problema de saúde como um acidente vascular cerebral ou um ataque cardíaco (OMS, 2021b).

Assim, é importante que serviços de emergência consigam atender em tempo hábil um indivíduo que sofreu uma queda a fim de identificar possíveis lesões e condições relacionadas, tomando as medidas necessárias para tratamento e encaminhamento do paciente. Ações que são iniciadas no local do incidente, como medidas de primeiros socorros, estabilização, imobilização e o transporte para a correta instituição de tratamento, são capazes de salvar vidas (OMS, 2021b).

Nesse contexto, dispositivos capazes de detectar quedas e emitir alertas são uma das possíveis precauções para minimizar as sequelas causadas por esse tipo de acidente (Xu; Zhou; Zhu, 2018). O princípio central de funcionamento de um sistema de detecção de queda inclui a aquisição e processamento de dados de sensores, a identificação de uma queda a partir dessas informações e a notificação de familiares, cuidadores ou serviço de emergência (Gharghan; Hashim, 2024).

Diferentes tipos de sensores podem ser utilizados para detecção de queda, como acelerômetros, giroscópios, câmeras, radares, microfones e sensores de pressão, sendo possível combinar mais de um tipo de sensor para identificar quedas com mais acurácia. Visto que são pequenos, leves, consomem pouca energia e são mais baratos

do que outros sensores, acelerômetros e giroscópios são amplamente utilizados nesse tipo de aplicação (Gharghan; Hashim, 2024).

O posicionamento desses dois tipos de sensores afeta não só a usabilidade do dispositivo, mas também o processamento necessário para a detecção. A cintura é vista como uma posição melhor em comparação ao punho ou joelho, já que é próxima ao centro de gravidade do corpo e não tem tanta variação de movimento em atividades do dia a dia quanto braços e pernas. Apesar disso, o punho é uma ótima opção no quesito usabilidade, visto que o dispositivo pode ser desenvolvido com formato semelhante a um relógio de pulso (Kangas *et al.*, 2008).

Uma vez que a localização dos sensores foi determinada, os dados obtidos são processados a fim de extrair características que serão utilizadas pelos algoritmos para realizar a detecção em si. Diferentes tipos podem ser aplicados, sendo os mais comuns os por limiar e os baseados em Inteligência Artificial (IA). Dentre as técnicas de IA, existem vários métodos de aprendizado de máquina e de aprendizado profundo que podem ser utilizados nessa situação (Gharghan; Hashim, 2024).

Sendo assim, neste trabalho busca-se avaliar o desempenho de diferentes métodos de IA para a detecção de quedas quando aplicados em dispositivos que utilizam acelerômetros e giroscópios. A base de dados escolhida possui informações de tais sensores posicionados no tornozelo, bolso, cintura, pescoço e punho, logo será analisado como cada método se comporta se forem utilizadas apenas uma das posições ou todas em conjunto. Em relação aos métodos estudados, foram considerados o K-Nearest Neighbors (KNN), Random Forest (RF), Multi-Layer Perceptron (MLP) e Support Vector Machine (SVM), por meio da biblioteca scikit-learn (Pedregosa *et al.*, 2011).

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Analisar o desempenho de diferentes métodos de aprendizado de máquina para detecção de queda baseada em acelerômetros e giroscópios posicionados em diferentes pontos do corpo.

1.1.2 Objetivos Específicos

- Implementar scripts de avaliação considerando os métodos escolhidos;
- Otimizar os hiperparâmetros de cada método com busca em grade;
- Analisar a influência de diferentes atributos calculados a partir dos dados dos sensores;
- Avaliar o impacto do posicionamento dos sensores no corpo;
- Verificar quais atividades apresentam a maior taxa de classificação incorreta.

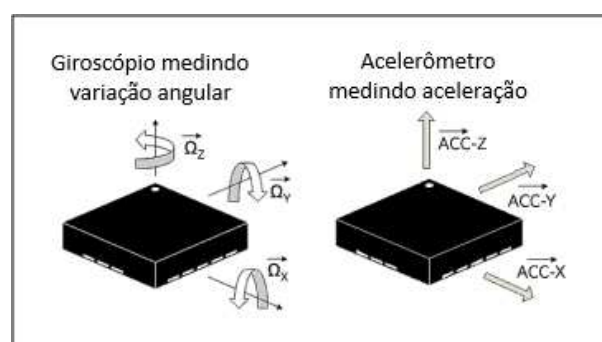
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão explicitados os principais fundamentos necessários para a compreensão da pesquisa sobre detecção de queda a partir de dados de sensores de aceleração e variação angular utilizando métodos de IA. Primeiro, será discorrido sobre o funcionamento de acelerômetros e giroscópios. Em seguida, como esses sensores podem ser aplicados em um sistema de detecção de queda, além de outras técnicas que podem ser utilizadas nesses sistemas. Na sequência, será definido o aprendizado de máquina, com foco no aprendizado supervisionado para tarefas de classificação, assim como os métodos utilizados, KNN, RF, MLP e SVM. Por fim, serão apresentados métodos e métricas para avaliar e otimizar modelos de classificação, e também serão analisados trabalhos relacionados à esta pesquisa.

2.1 ACELERÔMETROS E GIROSCÓPIOS

Acelerômetros e giroscópios são dispositivos capazes de medir, respectivamente, aceleração e velocidade angular em um ou mais eixos. Esses sensores são comumente integrados em conjunto em uma Unidade de Medição Inercial (IMU), podendo ser utilizados em diversas aplicações, como estabilização de imagem, reconhecimento de gestos e sistemas de navegação. A Figura 1 ilustra os sentidos de medição de uma IMU do tipo Sistema Microeletromecânico (MEMS) que possui um giroscópio e um acelerômetro de 3 eixos (Dadafshar, 2015).

Figura 1 – Sentidos de medição de giroscópios e acelerômetros de 3 eixos



Fonte: Adaptado de Dadafshar (2015).

O uso de MEMS tem aumentado cada vez mais na indústria de aparelhos eletrônicos de consumo, visto que conseguem combinar componentes eletrônicos e mecânicos em um pequeno encapsulamento. As IMUs que utilizam dessa tecnologia são geralmente construídas com um circuito de medição de posição de uma pequena massa, sendo essa medição utilizada para obter a aceleração e a velocidade angular

do sistema (Dadafshar, 2015).

2.2 SISTEMAS DE DETECÇÃO DE QUEDA

Os sistemas de detecção de queda podem ser definidos como dispositivos desenvolvidos para emitir alertas quando ocorre uma queda. Para alcançar esse objetivo, a principal tarefa consiste em diferenciar quedas de atividades do dia a dia. Essa distinção, entretanto, não é simples e direta de ser executada, visto que atividades comuns, como deitar e sentar, podem ser interpretadas como quedas. Logo, uma das características que torna um sistema assim robusto é a capacidade de classificar corretamente quedas e não-quedas (Igal; Medrano; Plaza, 2013).

Segundo Gharghan e Hashim (2024), a estrutura básica de um sistema de detecção de quedas engloba a coleta de dados por sensores, seguida pelo processamento desses dados para a extração de características relevantes à aplicação, o que possibilita identificar padrões relacionados a quedas e assim, determinar quando um alerta deve ser emitido. A Figura 2 apresenta um diagrama que ilustra a sequência dessas etapas, sendo a primeira e a terceira detalhadas a seguir.

Figura 2 – Diagrama de etapas de um sistema de detecção de quedas



Fonte: Elaborado pelo autor (2024).

A etapa inicial de obtenção dos dados pode ser realizada utilizando diferentes tipos de sensores. Os dispositivos são geralmente categorizados como *wearables*, quando são usados ou vestidos pelo usuário, ou relacionados ao contexto, quando são posicionados no ambiente onde o usuário se encontra. Cada categoria apresenta vantagens e limitações em relação a aspectos como facilidade de uso, preço, alcance de cobertura e privacidade (Liu *et al.*, 2023).

Câmeras, radares e sensores acústicos, ultrassônicos ou de infravermelho são alguns exemplos de sensores que podem ser aplicados em sistemas de detecção de queda relacionados ao contexto. Esses têm a vantagem de não serem utilizados junto ao corpo, o que facilita a adoção ao uso, principalmente, por parte dos idosos. Entretanto, como precisam ser posicionados em um local específico, o sistema é limitado a realizar a detecção apenas na área em que sensores estão instalados. Além disso, podem existir ruídos originados por objetos, pessoas ou animais que passem

pela área de medição. Existe também uma questão de privacidade, principalmente, quando são utilizadas câmeras, que impossibilita o uso desses dispositivos em certos ambientes, como banheiros (Liu *et al.*, 2023).

Já os dispositivos *wearables* podem sofrer resistência quanto ao uso, devido a necessidade de serem utilizados junto ao corpo para que consigam realizar a detecção. Essa abordagem, entretanto, não possui limitações no que diz respeito à área onde as quedas podem ser detectadas, uma vez que o detector acompanha o usuário. Os principais sensores utilizados nessa categoria são acelerômetros, giroscópios e IMUs, que, com os avanços tecnológicos dos MEMS, vem se tornando cada vez mais leves, compactos e acessíveis. Sendo assim, não há preocupações significativas com privacidade, pois nenhum dado sensível ou imagem são capturados (Liu *et al.*, 2023).

Além dos dispositivos dedicados à detecção de quedas, celulares também podem ser empregados nessa função, devido à presença de vários sensores embarcados, incluindo acelerômetros, combinado com alta taxa de adoção, capacidade de processamento e conectividade. Em contrapartida, como são equipamentos com múltiplas funcionalidades, podem ser usados de forma que sofram uma variação brusca de aceleração, o que pode levar a uma falsa detecção de queda. Além disso, não são transportados junto ao corpo de uma única forma, podendo ser colocados no bolso da calça, dentro de uma bolsa ou mochila ou segurados na mão, o que dificulta a generalização por parte dos algoritmos de detecção (Stampfler *et al.*, 2022).

Na etapa de detecção de quedas, são aplicados dois tipos principais de algoritmos, os por limiar e os que implementam técnicas de IA. A primeira abordagem consiste em definir valores limite para parâmetros extraídos dos dados dos sensores, de forma que a identificação de uma queda acontece quando esses limiares são ultrapassados. Esses algoritmos são considerados simples de implementar e interpretar e são eficientes computacionalmente, o que faz com não sejam necessários muitos recursos computacionais para a execução. Porém, são limitados em cenários complexos, uma vez que não é trivial estabelecer regras que atendam todos os tipos e intensidades de quedas e perfis de usuários (Gharghan; Hashim, 2024).

A segunda abordagem utiliza algoritmos de IA, que são capazes de reconhecer padrões a partir de características derivadas dos dados obtidos pelos sensores. Esses modelos precisam primeiro ser treinados com uma base de dados que associa as características disponíveis à ação executada, seja uma queda ou atividade rotineira. A qualidade e o tamanho dessa base são cruciais para a performance, pois impactam diretamente na capacidade de generalização. Com isso, os modelos conseguem aprender relações não lineares entre os dados e se adaptam a diferentes situações de quedas, apresentando melhores resultados em cenários complexos em comparação à primeira abordagem. Entretanto, o treinamento é uma tarefa computacionalmente complexa, que pode demandar alta capacidade de processamento e longo tempo de

execução (Gharghan; Hashim, 2024).

2.3 APRENDIZADO DE MÁQUINA

O aprendizado pode ser definido como o processo de aprimoramento em uma atividade com base em experiências anteriores. A partir disso, o aprendizado de máquina consiste em desenvolver métodos que possibilitem que computadores modifiquem suas ações a fim de realizar uma tarefa da forma mais precisa possível, sem de fato programá-los para tal. Esse processo inclui os dados de entrada e as características que podem ser extraídas deles, o algoritmo de aprendizado e seus parâmetros, o treinamento e a avaliação. A forma como o treinamento é conduzido, ou seja, como o algoritmo avalia seu progresso, define o tipo de aprendizado utilizado, podendo ser supervisionado, não supervisionado, por reforço ou evolutivo (Marsland, 2014).

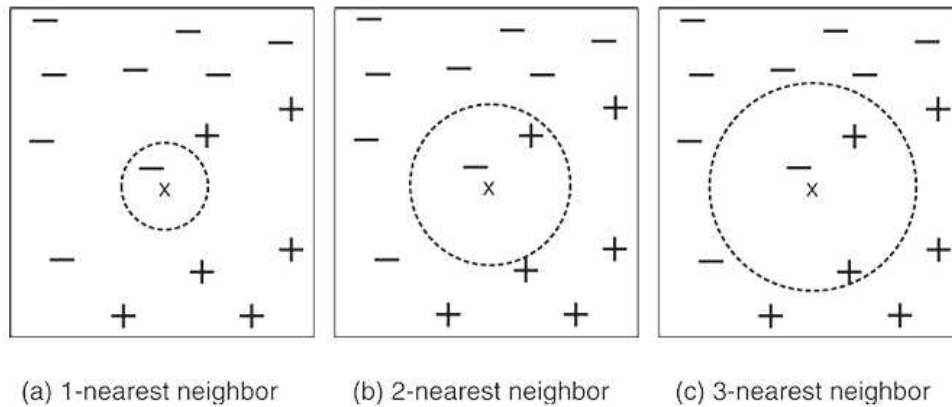
A aprendizagem supervisionada, foco deste trabalho, utiliza um conjunto de dados de treino que mapeia amostras de características de entrada às respectivas saídas esperadas. Conhecendo as respostas aguardadas, o algoritmo consegue ajustar o funcionamento interno de forma a calcular o resultado correto. Os ajustes são feitos com o objetivo de alcançar a generalização, o que implica no algoritmo ser capaz de chegar no valor esperado para amostras que não foram utilizadas durante o treinamento (Marsland, 2014).

Segundo Russell e Norvig (2021), o aprendizado supervisionado pode ser aplicado para resolver dois principais tipos de problema, os de regressão e os de classificação. No primeiro, o objetivo é prever um valor contínuo, como a previsão de temperatura, enquanto no segundo, o intuito é categorizar os dados de entrada em um conjunto determinado de classes, por exemplo a previsão de condições climáticas (ensolarado, nublado ou chuvoso) ou o caso abordado nesse estudo, a identificação da ocorrência ou não de uma queda. Este trabalho lida portanto, com um problema de classificação no contexto de AM supervisionado.

2.3.1 K-Nearest Neighbors

O KNN é um método de aprendizado supervisionado baseado em instâncias. Em vez de construir um modelo que generaliza explicitamente o problema, as amostras de treinamento são armazenadas e utilizadas para classificar novas amostras por semelhança. Cada dado é representado por um ponto em um espaço com n dimensões, sendo n o número de características, e para classificar um dado novo, a distância entre ele e os pontos de treino é calculada a fim de classificá-lo com base na classe das amostras próximas ou semelhantes (Tan *et al.*, 2019).

Figura 3 – Exemplos de KNN



Fonte: Tan *et al.* (2019).

Um parâmetro importante do KNN é o número de vizinhos levados em consideração para categorizar uma amostra. A Figura 3 ilustra a categorização de um mesmo exemplo utilizando 1, 2 e 3 vizinhos. Na primeira situação, como o ponto mais próximo é da classe negativa, o resultado seria a classe negativa. Já no segundo caso, há um empate entre as classes, assim, o resultado poderia ser escolhido de forma aleatória ou a distância para cada ponto poderia ser utilizada como critério. Por fim, com 3 vizinhos, se for considerada apenas a maioria dos rótulos vizinhos, a classe resultante seria a positiva. Outra opção, porém, seria atribuir pesos à distância de cada amostra de forma que pontos mais próximos influenciem mais na decisão. Isso é particularmente importante para que a escolha não seja feita baseada apenas no número de vizinhos (Tan *et al.*, 2019).

Apesar de não possuir uma etapa de treinamento complexa, visto que não há a construção de um modelo, a classificação de uma amostra pode exigir um alto custo computacional devido a necessidade de calcular a distância para todos os pontos de treino. Além disso, o KNN é sensível a características irrelevantes ao problema, já que elas podem distorcer as métricas de distância (Tan *et al.*, 2019).

2.3.2 Random Forest

Para descrever o RF, primeiro é preciso compreender o conceito de árvore de decisão. Segundo Tan *et al.* (2019), uma árvore de decisão é uma estrutura hierárquica que organiza uma série de perguntas e suas possíveis respostas que leva à conclusão de qual classe a amostra de entrada deve ser classificada. A árvore é construída de forma que os nós folha são sempre associados ao rótulo de uma classe, enquanto os demais nós internos possuem condições de teste relacionadas a alguma característica.

Nesse contexto, o RF é um método que combina árvores de decisão com o objetivo de melhorar a capacidade de generalização. A floresta é construída com base

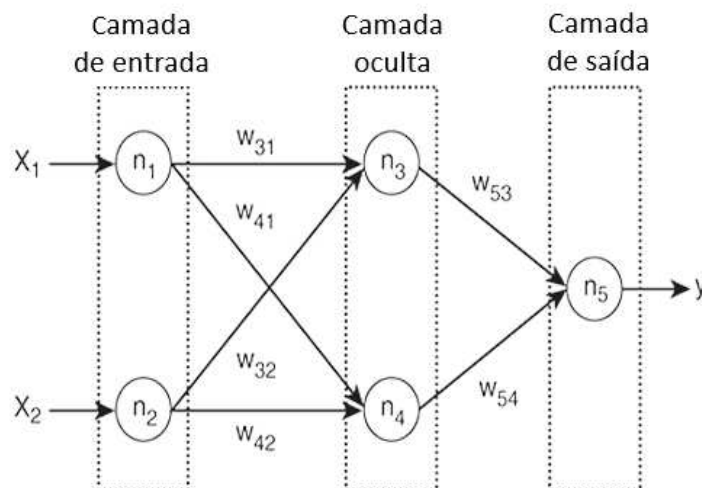
na técnica conhecida como *bagging*. Nela, para cada árvore, uma porção aleatória das instâncias da base de dados é removida e amostras que permaneceram são duplicadas a fim de manter o tamanho inicial da base. Dessa forma, cada árvore consegue generalizar sobre uma parte menor e distinta do problema, entendendo diferentes padrões e variações nos dados (Tan *et al.*, 2019).

Além disso, cada árvore é construída com um subconjunto aleatório das características, logo diferentes atributos das amostras serão considerados por cada uma, aumentando a diversidade da floresta. Sendo assim, a decisão final do RF é feita a partir da combinação das previsões individuais das árvores, sendo escolhido o valor previsto mais vezes. Essa composição de submodelos descorrelacionados torna o método robusto a *overfitting* (Tan *et al.*, 2019).

2.3.3 Multi-Layer Perceptron

O MLP é um método de Redes Neurais Artificiais capaz de generalizar problemas complexos por meio de uma estrutura de camadas sequenciais de neurônios, de forma que o resultado de uma camada é usado como a entrada da seguinte, assim, o problema é decomposto em níveis de abstração mais simples. Como pode ser observado na Figura 4, essas camadas são divididas em três tipos, de entrada, oculta e de saída (Tan *et al.*, 2019).

Figura 4 – Exemplo de topologia de um MLP



Fonte: Adaptado de Tan *et al.* (2019).

Cada nó da camada de entrada representa uma das características das amostras, de forma que esses dados são então transmitidos aos nós da camada oculta, que pode ser composta por uma ou mais camadas, sendo cada valor transmitido ponderado por um determinado peso e processado por uma função de ativação nos nós

que os recebem. Esse processo é repetido até que a camada de saída seja atingida, sintetizando os valores recebidos e produzindo como saída a previsão do modelo. Vale ressaltar que, para um problema de classificação binária, a camada de saída geralmente possui apenas um nó, representando os dois possíveis rótulos de forma binária (Tan *et al.*, 2019).

O ajuste dos pesos aplicados entre cada nó é feito utilizando a técnica de retropropagação de erro, a qual propaga as derivadas parciais do erro total da camada de saída às ocultas, a fim de minimizar o erro nas camadas intermediárias. Nesse contexto, o MLP é capaz de se adaptar a diversos cenários complexos, inclusive conseguindo reconhecer e desconsiderar atributos irrelevantes ao atribuí-los peso zero. Entretanto, necessita de um alto tempo computacional para o treinamento, principalmente, se forem utilizados muitos nós na camada oculta. Porém, uma vez treinado, o modelo classifica novas amostras rapidamente, visto que os pesos já foram otimizados (Tan *et al.*, 2019).

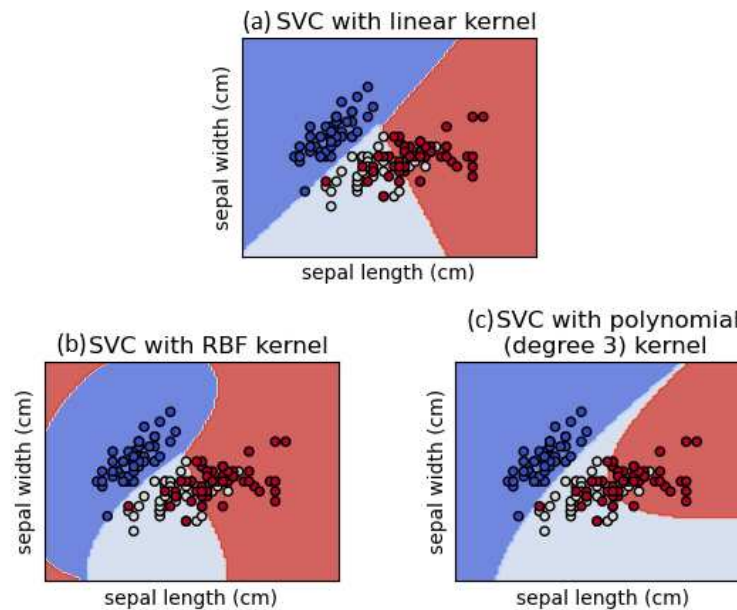
2.3.4 Support Vector Machine

O SVM é um método de IA que busca aprender os limites de decisão, sejam eles lineares ou não-lineares, que separam as classes em um hiperplano no espaço das características. Para tal, são utilizadas apenas as amostras de treino mais difíceis de classificar, conhecidas como vetores de suporte, assim, o modelo precisa generalizar o que diferencia uma classe da outra e não o que caracteriza cada uma (Tan *et al.*, 2019). A construção do hiperplano se dá por meio de uma função de kernel, de forma que o impacto que essa função tem para um mesmo caso pode ser observado na Figura 5.

De acordo com Marsland (2014), inúmeras funções podem ser utilizadas como *kernel*, sendo as mais comuns a linear, a polinomial, a sigmoide e a de base radial. Na Figura 5, é possível visualizar as áreas de classificação obtidas pelo mesmo modelo alterando apenas a função de *kernel*. A Figura 5a ilustra a utilização de uma função linear, resultando em retas como limites de decisão, enquanto a Figura 5b exibe o resultado da escolha de uma função de base radial, que são bordas curvas. Assim como o anterior, os limiares de decisão apresentados na Figura 5c são curvos, resultantes do uso de uma função polinomial de grau 3. Sendo assim, a escolha da função de *kernel* é comumente feita experimentando e validando com diferentes opções até encontrar uma que atenda os requisitos da aplicação, mesmo podendo ser aplicada a teoria da dimensão de Vapnik-Chernik (Marsland, 2014).

Em situações em que a base de dados é desbalanceada, ou seja, existe uma quantidade muito maior de amostras de uma classe do que de outra, os modelos podem não conseguir generalizar adequadamente a classe menos representada. Uma alternativa para reduzir o impacto desse problema é atribuir um peso maior aos exemplos da classe minoritária, o que, do ponto de vista do modelo, equivale a ter

Figura 5 – Exemplos de kernel para SVM



Fonte: Adaptado de scikit-learn (2024).

mais amostras no mesmo ponto (Tan *et al.*, 2019).

2.4 AVALIAÇÃO DE DESEMPENHO DE MODELOS DE CLASSIFICAÇÃO

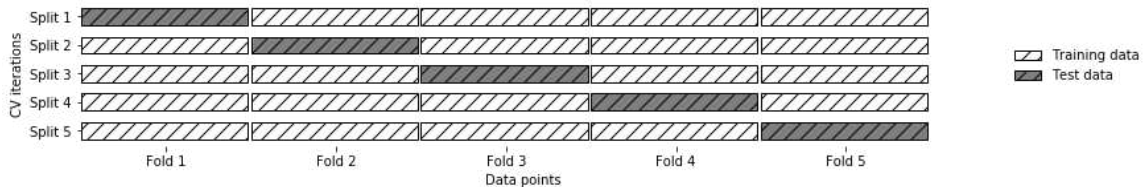
A validação de modelos é uma etapa essencial no processo de aprendizado de máquina. Nela, a performance de cada modelo é medida por meio de métricas para avaliar se os requisitos da aplicação são atendidos e, também, para selecionar qual performa melhor. Um detalhe importante sobre essa etapa é que precisam ser utilizados dados diferentes dos que foram empregados no treinamento do modelo, já que a intenção é medir a capacidade de generalização em amostras desconhecidas por ele e não o quanto ele se adapta ao conjunto em que foi treinado (Varoquaux; Colliot, 2023).

2.4.1 Stratified K-Fold Cross Validation

Visto a importância de separar os dados utilizados para treinar e testar o modelo, uma técnica que permite uma análise mais completa e detalhada do que fazer uma divisão simples da base de dados em dois grupos, estratégia conhecida como *hold-out*, é a validação cruzada. Dentre as variações dessa técnica, a mais comum é a validação cruzada K-Fold, onde K representa o número de grupos ou *folds* de mesmo tamanho em que as amostras serão divididas. A partir disso, K modelos são treinados, cada um utilizando uma combinação diferente de K - 1 *folds* para treinamento e um

para teste. Assim, em vez de apenas uma, obtém-se K medidas de desempenho do modelo (Muller; Guido, 2016). Na Figura 6, pode ser observado como fica a divisão da base de dados quando utilizados 5 *fold*s.

Figura 6 – Exemplo de validação cruzada K-Fold com K=5



Fonte: Muller e Guido (2016).

Com essa abordagem, todas as amostras são usadas tanto para treino quanto para teste, o que fornece uma visão sobre o quão sensível o modelo é à base de dados de treinamento e sobre como desempenha em cenários ideais e adversos ao ser exposto à novos dados. Por outro lado, como é necessário repetir o processo de treinamento e avaliação K vezes, são necessários mais recursos computacionais e tempo (Muller; Guido, 2016).

Entretanto, esse método pode não ser suficiente caso as classes não estejam representadas igualmente ao longo da base ou a quantidade de amostras de uma classe seja maior do que da outra, visto que ao realizar a divisão em *fold*s é possível que ocorra uma concentração maior de dados de uma classe em um deles, o que pode impactar na capacidade de generalização do modelo (Muller; Guido, 2016).

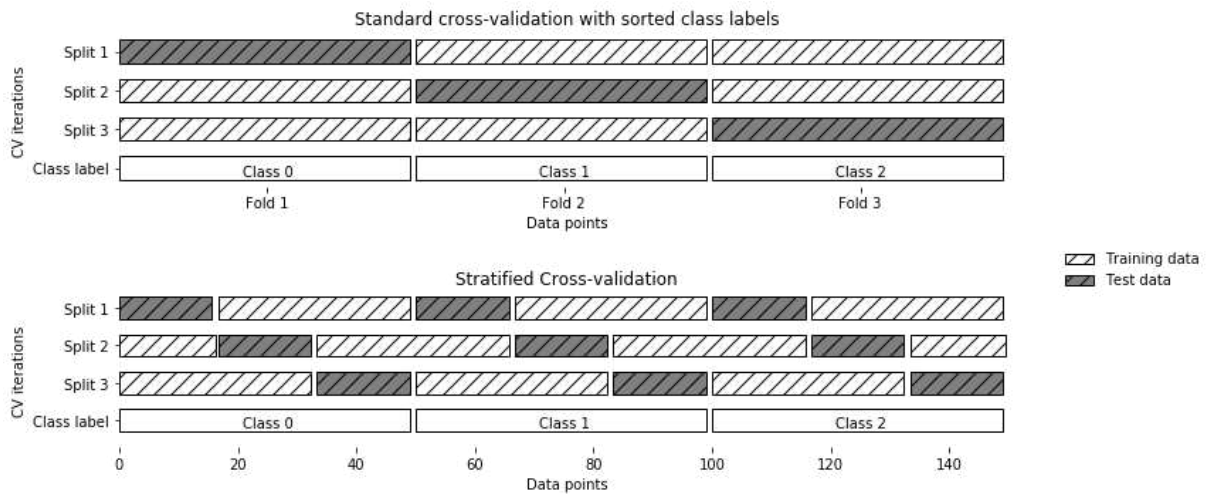
Conforme Muller e Guido (2016), nesse contexto, a validação cruzada estratificada K-Fold é uma alternativa para garantir estimativas de desempenho mais abrangentes, já que nela a proporção de amostras existentes de cada classe é mantida entre os *fold*s, assim, o modelo é exposto à mesma proporção entre as classes tanto no treinamento quanto no teste em todos os *fold*s.

A Figura 7 ilustra uma comparação entre esses dois tipos de validação cruzada em um caso em que as amostras estão ordenadas por classe. Na validação cruzada K-Fold, cada grupo iria conter informações somente sobre uma classe específica, o que resultaria no modelo ser avaliado em exemplos de uma classe que não foi vista durante o treinamento, levando a um desempenho pior se comparado à versão estratificada, na qual o modelo seria treinado e avaliado com conjuntos que mantém a proporção das classes da base original (Muller; Guido, 2016).

2.4.2 Grid Search Cross Validation

A maioria dos métodos de aprendizado de máquina possui hiperparâmetros que podem ser configurados a fim de otimizar o seu comportamento para a problemática

Figura 7 – Comparação entre validação cruzada K-Fold e estratificada K-Fold



Fonte: Muller e Guido (2016).

que pretende resolver. Encontrar tal configuração ótima não é uma tarefa simples, mas é importante para grande parte dos modelos e bases de dados, sendo o *grid search* uma estratégia comum para alcançar esse objetivo. Essa técnica consiste em definir valores de interesse para cada um dos parâmetros a serem otimizados e testar todas as possíveis combinações (Muller; Guido, 2016).

Sendo assim, é preciso treinar e avaliar modelos com cada uma das configurações e para isso, dividir a base de dados em treinamento e teste não é mais suficiente, visto que é necessário mais um grupo de amostras, comumente chamado de base de validação, para comparar o desempenho dos modelos com parâmetros diferentes. Após escolhido o modelo com a melhor performance, a base de teste é usada para uma avaliação final (Muller; Guido, 2016).

Além disso, para evitar que a escolha dos hiperparâmetros seja influenciada pela divisão da base de dados, é possível aplicar a validação cruzada estratificada K-Fold para realizar K avaliações de cada combinação. Ademais, pode-se englobar esse processo em outra validação cruzada de mesmo tipo, de forma que existirá um loop exterior separando os grupos de treinamento e teste e para cada iteração dele, é utilizada a validação cruzada nos dados de treino para avaliar os hiperparâmetros. Isso garante uma avaliação mais completa do modelo em toda a base de dados (Muller; Guido, 2016).

2.4.3 Métricas de Avaliação para Classificação Binária

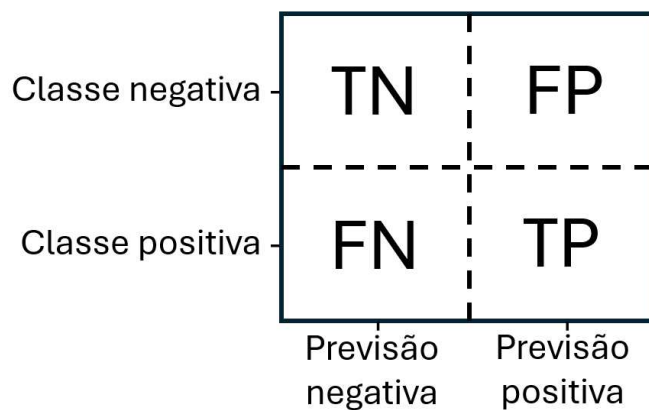
Em problemas de classificação binária, as classes são geralmente denominadas como positiva e negativa, sendo a positiva a que deseja-se identificar prioritariamente (Muller; Guido, 2016). No caso da detecção de quedas, a ocorrência

de uma queda é a classe positiva, uma vez que é o evento de maior interesse.

Segundo Muller e Guido (2016), uma das maneiras mais representativas de demonstrar resultados de uma classificação binária é utilizando matrizes de confusão. Nelas, as colunas correspondem à classe prevista e as linhas aos resultados esperados. A Figura 8 demonstra um exemplo genérico de uma matriz de confusão em que é possível visualizar em qual quadrante se localiza cada um dos quatro possíveis tipos de resultado, sendo eles:

- Falso Negativo (False Negative - FN): são os casos da classe positiva que foram classificados erroneamente como da negativa pelo modelo, representam quedas que foram identificadas como atividades do dia a dia;
- Falso Positivo (False Positive - FP): são os casos da classe negativa que foram classificados erroneamente como da positiva pelo modelo, representam atividades do dia a dia que foram identificadas como quedas;
- Verdadeiro Negativo (True Negative - TN): são os casos da classe negativa que foram classificados corretamente pelo modelo, representam atividades do dia a dia que foram identificadas como atividades do dia a dia;
- Verdadeiro Positivo (True Positive - TP): são os casos da classe positiva que foram classificados corretamente pelo modelo, representam quedas que foram identificadas como quedas.

Figura 8 – Matriz de confusão genérica



Fonte: Elaborado pelo autor (2024).

Apesar da matriz de confusão ser informativa, é possível resumir os resultados apresentados nela por meio de métricas quantitativas (Muller; Guido, 2016). Sendo algumas delas a revocação ou *recall*, calculada a partir da Equação (1), que representa a porcentagem das instâncias positivas que foram classificadas corretamente:

$$recall = \frac{TP}{TP + FN} \quad (1)$$

A precisão ou *precision*, calculada pela Equação (2), que expressa a proporção de amostras previstas como positivas que realmente são da classe positiva:

$$precision = \frac{TP}{TP + FP} \quad (2)$$

A acurácia ou *accuracy*, que demonstra o percentual de predições corretas em relação à quantidade total de amostras, encontrada a partir da Equação (3):

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

E o *f1-score*, o qual pode ser uma métrica mais representativa do que a acurácia em bases de dados desbalanceadas, pois combina o recall e a precisão, sendo calculado por meio da Equação (4):

$$f1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 2 \cdot \frac{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (4)$$

No contexto de detecção de quedas, o *recall* representa a porcentagem de quedas que foram detectadas entre todas que ocorreram, enquanto a precisão indica o percentual de instâncias classificadas como quedas que, de fato, eram quedas. Já o *f1-score*, por combinar essas duas métricas, possibilita a análise do equilíbrio entre a correta classificação de quedas e a emissão de falsos alarmes. A acurácia, por outro lado, pode não ser significativa nesse cenário, visto que a quantidade de quedas tende a ser muito menor do que a de atividades rotineiras, o que pode levar a uma alta acurácia, mesmo com uma baixa identificação de quedas.

2.5 TRABALHOS RELACIONADOS

A detecção e prevenção de quedas são tópicos de interesse na área da saúde, portanto, nos últimos anos, diversos estudos foram desenvolvidos sobre tecnologias relacionadas a esses assuntos. Em relação aos algoritmos de detecção de quedas, as pesquisas têm concentrado cada vez mais nas técnicas de aprendizado de máquina (Xu; Zhou; Zhu, 2018).

Wixley (2023) buscou desenvolver um sistema de detecção de quedas a partir da combinação de um sensor de batimentos cardíacos e um acelerômetro de 3 eixos posicionados na altura do peito. Para a escolha do algoritmo, foram analisados diversos modelos de aprendizado de máquina e de aprendizado profundo. Sendo que o que obteve o melhor desempenho geral nos dados de teste foi o ResNet152, arquitetura de aprendizado profundo baseada em redes neurais convolucionais de 152 camadas, com 87,27% de recall. Já considerando apenas as técnicas de aprendizado de máquina comum, o Bernoulli Naive Bayes, método baseado no Teorema de Bayes para dados que seguem a distribuição de Bernoulli, obteve 58,43% de recall.

Martínez-Villaseñor *et al.* (2019) tiveram como objetivo desenvolver uma base de dados com informações de tipos variados de sensores, incluindo cinco IMUs posicionadas em diferentes pontos do corpo, câmeras, sensores de infravermelho e de atividade cerebral. Para validar essa base, foi comparado, por meio de validação cruzada K-Fold com dez folds, o desempenho de quatro métodos de aprendizado de máquina na classificação de cinco variações de quedas e seis atividades rotineiras utilizando tanto os dados de cada tipo de sensor individualmente quanto combinados. As análises foram feitas com base na performance em relação à métrica f1-score, sendo o melhor resultado obtido de 70,44% de f1-score pelo MLP a partir dos dados combinados das IMUs, câmeras e sensor de ondas cerebrais. Quando utilizado apenas as IMUs, alcançou-se 70,31% de f1-score e 69,40% de recall. Vale ressaltar que foram usados os dados das cinco IMUs em conjunto, não sendo feita a análise individual de cada posicionamento.

Özdemir (2016) utilizou a validação cruzada K-Fold com 10 folds para comparar a performance de seis métodos de aprendizado de máquina na detecção de quedas baseada em dados de seis IMUs distribuídas entre cabeça, peito, cintura, punho direito, coxa direita e tornozelo direito. A partir de uma base de dados composta por 2520 amostras, sendo 1400 quedas, foram avaliadas todas as 63 possíveis combinações de agrupamento desses sensores, de forma que o melhor desempenho para um único sensor foi de 99,87% de acurácia e 99,96% de recall utilizando os dados da cintura e o método KNN, com K igual a 7. Já entre todas as combinações, foi alcançando 100% de recall e 99,94% de acurácia pelo mesmo método usando a combinação dos dados do tornozelo direito, punho direito, peito e cabeça.

3 MATERIAIS E MÉTODOS

Neste capítulo, será discorrido sobre os materiais e métodos utilizados para o desenvolvimento do trabalho. Primeiramente, serão explicitados os motivos da escolha da base de dados e como ela foi preparada para essa pesquisa. Na sequência, serão apresentados os métodos e como foram utilizados. Para finalizar, será explicada a implementação das técnicas para treinamento e avaliação dos modelos, assim como para a otimização dos hiperparâmetros de cada método.

3.1 BASE DE DADOS

Existem diversas bases de dados desenvolvidas para estudos relacionados à detecção de quedas, a maioria delas com quedas simuladas por pessoas jovens em ambientes controlados, visto o risco envolvido na atividade (Casilari; Silva, 2022). Dentre as disponíveis publicamente, foi considerada a utilização de KFall¹, UR Fall², UMAFall³ e UP-Fall⁴. As duas primeiras foram descartadas visto que foram construídas utilizando dados de apenas uma unidade de medição inercial posicionada na região da cintura, impossibilitando a comparação entre diferentes posicionamentos dos sensores. Já as outras duas possuem dados de cinco posições diferentes, porém, a quarta disponibiliza, de forma estruturada, além dos dados brutos de cada sensor, características já calculadas a partir de tais dados, isto é, atributos descritivos extraídos a partir das séries temporais medidas pelos sensores.

Sendo assim, a base de dados escolhida foi a UP-Fall. Para a sua construção, foram utilizadas unidades de medição inercial Mbiéntlab MetaSensor para coletar dados de acelerômetros e giroscópios de 3 eixos posicionadas no tornozelo, bolso, cintura, pescoço e punho de 17 indivíduos que simularam seis atividades do dia a dia, andar, ficar em pé, pegar um objeto, sentar, deitar e pular, e cinco tipos de quedas, para frente usando as mãos, para frente de joelhos, para trás, sentando em uma cadeira inexistente e de lado (Martínez-Villaseñor *et al.*, 2019). É importante citar que também existem dados de outros tipos de sensores nessa base de dados, porém esses não serão utilizados, visto o foco desse estudo em acelerômetros e giroscópios.

Além dos dados brutos de aceleração e velocidade angular, a UP-Fall também disponibiliza características calculadas a partir de tais dados no domínio do tempo e no domínio da frequência, mostradas no Quadro 1, para cada sensor calculadas para três janelamentos, um, dois e três segundos, todos com sobreposição de 50% dos dados em

¹ Disponível em: <https://sites.google.com/view/kfalldataset>

² Disponível em: <http://fenix.ur.edu.pl/~mkepski/ds/uf.html>

³ Disponível em: http://webpersonal.uma.es/de/ECASILARI/Fall_ADL_Traces/UMA_FALL_ADL_dataset.html

⁴ Disponível em: <https://sites.google.com/up.edu.mx/har-up/>

cada janela. Nas análises de detecção de queda utilizando algoritmos de aprendizado de máquina feitas pelos autores da base de dados, não houve diferenças significativas nos resultados obtidos entre os diferentes janelamentos (Martínez-Villaseñor *et al.*, 2019). Assim, visando a maior quantidade de dados, foi escolhido o de um segundo para esse trabalho, o que levou a uma base com as características composta por 32294 amostras, sendo 31339 referentes a atividades do dia a dia e 955 a quedas, o que corresponde, respectivamente, a 97% e 3% dos dados.

Quadro 1 – Características disponíveis na base de dados

Característica	Domínio
Média	Tempo
Desvio padrão	Tempo
Erro médio quadrático	Tempo
Amplitude máxima	Tempo
Amplitude mínima	Tempo
Mediana	Tempo
Número de intersecções no zero	Tempo
Assimetria	Tempo
Curtose	Tempo
Primeiro quartil	Tempo
Terceiro quartil	Tempo
Autocorrelação	Tempo
Frequência média	Frequência
Frequência mediana	Frequência
Entropia	Frequência
Energia	Frequência
Frequência principal	Frequência
Centróide espectral	Frequência

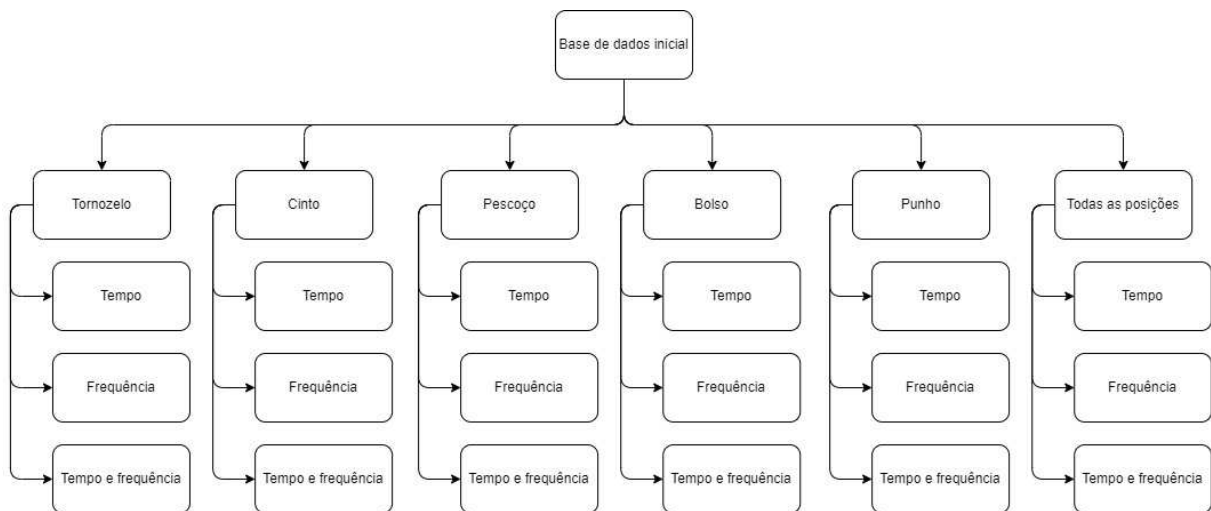
Fonte: Elaborado pelo autor (2024).

Nesse contexto, cada amostra contém tais características para cada um dos sensores, de forma que as no domínio do tempo são calculadas a partir de todos os sinais obtidos pelo respectivo sensor em cada janela de um segundo, enquanto as no domínio da frequência são extraídas dos valores absolutos do resultado da Transformada Rápida de Fourier aplicada sobre os sinais do mesmo janelamento de um segundo (Martínez-Villaseñor *et al.*, 2019).

3.1.1 Preparação dos Dados

A base inicial foi separada, como mostrado na Figura 9, em 18 sub-bases, de forma a ter: uma base para cada posicionamento dos sensores e uma composta de todos os sensores. Adicionalmente, para cada uma dessas, foram selecionadas as características a fim de obter uma base apenas com as informações no domínio do tempo, uma somente com as no domínio da frequência e outra com ambas. Apesar das amostras possuírem a informação de qual é a atividade ou queda sendo executada, optou-se por realizar o estudo considerando uma classificação binária entre queda ou não, visto que o foco não é comparar a performance dos métodos quanto a capacidade de classificar entre tipos de atividades e quedas, mas sim de detectá-las.

Figura 9 – Divisão da base de dados inicial



Fonte: Elaborado pelo autor (2024).

3.2 MÉTODOS ESCOLHIDOS

O critério principal para seleção dos métodos a serem comparados foi a recorrência da utilização de cada um em aplicações de detecção de queda na literatura, sendo eles KNN, RF e MLP e SVM. Todos foram aplicados por meio da biblioteca scikit-learn na versão 1.5.2 para a linguagem Python visando a facilidade de uso e padronização dos códigos. Para a aplicação das técnicas de otimização dos hiperparâmetros e de avaliação de desempenho, também foram usadas implementações da mesma biblioteca.

Para o primeiro método, foi aplicada a classe `KNeighborsClassifier`. Já a implementação do segundo foi feita por meio da classe `RandomForestClassifier`, que teve seu parâmetro `oob_score` modificado para utilizar o cálculo de recall na avaliação

do desempenho de cada árvore em amostras não utilizadas para o seu treinamento. Para o terceiro método, foi utilizada a classe SVC. Por fim, o quarto foi implementado com a classe MLPClassifier, ajustando o número máximo de iterações permitidas até a convergência do resolvidor para 1000, por meio do hiperparâmetro *max_iter*.

3.2.1 Treinamento dos Modelos e Otimização de Hiperparâmetros

Considerando o desbalanceamento entre as amostras de quedas e de atividades comuns e visando realizar uma análise mais confiável estatisticamente, a divisão dos dados para treinamento e teste dos modelos foi feita utilizando a técnica Stratified K-Fold a fim de garantir que todas as amostras fossem usadas tanto para treino quanto para teste, mantendo a proporção de cada classe entre os K grupos. A implementação foi feita por meio da classe StratifiedKFold, sendo 5 o número de *folds* escolhido, de forma que fosse obtido equilíbrio entre o número de divisões e a quantidade de amostras de quedas por grupo, visto que se a base de dados fosse dividida muitas vezes, haveria poucos exemplos de quedas por divisão, o que poderia impactar na capacidade dos modelos de se adequarem ao problema.

Diferentemente do que foi feito para os hiperparâmetros citados anteriormente, que foram especificados com valores fixos, para outras características de cada método foram definidos espaços de busca compostos por possíveis valores, conforme pode ser observado no Quadro 2. Tais espaços foram utilizados para realizar a otimização do respectivo hiperparâmetro com a técnica Grid Search Cross Validation, que foi implementada a partir da classe GridSearchCV, configurando o parâmetro *scoring* para usar a métrica recall na avaliação e escolha dos modelos, visto que, devido à natureza da aplicação, é crítico que o máximo de verdadeiros positivos, que representam as quedas, sejam detectados.

Também foi ajustado o parâmetro *cv* para 2, definindo a técnica interna de validação cruzada como do tipo Stratified K-Fold com 2 *folds*. Isso determina que os dados passados a classe serão divididos em dois grupos, sendo que a cada *fold* um é usado para treinamento e o outro para validação, utilizado para a escolha dos hiperparâmetros. Como já há uma divisão externa das amostras pela mesma técnica, tem-se que para cada um dos 5 *folds*, metade dos dados de treino, ou seja, 2/5 da base de dados, é efetivamente utilizada para treinamento dos modelos e a outra metade para validação, enquanto o quinto grupo, separado para teste, é usado para avaliar e extrair as métricas de desempenho do modelo naquele *fold*.

Em relação aos hiperparâmetros otimizados, para o KNN, foram escolhidos o *n_neighbors*, que determina o número de pontos vizinhos envolvidos na tomada de decisão, e o *weights*, referente a como é decidido o peso dado a cada ponto. No caso do RF, foi decidido pelo *n_estimators* e pelo *max_features*. O primeiro representa a quantidade de árvores de decisão utilizadas para compor o modelo, enquanto o

Quadro 2 – Espaço de busca dos hiperparâmetros

Método	Hiperparâmetro	Espaço de busca
KNN	n_neighbors	[3, 5, 7, 9, 11]
	weights	[uniform, distance]
RF	n_estimators	[50, 100, 200]
	max_features	[sqrt, log2]
MLP	hidden_layer_sizes	[(8,), (8, 8), (8, 16), (8, 32), (8, 64), (8, 128), (16,), ..., (32,), ..., (64,) ..., (128, 128)]
SVM	class_weight	{0: 1, 1: 10}, {0: 1, 1: 20}, {0: 1, 1: 1}

Fonte: Elaborado pelo autor (2024).

segundo especifica o número máximo de características consideradas em um nó ou folha, limitando a escolha de variáveis avaliadas a cada divisão.

Quanto ao SVM, foi otimizado somente o *class_weight*, que define o peso de cada classe. Já para o MLP, foi selecionado apenas o *hidden_layer_sizes*, o qual estabelece a estrutura de camadas internas e de neurônios em cada uma delas. O espaço de busca para esse parâmetro é composto por combinações de uma ou duas camadas de 8, 16, 32, 64 e 128 neurônios, sendo a primeira camada sempre menor que a segunda.

No caso dos hiperparâmetros que não foram nem definidos com um valor fixo nem otimizados, assim como os parâmetros das classes referentes às técnicas de treinamento e otimização que não foram citados, foi utilizado o valor padrão determinado pela própria biblioteca. Vale ressaltar que não é objetivo desse trabalho definir os hiperparâmetros ótimos, o foco é apenas aumentar o desempenho em cada *fold* a fim de comparar os melhores resultados de cada método com os hiperparâmetros utilizados.

4 RESULTADOS

Neste capítulo, serão apresentados e discutidos os resultados de desempenho de cada um dos métodos analisados. Inicialmente, serão expostas as configurações de hiperparâmetros de cada método mais recorrentes durante as etapas de treinamento e avaliação por validação cruzada. Em seguida, serão comparadas as métricas obtidas a partir da matriz de confusão considerando o uso das características no domínio do tempo, da frequência e a combinação de ambas. Na sequência, serão analisados os melhores resultados alcançados para cada posicionamento dos sensores, avaliando a viabilidade prática de aplicá-los em um sistema de detecção de quedas. Finalmente, será discutido em quais atividades os métodos apresentaram mais dificuldades na identificação de quedas.

Com a utilização da técnica Grid Search Cross Validation, a partir dos espaços de busca definidos no Quadro 2, obteve-se uma configuração de hiperparâmetros para cada um dos cinco folds da validação cruzada estratificada K-Fold realizados para cada uma das 18 combinações de posicionamento dos sensores e características utilizadas, totalizando 90 configurações para cada método. Sendo assim, a configuração mais recorrente de cada hiperparâmetro é disponibilizada no Quadro 3.

Quadro 3 – Hiperparâmetros mais frequentes

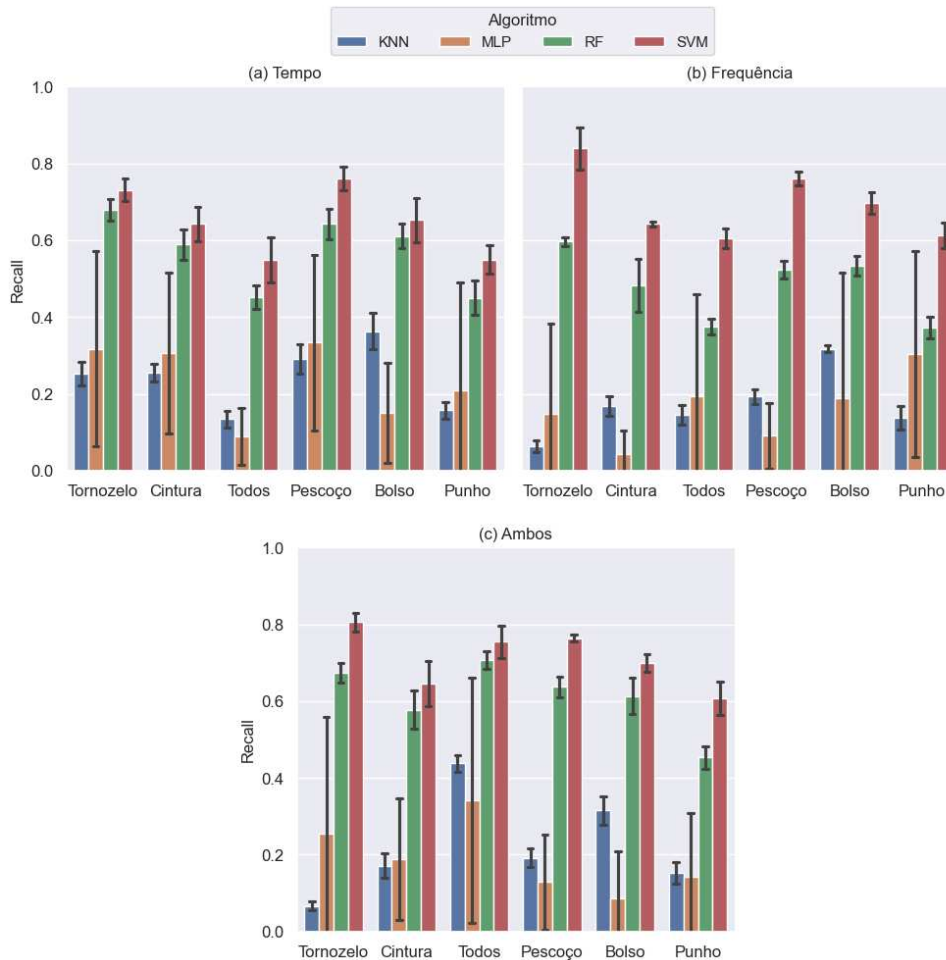
Método	Hiperparâmetro	Configuração
KNN	n_neighbors	3
	weights	distance
RF	n_estimators	200
	max_features	sqrt
MLP	hidden_layer_sizes	(128,)
SVM	class_weight	{0: 1, 1: 20}

Fonte: Elaborado pelo autor (2024).

Nesse contexto, foi obtido a média e o desvio padrão das métricas recall, f1-score, acurácia e precisão nos 90 folds realizados com cada método para cada conjunto de características a fim de verificar a influência do domínio do tempo e da frequência na detecção de quedas, além do desempenho dos métodos. De forma abrangente, é possível observar na Figura 10 que não há uma correlação direta entre o recall obtido em cada posicionamento dos sensores e as características utilizadas, visto que em algumas situações, como o KNN aplicado aos dados do tornozelo e o MLP aos do

pescoço, o desempenho apenas no tempo, ainda que não satisfatório à aplicação, é superior ao da frequência e ao de ambos em conjunto e em outras, como o KNN com as amostras de todos os sensores, o resultado com ambos os domínios superou os individuais.

Figura 10 – Recall por posicionamento e características



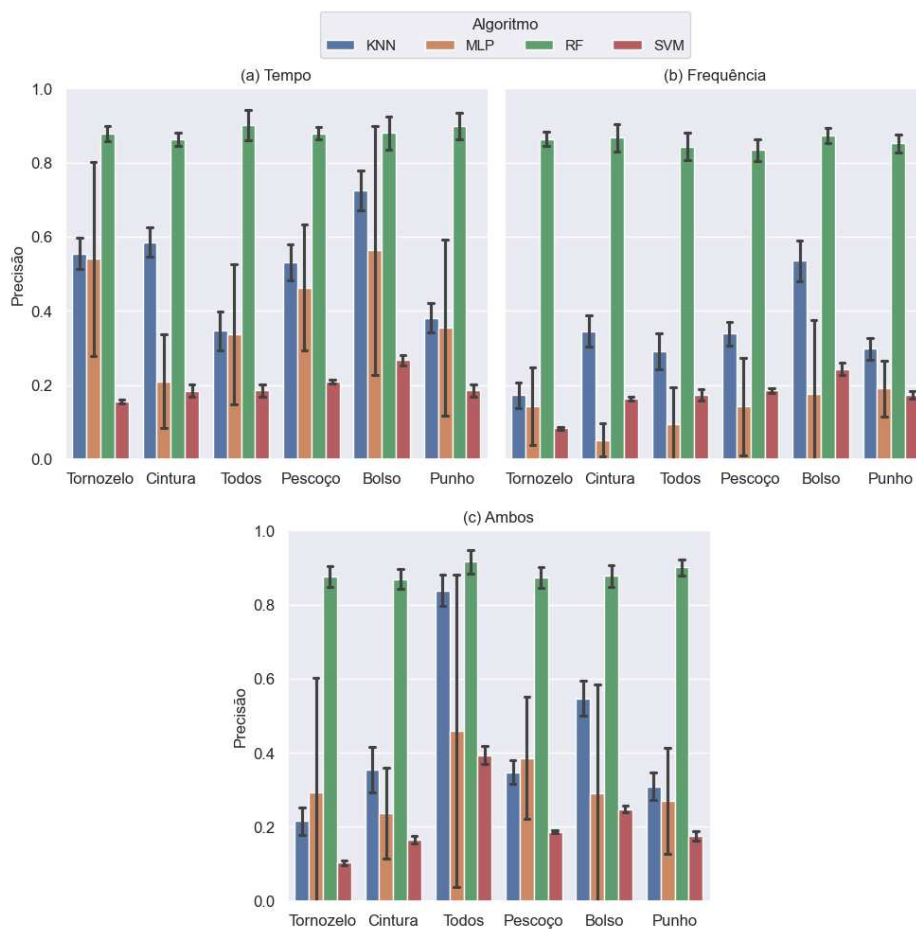
Fonte: Elaborado pelo autor (2024).

Analisando apenas o recall, o RF e o SVM foram os métodos com os melhores desempenhos, de forma que o primeiro conseguiu ultrapassar ou ficar próximo da marca de 60% na maioria dos casos, enquanto o segundo consistentemente teve resultados entre 60% e 80%, inclusive superando 80% em duas situações. Por outro lado, os outros não conseguiram generalizar o problema da mesma forma, especialmente o MLP, que, durante a avaliação, não identificou nenhuma das quedas em alguns folds, levando a um recall de 0% e, por consequência, aumentando o desvio padrão desses cenários. Sendo assim, visando somente a detecção de quedas, o RF e o SVM seriam os algoritmos mais adequados para uma aplicação com essa finalidade.

A Figura 11, da mesma forma que a anterior, expõe a precisão dos métodos em

cada posição por conjunto de características. Nessa métrica, é possível observar que os resultados obtidos a partir das características no domínio do tempo, demonstrados na Figura 11a, foram similares ou superiores aos obtidos com base nas no domínio da frequência, que podem ser vistos na Figura 11b, assim como em relação aos que utilizam os dois domínios, exibidos na Figura 11c, exceto no caso do KNN utilizando os dados de todos os sensores. Isso indica que as características no domínio da frequência podem estar impactando negativamente na precisão dos modelos por meio da redução do número de TPs ou do aumento de FPs.

Figura 11 – Precisão por posicionamento e características



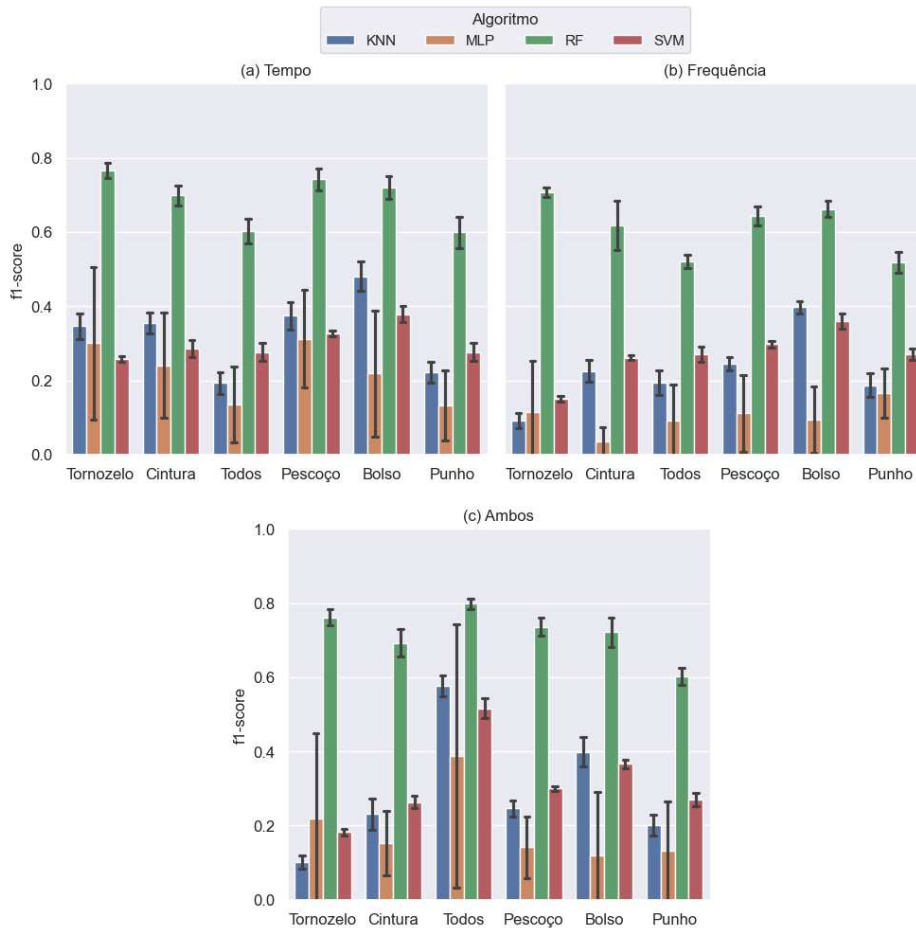
Fonte: Elaborado pelo autor (2024).

Em relação ao desempenho de cada método, o RF também apresentou bons resultados, chegando próximo a 90% de precisão em todos os cenários, enquanto o SVM foi o pior na maioria dos casos. Em contrapartida, o KNN apresentou melhores resultados nessa métrica quando comparada ao recall e novamente, por não detectar TPs em alguns folds, o MLP apresentou um alto desvio padrão em algumas situações.

A baixa precisão unida ao alto recall visto na Figura 10 sugere que o SVM esteja enviesado a classificar novas amostras como quedas, levando tanto a um

alto número de TPs quanto de FPs. Por outro lado, o menor recall e maior precisão apresentados pelo KNN podem indicar que há um viés nesse modelo em identificar instâncias como não-quedas, diminuindo TPs e FPs. Essa análise pode ser embasada pela métrica f1-score, disponibilizada na Figura 12 seguindo o mesmo formato das métricas anteriores.

Figura 12 – f1-score por posicionamento e características



Fonte: Elaborado pelo autor (2024).

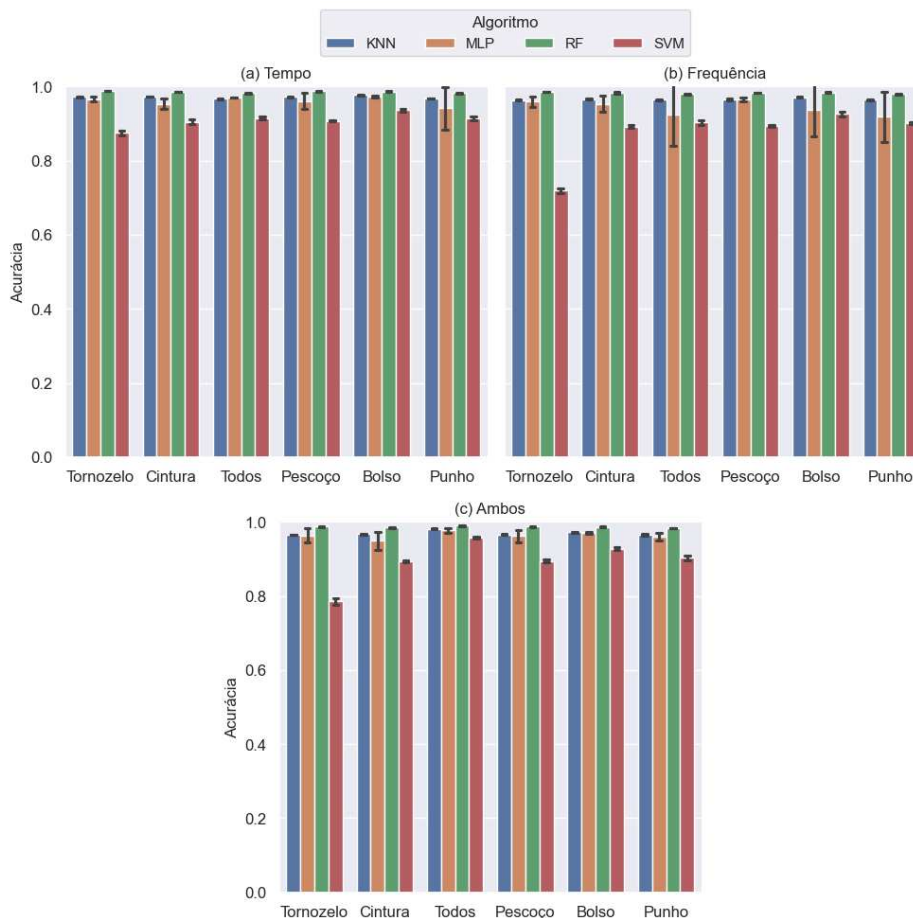
O f1-score, por considerar simultaneamente o recall e a precisão em seu cálculo, é uma métrica apropriada para analisar o equilíbrio entre esses dois aspectos. Sendo assim, é possível verificar na Figura 12 que o RF apresentou o melhor desempenho, alcançando entre 60% e 80% na maioria dos cenários avaliados, enquanto os outros métodos não atingiram a faixa de 60% em nenhum momento. Dessa forma, do ponto de vista de uma sistema de detecção de quedas, o método mais apropriado seria o RF, já que apresenta uma alta taxa de detecção de quedas, ao mesmo tempo que mantém um número baixo de falsos alarmes, que poderiam comprometer a confiança no sistema.

Igual ocorreu com a precisão, os resultados para o f1-score com as

características no domínio do tempo, exibidos na Figura 12a, foram próximos ou superiores aos a partir das no domínio da frequência, apresentados na Figura 12b. Já o f1-score alcançado com a combinação dos dois domínios, demonstrado na Figura 12c, ultrapassou os resultados individuais do tempo com todos os métodos quando utilizados os dados de todos os sensores, porém foi inferior ou similar nas outras situações.

A Figura 13 ilustra a acurácia alcançada em cada posição dos sensores por cada método com cada conjunto de características. Observa-se que, exceto para o SVM com os dados do tornozelo, todos os cenários atingiram resultados semelhantes independente das características envolvidas, logo não é possível visualizar nenhuma influência de algum dos domínios nessa métrica. Da mesma forma, não há discrepâncias entre o desempenho dos métodos, de forma que em apenas dois cenários não foi ultrapassada a faixa de 80% de acurácia, sendo a maioria acima de 90%. Entretanto, a análise isolada da acurácia pode levar a conclusões enganosas em situações que utilizam bases de dados desbalanceadas, visto que o modelo pode ter uma alta acurácia apenas classificando todas as amostras como sendo da classe majoritária (Varoquaux; Colliot, 2023).

Figura 13 – Acurácia por posicionamento e características



Fonte: Elaborado pelo autor (2024).

Visto a influência de cada configuração de características e a comparação geral do desempenho dos métodos, os resultados por posição e característica do método com o melhor f1-score em cada cenário foram compilados na Tabela 1 a fim de avaliar o posicionamento dos sensores em relação ao melhor desempenho obtido e à viabilidade de sua aplicação em sistemas de detecção de quedas. É importante ressaltar que não foi utilizado o melhor recall para a determinação dos resultados exibidos, pois, olhando apenas essa métrica, poderia ser escolhido um método que teve um viés de classificar amostras como quedas.

Como observado na Figura 12, o RF apresentou o melhor f1-score em todas as situações, portanto são expostos na Tabela 1 a média e o desvio padrão dos resultados obtidos nos 5 folds executados com esse método para cada combinação de posição dos sensores e características.

Tabela 1 – Melhores resultados

Posição	Características	Recall (%)	Precisão (%)	F1 (%)	Acurácia (%)
Tornozelo	Tempo	67,9 ± 2,9	87,7 ± 2,0	76,6 ± 2,0	98,8 ± 0,1
	Frequência	59,7 ± 1,2	86,4 ± 2,0	70,6 ± 1,3	98,5 ± 0,1
	Ambos	67,4 ± 2,6	87,6 ± 2,9	76,2 ± 2,1	98,7 ± 0,1
Bolso	Tempo	61,1 ± 3,1	88,0 ± 4,6	72,1 ± 3,0	98,6 ± 0,2
	Frequência	53,3 ± 2,5	87,3 ± 2,1	66,2 ± 2,1	98,4 ± 0,1
	Ambos	61,3 ± 4,7	87,7 ± 3,0	72,1 ± 4,0	98,6 ± 0,2
Cintura	Tempo	58,8 ± 3,9	86,2 ± 1,9	69,9 ± 2,7	98,5 ± 0,1
	Frequência	48,2 ± 6,9	86,7 ± 3,8	61,8 ± 6,7	98,2 ± 0,2
	Ambos	57,7 ± 4,9	86,8 ± 2,7	69,2 ± 3,7	98,5 ± 0,1
Punho	Tempo	45,0 ± 4,5	89,8 ± 3,5	59,9 ± 4,3	98,2 ± 0,1
	Frequência	37,3 ± 2,7	85,2 ± 2,4	51,8 ± 2,7	97,9 ± 0,1
	Ambos	45,3 ± 3,0	90,0 ± 2,2	60,2 ± 2,3	98,2 ± 0,1
Pescoço	Tempo	64,2 ± 3,9	87,9 ± 1,6	74,2 ± 3,0	98,7 ± 0,1
	Frequência	52,5 ± 2,3	83,4 ± 2,9	64,3 ± 2,6	98,3 ± 0,1
	Ambos	63,7 ± 2,6	87,2 ± 2,8	73,6 ± 2,5	98,6 ± 0,1
Todas	Tempo	45,2 ± 3,0	90,0 ± 4,2	60,2 ± 3,3	98,2 ± 0,1
	Frequência	37,6 ± 2,0	84,3 ± 3,7	51,9 ± 1,8	97,9 ± 0,1
	Ambos	70,7 ± 2,2	91,6 ± 3,2	79,8 ± 1,5	98,9 ± 0,1

Fonte: Elaborado pelo autor (2024).

Assim, é possível verificar que o melhor desempenho para as quatro métricas foi alcançado quando utilizados os dados de todos os sensores e a combinação dos dois domínios das características, de forma que aproximadamente sete em cada dez quedas seriam detectadas e apenas uma em dez detecções não seria proveniente de

uma queda. Entretanto, para a implementação de um sistema com essa configuração são necessários que os sensores sejam aplicados nas cinco posições, o que além de aumentar a complexidade e o custo de implementação, pode afetar negativamente a usabilidade, já que usar tal sistema pode ser desconfortável.

Logo, a utilização de apenas um posicionamento de sensores pode facilitar o uso de um dispositivo para detecção de quedas. Nesse sentido, posicionar uma IMU no tornozelo e utilizar ou as características do tempo exclusivamente ou em conjunto com as da frequência é uma opção que leva à diminuição de menos de 5% nas métricas, assim como, com as mesmas opções de características, posicionar no pescoço ou no bolso diminui o desempenho em menos de 10%. Diferentemente do que foi concluído por Özdemir (2016), a cintura não foi o local que trouxe os melhores resultados, apresentando um desempenho aproximadamente 10% inferior do que o tornozelo e sendo superior apenas ao punho.

Finalmente, para verificar em quais atividades os métodos apresentaram mais dificuldades para classificar corretamente, calculou-se a porcentagem média de erros de cada um em cada atividade nos 15 *folds* de validação cruzada executados para cada posicionamento dos sensores analisado, agrupando os três conjuntos de características. Primeiramente, são elencadas no Quadro 4 as atividades disponíveis na base de dados utilizada, juntamente com os respectivos identificadores que serão usados para identificá-las nas tabelas subsequentes.

Quadro 4 – Lista de atividades

Identificador	Atividade
1	Queda para frente apoiando com as mãos
2	Queda para frente apoiando com os joelhos
3	Queda para trás
4	Queda para o lado
5	Queda sentando em uma cadeira inexistente
6	Andar
7	Ficar parado em pé
8	Sentar
9	Pegar um objeto
10	Pular
11	Deitar

Fonte: Elaborado pelo autor (2024).

A Tabela 2 mostra a porcentagem média de erros que o método KNN apresentou em cada atividade para cada posicionamento dos sensores. É possível

observar que, de forma geral, ocorreram poucos erros na classificação de atividades do dia a dia como não-quedas, visto que a maior porcentagem foi de 1,63% quando a ação executada era pegar um objeto e os dados utilizados eram da IMU posicionada no bolso. Mesmo tratando-se de valores pequenos, a atividade rotineira que apresentou maior porcentagem de identificação incorreta, independente do sensor empregado, foi a de pegar um objeto. Já em relação à identificação das quedas, o método não teve o mesmo desempenho, uma vez que classificou incorretamente mais de 15% de todos os tipos de quedas na maioria das situações, não existindo uma discrepância considerável entre as variedades de quedas.

Tabela 2 – Porcentagem média de erros de identificação do KNN

Atividade	Tornozelo	Bolso	Cintura	Punho	Pescoço	Todos
1	17,09%	15,38%	16,94%	17,17%	16,47%	15,66%
2	17,37%	15,48%	16,53%	17,87%	16,08%	16,28%
3	17,51%	11,05%	15,15%	16,56%	16,14%	14,41%
4	16,45%	13,64%	16,17%	17,12%	12,94%	14,99%
5	18,46%	12,21%	15,71%	16,61%	15,65%	15,01%
6	0,41%	0,13%	0,34%	0,38%	0,14%	0,28%
7	0,03%	0,05%	0,05%	0,11%	0,06%	0,08%
8	0,03%	0,03%	0,04%	0,07%	0,03%	0,05%
9	0,76%	1,63%	0,76%	0,21%	1,62%	0,25%
10	0,37%	0,72%	0,64%	0,57%	1,31%	0,37%
11	0,07%	0,06%	0,06%	0,08%	0,08%	0,08%

Fonte: Elaborado pelo autor (2024).

De maneira similar, a Tabela 3 exibe a porcentagem média de erros do RF, sendo possível constatar que ocorreram ainda menos erros ao identificar atividades do dia a dia e que o ato de pegar um objeto continua sendo o mais confundido com uma queda. A porcentagem de quedas classificadas incorretamente também diminuiu, de forma que apenas as médias de erro de quando utilizados os dados do punho ficaram consistentemente acima de 10%. Apesar da quantidade de erros na identificação das quedas variar dependendo do posicionamento dos sensores analisado, a queda sentando em uma cadeira inexistente demonstrou ser a mais difícil de ser detectada na maioria dos casos.

A Tabela 4, seguindo o mesmo formato das anteriores, expõe a porcentagem média de erros do MLP, possibilitando visualizar que, mais um vez, a ação de pegar um objeto é a qual o método teve mais dificuldade para identificar corretamente, nesse caso chegando a 2,76% de média de erro quando usados os dados da cintura. Quanto aos tipos de queda, a porcentagem média fica em torno de 16% de erro, com os três maiores valores ocorrendo com os dados da IMU posicionada no bolso referentes às

Tabela 3 – Porcentagem média de erros de identificação do RF

Atividade	Tornozelo	Bolso	Cintura	Punho	Pescoço	Todos
1	8,64%	9,54%	9,07%	10,31%	6,67%	9,56%
2	4,72%	10,66%	10,51%	12,04%	8,57%	9,74%
3	5,77%	6,56%	7,46%	10,75%	7,94%	8,93%
4	6,55%	8,07%	9,31%	10,85%	7,73%	9,28%
5	8,93%	7,31%	9,04%	13,10%	8,78%	11,04%
6	0,00%	0,00%	0,00%	0,01%	0,00%	0,01%
7	0,00%	0,02%	0,01%	0,01%	0,01%	0,01%
8	0,00%	0,00%	0,01%	0,01%	0,00%	0,01%
9	0,00%	0,60%	0,56%	0,03%	0,65%	0,03%
10	0,01%	0,07%	0,06%	0,01%	0,08%	0,07%
11	0,20%	0,12%	0,14%	0,06%	0,15%	0,09%

Fonte: Elaborado pelo autor (2024).

quedas para frente apoiando com os joelhos, para o lado e para a frente apoiando com as mãos.

Tabela 4 – Porcentagem média de erros de identificação do MLP

Atividade	Tornozelo	Bolso	Cintura	Punho	Pescoço	Todos
1	14,96%	17,21%	15,47%	15,23%	16,94%	16,09%
2	13,94%	18,20%	16,65%	16,04%	16,53%	15,58%
3	14,70%	16,58%	16,43%	15,69%	16,78%	15,39%
4	15,74%	17,41%	17,08%	15,66%	14,95%	16,06%
5	16,41%	16,78%	16,49%	15,57%	16,26%	16,13%
6	0,73%	0,88%	1,26%	1,50%	0,35%	0,73%
7	0,02%	0,03%	0,12%	0,41%	0,14%	0,25%
8	0,03%	0,07%	0,22%	0,39%	0,07%	0,19%
9	0,35%	2,07%	2,76%	1,29%	2,09%	1,13%
10	0,44%	0,75%	0,86%	1,74%	1,04%	0,89%
11	0,37%	0,09%	0,29%	0,46%	0,20%	0,26%

Fonte: Elaborado pelo autor (2024).

O SVM foi o método que apresentou os resultados mais destoantes em relação aos demais. Como pode ser observado na Tabela 5, que segue o mesmo padrão das anteriores, a porcentagem média de erros desse método não chegou a 10% em nenhum cenário dos cinco tipos de quedas, entretanto ultrapassou essa mesma porcentagem em algumas situações de atividades do dia a dia, o que não aconteceu com os outros métodos. Além disso, a atividade com as maiores taxas de erro foi pular, seguida por pegar um objeto, enquanto no quesito de identificação de quedas, não há uma distinção discrepante entre os resultados de cada tipo.

Tabela 5 – Porcentagem média de erros de identificação do SVM

Atividade	Tornozelo	Bolso	Cintura	Punho	Pescoço	Todos
1	3,60%	8,71%	7,79%	6,43%	3,68%	5,61%
2	3,16%	8,64%	7,66%	7,70%	5,30%	7,18%
3	4,11%	4,99%	6,73%	9,19%	5,61%	8,10%
4	3,55%	5,66%	6,31%	8,17%	5,80%	7,75%
5	5,86%	4,67%	7,22%	9,07%	3,68%	7,57%
6	12,17%	2,32%	5,17%	3,88%	1,06%	2,61%
7	0,30%	0,38%	0,11%	0,78%	0,34%	0,57%
8	0,26%	0,18%	0,14%	0,49%	0,15%	0,35%
9	8,08%	11,37%	4,82%	2,94%	14,01%	6,36%
10	15,04%	5,54%	7,96%	4,96%	14,52%	4,56%
11	0,63%	0,28%	0,26%	0,58%	0,57%	0,50%

Fonte: Elaborado pelo autor (2024).

Comparando as porcentagens médias de erro dos quatro métodos, é possível constatar que a atividade rotineira que mais foi classificada incorretamente foi a ação de pegar um objeto. No entanto, para os tipos de queda, não foi possível identificar um padrão específico de erro mais frequente, visto a variação dos resultados para a mesma variedade de queda tanto entre os diferentes posicionamentos dos sensores quanto entre os métodos analisados.

5 CONCLUSÕES

Diante das consequências que as quedas trazem às pessoas, sobretudo aos idosos, e aos sistemas de saúde, principalmente em situações em que o indivíduo não é capaz de se levantar sem auxílio, os sistemas de detecção de queda surgem como uma ferramenta para minimizar os problemas causados por esses acidentes. Um dos componentes essenciais para esses sistemas é o algoritmo que determina quando há uma queda a partir dos dados provindos de sensores.

Nesse contexto, este trabalho buscou comparar quatro métodos de aprendizado de máquina supervisionado para sistemas de detecção de queda baseados em acelerômetros e giroscópios. Foi utilizada uma base de dados com amostras de atividades do dia a dia e de quedas simuladas por pessoas jovens que usaram tais sensores em cinco pontos do corpo. Tal base também contém características no domínio do tempo e da frequência extraídas das informações brutas, o que facilitou o processamento dos dados.

A partir disso, foram treinados, realizando a otimização de determinados hiperparâmetros, modelos de cada método e comparado o desempenho de cada um com base em métricas comuns na área de IA e avaliada a influência das características de cada domínio e o impacto do posicionamento dos sensores na detecção de quedas. Assim como, foi analisado em quais atividades cada método apresentou mais dificuldades ao diferenciar quedas de atividades rotineiras.

Sendo assim, com base na análise dos resultados de cada método, é possível concluir que, considerando a implementação em um sistema de detecção de quedas, o RF apresentou o melhor desempenho, atingindo 67,9% de recall, 87,7% de precisão, 76,6% de f1-score e 98,8% de acurácia quando utilizado com as características no domínio do tempo extraídas dos dados de uma IMU posicionada no tornozelo.

Em relação à contribuição de cada domínio das características, pode-se notar que foram alcançados melhores resultados, na maioria dos cenários avaliados, aplicando exclusivamente as características no domínio do tempo e quanto ao posicionamento dos sensores, o tornozelo proporcionou constantemente o melhor desempenho quando analisados os locais individualmente. Por fim, referente às atividades mais frequentemente identificadas incorretamente, não foi observado um padrão consistente entre os tipos de quedas, já entre as atividades do dia a dia, a ação de pegar um objeto foi a que mais gerou erros nos métodos avaliados.

Visto que apenas o método RF apresentou bons resultados simultaneamente nas métricas de recall e precisão, futuros trabalhos podem explorar técnicas de sobreamostragem e subamostragem para balancear a quantidade de amostras de cada classe, a fim de mitigar os efeitos do desbalanceamento, o que pode reduzir o viés

dos modelos em relação à classe majoritária. Além disso, com o intuito de identificar as características mais relevantes dentro de cada domínio, podem ser investigados os efeitos causados por uma seleção mais criteriosa das características utilizadas. Também podem ser estudadas a combinação de diferentes posicionamentos dos sensores de aceleração e variação angular e a utilização de distintos tipos de sensores, como de frequência cardíaca ou cerebral, para ampliar o contexto fornecido aos métodos.

REFERÊNCIAS

- CASILARI, E.; SILVA, C. A. An analytical comparison of datasets of real-world and simulated falls intended for the evaluation of wearable fall alerting systems. **Measurement**, v. 202, p. 111843, 2022.
- DADAFSHAR, M. **Accelerometer and Gyroscopes Sensor: Operation, sensing and applications**. Estados Unidos, 2015. Disponível em: <https://www.analog.com/en/resources/technical-articles/accelerometer-and-gyroscopes-sensors-operation-sensing-and-applications.html>. Acesso em: 10 nov. 2024.
- GHARGHAN, S. K.; HASHIM, H. A. A comprehensive review of elderly fall detection using wireless communication and artificial intelligence techniques. **Measurement**, v. 226, p. 114186, 2024.
- IGUAL, R.; MEDRANO, C.; PLAZA, I. Challenges, issues and trends in fall detection systems. **BioMedical Engineering OnLine**, Springer Nature, v. 12, n. 1, p. 66, jul. 2013.
- KANGAS, M. *et al.* Comparison of low-complexity fall detection algorithms for body attached accelerometers. **Gait Posture**, v. 28, n. 2, p. 285–291, ago. 2008.
- LIMA, J. d. S. *et al.* Custos das autorizações de internação hospitalar por quedas de idosos no Sistema Único de Saúde, Brasil, 2000-2020: um estudo descritivo. **Epidemiologia e Serviços de Saúde**, v. 31, n. 1, p. e2021603, 2022.
- LIU, J. *et al.* A review of wearable sensors based fall-related recognition systems. **Engineering Applications of Artificial Intelligence**, v. 121, p. 105993, 2023.
- MARSLAND, S. **Machine Learning: An algorithm perspective**. 2. ed. Philadelphia, PA: Chapman & Hall/CRC, 2014.
- MARTÍNEZ-VILLASEÑOR, L. *et al.* UP-Fall detection dataset: A multimodal approach. **Sensors**, v. 19, n. 9, p. 1988, 2019.
- MULLER, A.; GUIDO, S. **Introduction to Machine Learning: A guide for data scientists**. 1. ed. Sebastopol, CA: O'Reilly, 2016.
- ORGANIZAÇÃO MUNDIAL DE SAÚDE. **Falls: Key facts**. Geneva, Suíça, 2021. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/falls>. Acesso em: 13 out. 2024.
- ORGANIZAÇÃO MUNDIAL DE SAÚDE. **Step safely: Strategies for preventing and managing falls across the life-course**. Geneva, Suíça, 2021. Disponível em: <https://www.who.int/publications/i/item/978924002191-4>. Acesso em: 13 out. 2024.
- ÖZDEMİR, A. T. An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice. **Sensors**, v. 16, n. 8, p. 1161, 2016.
- PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A modern approach**. 4. ed. Estados Unidos: Pearson Education, 2021.

SCIKIT-LEARN. **1.4. Support Vector Machines**. 2024. Disponível em: <https://scikit-learn.org/1.5/modules/svm.html>. Acesso em: 14 nov. 2024.

STAMPFLER, T. *et al.* Fall detection using accelerometer-based smartphones: Where do we go from here? **Frontiers in Public Health**, v. 10, p. 996021, out. 2022.

TAN, P.-N. *et al.* **Introduction to Data Mining**. 2. ed. Nova Iorque, Estados Unidos: Pearson, 2019.

VAROQUAUX, G.; COLLIOT, O. Evaluating machine learning models and their diagnostic value. *In*: COLLIOT, O. (ed.). **Machine Learning for Brain Disorders**. New York, NY: Springer US, 2023. p. 601–630.

WIXLEY, H. **Smart Application for Fall Detection Using Wearable ECG Accelerometer Sensors**. 2023. 4th Year Project Report (Artificial Intelligence and Computer Science) — School of Informatics, University of Edinburgh, 2023.

XU, T.; ZHOU, Y.; ZHU, J. New advances and challenges of fall detection systems: A survey. **Applied Sciences**, v. 8, n. 3, 2018.