

Wearable IoT Devices: Feasibility and Effectiveness in Implementing Local Decision-Making for Elderly Fall Detection

Vitor Miller de Toledo
Federal Institute of Espírito Santo
Cachoeiro de Itapemirim, Espírito
Santo, Brazil
vitormillerdetoledo@gmail.com

Cristiano da Silveira Colombo
Federal Institute of Espírito Santo
Cachoeiro de Itapemirim, Espírito
Santo, Brazil
cristianos@ifes.edu.br

Igor Henrique Beloti Pizetta
Federal Institute of Espírito Santo
Cachoeiro de Itapemirim, Espírito
Santo, Brazil
igorpizetta@gmail.com

Lucas Poubel Timm do Carmo
Federal Institute of Espírito Santo
Cachoeiro de Itapemirim, Espírito
Santo, Brazil
lucas.carmo@ifes.edu.br

Rafael Silva Guimarães
Federal Institute of Espírito Santo
Cachoeiro de Itapemirim, Espírito
Santo, Brazil
rafaelg@ifes.edu.br

ABSTRACT

Context: The growing use of IoT wearables in healthcare highlights their potential for elderly fall detection. However, wearable devices based on inertial sensors have processing and energy consumption limitations, making them unsuitable for complex algorithms. Traditional cloud systems face latency and bandwidth issues when processing inertial data, which may delay caregiver responses. Edge computing, with local decision-making, emerges as an efficient and energy-conscious alternative.

Problem: The overload in transmitting inertial sensor data to remote servers can delay notifications, putting elderly individuals whose falls are undetected at risk. Rapid intervention is crucial to prevent severe consequences.

Solution: This study implements a decision tree algorithm processed locally on the ESP32 microcontroller, enabling real-time fall detection while reducing energy consumption and response times.

SI Theory: The study applies General Systems Theory by integrating inertial sensor data, software algorithms, and human behavior. This systemic approach optimizes device performance, promoting greater efficiency in real-time health monitoring and improving elderly safety.

Method: This applied, quantitative research pre-processed accelerometer and gyroscope data into statistical metrics. A decision tree algorithm was developed and validated for local processing on the ESP32, using a dataset tailored for fall detection.

Summary of Results: The proposed solution on the ESP32 detected falls with precision, without false positives, distinguishing daily activities in controlled tests. Additionally, it demonstrated high energy efficiency, validating its viability for IoT wearable devices.

Contributions and Impact in the IS area: This study validates the integration of edge computing with wearable devices, balancing computational efficiency, energy consumption, and safety, offering effective solutions for health monitoring.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Computer systems organization** → **Embedded and cyber-physical**

systems; • **Applied computing** → **Life and medical sciences**; • **Hardware** → **Power estimation and optimization**.

KEYWORDS

Elderly, Fall detection, IOT, Edge-Computing, Decision Tree, Wearables

1 INTRODUÇÃO

Nos últimos anos, observou-se um avanço notável na integração de tecnologias que buscam aprimorar os cuidados com a saúde. Nesse contexto, dispositivos de Internet das Coisas (IoT) têm se destacado como ferramentas inovadoras e cada vez mais recorrentes no monitoramento e gestão da saúde. Tais métodos, se manifestam vantajosos sobre métodos tradicionais pois permitem a coleta e transmissão de dados em tempo real e inclusive fora do ambiente hospitalar[37].

Uma consideração fundamental neste contexto é o método pelo qual os dados são processados e aplicados na tomada de decisões. Comumente, muitos sistemas optam por enviar essas informações para processamento em nuvem com o fim de aplicar métodos de inteligência computacional que exigem grande carga de processamento. Contudo, a distância física entre os dispositivos finais de IoT e os servidores de computação em nuvem podem representar um desafio significativo para aplicações que demandam respostas imediatas. Isso é particularmente crítico em cenários como o gerenciamento de catástrofes e a transferência de conteúdo[32], destacando o monitoramento de saúde como uma área particularmente suscetível a essas limitações.

Um exemplo concreto que ilustra a problemática mencionada é o cenário de quedas em idosos, especialmente aqueles que vivem sozinhos. Tendo em vista que uma queda não sinalizada dentro do intervalo de uma hora pode levar a sérias consequências, como deficiência funcional[20] e declínio na saúde geral[29], esse recorte traz à tona a necessidade de otimizar o gerenciamento de tais catástrofes. Alternativas já estudadas e utilizadas para monitorar esses eventos envolvem o uso de *smartphones*[31], *smartwatches*[8] e até mesmo cintos[9] - ambos associados à sensores inerciais, como acelerômetros e giroscópios. Esses sensores são utilizados para registrar e monitorar os movimentos dos idosos e, em grande parte

das soluções já existentes, enviam os dados brutos para processamento em nuvem. No entanto, a transmissão contínua de grandes volumes de dados gerados pelo acelerômetro e giroscópio pode retardar a notificação aos cuidadores, tendo em vista limitantes como largura de banda, por exemplo, comprometendo a rapidez da intervenção necessária em casos de emergência. Por isso, uma abordagem que vem ganhando destaque é a execução da tomada de decisão diretamente no microprocessador do dispositivo IoT.

Esta abordagem, conhecida como *Edge Computing*, destaca-se por sua capacidade de realizar o processamento e a análise de dados diretamente no dispositivo, eliminando a necessidade de envio para servidores remotos. Essa estratégia reduz significativamente o tempo de latência, o que é particularmente crítico em contextos médicos, onde a rapidez na tomada de decisões pode ser determinante para a segurança e o bem-estar dos pacientes.

Neste cenário, os dispositivos vestíveis, também conhecidos como "*wearables*", têm se tornado uma alternativa cada vez mais popular para o monitoramento contínuo e a coleta de dados de pacientes[5]. Esses dispositivos oferecem um método pouco invasivo, graças à sua fácil integração com o usuário, abrangendo produtos como relógios, pulseiras e óculos. Quando combinados com a tomada de decisão local e sensores embutidos, os *wearables* apresentam vantagens significativas, permitindo a obtenção de métricas precisas e tomadas de decisão ágeis, essenciais para contextos de monitoramento em tempo real.

Sabendo disso, para a resolução da problemática e implementação dos princípios da *Edge Computing*, será realizado um estudo de caso no microcontrolador ESP32, um elemento cada vez mais reconhecido como fundamental nesse panorama tecnológico[26]. A análise se concentrará na avaliação das limitações e viabilidades da implementação de inteligência computacional aplicada à detecção de quedas em idosos através deste dispositivo. Concentrando-se numa solução integrada do ESP32 com sensores inerciais, o objetivo é estabelecer um sistema de monitoramento de quedas de idosos com tomada de decisão local, buscando-se obter redução no tempo de resposta, algo de suma importância em contextos médicos. Este estudo visa verificar o consumo energético e a eficácia dessas soluções em ambientes reais, vislumbrando assim as possibilidades de implementações práticas na área da saúde.

Por fim, a solução proposta, ao adotar uma abordagem pouco invasiva por meio de dispositivos vestíveis para monitoramento da saúde de idosos, alinha-se à visão sociotécnica dos Sistemas de Informação (SI), que busca unir avanços tecnológicos a benefícios sociais concretos. A interoperabilidade, um dos desafios fundamentais dos SI, ressalta a necessidade de soluções capazes de operar em ambientes heterogêneos e distribuídos. A aplicação de protocolos IoT, utilizados na arquitetura do sistema de validação deste estudo, bem como a abordagem de tomada de decisão local, reforçam essa premissa ao viabilizar um modelo mais eficiente e descentralizado. Nesse contexto, o uso do ESP32 aliado ao paradigma de *Edge Computing* e às otimizações energéticas demonstradas nesta pesquisa enfatiza a viabilidade do sistema dentro de um cenário cada vez mais interconectado, no qual arquiteturas eficientes são essenciais para gerenciar a complexidade crescente dos SI.

2 REFERENCIAL TEÓRICO

Nesta seção, apresentam-se os conceitos fundamentais que sustentam o desenvolvimento deste trabalho.

2.1 Internet das Coisas

A Internet das Coisas (IoT) é um conceito tecnológico que descreve a interconexão de dispositivos físicos, objetos e sistemas por meio da internet. Sua premissa básica é a presença ubíqua de uma variedade de dispositivos – como etiquetas de Identificação por Radiofrequência (RFID), sensores, atuadores, e telefones celulares – que, por meio de esquemas de endereçamento únicos, são capazes de interagir entre si e cooperar com seus pares para alcançar objetivos comuns[1].

A IoT tem desempenhado um papel fundamental na transformação digital em diversos setores, como saúde, infraestrutura e transporte[10]. Ela possibilita a coleta de dados em tempo real, a automação de processos, a melhoria da eficiência operacional e a geração de insights valiosos que auxiliam na tomada de decisões informadas[21].

2.2 Edge Computing

O *Edge Computing* é um paradigma de computação distribuída que realiza o processamento e a análise de dados próximo aos dispositivos de coleta, reduzindo a dependência de servidores remotos na nuvem[34]. Essa abordagem diminui significativamente a latência e a sobrecarga na rede, proporcionando respostas mais rápidas e eficientes em tempo real.

Ao possibilitar que dispositivos IoT executem tarefas computacionais e análises localmente, o *Edge Computing* combina eficiência operacional com a flexibilidade da computação em nuvem. Essa tecnologia destaca-se em aplicações críticas, como o monitoramento de saúde, em que a rapidez e a precisão são determinantes para assegurar a segurança e o bem-estar dos indivíduos.

2.3 Redes de Sensores

Sensores são dispositivos capazes de converter fenômenos físicos, químicos ou biológicos em sinais elétricos ou digitais. Esses dispositivos são amplamente utilizados para medir diversas grandezas, como temperatura, pressão, umidade, luminosidade e movimento. Com base nessas funcionalidades, as redes de sensores emergem como uma solução promissora para o processamento de dados em larga escala e em tempo real em ambientes complexos, ajudando a proteger e monitorar infraestruturas e recursos militares, ambientais, críticos para segurança e domésticos[40].

2.3.1 Acelerômetro. O acelerômetro é um sensor utilizado para medir a aceleração de um objeto em relação à gravidade ou a uma aceleração específica. Esse dispositivo é capaz de detectar movimentos lineares, como deslocamentos ou mudanças na velocidade, operando com base em princípios físicos, como a força de inércia[11].

A tecnologia de acelerômetros é amplamente empregada em diversas aplicações, incluindo dispositivos eletrônicos portáteis[39], sistemas de navegação[36], monitoramento de saúde[17] e pesquisas científicas. No contexto da detecção de quedas, os acelerômetros

desempenham um papel essencial, pois podem identificar rapidamente variações bruscas na aceleração de um indivíduo, acionando alertas e procedimentos de emergência quando necessário.

2.3.2 Giroscópio. Um giroscópio é um sensor que mede a taxa de rotação ou a velocidade angular de um objeto em torno de um eixo específico. Ele detecta movimentos de rotação, como giros ou curvas, e opera com base em princípios de conservação do momento angular[11]. Os giroscópios são utilizados em uma ampla gama de aplicações, em contextos de segurança, por exemplo, podem ser empregados na identificação de quedas[7], auxiliando na avaliação da orientação e na determinação da gravidade do impacto durante um evento de queda.

2.4 Microcontroladores

Microcontroladores são dispositivos compactos que integram uma unidade central de processamento (CPU), memória, periféricos de entrada/saída e, frequentemente, interfaces de comunicação em um único chip. Sua aplicação principal está em sistemas embarcados, nos quais desempenham funções específicas de controle e processamento[16].

2.4.1 ESP32. O ESP32 é um microcontrolador de baixo custo e alto desempenho, desenvolvido pela Espressif Systems, projetado para aplicações de Internet das Coisas (IoT). Ele integra CPU de dois núcleos, conectividade Wi-Fi e Bluetooth, e suporte para diversos periféricos.

Amplamente utilizado em protótipos e aplicações como casas inteligentes, automação, dispositivos vestíveis e soluções IoT baseadas em nuvem, o ESP32 permite tanto o uso de kits de desenvolvimento quanto o design de sistemas embarcados personalizados[2].

2.5 Wearables

Os dispositivos vestíveis, ou "*wearables*", representam uma categoria inovadora de tecnologia projetada para integrar-se de forma contínua e discreta ao cotidiano dos usuários. Esses dispositivos são projetados para serem usados de maneira contínua, integrando-se ao vestuário ou acessórios diários, o que contribui para sua aceitação generalizada. Por isso, ao longo dos últimos anos, uma ampla variedade de dispositivos vestíveis tem aparecido no mercado, oferecendo diferentes funcionalidades e opções de uso, proporcionando funcionalidades diversas em formatos compactos e convenientes[33].

Ademais, a constante evolução em *hardware* e *software* amplia as possibilidades de um futuro promissor desses dispositivos em ambientes de monitoramento remoto e assistência à saúde[27].

2.6 Machine Learning para IoT na saúde

O aprendizado de máquina (*Machine Learning* - ML), parte da inteligência artificial, desenvolve algoritmos capazes de aprender e tomar decisões com base em dados.

No monitoramento de saúde, ML permite tarefas como classificação de padrões, detecção de anomalias e diagnósticos assistidos, contribuindo para uma assistência médica mais precisa e personalizada[30]. A partir da integração de ML com IoT e *Edge Computing*, os dispositivos podem não apenas coletar, mas também analisar

dados localmente, oferecendo respostas rápidas e autônomas em situações críticas.

2.6.1 Árvores de Decisão. As Árvores de Decisão são algoritmos de aprendizado de máquina que utilizam uma estrutura hierárquica semelhante a uma árvore, composta por nós (atributos), ramificações (valores dos atributos) e folhas (classes ou resultados). Essa estrutura segue uma abordagem de divisão e conquista[13]. O nó raiz, localizado no topo, inicia o processo de decisão, enquanto os ramos conectam outros nós estruturais ou de decisão, e as folhas representam as classes finais.

Operando sob aprendizado supervisionado, as Árvores de Decisão transformam conjuntos de dados analisados em regras claras do tipo "se/então", facilitando previsões com base nas características fornecidas. Utilizando um conjunto de dados específico, constrói-se uma árvore onde testes são aplicados às características dos dados de entrada e, ao atingir uma folha, é possível prever a etiqueta correspondente[14]. Essa característica torna a implementação de Árvores de Decisão em dispositivos IoT particularmente vantajosa, pois basta codificar diretamente as regras "se/então" no microcontrolador, eliminando a necessidade de processamentos complexos ou dependência de sistemas externos.

Por isso, devido à sua eficiência energética e operacional, Árvores de Decisão são adequadas para identificar eventos críticos, como quedas, maximizando a eficácia e a praticidade em contextos de uso real.

3 TRABALHOS CORRELATOS

Os trabalhos correlatos exploram diversas abordagens e técnicas desenvolvidas para a detecção de quedas em pessoas, oferecendo diferentes métodos e estratégias para aprimorar a precisão e a eficiência dos sistemas de monitoramento.

Os autores em [8] propuseram um método utilizando dados de acelerômetros presentes em *smartwatches* de pulso. O sistema, baseado em *autoencoders* empilhados (ESAEs) e classificação por *convex hull* (OCCCH), obteve especificidade de 98,92% e sensibilidade de 96,09%, superando métodos convencionais. Por outro lado, [9] desenvolveram um cinto vestível inteligente equipado com um sensor IMU, combinando acelerômetro e giroscópio triaxiais, posicionado na cintura. Esse sistema detectou quedas em tempo real e envia alertas via GSM, alcançando 100% de precisão nos testes realizados.

No trabalho [28], foi criado o sistema *Home e-Care*, que utiliza acelerômetros embutidos em *smartwatches Android Wear* posicionados no pulso. O sistema, baseado em algoritmos de limiar, notifica contatos de emergência e apresentou uma acurácia de 83,33%, com sensibilidade de 75% e especificidade de 89,29%. De maneira complementar, [24] exploraram sensores vestíveis da plataforma SHIMMER, equipados com acelerômetros e giroscópios triaxiais, posicionados no tórax e na coxa. O sistema é capaz de classificar quedas frontais, traseiras e laterais, com uma precisão variando entre 72% e 81%, dependendo do tamanho do conjunto de treinamento utilizado.

Já [25] realizou um estudo comparativo do desempenho de algoritmos de aprendizagem de máquina, como Árvore de Decisão, *K-Nearest Neighbors* (KNN) e Perceptron Multicamadas (MLP), aplicados à detecção de quedas em idosos. O trabalho utilizou o *dataset*

SisFall, composto por dados de acelerômetros capturados em diferentes simulações de quedas e atividades diárias. O dispositivo vestível utilizado para a captura de dados foi fixado na região da cintura dos participantes. Após o pré-processamento e a extração de características estatísticas, o modelo MLP apresentou os melhores resultados, com uma revocação de 97,92% nos testes e 100% durante a validação.

Dispositivos vestíveis podem ser integrados e posicionados de diversas maneiras, podendo ser inclusive fixados na região da cintura ou do tórax. Essas localizações facilitam a detecção de quedas, pois os movimentos nessas áreas geralmente são pequenos em atividades normais. Durante uma queda, no entanto, os movimentos nessas regiões se tornam muito mais intensos, resultando em mudanças drásticas nos dados dos sensores de movimento[18]. Apesar de sua eficácia, dispositivos posicionados na cintura ou no tórax podem ser considerados pouco estéticos ou inconvenientes, especialmente em situações que exigem sua remoção e recolocação frequentes.

Por outro lado, dispositivos posicionados no pulso, devido à sua semelhança com relógios convencionais, não enfrentam as limitações relacionadas à usabilidade. Esses dispositivos são pequenos, discretos e permanecem no lugar com facilidade, o que contribui para uma aceitação mais ampla entre os usuários. No entanto, o pulso é uma região mais instável, devido ao constante movimento das mãos. Essa característica pode resultar em um aumento na incidência de falsos positivos na identificação de quedas, especialmente quando são utilizadas abordagens baseadas em limiares.

Este trabalho se conecta diretamente aos estudos anteriores que exploram dispositivos vestíveis fixados no pulso, como os de [8] e [28], considerando as vantagens e limitações dessa abordagem. No entanto, ele se diferencia ao utilizar o modelo de Árvore de Decisão, a partir de pré-processamento estatístico, como o explorado por [25], implementados diretamente no dispositivo e, além disso, avalia impactos energéticos e viabilidade prática da solução.

4 METODOLOGIA DE PESQUISA

Esta pesquisa adota uma abordagem aplicada, situando-se entre a ciência básica e a tecnologia [23]. Baseada na metodologia quantitativa [22], a investigação segue um processo formal e sistemático, quantificando variáveis e analisando-as estatisticamente para garantir resultados confiáveis e generalizáveis.

O estudo utiliza o método experimental [15], no qual o ESP32 foi configurado para coletar e processar dados de sensores inerciais (acelerômetro e giroscópio) para detecção de quedas em idosos. No processamento dos dados, foram extraídas estatísticas como média, desvio padrão, valores máximos e mínimos (subseção 7.2), usadas como entrada para um modelo de Árvore de Decisão executado localmente.

A avaliação do modelo foi realizada por meio da Matriz de Confusão e métricas como Acurácia, Sensibilidade, Precisão e F1-Score (seção 9), garantindo uma análise abrangente do desempenho. Formulou-se a hipótese de que a execução local do modelo reduziria latência e consumo energético, sem comprometer a precisão da detecção. Essa hipótese foi testada comparando-se cenários de processamento local e remoto, analisando os ganhos da abordagem Edge Computing na eficiência da solução.

5 MATERIAIS E MÉTODOS

Nesta seção serão apresentados os materiais, métodos e tecnologias utilizadas neste trabalho.

5.1 Hardware Principal

O ESP32-S3-Touch-LCD-1.69, desenvolvido pela *Waveshare*, é um microcontrolador projetado para aplicações de IoT e dispositivos inteligentes. Este módulo se destaca por sua capacidade de processamento, conectividade integrada e, além disso, por integrar diversos recursos em um formato pequeno - facilitando a adequação visual, ergonômica e prática à temática de dispositivos vestíveis -, incluindo um *display* capacitivo de 1,69 polegadas com resolução de 240×280 *pixels*, controlado pelo driver ST7789V2 e o chip de toque CST816T. Seu núcleo de processamento é o ESP32-S3R8, com arquitetura Xtensa® 32-bit LX7 dual-core, operando a até 240 MHz, e suportando conectividade Wi-Fi 2.4GHz e Bluetooth 5.0 (LE), possibilitando a comunicação contínua com outros dispositivos e serviços de monitoramento remoto.

Entre os recursos integrados, destaca-se o QMI8658, um sensor inercial de 6 eixos que combina um acelerômetro triaxial e um giroscópio triaxial, capaz de auxiliar na medição de movimentos, bem como na detecção de gestos e monitoramento de atividades, tornando possível a aplicação na análise de padrões de movimento. Além disso, o dispositivo possui uma memória expansiva de 8 MB de PSRAM e 16 MB de Flash, um RTC com chip PCF85063 para funcionalidades de tempo real, e suporte a bateria de lítio recarregável, o que facilita o desenvolvimento de dispositivos portáteis. Com interfaces adicionais de GPIO, I2C e UART reservadas [38].



Figura 1: Hardware Principal.
Fonte: Fabricante [38].

5.2 Plataforma de Programação do Hardware

A plataforma Arduino, conforme descrito por [4], é um ecossistema de *hardware* e *software* projetado para simplificar o desenvolvimento de projetos eletrônicos interativos, utilizando uma linguagem de programação baseada em C/C++ e uma IDE de simples acesso. A plataforma é amplamente utilizada devido à sua vasta coleção de bibliotecas, comunidade ativa e suporte a dispositivos como o ESP32. Essa compatibilidade permite tirar proveito dos recursos avançados do ESP32, como conectividade Wi-Fi e Bluetooth, e além disso, controlar a interação com os sensores.

5.3 Ambiente de manipulação de *dataset* e modelo

Google Collaboratory, ou simplesmente Google Colab, um projeto que busca promover a educação e pesquisa em aprendizado de máquina. Esta plataforma baseia-se na tecnologia dos *Jupyter Notebooks*, uma ferramenta de código aberto e baseada em navegador que integra linguagens interpretadas, bibliotecas e ferramentas de visualização. A facilidade de operação tanto em ambiente local quanto na nuvem e a estrutura modular dos documentos, que são compostos por múltiplas células capazes de conter tanto código quanto markdown, tornam os *Jupyter Notebooks* ideais para documentar experimentos e resultados de forma dinâmica e interativa [6].

A plataforma oferece ambientes de execução para Python já pré-configurados com bibliotecas amplamente utilizadas em manipulação de dados, aprendizado de máquina e visualização de métricas, como *Pandas*, *Scikit-Learn* e *Matplotlib*, o que simplifica significativamente o desenvolvimento e a execução de modelos de aprendizado de máquina [6]. No presente trabalho, o Google Colab foi utilizado como ferramenta central para o pré-processamento de dados, a definição, o treinamento e os testes do algoritmo de Árvore de Decisão.

5.4 Dataset

O WEDA-FALL - *Wrist Elderly Daily Activity and Fall Dataset* foi utilizado neste trabalho como base para a construção do modelo de Árvore de Decisão, entendimento da mecânica e os aspectos quantitativos de uma ocorrência de queda. Este *dataset* é um recurso público disponível no *GitHub*¹ - uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git - e foi desenvolvido no âmbito do projeto *IntelligentCare – Intelligent Multimorbidity Management System*, cofinanciado pelo Fundo Europeu de Desenvolvimento Regional. Ele foi coletado utilizando o *smartwatch Fitbit Sense 2* - desenvolvido pela Google -, posicionado no pulso, empregando o mecanismo de captura de dados do próprio dispositivo [12]. Ademais, foi disponibilizado em múltiplas frequências (50Hz, 40Hz, 25Hz, 10Hz e 5Hz), sendo a frequência de 25Hz a selecionada para o presente estudo e análise.

O *dataset* abrange duas categorias principais de movimentos: quedas (*Falls*) e atividades de vida diária (*ADLs*). As quedas incluem eventos como quedas para frente e para trás durante caminhada, quedas laterais ao estar sentado, entre outras situações. Já as *ADLs* foram selecionadas com base na frequência de atividades reais e na semelhança com padrões de movimento que poderiam gerar falsos positivos. Essas atividades incluem caminhar, correr, abrir e fechar portas, subir e descer escadas, entre outras.

O WEDA-FALL destaca-se por seu foco no público idoso, com dados divididos entre participantes jovens e idosos. Para garantir a segurança dos idosos, as atividades foram limitadas a *ADLs* sem risco, como caminhar e sentar. O *dataset* possui dados diversificados, categorizados por usuário, repetição e tipo de movimento, com eventos segmentados em fases específicas (pré-queda, impacto, ajuste corporal e pós-queda).

No total, o grupo de participantes jovens gerou 350 sinais de quedas e 462 sinais de *ADLs*. Para os participantes idosos, as repetições variaram conforme a mobilidade, conforto e fadiga de cada

indivíduo, resultando em 157 sinais específicos de participantes idosos, totalizando 619 sinais de *ADLs*. Essa abordagem permitiu uma coleta ampla e diversificada de dados, respeitando as limitações físicas do público idoso.

A escolha desse *dataset* deve-se à sua riqueza de dados quantitativos, à alta quantidade de eventos de queda capturados e à categorização detalhada das atividades. Além disso, a utilização de dispositivos no pulso para a captura de dados alinha-se ao objetivo de monitorar quedas de forma pouco invasiva com dispositivos vestíveis.

6 CONFIGURAÇÕES E AJUSTES PARA COMPATIBILIDADE ENTRE DATASET E DISPOSITIVO

A partir da análise e estudo dos valores do *dataset*, bem como a leitura da documentação dos dispositivos e sensores - QMI8658, ESP32-S3 e *Fitbit Sense* - foram observadas algumas diferenciações e incompatibilidades que resultaram em correções e calibrações necessárias. Portanto, para a compatibilização foram feitas alterações tanto no *dataset* quanto no dispositivo de captura (ESP32-S3).

6.1 Ajustes e calibrações na captura de dados pelo sensor

- Aplicação de *Offsets*: *Offsets* foram aplicados para corrigir os desvios existentes nos sensores e garantir que as medições capturadas em repouso ou movimento representem valores reais. Para determinar os *offsets* de cada eixo (x,y,z) tanto do acelerômetro quanto do giroscópio, foi realizado um procedimento em que os sensores foram mantidos em uma condição estática e normalizada por 10 segundos enquanto ocorria a captura dos dados pelos sensores. A média de cada eixo foi calculada a partir dos dados coletados e utilizada como *offset*.
- Conversão e Processamento dos Dados do Acelerômetro: Os valores do acelerômetro são fornecidos em *Least Significant Bit* (LSB), que é a menor unidade digital correspondente a um valor físico no sensor. Para o acelerômetro configurado na escala de $\pm 16g$ - que a partir de análise dos padrões e limites de valores do *dataset* averiguou-se ser o mesmo adotado pelo *FitBit Sense*, o *datasheet* do sensor inercial² especifica que $1g$ equivale a 2048 em LSB. A conversão dos valores brutos para unidades físicas (m/s^2) foi realizada dividindo os valores por 2048 e, em seguida, multiplicando por $9.81 m/s^2$, correspondente à aceleração gravitacional. Os dados retornados pelo sensor de acelerômetro, em estado estacionário e alinhados com o vetor da aceleração gravitacional devem resultar em aproximadamente $9.8 m/s^2$, correspondendo à influência da aceleração gravitacional sobre o dispositivo. Além disso, os valores foram limitados a $\pm 39.2 m/s^2$, com base na análise do *dataset*, que revelou que os valores mínimos e máximos atingiam esse teto intervalar.
- Conversão e Processamento dos dados do Giroscópio: Os valores brutos do giroscópio também são fornecidos em LSB (*Least Significant Bit*), sendo que, para a configuração de

¹<https://github.com/joaojtmarques/WEDA-FALL>

²https://www.waveshare.com/w/upload/3/36/QMI8658C_datasheet_rev_0.8.pdf

± 256 dps (graus por segundo), 1 dps equivale a 128 LSB. A conversão dos valores para rad/s foi realizada dividindo os valores por 128 e, em seguida, multiplicando pelo fator de conversão $\frac{\pi}{180} \approx 0.0174533$, que transforma graus em radianos.

6.2 Ajustes e Operações no Dataset

A partir da documentação do *smartwatch FitBit Sense*³ e do *datasheet* do sensor QMI8658, averiguou-se uma diferença de orientação entre os eixos, como pode ser visto na Figura 2.

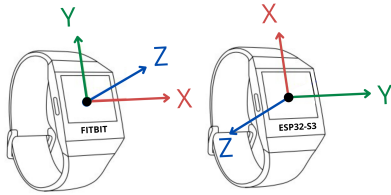


Figura 2: Orientação dos Eixos dos Sensores dos Dispositivos.
Fonte: Adaptado de Guia do Fabricante Fitbit®

Tendo em vista que a diferença citada poderia ocasionar em uma adequação ruim entre dispositivo, *dataset* e modelo de Árvore de Decisão, foi aplicada uma transformação conceitual baseada em uma matriz de rotação [19] diretamente nas colunas do *dataset*, deixando-as compatíveis com a orientação dos eixos do dispositivo estudado. Essa transformação envolveu a troca dos valores das colunas relativas aos eixos x e y , além da inversão do sinal dos valores das colunas do eixo z , ambas alterações foram aplicadas nas colunas dos eixos do giroscópio e do acelerômetro presentes no *dataset*. Embora simplificada, essa abordagem respeita o princípio da rotação e da transformação linear, ajustando os valores para análise consistente.

7 ANÁLISE E PRÉ-PROCESSAMENTO DOS DADOS

O Pré-Processamento possui uma função crucial na etapa de discretizar os dados do conjunto e, assim, tornar possível a aplicação efetiva em um modelo de classificação. Para isso foram adotadas duas principais ações, a primeira envolve uniformização temporal e limpeza de dados, a segunda envolve discretização dos dados baseada em análise estatística.

7.1 Uniformização Temporal e Limpeza de Dados

Inicialmente, foi necessário segmentar os eventos de "quedas" e "não quedas" com base nas informações fornecidas pelos autores do *dataset*, disponíveis no arquivo *fall_timestamps.csv*⁴. Esse arquivo apresenta os instantes de início e término das quedas, correspondentes às quatro fases descritas: pré-queda, impacto, ajuste corporal e pós-queda. Os *timestamps*, determinados manualmente, indicaram

que os eventos de queda têm, em média, duração de 3 segundos. Com base nessa informação, todas as sequências de quedas foram recortadas para esta faixa temporal de 3 segundos, concentrando-se no núcleo do movimento de queda. Essa abordagem foi adotada devido à irregularidade dos tempos nos arquivos completos originais, que variavam entre 10 e 20 segundos.

Para manter a consistência temporal entre "quedas" e "não quedas", os eventos de "não quedas" também foram recortados em divisões menores de 3 segundos. Nos casos em que a duração não era divisível por 3, foram removidos registros alternadamente do início e do fim. Essa remoção não foi prejudicial à integridade dos dados, pois os intervalos correspondentes geralmente representavam momentos em que o usuário ativava ou pausava a captura dos movimentos.

O recorte dos dados pode ser exemplificado nos gráficos presentes nas Figuras 3 e 4. A Figura 3 possui o gráfico de movimento total capturado, enquanto a Figura 4 possui os gráficos que mostram o evento após o recorte.

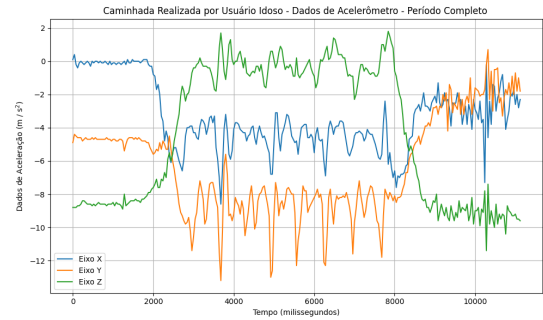


Figura 3: Dados de Acelerômetro - Captura Completa.

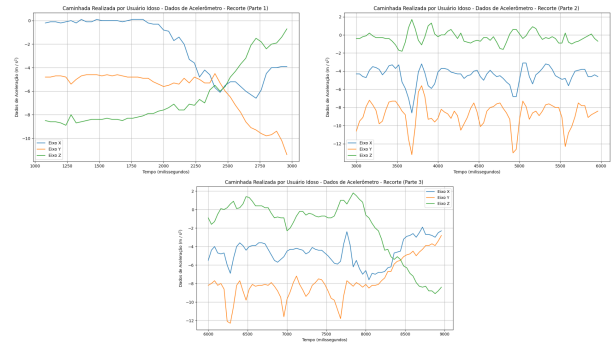


Figura 4: Dados de Acelerômetro - Recortes.

Para ajustar os intervalos de tempo irregulares fornecidos pelo *dataset* e melhorar a visualização dos gráficos, os valores da coluna de tempo foram alterados para uma faixa de múltiplos de 40 milissegundos, de acordo com a frequência escolhida de 25 Hz do *dataset* (1 envio a cada 40 milissegundos). Essa abordagem assegurou que o tempo de análise fosse consistente para eventos de "quedas" e "não quedas".

³<https://dev.fitbit.com/build/guides/sensors/accelerometer/>

⁴Disponível em: <https://github.com/joaojtmarques/WEDA-FALL>.

Além disso, a abordagem resultou em um aumento substancial no número de amostras para "não quedas", uma vez que foram recortadas, gerando um total de 2358 amostras.

7.2 Discretização baseada em Análise Estatística

O método de Pré-Processamento estatístico selecionado é pautado na extração de métricas e medidas básicas para compreensão da distribuição e dispersão do conjunto de dados, a saber, média (*mean*) - soma de todos os valores dividida pelo número de valores, representando o valor central do conjunto -, desvio padrão (*std*) - mede a dispersão dos valores em relação à média; quanto maior, mais espalhados os dados -, mínimo (*min*) - menor valor no conjunto de dados, representando o limite inferior observado - e máximo (*max*) - maior valor no conjunto de dados, indicando o limite superior registrado - seguindo o proposto por [25].

As características citadas foram extraídas dos arquivos com os dados brutos provenientes dos sensores inerciais e utilizadas como entrada do modelo de Árvore de Decisão.

8 DESENVOLVIMENTO DE MODELO DE ÁRVORE DE DECISÃO

A priori, o desenvolvimento do modelo implica na separação do conjunto de dados em treino e teste. Essa separação é fundamental para garantir que o desempenho do modelo seja avaliado de forma imparcial e confiável. Sabendo que a separação habitual ocorre respeitando a faixa de 20-30% dos dados para teste, para o desenvolvimento do modelo de Árvore de Decisão adotou-se 25% para teste e 75% para treinamento.

Além disso, o modelo foi configurado com a utilização dos seguintes parâmetros ⁵:

- *criterion='entropy'*: Define o critério utilizado para medir a qualidade da divisão nos nós da Árvore de Decisão. Neste caso, é usada a entropia, que avalia o grau de desordem ou impureza nos dados. A divisão busca minimizar a entropia, resultando em nós mais homogêneos.
- *random_state=42*: Controla a aleatoriedade do estimador, fixando a semente do gerador aleatório.
- *class_weight='balanced'*: Ajusta automaticamente os pesos das classes no caso de dados desbalanceados. Os pesos são calculados como o inverso da frequência das classes, garantindo que o modelo dê maior importância às classes menos representadas.
- *min_samples_split=5*: Especifica o número mínimo de amostras necessário para dividir um nó interno da árvore. Neste caso, um nó só será dividido se contiver pelo menos 5 amostras, evitando um aumento desnecessário de complexidade do modelo e a propensão ao *overfitting*.

9 MÉTRICAS DE AVALIAÇÃO (MODELO E CONSUMO ENERGÉTICO)

Para avaliação de desempenho do modelo, adotou-se a análise a partir da matriz de confusão, ao se manifestar como princípio base

para outras métricas definindo e identificando os valores como verdadeiros positivos, falsos negativos, falsos positivos e verdadeiros negativos.

A matriz de confusão é uma tabela que organiza os resultados de classificação de um modelo, comparando os valores previstos com os valores reais. Para problemas binários, ela é geralmente representada em um formato 2×2, enquanto para múltiplas classes é expandida para N×N, onde N é o número de classes. [35]

A partir dessa definição, podem ser extraídas métricas como:

- **Acurácia**: Mede a proporção total de classificações corretas.
- **Sensibilidade**: Mede a capacidade do modelo de identificar corretamente as amostras da classe positiva. Consiste na divisão do número de verdadeiros positivos pela soma entre falsos negativos e verdadeiros positivos.
- **Precisão**: Mede a proporção de previsões positivas que são realmente positivas. Consiste na divisão do número de verdadeiros positivos pela soma entre falsos positivos e verdadeiros positivos.
- **F1-Score**: Média harmônica entre precisão e sensibilidade, ideal para cenários com classes desbalanceadas. O *F1-Score* é calculado como duas vezes o produto entre a precisão e a sensibilidade, dividido pela soma desses mesmos valores.

Já para a marcação de consumo energético, foram utilizados dois valores principais:

- **Miliamperes-hora (mAh)**: Mede a quantidade total de carga elétrica consumida ao longo do tempo. Este valor indica a capacidade de energia usada pelo dispositivo e é fundamental para estimar a duração da bateria.
- **Amperes (A)**: Representa a corrente elétrica instantânea consumida pelo dispositivo em funcionamento. Este valor permite avaliar picos de consumo durante diferentes operações, auxiliando na identificação de estados ou atividades que exigem maior energia.

10 RESULTADOS DO MODELO

O algoritmo de Árvore de Decisão foi treinado com os dados e amostras do *dataset* processados como explicado na seção anterior. Os resultados obtidos foram apresentados na Tabela 1 e os valores aplicados nas métricas podem ser visualizados também através da matriz de confusão disponível na Figura 5.

Tabela 1: Avaliação do Modelo

Métrica	Valor
Acurácia	0.959
Sensibilidade	0.958
Precisão	0.961
F1-Score	0.959

⁵<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

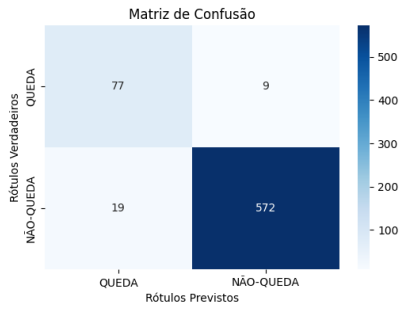


Figura 5: Matriz de confusão relativa à fase de teste

A partir da análise das métricas estabelecidas, o desempenho do algoritmo mostrou-se promissor. Ao compará-los com trabalhos correlatos verificou-se que o modelo de Árvore de Decisão desenvolvido obteve resultados próximos ou até mesmo superiores aos demais métodos - mesmo que em parte do grupo de trabalhos analisado tenham sido envolvidos modelos de *Machine Learning* mais robustos.

11 IMPLEMENTAÇÃO NO DISPOSITIVO

A implementação da Árvore de Decisão no dispositivo ESP32-S3 envolveu o desenvolvimento de uma solução que pudesse capturar os dados provenientes dos sensores, realizar o pré-processamento, e por fim, classificar localmente as amostras com os rótulos de "Queda" ou "Não Queda". Ademais, a solução foi implementada na linguagem de programação C++, utilizando a plataforma Arduino. Sabendo disso, os passos para realização das citadas etapas serão especificados nos próximos itens.

- (1) Captura de Dados dos Sensores: Para cada ciclo de análise, foram capturados 3 segundos de dados com uma frequência de 25Hz (1 envio a cada 40 milissegundos), totalizando 75 amostras por eixo de cada sensor. Para cada eixo, os dados foram armazenados num vetor para que posteriormente pudesse ser aplicado o pré-processamento. Destaca-se também que foram aplicadas as configurações previstas na seção 6.1.
- (2) Pré-Processamento no Dispositivo: Para cada janela de dados, foram calculadas as medidas necessárias para a análise estatística do conjunto (média, máximo, mínimo e desvio padrão). Tais medidas foram calculadas para os três eixos do acelerômetro (x,y,z) e do giroscópio (x,y,z), totalizando 12 entradas para a função da Árvore de Decisão.
- (3) Implementação da Árvore de Decisão: O modelo de Árvore de Decisão foi implementado no dispositivo por meio de sua transcrição para uma função programada. Essa função representa cada nó da árvore como uma condição, resultando em uma estrutura encadeada de blocos condicionais do tipo "if-else". A codificação foi viabilizada pela análise e visualização gráfica do modelo gerado durante o treinamento. Essa visualização foi obtida utilizando ferramentas gráficas do pacote *sklearn.tree*, como, por exemplo, a função *plot_tree*. A função recebe como parâmetros as métricas calculadas e, baseado nas regras de decisão, ao chegar em uma folha, retorna o rótulo "Queda" ou "Não Queda".

Por fim, acerca da captura de dados, é importante destacar que foi adotada uma janela intermediária de 1.5 segundos para complementar o ciclo de captura e classificação. Essa abordagem foi implementada para lidar com possíveis eventos que ocorram na transição entre ciclos consecutivos de captura. A janela intermediária funciona comparando 1.5 segundos finais do ciclo anterior com os 1.5 segundos iniciais do ciclo atual. Essa estratégia é particularmente importante devido a estratégia adotada para classificação das quedas, baseada em discretização de períodos e amostras.

12 ARQUITETURA E MÉTODOS DE VALIDAÇÃO

Para validar a integração do dispositivo com o modelo foi desenvolvida uma arquitetura de serviço e comunicação baseada no protocolo MQTT⁶ - um protocolo leve de comunicação baseado no modelo *publish-subscribe* (publicador-assinante), ideal para dispositivos IoT, uma vez que exige largura de banda mínima[3]. O protocolo utiliza um *Broker* central que recebe mensagens publicadas em tópicos específicos e as distribui para os clientes que se inscrevem nesses tópicos. No caso deste sistema, uma API desenvolvida utilizando Node.js⁷ assina o tópico para receber as publicações em tempo real.

No caso de estudo, para o sistema desenvolvido, o dispositivo ESP32 se conecta à rede Wi-fi e publica o rótulo da classificação no tópico definido. A API, por sua vez, recebe os dados publicados no *Broker* e salva os resultados transmitidos para análise. O esquema arquitetural do sistema pode ser visto na Figura 6.



Figura 6: Arquitetura de serviço desenvolvido para validação

13 RESULTADOS DA SOLUÇÃO

Os resultados obtidos foram organizados em diferentes etapas para avaliar a eficácia da solução desenvolvida. Primeiramente, realizou-se a validação do algoritmo em um ambiente controlado, com o objetivo de verificar sua capacidade de identificar corretamente as quedas e distinguir Atividades da Vida Diária (AVDs). Além disso, foram realizadas análises complementares, como o consumo energético do dispositivo, para avaliar sua viabilidade em contextos reais. A seguir, cada etapa é detalhada com base nos experimentos realizados e nos dados obtidos.

13.1 Validação do Algoritmo

A solução desenvolvida foi testada em um ambiente controlado, por meio de simulações de novas AVDs e quedas, detalhadas na Tabela 2. Para os testes, o dispositivo foi fixado no pulso esquerdo de um participante do sexo masculino, com 23 anos de idade, 70 kg de peso e 1,70 m de altura. A Tabela 2 apresenta uma breve descrição das simulações realizadas, a quantidade de ocorrências da ação e a duração de cada ocorrência.

⁶<https://mqtt.org/>

⁷<https://nodejs.org/pt>

Tabela 2: Tipos de atividades simuladas

Atividade	Repetições	Duração
Caminhada	2	30s
Saltar suavemente sem queda, com o objetivo de alcançar um objeto em altura	4	6s
Golpear a superfície de uma mesa com a mão	4	6s
Acenar com a mão em gesto de despedida	4	3s
Queda para trás ao caminhar causada por escorregão	4	12s
Queda para trás ao tentar sentar-se	2	9s
Queda para frente ao caminhar causada por um tropeço	2	12s
Queda lateral enquanto sentado, causada por desmaio ou adormecimento	4	9s

Para avaliar o desempenho do algoritmo, realizou-se a análise a partir dos retornos de rótulos enviados pelo dispositivo para o tópico no *Broker* e, consequentemente, o valor lido pela API, verificando sua adequação em relação aos verdadeiros positivos e negativos.

Durante os eventos classificados como quedas, foi verificado se o marcador "Queda" foi enviado corretamente pelo dispositivo. A ausência desse marcador em um evento de queda seria considerada um falso negativo, comprometendo a capacidade do sistema de detectar quedas de forma confiável.

Para atividades classificadas como AVDs, analisou-se se, durante todo o período de captura e movimentação, o dispositivo retornou exclusivamente rótulos de "Não Queda". Caso, em qualquer momento, fosse enviado o marcador "Queda" durante uma atividade que não representava uma queda, o evento foi configurado como um falso positivo.

Com base nos resultados obtidos no experimento, pode-se observar que o sistema apresentou um desempenho consistente ao identificar corretamente as quedas e distinguir as AVDs. Durante as simulações realizadas, o sistema não registrou nenhum falso positivo, mesmo em atividades que poderiam gerar dados semelhantes aos de uma queda, como saltos ou gestos com as mãos.

Apesar do desempenho assertivo alcançado na etapa de validação, é importante reconhecer possíveis limitações do experimento. A restrição a um número reduzido de tipos de atividades e quedas pode ter favorecido os resultados obtidos, uma vez que as condições simuladas não refletiram todas as possibilidades e complexidades de cenários encontrados no mundo real.

Além disso, embora os sensores do dispositivo tenham sido ajustados para alinhar seus dados ao *dataset* utilizado, essa adequação, mesmo aparentemente precisa, pode apresentar diferenças em contextos extremos ou menos controlados. Pequenas variações na calibração ou no comportamento dos sensores podem levar a discrepâncias em ambientes onde os movimentos não se comportam conforme o padrão esperado.

13.2 Consumo Energético

Para avaliar o consumo energético da solução desenvolvida, foi utilizado um wattímetro USB, um dispositivo capaz de medir o consumo de energia elétrica em tempo real, fornecendo dados como corrente (em amperes), energia consumida (em miliamperes-hora, mAh), entre outros.

O dispositivo foi conectado ao wattímetro USB por um período de 5 minutos em cada situação de teste. Os cenários avaliados foram projetados para medir o consumo de energia em três condições principais: funcionamento básico da placa, execução da lógica de tomada de decisão localmente e envio de dados pela rede.

Cada situação testada, bem como seu consumo, podem ser visualizados na Tabela 3.

Tabela 3: Análise de Consumo Energético.

Situação Teste	Corrente	Consumo
Funcionamento da Placa (sem código implementado)	0.07 A	6 mAh
Captura dos Dados e Tomada de Decisão (árvore implementada)	0.13 A	11 mAh
Captura e Envio de Dados Brutos pela Rede (1 envio a cada 40 ms)	0.20 A	17 mAh
Captura dos Dados, Tomada de Decisão e Envio de Resultados (1 envio a cada 3 segundos)	0.13 A ~ 0.18 A	13 mAh

A variação na corrente observada durante o cenário de Captura dos Dados, Tomada de Decisão e Envio de Resultados (0.13 A a 0.18 A) está relacionada às diferentes fases de operação do dispositivo. Durante os períodos em que apenas a captura dos dados e a execução da árvore de decisão local ocorrem, o consumo de corrente é menor, refletindo a baixa demanda de processamento. Já nos momentos em que o dispositivo realiza o envio dos resultados pela rede, a corrente aumenta devido ao maior esforço energético requerido pelo módulo Wi-Fi para transmitir os dados.

A análise energética comprova que a tomada de decisão local, com envio apenas dos resultados, é a solução mais eficiente do ponto de vista de consumo de bateria. O consumo total de energia nesse cenário (13 mAh) foi substancialmente menor do que o registrado para o envio contínuo de dados brutos (17 mAh).

14 CONCLUSÕES

Este estudo destacou a viabilidade e os desafios da implementação de algoritmos de aprendizado de máquina em dispositivos IoT vestíveis para a detecção local de quedas em idosos.

A implementação de uma Árvore de Decisão no ESP32 mostrou-se eficiente e adequada para as limitações de *hardware*, demonstrando que soluções de inteligência artificial podem ser ajustadas para operar em dispositivos com recursos computacionais limitados. As análises realizadas evidenciaram um equilíbrio positivo entre precisão e consumo energético, destacando o potencial desta solução para aplicações práticas no monitoramento de saúde.

Embora os resultados sejam promissores, as condições controladas dos testes indicam a necessidade de validações adicionais em cenários mais diversificados e dinâmicos. Futuros esforços poderão explorar abordagens como aprendizado incremental, detecção de padrões de postura, considerar a integração com novos tipos de sensores e avaliar a eficácia do sistema em ambientes reais, com usuários de diferentes perfis e níveis de mobilidade, objetivando uma menor incidência de falsos positivos e falsos negativos e, além disso, aprimorando a robustez da solução proposta em situações do cotidiano.

AGRADECIMENTOS

Este trabalho recebeu financiamento da Fapes, apoio do IFES Campus Cachoeiro de Itapemirim e Unimed Sul Capixaba.

REFERÊNCIAS

- [1] Luigi Atzori, Antonio Iera e Giacomo Morabito. 2010. The internet of things: a survey. *Computer Networks*, 54, 15, 2787–2805. doi: <https://doi.org/10.1016/j.comnet.2010.05.010>.
- [2] Marek Babiuch, Petr Foltýnek e Pavel Smutný. 2019. Using the esp32 microcontroller for data processing. Em *2019 20th International Carpathian Control Conference (ICCC)*, 1–6. doi: [10.1109/CarpathianCC.2019.8765944](https://doi.org/10.1109/CarpathianCC.2019.8765944).
- [3] Andrew Banks, Ed Briggs, Ken Borgendale e Rahul Gupta. 2019. MQTT Version 5.0. Andrew Banks, Ed Briggs, Ken Borgendale e Rahul Gupta, editores. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>. OASIS Standard. (Mar. de 2019). <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>.
- [4] Massimo Banzi e Michael Shiloh. 2022. *Getting Started with Arduino*. (3rd ed.). O'Reilly Media, Sebastopol, CA. ISBN: 978-1-491-95759-0.
- [5] Armir Bujari, Ombretta Gaggi, Claudio Enrico Palazzi e Giacomo Quadrio. 2018. Smart wearable sensors: analysis of a real case study. *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 37–41. <https://api.semanticscholar.org/CorpusID:56596566>.
- [6] Tiago Carneiro, Raul Victor Medeiros Da Nóbrega, Thiago Nepomuceno, Gui-Bin Bian, Victor Hugo C. De Albuquerque e Pedro Pedrosa Rebouças Filho. 2018. Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677–61685. doi: [10.1109/ACCESS.2018.2874767](https://doi.org/10.1109/ACCESS.2018.2874767).
- [7] Eduardo Casilari, Moisés Álvarez-Marco e Francisco García-Lagos. 2020. A study of the use of gyroscope measurements in wearable fall detection systems. *Symmetry*, 12, 4. doi: [10.3390/sym12040649](https://doi.org/10.3390/sym12040649).
- [8] Lin Chen, Rong Li, Hang Zhang, Lili Tian e Ning Chen. 2019. Intelligent fall detection method based on accelerometer data from a wrist-worn smart watch. *Measurement*, 140, 215–226. doi: <https://doi.org/10.1016/j.measurement.2019.03.079>.
- [9] Kimaya Desai, Pritam Mane, Manish Dsilva, Amogh Zare, Parth Shingala e Dayanand Ambawade. 2020. A novel machine learning based wearable belt for fall detection. Em *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, 502–505. doi: [10.1109/GUCON48875.2020.9231114](https://doi.org/10.1109/GUCON48875.2020.9231114).
- [10] A. Elhashar. 2018. Iot evolution towards a super-connected world. *Practical Guide to LTE-A, VoLTE and IoT*. doi: [10.1002/9781119063407.ch7](https://doi.org/10.1002/9781119063407.ch7).
- [11] I Arun Faisal, T Waluyo Purboyo e A Siswo Raharjo Ansori. 2019. A review of accelerometer sensor and gyroscope sensor in imu sensors on motion capture. *J. Eng. Appl. Sci*, 15, 3, 826–829.
- [12] Vanilson Fula e Plinio Moreno. 2024. Wrist-based fall detection: towards generalization across datasets. *Sensors*. doi: [10.3390/s24051679](https://doi.org/10.3390/s24051679).
- [13] Eva Garcia-Martín. 2017. *Extraction and energy efficient processing of streaming data*. Tese de doutoramento. Blekinge Tekniska Högskola.
- [14] Aurélien Gerón. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. English. O'Reilly Media; 856 pp. ISBN: 9781492032649.
- [15] Antônio Carlos Gil e outros. 2002. *Como elaborar projetos de pesquisa*. Vol. 4. Atlas São Paulo, [S.l.]
- [16] Gunther Gridling e Bettina Weiss. 2007. Introduction to microcontrollers. *Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group*, 25.
- [17] Yujin Hong, Ig-Jae Kim, Sang Ahn e Hyoung-Gon Kim. 2010. Mobile health monitoring system based on activity recognition using accelerometer. *Simulation Modelling Practice and Theory*, 18, (abr. de 2010), 446–455. doi: [10.1016/j.simpat.2009.09.002](https://doi.org/10.1016/j.simpat.2009.09.002).
- [18] Shang-Lin Hsieh, Chun-Che Chen, Shin-Han Wu e Tai-Wen Yue. 2014. A wrist-worn fall detection system using accelerometers and gyroscopes. Em *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, 518–523. doi: [10.1109/ICNSC.2014.6819680](https://doi.org/10.1109/ICNSC.2014.6819680).
- [19] Emőd Kovács. 2012. Rotation about an arbitrary axis and reflection through an arbitrary plane. Em *Annales Mathematicae et Informaticae*. Vol. 40, 175–186.
- [20] Stephen R. Lord, Catherine Sherrington e Hylton B. Menz. 2001. *Falls in Older People: Risk Factors and Strategies for Prevention*. First published 2001. Includes index. Cambridge University Press, Cambridge, United Kingdom. ISBN: 0-521-58964-9. <http://www.cambridge.org>.
- [21] Emna Mezghani, Ernesto Exposito e Khalil Drira. 2017. A model-driven methodology for the design of autonomic and cognitive iot-based systems: application to healthcare. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1, 3, 224–234. doi: [10.1109/TETCI.2017.2699218](https://doi.org/10.1109/TETCI.2017.2699218).
- [22] H. Mohajan. 2020. Quantitative research: a successful investigation in natural and social sciences. *Journal of Economic Development, Environment and People*. doi: [10.26458/jedep.v9i4.679](https://doi.org/10.26458/jedep.v9i4.679).
- [23] Ilkka Niiniluoto. 1993. The aim and structure of applied research. *Erkenntnis*, 38, 1, 1–21. doi: [10.1007/BF01129020](https://doi.org/10.1007/BF01129020).
- [24] Olukunle Ojetola, Elena I. Gaura e James Brusey. 2011. Fall Detection with Wearable Sensors—Safe (Smart Fall Detection). Em *Proceedings of the 7th International Conference on Intelligent Environments (IE)*. IEEE, Nottingham, UK. doi: [10.1109/IE.2011.38](https://doi.org/10.1109/IE.2011.38).
- [25] Camila Pereira de Oliveira. 2023. *Aprendizagem de máquina aplicada à detecção de queda de idosos*. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação). Instituto Federal do Espírito Santo (IFES), Cachoeiro de Itapemirim, (dez de 2023), 68. <https://repositorio.ifes.edu.br/handle/123456789/4027>.
- [26] Tasnim Hossain Orpa, Adil Ahnaf, Tareque Bashir Ovi e Mubdiul Islam Rizul. 2022. An iot based healthcare solution with esp32 using machine learning model. Em *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 1–6. doi: [10.1109/STI56238.2022.10103231](https://doi.org/10.1109/STI56238.2022.10103231).
- [27] Alexandros Pantelopoulos e Nikolaos G. Bourbakis. 2010. Prognosis—a wearable health-monitoring system for people at risk: methodology and modeling. *IEEE Transactions on Information Technology in Biomedicine*, 14, 3, 613–621. doi: [10.1109/TITB.2010.2040085](https://doi.org/10.1109/TITB.2010.2040085).
- [28] Mauricio César Pinto Pessoa, Geraldo Braz Junior e Tiago Bonini Borchardt. 2016. Home e-Care: Monitoramento de quedas em idosos através de smartwatches. Em *XV Congresso Brasileiro de Informática em Saúde (CBIS)*. SBIS, Goiânia, Brasil, 265–276. <http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis>.
- [29] Sarah Brandão Pinheiro, ALS SILVA, CJ CÁRDENAS e ML SILVA. 2015. A síndrome do pós-queda em idosos que sofrem fratura de fêmur. *Cadernos de Estudos e Pesquisas*, 19, 41.
- [30] Alvin Rajkomar, Jeffrey Dean e Isaac Kohane. 2019. Machine learning in medicine. *New England Journal of Medicine*, 380, 14, 1347–1358. PMID: 30943338. eprint: <https://www.nejm.org/doi/pdf/10.1056/NEJMra1814259>. doi: [10.1056/NEJMra1814259](https://doi.org/10.1056/NEJMra1814259).
- [31] Arkham Zahri Rakhman, Lukito Edi Nugroho, Widyawan e Kurnianingsih. 2014. Fall detection system using accelerometer and gyroscope based on smartphone. Em *2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering*, 99–104. doi: [10.1109/ICITACEE.2014.7065722](https://doi.org/10.1109/ICITACEE.2014.7065722).
- [32] Ali Akbar Sadri, Amir Masoud Rahmani, Morteza Saberikamarposhti e Mehdi Hosseinzadeh. 2021. Fog data management: a vision, challenges, and future directions. *Journal of Network and Computer Applications*, 174, 102882. doi: <https://doi.org/10.1016/j.jnca.2020.102882>.
- [33] Suranga Seneviratne, Yining Hu, Tham Nguyen, Guohao Lan, Sara Khalifa, Kanchana Thilakarathna, Mahbub Hassan e Aruna Seneviratne. 2017. A survey of wearable devices and challenges. *IEEE Communications Surveys Tutorials*, 19, 4, 2573–2620. doi: [10.1109/COMST.2017.2731979](https://doi.org/10.1109/COMST.2017.2731979).
- [34] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li e Lanyu Xu. 2016. Edge computing: vision and challenges. *IEEE Internet of Things Journal*, 3, 5, 637–646. doi: [10.1109/JIOT.2016.2579198](https://doi.org/10.1109/JIOT.2016.2579198).
- [35] Robert Susmaga. 2004. Confusion matrix visualization. Em *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM '04 Conference held in Zakopane, Poland, May 17–20, 2004*. Springer, 107–116.
- [36] Chin-Woo Tan e Sungsu Park. 2005. Design of accelerometer-based inertial navigation systems. *IEEE Transactions on Instrumentation and Measurement*, 54, 6, 2520–2530. doi: [10.1109/TIM.2005.858129](https://doi.org/10.1109/TIM.2005.858129).
- [37] Prajona Valsalan, Tariq Ahmed Barham Baomar e Ali Hussain Omar Baabood. 2020. Iot based health monitoring system. *Journal of critical reviews*, 7, 4, 739–743.
- [38] Waveshare. 2024. Esp32-s3-touch-lcd-1.69. Acesso em: 14 nov. 2024. (2024). <https://www.waveshare.com/wiki/ESP32-S3-Touch-LCD-1.69>.
- [39] Thomas C. Wong, John G. Webster, Henry J. Montoye e Richard Washburn. 1981. Portable accelerometer device for measuring human energy expenditure. *IEEE Transactions on Biomedical Engineering*, BME-28, 6, 467–471. doi: [10.1109/TBME.1981.324820](https://doi.org/10.1109/TBME.1981.324820).
- [40] A.D. Wood e J.A. Stankovic. 2002. Denial of service in sensor networks. *Computer*, 35, 10, 54–62. doi: [10.1109/MC.2002.1039518](https://doi.org/10.1109/MC.2002.1039518).