

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**GABRIEL CORTES PIRES LEOPOLSKI**

**LEONARDO ZUCCO**

**RODRIGO BALDISSERA CERQUEIRA LEITE**

**DESENVOLVIMENTO DE UMA SMARTBAND COM INTELIGÊNCIA  
ARTIFICIAL EMBARCADA PARA DETECÇÃO DE QUEDAS EM INDIVÍDUOS  
COM COMPROMETIMENTO COGNITIVO**

**CURITIBA**

**2026**

**GABRIEL CORTES PIRES LEOPOLSKI  
LEONARDO ZUCCO  
RODRIGO BALDISSERA CERQUEIRA LEITE**

**DESENVOLVIMENTO DE UMA SMARTBAND COM INTELIGÊNCIA  
ARTIFICIAL EMBARCADA PARA DETECÇÃO DE QUEDAS EM INDIVÍDUOS  
COM COMPROMETIMENTO COGNITIVO**

**Development of a Smartband with Embedded Artificial Intelligence for Fall  
Detection in Individuals with Cognitive Impairment**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Controle e Automação do curso de Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná(UTFPR).

Orientador: Prof. Dr. Amauri Amorin Assef

Coorientador: Prof. Dr. Fábio Pires Itturiet

**CURITIBA**

**2026**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**GABRIEL CORTES PIRES LEOPOLSKI  
LEONARDO ZUCCO  
RODRIGO BALDISSERA CERQUEIRA LEITE**

**DESENVOLVIMENTO DE UMA SMARTBAND COM INTELIGÊNCIA  
ARTIFICIAL EMBARCADA PARA DETECÇÃO DE QUEDAS EM INDIVÍDUOS  
COM COMPROMETIMENTO COGNITIVO**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Controle e Automação do curso de Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 22/junho/2026

---

Alexandre Francisco de Moraes Filho  
Doutorado  
Instituto Federal do Paraná

---

Renata Coelho Borges  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Adriano Ruseler  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Fábio Pires Itturiet  
Doutorado  
Universidade Tecnológica Federal do Paraná

**CURITIBA**

**2026**

## **AGRADECIMENTOS**

Os autores agradecem às instituições de fomento à pesquisa cujo apoio financeiro foi fundamental para a realização deste trabalho. À Fundação Araucária, à Secretaria de Estados da Ciência, Tecnologia e Ensino Superior do Paraná (SETI), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), à Universidade Tecnológica Federal do Paraná (UTFPR) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo suporte que viabilizou o desenvolvimento deste projeto.

Agradecemos imensamente aos professores orientadores Amauri Amorin Assef e Fábio Pires Itturiet, pela dedicação, paciência e orientação ao longo de todo o processo. A generosidade com que compartilharam seus conhecimentos foi determinante para a qualidade deste trabalho.

À nossa família, expressamos gratidão profunda pelo amor incondicional, pela compreensão nos momentos de ausência e por todo o incentivo que nos sustentou durante essa jornada. Sem esse alicerce, nada disso seria possível.

Aos nossos cônjuges, agradecemos pelo companheirismo, pela paciência e pelo afeto que renovaram nossas energias em cada etapa deste caminho. A presença de vocês foi, e sempre será, um ponto de equilíbrio essencial.

Aos amigos que acompanharam essa trajetória de perto, obrigado pelas palavras de encorajamento, pelos momentos de leveza e por tornarem esse percurso mais leve e significativo.

## RESUMO

O envelhecimento populacional e o aumento da prevalência de doenças neurodegenerativas têm intensificado a necessidade de soluções tecnológicas capazes de auxiliar no monitoramento da saúde e na prevenção de eventos adversos, como quedas, especialmente em indivíduos com comprometimento cognitivo. As quedas representam um dos principais fatores de morbidade e mortalidade nessa população, tornando essencial o desenvolvimento de sistemas confiáveis, acessíveis e de resposta rápida. Diante desse contexto, este trabalho teve como objetivo o desenvolvimento de uma smartband equipada com inteligência artificial embarcada para a detecção automática de quedas, visando contribuir para a segurança e o monitoramento contínuo de indivíduos com comprometimento cognitivo. A metodologia adotada envolveu a concepção do hardware do dispositivo vestível, utilizando sensores inerciais para a aquisição de dados de movimento, bem como o desenvolvimento do software embarcado responsável pelo processamento dos sinais e pela execução de algoritmos de aprendizado de máquina diretamente no dispositivo. Os dados coletados foram utilizados para o treinamento e validação de modelos de classificação capazes de distinguir eventos de queda de atividades cotidianas, priorizando baixo consumo de energia e processamento local, sem dependência constante de servidores externos. O sistema desenvolvido foi testado em diferentes cenários de movimento, permitindo a avaliação do desempenho do modelo quanto à precisão, confiabilidade e tempo de resposta. Os resultados demonstram que a utilização de inteligência artificial embarcada em dispositivos vestíveis é viável e eficaz para a detecção de quedas. O modelo atingiu acurácia de 95,2% e ROC-AUC de 0,951 na versão embarcada, com tempo médio de inferência de 7 ms e carga computacional de 0,7%. Nos testes práticos, o sistema detectou 100% das quedas frontais, 80% das quedas de costas e 70% das quedas laterais. Conclui-se que a smartband proposta constitui uma solução promissora para aplicações em monitoramento de saúde, com potencial de expansão para integração com sistemas de alerta e plataformas de acompanhamento remoto, contribuindo assim para a melhoria da qualidade de vida e da segurança de indivíduos com comprometimento cognitivo. Os resultados obtidos motivaram a submissão de um artigo científico ao Congresso Brasileiro de Engenharia Biomédica (CBEB) 2026.

Palavras-chave: detecção de quedas; inteligência artificial embarcada; dispositivos vestíveis; internet das coisas; monitoramento de idosos.

## ABSTRACT

Population aging and the increasing prevalence of neurodegenerative diseases have intensified the need for technological solutions capable of supporting health monitoring and preventing adverse events, such as falls, especially among individuals with cognitive impairment. Falls represent one of the main factors of morbidity and mortality in this population, making the development of reliable, accessible, and fast-response systems essential. In this context, this study aimed to develop a smartband equipped with embedded artificial intelligence for automatic fall detection, contributing to the safety and continuous monitoring of individuals with cognitive impairment. The proposed methodology involved the design of the wearable device hardware, using inertial sensors for motion data acquisition, as well as the development of embedded software responsible for signal processing and the execution of machine learning algorithms directly on the device. The collected data were used for training and validating classification models capable of distinguishing fall events from daily activities, prioritizing low power consumption and on-device processing, without constant dependence on external servers. The developed system was tested under different movement scenarios, enabling the evaluation of model performance in terms of accuracy, reliability, and response time. The results demonstrate that embedded artificial intelligence in wearable devices is feasible and effective for fall detection. The embedded model achieved 95.2% accuracy and a ROC-AUC of 0.951, with an average inference time of 7 ms and a computational load of 0.7%. In practical tests, the system detected 100% of forward falls, 80% of backward falls, and 70% of lateral falls. It is concluded that the proposed smartband represents a promising solution for health monitoring applications, with potential expansion for integration with alert systems and remote monitoring platforms, thereby contributing to improved quality of life and safety for individuals with cognitive impairment. The results obtained motivated the submission of a scientific article to the Brazilian Congress of Biomedical Engineering (CBEB) 2026.

Keywords: fall detection; embedded artificial intelligence; wearable devices; internet of things; elderly monitoring systems.

## LISTA DE FIGURAS

Figura 1 – Representação do <i>geofencing</i> baseado em Wi-Fi em ambiente residencial.	17
Figura 2 – Fluxo de dados em sistemas embarcados IoT.	19
Figura 3 – Diagrama de funcionamento do sistema.	21
Figura 4 – Diagrama detalhado do fluxo de dados, do dispositivo vestível ao aplicativo <i>Flutter</i> .	23
Figura 5 – Comparação entre os <i>hardwares</i> pesquisados em outubro de 2025.	24
Figura 6 – Esquema do fluxo de funcionamento resumido do protocolo MQTT.	26
Figura 7 – Fluxograma de funcionamento do sistema utilizando método K de M.	32
Figura 8 – <i>Dashboard</i> do Monitor de Segurança em três estados: a) operação normal, com <i>hardware</i> conectado e monitoramento ativo; b) queda detectada, exibindo alerta e opção de confirmação da ocorrência; e c) <i>hardware</i> desconectado, indicando perda de comunicação com a <i>smartband</i> .	34
Figura 9 – Esquema de funcionamento do sistema antes do ML e depois.	35
Figura 10 – Modelo 3D com medidas em milímetros da <i>case</i> para o microcontrolador.	36
Figura 11 – Pulseira posicionada verticalmente.	36
Figura 12 – Pulseira posicionada lateralmente.	36
Figura 13 – Foto lateral da pulseira em sua posição de operação.	37
Figura 14 – Foto vertical da pulseira em sua posição de operação.	37
Figura 15 – Sinais não filtrados do acelerômetro e do giroscópio ( $N = 1$ ).	39
Figura 16 – Sinais filtrados do acelerômetro e do giroscópio com filtro de média móvel ( $N = 5$ ).	39
Figura 17 – Sinais filtrados do acelerômetro e do giroscópio com filtro de média móvel ( $N = 10$ ).	40
Figura 18 – Sinais filtrados do acelerômetro e do giroscópio com filtro de média móvel ( $N = 20$ ).	40
Figura 19 – Visualização dos sinais do acelerômetro e do giroscópio durante o protocolo experimental de queda frontal.	42
Figura 20 – Visualização dos sinais do acelerômetro e do giroscópio durante o protocolo experimental de queda lateral.	42
Figura 21 – Visualização de movimento de caminhada contínua.	43
Figura 22 – Visualização de movimento de corrida leve contínua.	43
Figura 23 – Visualização de registro de um pulso realizado pelo indivíduo.	44
Figura 24 – Visualização de registro do indivíduo sentando em uma cadeira.	44
Figura 25 – Interface de configuração do <i>pipeline</i> no <i>Edge Impulse</i> , em que: a) define os parâmetros de aquisição da IMU (6 eixos, janela de 1000 ms, incremento de 250 ms e frequência de 20 Hz); b) apresenta o bloco de dados brutos, com os 6 eixos selecionados como entrada; c) configura o bloco de classificação com 2 classes de saída (normal e queda); e d) resume as características de saída do modelo.	45
Figura 26 – Arquitetura da rede neural configurada no <i>Edge Impulse</i> , composta por uma camada de entrada com 120 características, seguida de reorganização dos dados ( <i>reshape</i> ), camadas convolucionais 1D com operações de <i>pooling</i> , camada de <i>flatten</i> , camada densa com 32 neurônios e camada de saída para classificação em duas classes.	47
Figura 27 – Matriz de confusão do modelo quantizado em <i>int8</i> , exibindo o desempenho de classificação para as classes normal e queda no conjunto de validação.	50

<b>Figura 28 – Curvas de acurácia e de <i>loss</i> ao longo das épocas de treinamento do modelo supervisionado no <i>Edge Impulse</i>. .....</b>	<b>51</b>
<b>Figura 29 – Projeção bidimensional das amostras do conjunto de validação gerada pela técnica t-SNE, exibindo a separação entre as classes normal e queda e a distribuição dos erros de classificação. ....</b>	<b>52</b>

## LISTA DE TABELAS

<b>Tabela 1 – Resultados dos testes práticos de detecção de quedas. ....</b>	<b>55</b>
<b>Tabela 2 – Resumo do desempenho do modelo supervisionado na detecção dos diferentes tipos de queda. ....</b>	<b>55</b>
<b>Tabela 3 – Resumo dos principais parâmetros de desempenho computacional observados durante os testes embarcados.....</b>	<b>56</b>

## LISTA DE QUADROS

- Quadro 1 – Comparação entre protocolos de comunicação em IoT aplicados à saúde 18**  
**Quadro 2 – Síntese de estudos relacionados ao uso de dispositivos vestíveis e técnicas de aprendizagem de máquina na detecção de quedas ..... 20**

## LISTA DE SIGLAS E ABREVIATURAS

FCM *Firebase Cloud Messaging*.

GPS Sistema de Posicionamento Global.

IMU Unidade de Medida Inercial.

IoT *Internet of Things*.

LWT *Last Will and Testament*.

ML *Machine Learning*.

MQTT *Message Queuing Telemetry Transport*.

OMS Organização Mundial da Saúde.

QoS Nível de qualidade de serviço.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>1.1</b>	<b>OBJETIVOS</b>	<b>13</b>
1.1.1	Objetivo Geral	13
1.1.2	Objetivos Específicos	14
<b>1.2</b>	<b>ORGANIZAÇÃO DO TRABALHO</b>	<b>14</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>15</b>
<b>2.1</b>	<b>DISPOSITIVOS VESTÍVEIS E MONITORAMENTO NA SAÚDE</b>	<b>15</b>
2.1.1	Monitoramento de idosos e quedas	15
<b>2.2</b>	<b>TÉCNICAS DE DETECÇÃO DE QUEDAS</b>	<b>15</b>
<b>2.3</b>	<b>SENSORES DE UNIDADE INERCIAL</b>	<b>16</b>
<b>2.4</b>	<b>GEOFENCING E MONITORAMENTO DE PERÍMETRO</b>	<b>17</b>
<b>2.5</b>	<b>PROTOCOLOS DE COMUNICAÇÃO IOT NA SAÚDE</b>	<b>17</b>
<b>2.6</b>	<b>MICROCONTROLADORES E SISTEMAS EMBARCADOS EM IOT</b>	<b>18</b>
<b>2.7</b>	<b>ESTUDOS CORRELATOS</b>	<b>19</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>21</b>
<b>3.1</b>	<b>ARQUITETURA DO SISTEMA</b>	<b>21</b>
<b>3.2</b>	<b>COMPONENTES DO SISTEMA</b>	<b>24</b>
3.2.1	<i>Hardware</i>	24
3.2.2	Protocolo de comunicação	25
3.2.2.1	<u>MQTT</u>	25
3.2.3	<i>Firmware</i>	27
3.2.3.1	<u>Manipulação e tratamento de dados</u>	28
3.2.4	<i>Machine Learning</i> Embarcado	28
3.2.4.1	<i>Edge Impulse</i>	29
3.2.4.2	<u>Classificação Supervisionada</u>	29
3.2.4.3	<u>Método de verificação baseado em janelas</u>	30
3.2.5	Interface gráfica	32
<b>3.3</b>	<b>METODOLOGIA EXPERIMENTAL</b>	<b>34</b>
3.3.1	PREPARAÇÃO PARA EXPERIMENTAÇÃO	35
<b>3.4</b>	<b>COLETA INICIAL DE DADOS</b>	<b>38</b>
3.4.1	ANÁLISE DOS DADOS OBTIDOS	41
3.4.1.1	<u>CATEGORIZAÇÃO DE MOVIMENTOS EM DADOS TABULARES</u>	42
3.4.1.2	<u>Configuração para Classificação Supervisionada</u>	44
<b>4</b>	<b>RESULTADOS</b>	<b>49</b>
<b>4.1</b>	<b>Desempenho do Modelo durante Treinamento</b>	<b>49</b>
4.1.1	Resultados dos Testes Práticos	52
4.1.1.1	Desempenho na Detecção de Quedas	52

4.1.1.2	Desempenho do <i>geofencing</i> .....	55
4.1.1.3	Tempo de Processamento e Carga Computacional .....	56
4.1.2	Comparação com Trabalhos Relacionados .....	57
4.1.2.1	<i>Hardware</i> e Posicionamento do Sensor .....	57
4.1.2.2	Desempenho dos Modelos de Classificação .....	58
4.1.2.3	Falsos Positivos .....	58
4.1.2.4	Implementação Embarcada .....	59
<b>5</b>	<b>CONCLUSÃO</b> .....	<b>60</b>
<b>5.1</b>	<b>SUGESTÕES DE TRABALHOS FUTUROS</b> .....	<b>60</b>
	<b>REFERÊNCIAS</b> .....	<b>61</b>
	<b>APÊNDICE A – Wrist-Worn Smartband with Embedded Artificial Intelligence for Fall Detection in Individuals with Cognitive Impairment</b> .....	<b>64</b>

## 1 INTRODUÇÃO

O envelhecimento populacional é uma realidade crescente em todo o mundo e, de forma ainda mais acelerada, no Brasil. De acordo com a Organização Mundial da Saúde (OMS), a proporção de pessoas com 60 anos ou mais deve dobrar até 2050, tornando essa faixa etária um dos principais focos das políticas públicas de saúde. Dentro desse grupo, estima-se que 8,5% dos idosos brasileiros convivam com algum tipo de demência, o que representa aproximadamente 1,8 milhão de pessoas, podendo chegar a 5,7 milhões em 2050 (Ministério da Saúde do Brasil, 2024).

As doenças neurodegenerativas estão entre as principais causas de comprometimento cognitivo, afetando a memória, a orientação espacial e a autonomia dos indivíduos. Essa condição aumenta substancialmente o risco de quedas, que, segundo a OMS, atingem cerca de um terço dos idosos globalmente a cada ano (World Health Organization, 2021). Quando há déficits cognitivos, a prevalência é ainda maior: estudos apontam índices que variam de 40% a 60% de quedas anuais em idosos com demência (Bruhn *et al.*, 2021; Yu *et al.*, 2022). No Brasil, uma pesquisa realizada em Juiz de Fora (MG) identificou 42% de frequência de quedas entre idosos com comprometimento cognitivo, evidenciando a gravidade da situação (Santos *et al.*, 2017).

Além das quedas, comportamentos como *wandering* — caracterizados por episódios de desorientação, fuga de casa ou de instituições — também estão presentes em uma parcela significativa dessa população, aumentando a vulnerabilidade e o risco de acidentes ou desaparecimentos. Embora ainda pouco quantificados no Brasil, estudos internacionais reforçam a necessidade de estratégias de monitoramento e prevenção desse tipo de ocorrência (Wu *et al.*, 2024).

Diante desse cenário, torna-se urgente o desenvolvimento de soluções tecnológicas que integrem sensores e algoritmos de *Machine Learning* (ML), capazes de monitorar continuamente a rotina do idoso. Um dispositivo vestível em formato de pulseira pode ser utilizado para detectar automaticamente quedas e identificar quando o usuário ultrapassa o perímetro da residência, fornecendo alertas preventivos e ampliando a segurança, ao mesmo tempo em que oferece maior tranquilidade a familiares e cuidadores.

### 1.1 OBJETIVOS

Os objetivos deste trabalho estão listados a seguir.

#### 1.1.1 Objetivo Geral

Projetar uma *smartband* com ML e um aplicativo móvel para monitoramento contínuo, capaz de detectar quedas e saídas de perímetro, gerando alertas imediatos para aumentar a segurança de idosos com déficit cognitivo.

### 1.1.2 Objetivos Específicos

- Configurar e calibrar os sensores inerciais (acelerômetro e giroscópio) do dispositivo vestível para coleta de dados de movimento.
- Construir um banco de dados experimental, incluindo quedas simuladas e atividades diárias, com protocolo de coleta e rotulagem.
- Projetar e treinar modelos de *Machine Learning* (ML) para detecção de quedas, avaliando métricas como precisão, sensibilidade e taxa de falso alarme.
- Otimizar e embarcar o modelo de ML para execução no próprio dispositivo.
- Desenvolver um mecanismo de *geofencing* que permita configurar a área de detecção enquanto o dispositivo estiver conectado ao Wi-Fi doméstico, definindo-a como o perímetro aceitável.
- Criar um aplicativo móvel para monitoramento em tempo real, registro do histórico e envio de notificações a cuidadores.
- Testar o protótipo em situações controladas, verificando a confiabilidade do sistema e a usabilidade para o público-alvo.
- Documentar as limitações e propor melhorias futuras, incluindo a integração com outros sensores e a expansão de funcionalidades.

## 1.2 ORGANIZAÇÃO DO TRABALHO

O presente Trabalho de Conclusão de Curso foi estruturado em cinco capítulos principais. No Capítulo 1, apresenta-se a introdução ao tema, bem como os objetivos gerais e específicos do projeto. O Capítulo 2 reúne a fundamentação teórica, abordando os conceitos essenciais relacionados a dispositivos vestíveis, detecção de quedas, sensores inerciais, detecção de fuga de perímetro pré-delimitado, protocolos de comunicação em IoT e sistemas embarcados. O Capítulo 3 descreve detalhadamente os materiais e métodos utilizados, incluindo a arquitetura do sistema, a seleção dos componentes, o desenvolvimento do *firmware*, o processo de implementação do ML embarcado e a metodologia experimental adotada. O Capítulo 4 apresenta os resultados parciais obtidos até esta etapa do desenvolvimento, destacando as análises práticas, as limitações observadas e o desempenho inicial dos modelos utilizados. Por fim, o Capítulo 5 apresenta as conclusões do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são abordados os conceitos que embasam o projeto, incluindo dispositivos vestíveis para saúde, técnicas de detecção de quedas, *geofencing*, protocolos de comunicação em *Internet of Things* (IoT) e a justificativa da escolha dos componentes de *hardware*.

### 2.1 DISPOSITIVOS VESTÍVEIS E MONITORAMENTO NA SAÚDE

Os dispositivos vestíveis (*wearables*) são sistemas eletrônicos portáteis que podem ser assimilados ao corpo ou às vestimentas e capturam dados fisiológicos ou de movimento, por meio de sensores como Unidade de Medida Inercial (IMU), sensores de frequência cardíaca ou até mesmo termômetros. Esses equipamentos têm sido amplamente explorados na saúde para monitoramento contínuo e remoto, possibilitando intervenções mais rápidas e acompanhamento (Olmedo-aguirre *et al.*, 2022).

Na área da saúde, os *wearables* têm sido aplicados em diferentes contextos, como o monitoramento da atividade física, o acompanhamento da reabilitação e a emissão de alertas em situações emergenciais. Sua utilização também favorece a implementação de modelos de atenção preventiva, nos quais o paciente passa a ser monitorado mesmo fora de instituições médicas, reduzindo custos e sobrecarga nos sistemas de saúde (Gonzaga; Pereira; Santos, 2022).

#### 2.1.1 Monitoramento de idosos e quedas

No contexto dos idosos, os *wearables* assumem papel estratégico, pois combinam mobilidade e usabilidade com funcionalidades de segurança. Esses dispositivos podem capturar alterações sutis no padrão de movimento, detectar quedas súbitas e monitorar deslocamentos para fora de áreas pré-definidas, emitindo alertas automáticos para cuidadores ou familiares. Estudos demonstram que IMUs conseguem extrair parâmetros de marcha e de equilíbrio que se correlacionam diretamente com o risco de queda (Santos *et al.*, 2024).

Além disso, o uso de dispositivos vestíveis contribui para reduzir o tempo de resposta a emergências e aumentar a autonomia do idoso. Revisões recentes destacam que tecnologias desse tipo fortalecem tanto a prevenção de acidentes quanto a confiança de familiares e cuidadores no processo de envelhecimento ativo (Ferreira; Ribeiro; Santos, 2022).

### 2.2 TÉCNICAS DE DETECÇÃO DE QUEDAS

A detecção de quedas por meio de dispositivos vestíveis baseia-se principalmente no uso de sensores inerciais, como acelerômetros e giroscópios, que registram padrões de movimento e variações bruscas na aceleração do corpo. Esses dados podem ser analisados para

diferenciar atividades cotidianas de episódios de queda, fornecendo informações valiosas para sistemas de monitoramento remoto (Ferreira; Ribeiro; Santos, 2022).

Com o avanço das técnicas computacionais, modelos de aprendizado de máquina têm sido incorporados a esses sistemas, possibilitando maior precisão na identificação de eventos críticos. Tais modelos permitem explorar parâmetros derivados dos sensores, como estabilidade de marcha, velocidade angular e impacto, que estão diretamente associados ao risco de quedas em idosos (Santos *et al.*, 2024).

Dessa forma, as técnicas atuais combinam *hardware* de baixo custo com algoritmos inteligentes, resultando em sistemas de monitoramento mais confiáveis e capazes de reduzir tanto os falsos positivos quanto os falsos negativos na detecção de quedas. Além disso, por meio do ML, é possível treinar redes neurais para identificar movimentação normal ou situações anômalas. Enquanto os casos normais correspondem às variações normais de velocidade e posição da pessoa, os casos anômalos correspondem a eventos característicos de queda. Essa abordagem possibilita um reconhecimento mais preciso, reforçando a eficiência do sistema.

### 2.3 SENSORES DE UNIDADE INERCIAL

A escolha dos sensores é um dos aspectos mais importantes no desenvolvimento de dispositivos vestíveis voltados à detecção de quedas. Para essa finalidade, destacam-se os sensores inerciais, compostos por acelerômetros e giroscópios, frequentemente integrados em um único módulo conhecido como IMU.

O acelerômetro mede a aceleração linear em três eixos (X, Y e Z), permitindo identificar movimentos bruscos e impactos. Entretanto, o uso exclusivo do acelerômetro pode gerar falsos positivos, especialmente em situações de movimentos abruptos que não correspondem a uma queda real, como deitar-se rapidamente ou abaixar-se para pegar um objeto. Por isso, é comum associar o acelerômetro ao giroscópio, sensor que mede a velocidade angular e fornece informações sobre a rotação do corpo em torno de seus eixos.

A combinação desses dois sensores possibilita uma análise mais robusta do movimento, pois o giroscópio complementa o acelerômetro ao distinguir entre mudanças de orientação e deslocamentos lineares. Essa integração permite identificar padrões complexos, como a inclinação súbita seguida de impacto, que caracteriza com maior precisão uma queda real (Santos *et al.*, 2024).

Os dados obtidos pelos sensores de aceleração e velocidade angular permitem identificar de forma mais confiável as mudanças bruscas de movimento e de orientação do corpo. Enquanto o acelerômetro detecta variações lineares associadas a impactos ou quedas, o giroscópio fornece informações sobre a rotação e a inclinação, possibilitando distinguir uma queda real de movimentos cotidianos. Dessa forma, o uso conjunto desses sensores em uma IMU resulta em medições mais completas e robustas, mantendo baixo custo e simplicidade de implementação.

## 2.4 GEOFENCING E MONITORAMENTO DE PERÍMETRO

O termo *geofencing* refere-se ao estabelecimento de "cercas virtuais" que delimitam áreas geográficas específicas, dentro das quais um indivíduo ou objeto pode se deslocar livremente. Essa técnica pode ser implementada de diferentes formas, a partir de tecnologias como GPS, redes Wi-Fi ou *Bluetooth*. Em todos os casos, o objetivo é emitir alertas automáticos quando o usuário ultrapassa os limites definidos. A Figura 1 ilustra esse conceito, em que o idoso representado pela cor verde está dentro da área de alcance do Wi-Fi e encontra-se em situação segura, enquanto a saída do perímetro resulta na emissão de um alerta automático para cuidadores ou familiares, representado pela cor vermelha.

No cuidado de idosos, o *geofencing* assume papel essencial, especialmente para aqueles que apresentam episódios de desorientação e comportamento de *wandering*. Nessas situações, o idoso pode sair inadvertidamente de sua residência ou de instituições de acolhimento, expondo-se a riscos de acidentes ou desaparecimentos. A utilização de *wearables* com sistemas de cercamento virtual possibilita detectar essas situações em tempo real, ampliando a segurança e a autonomia do idoso, trazendo maior tranquilidade a familiares e cuidadores (Wu *et al.*, 2024). Vale ressaltar que, embora o foco do presente trabalho seja a população idosa, tal aplicação é igualmente válida para qualquer indivíduo que apresente limitações de natureza cognitiva.

**Figura 1 – Representação do *geofencing* baseado em Wi-Fi em ambiente residencial.**



Fonte: Autoria própria.

## 2.5 PROTOCOLOS DE COMUNICAÇÃO IOT NA SAÚDE

Na área de IoT, a escolha do protocolo de comunicação é determinante para a eficiência e a confiabilidade dos sistemas. Entre as opções mais utilizadas destacam-se *Bluetooth*, LoRa e Wi-Fi, cada um apresentando vantagens específicas em termos de alcance, consumo energético e capacidade de transmissão de dados (Ferreira; Ribeiro; Santos, 2022). O Quadro 1

apresenta uma comparação entre esses protocolos, evidenciando suas principais características.

No contexto da saúde, o Wi-Fi se mostra amplamente empregado por oferecer maior largura de banda, compatibilidade com dispositivos já consolidados e facilidade de integração com servidores locais e plataformas na nuvem. Apesar de apresentar consumo energético superior a alternativas de baixo consumo, sua ampla disponibilidade em ambientes residenciais e institucionais torna-o uma solução prática e acessível para sistemas de monitoramento contínuo (Santos *et al.*, 2024). Além disso, a integração com aplicativos móveis amplia as funcionalidades desses sistemas, permitindo que cuidadores e familiares recebam notificações em tempo real diretamente em seus *smartphones*. Essa interface garante maior acessibilidade e rapidez na resposta a situações críticas, consolidando o papel dos dispositivos vestíveis como ferramentas eficazes de apoio à saúde preventiva.

**Quadro 1 – Comparação entre protocolos de comunicação em IoT aplicados à saúde, incluindo valores médios de consumo em mA.**

Protocolo	Alcance típico	Consumo de energia	Taxa de transmissão
Bluetooth	Até 100 m	Muito baixo (~10–20 mA)	~1 Mbps
LoRa	Até 10 km (áreas abertas)	Muito baixo (~30–40 mA)	0,050 Mbps
Wi-Fi	50–100 m (interno)	Médio/alto (~120–250 mA)	Até centenas de Mbps

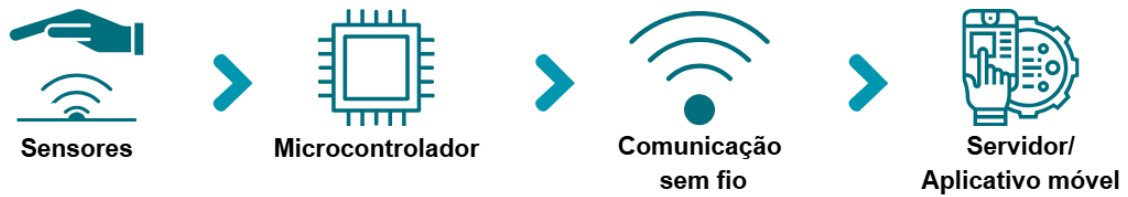
Fonte: Adaptado de (NORDIC SEMICONDUCTOR, 2021) (BLUETOOTH SIG, 2016) (SEMTECH CORPORATION, 2022) (LORA ALLIANCE, 2020) (ESPRESSIF SYSTEMS, 2023) (STMICROELECTRONICS, 2020).

## 2.6 MICROCONTROLADORES E SISTEMAS EMBARCADOS EM IOT

Os microcontroladores são circuitos integrados que reúnem em um único chip unidade de processamento, memória e interfaces de entrada e saída. Diferenciam-se dos microprocessadores por integrarem periféricos como conversores analógico-digitais e módulos de comunicação, o que possibilita a execução de tarefas específicas de forma mais eficiente e com menor consumo energético. Essas características os tornam fundamentais em aplicações que exigem processamento em tempo real e operação contínua em dispositivos compactos (Verma *et al.*, 2022).

No contexto de IoT, os microcontroladores assumem papel central na aquisição, pré-processamento e transmissão de dados provenientes de sensores. Em dispositivos vestíveis, destacam-se pela combinação de baixo consumo, conectividade sem fio (Wi-Fi, *Bluetooth*) e integração simplificada com servidores locais ou aplicativos móveis (Olmedo-aguirre *et al.*, 2022). A Figura 2 ilustra de forma simplificada esse fluxo, no qual os sensores coletam informações, o microcontrolador processa os dados e a transmissão ocorre para aplicações móveis ou plataformas em nuvem.

Figura 2 – Fluxo de dados em sistemas embarcados IoT.



Fonte: Autoria própria.

Além disso, as restrições de memória e consumo em microcontroladores exigem soluções de inferência enxuta, de modo que algoritmos de ML possam rodar diretamente no dispositivo sem depender inteiramente de servidores remotos. Tais abordagens demonstram como modelos leves de redes neurais, combinados com otimizações de uso de memória e energia, tornam possível executar inferência em microcontroladores comuns, ampliando a autonomia do sistema embarcado (Lin *et al.*, 2020).

## 2.7 ESTUDOS CORRELATOS

Diversos estudos exploram soluções baseadas em dispositivos vestíveis para detecção automática de quedas, com ênfase especial em sensores posicionados no pulso e no uso de técnicas de ML para análise de movimento. Essas pesquisas ajudam a contextualizar o desenvolvimento do presente projeto, permitindo comparar arquiteturas, sensores utilizados, algoritmos de classificação, limitações e avanços tecnológicos. Em geral, os trabalhos combinam acelerômetros e giroscópios com ML, avaliando desde modelos clássicos (*Random Forest, SVM, k-NN*) até arquiteturas avançadas de *deep learning*.

O Quadro 2 sumariza cinco estudos diretamente relacionados ao escopo do projeto: detecção de quedas, uso de *wearables* posicionados no punho ou na cintura e aplicação de ML. Essa síntese permite identificar o estado da arte, destacar lacunas e justificar as escolhas técnicas adotadas no desenvolvimento da nossa solução.

**Quadro 2 – Síntese de estudos relacionados ao uso de dispositivos vestíveis e técnicas de aprendizagem de máquina na detecção de quedas.**

Estudo	Tipo de dispositivo / posição	Algoritmos de ML	Principais contribuições
(Marques <i>et al.</i> , 2023)	<i>Wearable</i> no punho ( <i>wrist device</i> ).	Avaliação <i>offline</i> de múltiplos algoritmos clássicos de ML.	Seleção de modelo ótimo e experimentos sob diferentes cenários de queda.
(Nunez <i>et al.</i> , 2024)	<i>Wearables</i> na cintura e no punho.	<i>Random Forest</i> e modelos tradicionais.	Compara desempenho entre posições (cintura × punho), auxiliando na decisão do posicionamento ideal e identifica o modelo ótimo de ML.
(Jang <i>et al.</i> , 2024)	<i>Wearables</i> na cintura.	<i>Dual-stream CNN + Self-Attention</i> .	<i>Deep learning</i> que melhora reconhecimento de padrões complexos de queda.
(Li <i>et al.</i> , 2025)	Revisão ampla de <i>wearables</i> (inclui sensores de punho).	Modelos clássicos e redes neurais modernas.	Revisão de 10 anos com desafios, evolução tecnológica e tendências (energia, miniaturização, IA embarcada).
(Volochtchuk; Bassai, 2023)	<i>Wearable</i> no peito (colar).	Modelos de <i>Machine Learning</i> supervisionado e Redes Neurais via <i>Edge Impulse</i> .	Desenvolvimento de sensor de monitoramento de quedas com foco em autonomia, eficiência de processamento e baixo custo.

**Fonte: Autoria própria, baseado nos estudos referenciados.**

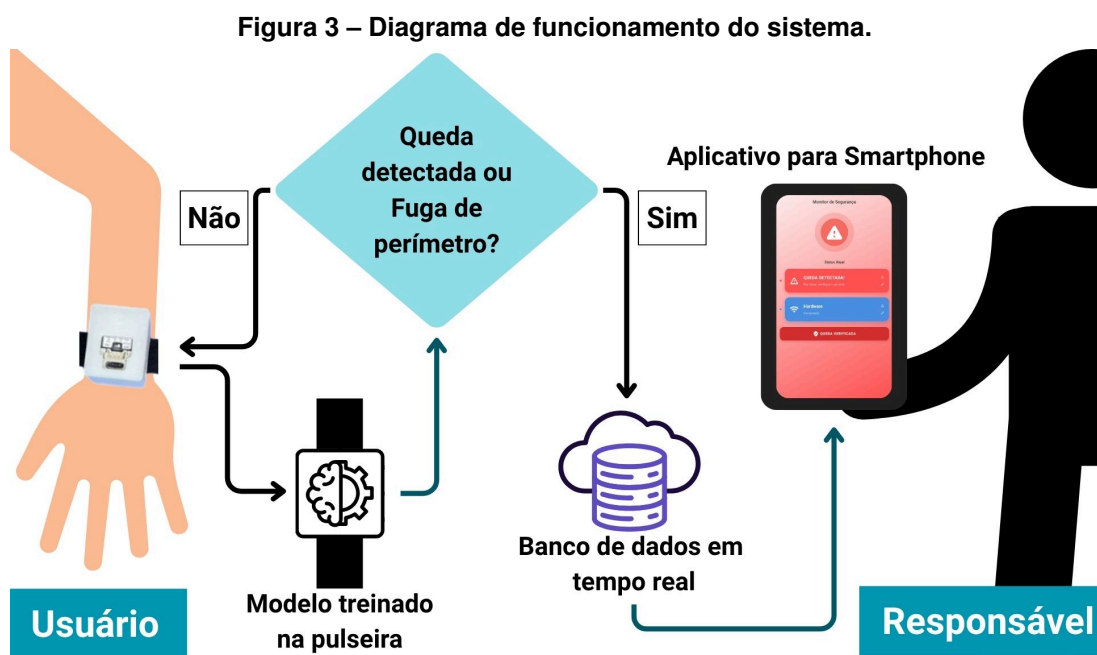
O presente trabalho, por sua vez, teve a pretensão de aplicar o aprendizado de máquina em um sistema embarcado, além de propor uma solução vestível focada no pulso, buscando equilibrar o conforto do usuário com a alta precisão na detecção de anomalias. O diferencial reside na execução em tempo real diretamente no *hardware* M5Stack, utilizando o *Edge Impulse* para otimizar o consumo de recursos e o protocolo MQTT para a comunicação de alertas imediatos, funcionalidades que serão apresentadas posteriormente no capítulo 3. Esta abordagem permite não apenas a detecção de quedas em idosos, mas também o monitoramento proativo de pacientes com necessidades de natureza cognitiva, garantindo uma assistência contínua e não invasiva.

### 3 MATERIAIS E MÉTODOS

Neste capítulo são apresentadas as etapas realizadas para a construção e escolha dos componentes do sistema: dispositivo de coleta, protocolos de comunicação, classificação e armazenamento de dados e interface gráfica para visualização dos mesmos. Este capítulo apresenta também como foram realizados os testes em campo.

#### 3.1 ARQUITETURA DO SISTEMA

O diagrama de funcionamento do sistema é apresentado na Figura 3.



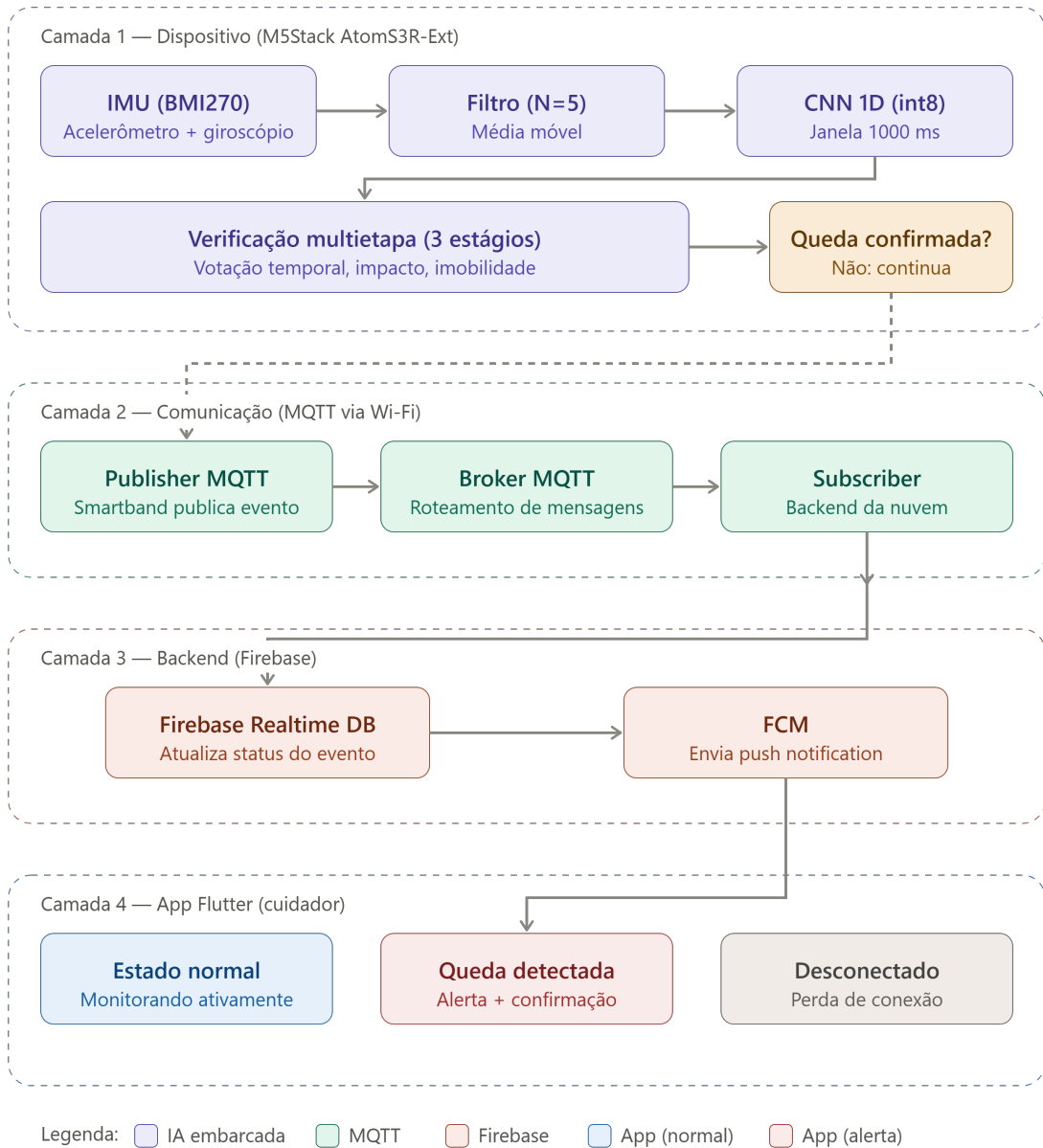
Fonte: Autoria própria.

O sistema proposto realiza aquisição periódica de dados inerciais, processamento embarcado e transmissão estruturada via *Message Queuing Telemetry Transport* (MQTT), que é descrito a seguir na subseção de protocolo de comunicação. Inicialmente, a unidade embarcada é vestida pelo portador, a qual amostrará acelerações e velocidades angulares com seus sensores nativos a uma frequência de amostragem  $f_s$ , sendo esses valores filtrados com o objetivo de minimizar ruídos/trepidações fora da faixa de interesse. Após a amostragem, sobre os dados é aplicado um classificador de movimento através de um modelo de ML que classificará o movimento como queda ou não queda. Durante condições normais, ou seja, realizando atividades cotidianas, nenhum dado é publicado ao *broker*, reduzindo tráfego e consumo energético. Entretanto, quando o classificador detecta uma queda, o *firmware* dispara um sinal de comunicação: publica apenas uma mensagem compacta indicando a ocorrência do evento, sem transmitir sinais brutos, permitindo que o *backend* do aplicativo registre o incidente, acione notificações e atualize o estado do usuário na plataforma. Além da detecção de quedas, o dispositivo

executa rotinas periódicas de *geofencing* baseadas em Wi-Fi. A partir de varreduras do ambiente, o sistema identifica a presença ou ausência do conjunto de pontos de acesso associados à residência, determinando se o usuário permanece dentro do perímetro configurado. Caso o dispositivo detecte a saída do perímetro, uma mensagem específica é enviada ao *backend*, permitindo ações rápidas de acompanhamento.

Enquanto a Figura 3 apresenta uma visão conceitual do funcionamento do sistema, a Figura 4 detalha o fluxo completo de dados desde a aquisição no dispositivo vestível até a exibição dos eventos no aplicativo do cuidador. O diagrama está organizado em quatro camadas: o dispositivo embarcado, responsável pela aquisição dos sinais da IMU, filtragem, inferência do modelo de ML e verificação multietapa; a camada de comunicação via MQTT sobre Wi-Fi; o *backend* na nuvem baseado no Firebase; e o aplicativo *Flutter* do cuidador, que reflete os diferentes estados do sistema.

**Figura 4 – Diagrama detalhado do fluxo de dados, do dispositivo vestível ao aplicativo Flutter.**



**Fonte: Autoria própria.**

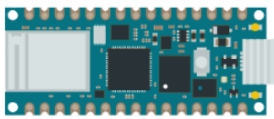

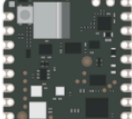
## 3.2 COMPONENTES DO SISTEMA

Nesta subseção, detalham-se os componentes do sistema, tanto *software* como *hardware*, sendo a parte de *software* dividida entre *firmware*, ML, protocolo de comunicação e interface gráfica.

### 3.2.1 Hardware

Como premissa, o sistema necessita ser compacto para possibilitar que o usuário possa portá-lo com facilidade e não ter acesso às funcionalidades do sistema, possibilitando que apenas o responsável as manuseie. Inicialmente, foram selecionadas três placas de desenvolvimento como possíveis candidatas: Arduino Nicla Sense ME, Arduino Nano 2040 e M5Stack AtomS3R-Ext. As placas escolhidas compartilham três características fundamentais: presença de Unidade de Medida Inercial (IMU) embutida, tamanho reduzido e ausência de tela. O quadro da Figura 5 mostra de forma resumida as principais diferenças entre os microcontroladores citados.

**Figura 5 – Comparação entre os *hardwares* pesquisados em outubro de 2025.**

	 Arduino Nano 2040	 M5Stack AtomS3R-Ext	 Arduino Nicla Sense ME
Wi-Fi	✓	✓	✗
Dimensões (mm)	43,18 x 17,77	24 x 24 x 12,9	22,85 x 22,86
Custo (US\$)	35,00	14,50	51,56
IMU	✓	✓	✓
OBS	Dimensões maiores	Baixo custo	Sem conectividade Wi-Fi

**Fonte: Autoria própria.**

A Arduino Nicla Sense ME, apesar de seu tamanho extremamente compacto de 22,85 × 22,86 × 3 mm, não possui conectividade Wi-Fi, recurso essencial para a operação do sistema proposto, visto que a grande maioria das residências possui roteador de conexão Wi-Fi. Além disso, apresenta custo elevado, em torno de US\$ 51,56 (orçado em outubro de 2025), o que torna sua adoção inviável para o projeto.

A placa Arduino Nano 2040 foi a segunda opção considerada. Diferentemente da Arduino Nicla Sense ME, essa placa oferece conectividade Wi-Fi, porém suas dimensões maiores (43,18 × 17,77 mm) dificultam sua aplicação em um dispositivo vestível, podendo comprometer o conforto do usuário. Além disso, possui custo relativamente alto, cerca de US\$ 35,00.

Por fim, a M5Stack AtomS3R-Ext foi a opção escolhida para o desenvolvimento do sistema. A placa possui uma ESP32-S3-PICO-1-N8R8, que integra um processador dual-core Xtensa® LX7 de 32 bits com frequência de operação de até 240 MHz, que permite executar múltiplas tarefas com grande velocidade de processamento, e conta com Wi-Fi embarcado. Outro fator determinante foi que a placa dispõe de 512 kB de SRAM interna, complementada por 8 MB de PSRAM e 8 MB de Flash, garantindo recursos suficientes para o armazenamento e execução de *firmwares* complexos. Além disso, ela possui dimensões reduzidas (24,0 × 24,0 × 12,9 mm). Outro ponto favorável é sua facilidade de integração com baterias externas, também disponibilizadas pela fabricante M5Stack, o que simplifica o desenvolvimento de um sistema portátil. Além disso, apresenta custo mais acessível, em torno de US\$ 14,50 e compatibilidade com o ambiente Arduino, facilitando o processo de programação e prototipagem.

Com a placa de desenvolvimento selecionada, tornou-se necessária a integração de uma fonte de alimentação autônoma, de modo a viabilizar o funcionamento do sistema sem o uso de cabos. Considerando a escolha da M5Stack AtomS3R-Ext, foi identificado um módulo de bateria compatível e de fácil conexão com a placa. O módulo *Atomic Battery Base* apresenta dimensões compactas de 24,0 × 24,0 × 23,95 mm e incorpora uma bateria recarregável de 200 mAh que pode ficar permanentemente conectada à placa e carregada mesmo enquanto acoplada, características adequadas para a implementação de um protótipo funcional.

Dessa forma, a M5Stack AtomS3R-Ext demonstrou o melhor equilíbrio entre tamanho, conectividade, custo e facilidade de implementação, sendo, portanto, a placa mais adequada para o projeto proposto.

### 3.2.2 Protocolo de comunicação

Para viabilizar o envio remoto de dados, é necessário implementar um protocolo de comunicação entre as entidades do sistema, definindo um conjunto de regras e formatos que padronizem a representação, o endereçamento, a transmissão e o reconhecimento das informações. Em sistemas IoT, a escolha do protocolo afeta diretamente o consumo de energia e dados, latência, escalabilidade e robustez da solução, sendo adotado nesse trabalho o protocolo MQTT.

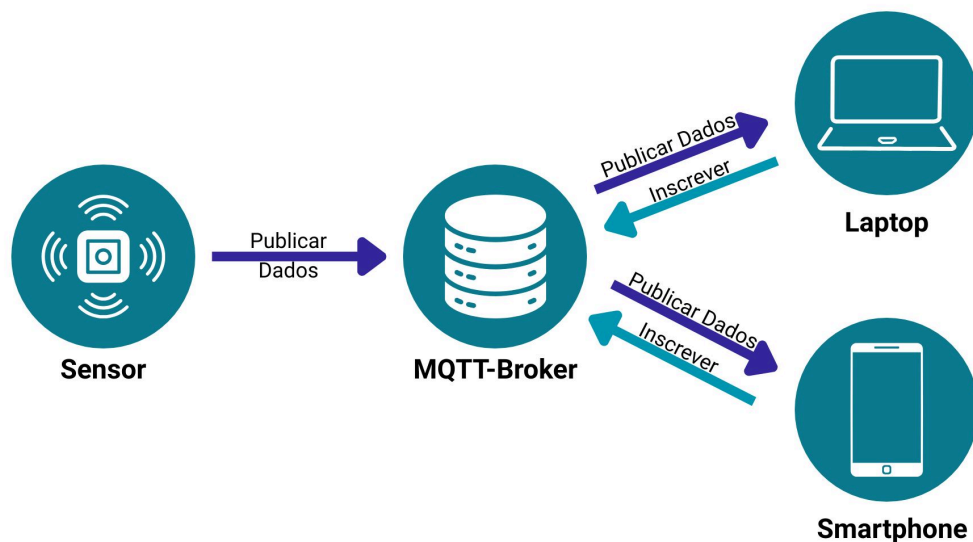
#### 3.2.2.1 MQTT

O MQTT é um protocolo de comunicação leve amplamente utilizado em aplicações de IoT. Ele foi projetado para cenários com dispositivos de recursos limitados e redes com baixa largura de banda ou conectividade instável, características comuns em sistemas embarcados

e dispositivos vestíveis. Diferentemente de protocolos tradicionais baseados em requisição e resposta, o MQTT utiliza o modelo de comunicação *publish/subscribe*, no qual os dispositivos publicam mensagens em tópicos específicos gerenciados por um servidor central denominado *broker*. Outros dispositivos ou aplicações podem se inscrever nesses tópicos para receber automaticamente as mensagens publicadas (OASIS, 2014).

A Figura 6 ilustra de forma simplificada o fluxo de comunicação do protocolo MQTT. No exemplo apresentado, o sensor publica dados no *broker*, que atua como intermediário responsável por distribuir essas informações aos clientes interessados. Dispositivos como *smartphones* ou computadores podem se inscrever (*subscribe*) em determinados tópicos e receber automaticamente os dados publicados (*publish*), sem que haja comunicação direta entre os dispositivos finais. Esse modelo promove um desacoplamento entre produtores e consumidores de dados, permitindo maior escalabilidade, flexibilidade e modularidade na arquitetura do sistema.

**Figura 6 – Esquema do fluxo de funcionamento resumido do protocolo MQTT.**



**Fonte: Autoria própria.**

No contexto do sistema desenvolvido neste trabalho, o uso do MQTT apresenta vantagens relevantes em relação a outros protocolos de comunicação frequentemente utilizados em aplicações conectadas. Protocolos baseados em HTTP, por exemplo, utilizam um modelo cliente-servidor síncrono no qual cada transmissão depende de uma requisição explícita do cliente. Além disso, o HTTP possui cabeçalhos relativamente extensos e maior sobrecarga de comunicação, o que pode aumentar o consumo de dados e de energia em dispositivos alimentados por bateria. Estudos comparativos indicam que o MQTT tende a apresentar menor consumo energético e menor quantidade de bytes transmitidos por mensagem quando comparado ao HTTP em aplicações IoT (Jara-ochoa *et al.*, 2023).

Outra alternativa frequentemente discutida em sistemas IoT é o protocolo *Constrained Application Protocol* (CoAP). O CoAP foi desenvolvido para dispositivos restritos e opera nor-

malmente sobre *User Datagram Protocol* (UDP) e depende de mecanismos adicionais para garantir confiabilidade e segurança. Em muitos casos, isso aumenta a complexidade de implementação e gerenciamento do sistema. Além disso, o modelo de comunicação CoAP permanece predominantemente baseado em requisição e resposta, enquanto o MQTT oferece nativamente um modelo orientado a eventos, mais adequado para aplicações de monitoramento contínuo e envio imediato de alertas (SHELBY, Z. AND HARTKE, K. AND BORMANN, C., 2014; Bayılmış *et al.*, 2022).

Outro aspecto relevante do MQTT é o suporte a diferentes níveis de qualidade de serviço (QoS). O protocolo define três níveis principais: 0 (no máximo uma vez), 1 (pelo menos uma vez) e 2 (exatamente uma vez). Esses níveis permitem ajustar o grau de confiabilidade na entrega das mensagens conforme a criticidade da informação transmitida. Em aplicações de monitoramento, por exemplo, dados periódicos de sensores podem ser enviados com níveis menores de garantia de entrega, enquanto eventos críticos, como a detecção de uma possível queda, podem utilizar níveis superiores de QoS para garantir maior confiabilidade na comunicação (OASIS, 2014).

Um recurso particularmente importante para sistemas de monitoramento remoto é o mecanismo conhecido como *Last Will and Testament*. Durante o estabelecimento da conexão com o *broker*, o cliente pode registrar previamente uma mensagem de "última vontade", associada a um tópico específico. Caso a conexão seja encerrada de forma inesperada, por exemplo, devido à perda de energia, falha do dispositivo ou perda de conectividade, o *broker* publica automaticamente essa mensagem para os clientes inscritos no tópico correspondente. Dessa forma, outros componentes do sistema podem ser notificados imediatamente sobre a perda de conexão do dispositivo monitorado.

No sistema desenvolvido neste trabalho, o mecanismo de LWT foi utilizado para indicar o estado de conectividade do dispositivo vestível responsável pela detecção de quedas. Caso o dispositivo se desconecte de forma abrupta da rede, o *broker* publica automaticamente uma mensagem indicando o estado *offline*. Isso permite que o servidor intermediário, o banco de dados em nuvem e a aplicação móvel identifiquem rapidamente a indisponibilidade do dispositivo, aumentando a robustez e a confiabilidade do sistema de monitoramento.

Dessa forma, considerando a baixa sobrecarga de comunicação, o modelo assíncrono de troca de mensagens, a flexibilidade nos níveis de confiabilidade e os mecanismos de monitoramento de conexão, como o LWT, o protocolo MQTT mostrou-se particularmente adequado para o sistema proposto neste trabalho, que envolve um dispositivo vestível de detecção de quedas integrado a uma infraestrutura de comunicação baseada em IoT (Bayılmış *et al.*, 2022; Jara-ochoa *et al.*, 2023).

### 3.2.3 *Firmware*

O *firmware* desenvolvido tem como finalidade realizar a aquisição dos dados do módulo IMU integrado à placa M5Stack AtomS3R-Ext, organizar e filtrar essas informações e transmití-

las via conexão Wi-Fi para um servidor MQTT. O código foi escrito em linguagem C/C++, utilizando a plataforma *PlatformIO* e faz uso das bibliotecas oficiais da Arduino, Espressif, M5Stack e de terceiros para a configuração e controle dos periféricos do microcontrolador.

### 3.2.3.1 Manipulação e tratamento de dados

Os dados obtidos pelos sensores necessitam de manipulação e tratamento para gerarem informações úteis para o sistema. O IMU utilizado gera dados brutos em unidades que não precisam de uma conversão prévia e acelerômetros em G, que correspondem a múltiplos de aceleração da gravidade (9,81 m/s<sup>2</sup>). Já o giroscópio fornece dados em %/s.

O sensor inercial utilizado apresenta boa precisão intrínseca nas medições, sendo capaz de registrar variações de aceleração e rotação com elevada sensibilidade, segundo a distribuidora Bosch. Dessa forma, não foi necessária a aplicação de algoritmos complexos de fusão sensorial, uma vez que o ruído presente nos sinais é relativamente baixo para a finalidade proposta. Ainda assim, com o objetivo de suavizar pequenas flutuações e eliminar ruídos de alta frequência, aplicou-se um filtro de média móvel, que realiza uma atenuação simples e eficiente das variações abruptas, preservando as características principais do movimento. Esse procedimento garante maior estabilidade nos dados analisados, sem comprometer a resposta dinâmica do sistema. A equação da média móvel é dada por:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \quad (1)$$

em que  $y[n]$  representa o valor filtrado na amostra  $n$ ,  $x[n]$  é o sinal original medido pelo sensor e  $N$  corresponde ao tamanho da janela de amostras utilizada na média. Esse método reduz ruídos de alta frequência e mantém a forma geral do sinal de movimento. Usando essa filtragem, foram avaliados diferentes valores de  $N$  a fim de determinar o melhor compromisso entre suavização do sinal e preservação da resposta dinâmica.

### 3.2.4 *Machine Learning* Embarcado

Neste projeto, o ML foi empregado com a finalidade de reconhecer padrões nos sinais obtidos pela IMU e distinguir atividades cotidianas de possíveis eventos de queda. Nesse contexto, adotou-se a abordagem de *Edge Artificial Intelligence (Edge AI)*, na qual o processamento do modelo é executado diretamente no microcontrolador, dispensando a necessidade de comunicação contínua com servidores remotos. Essa metodologia proporciona menor latência, redução no consumo de energia e maior preservação da privacidade dos dados, características muito interessantes para aplicações voltadas ao monitoramento de indivíduos (Ibm, 2025).

#### 3.2.4.1 Edge Impulse

Diversas plataformas de ML voltadas a dispositivos embarcados estão disponíveis atualmente, como o *TensorFlow Lite for Microcontrollers* e o *TinyML Framework*, além do *Edge Impulse*. Essas ferramentas possibilitam a implementação de modelos de inteligência artificial em sistemas de recursos limitados, porém demandam, em geral, um nível mais elevado de programação, além de configurações manuais nas etapas de coleta de dados, extração de atributos e otimização dos modelos para *hardware* específico.

O *TensorFlow Lite*, por exemplo, é amplamente utilizado em aplicações embarcadas e oferece desempenho satisfatório em dispositivos com maior capacidade de processamento. Porém, exige um fluxo de desenvolvimento mais complexo, que inclui a conversão manual dos modelos, ajustes finos de quantização e integração direta via código. Já *frameworks* genéricos de *TinyML* fornecem bibliotecas leves e modulares, mas não oferecem uma interface gráfica ou ambiente de desenvolvimento integrado, o que aumenta o tempo de configuração e dificulta a visualização das métricas de desempenho durante o treinamento.

O *Edge Impulse*, por sua vez, destaca-se por disponibilizar um ambiente integrado que abrange as etapas de processamento, treinamento, validação e implantação de modelos de ML embarcados. Embora a aquisição de dados seja realizada de forma independente, a plataforma oferece ferramentas completas para importação, organização e rotulagem dos conjuntos coletados, permitindo a aplicação direta de técnicas de extração de atributos e análise espectral.

A compatibilidade do *Edge Impulse* com dispositivos baseados em ESP32 (microcontrolador de baixo custo, baixo consumo de energia e alto desempenho desenvolvido pela Espressif Systems), aliada à possibilidade de conversão automática dos modelos de ML em bibliotecas otimizadas em C/C++, favorece a integração com o *firmware*, reduzindo a sobrecarga computacional e o consumo de memória em sistemas embarcados (ESPRESSIF, 2023). Além disso, o *Edge Impulse* disponibiliza módulos específicos para tarefas de classificação de dados, bem como ferramentas de visualização e avaliação de desempenho dos modelos, permitindo análises detalhadas de acurácia, latência e uso de recursos, o que o torna uma solução tecnicamente viável para aplicações embarcadas que demandam baixa latência, eficiência energética e autonomia operacional. Além das informações citadas, detalhes adicionais sobre as funcionalidades e a documentação da plataforma estão disponíveis em sua página oficial (EDGE IMPULSE, 2025).

#### 3.2.4.2 Classificação Supervisionada

Para a identificação de eventos de queda a partir dos dados provenientes da IMU, foi adotada a abordagem de classificação supervisionada. Nesse método, um modelo de ML é treinado utilizando um conjunto de dados previamente rotulado, permitindo que o sistema aprenda a distinguir diferentes padrões de movimento com base nas características extraídas dos sinais dos sensores. Dessa forma, o algoritmo é capaz de identificar automaticamente eventos asso-

ciados a quedas ou a movimentos considerados normais, a partir dos padrões presentes nos dados de aceleração e rotação coletados pela IMU.

Inicialmente, os sinais provenientes do acelerômetro e do giroscópio são coletados ao longo do tempo e organizados em janelas temporais, que representam segmentos do comportamento dinâmico do usuário durante a execução de determinadas atividades. Cada uma dessas janelas é então associada a um rótulo que identifica o tipo de movimento correspondente. No contexto deste trabalho, foram consideradas duas classes principais: movimento normal e queda. Essa rotulagem permite que o modelo aprenda as características que diferenciam os padrões associados a atividades cotidianas daqueles relacionados a eventos de queda.

A partir dessas informações, o algoritmo ajusta seus parâmetros internos de modo a estabelecer fronteiras de decisão entre as classes analisadas. Após o treinamento, o modelo é capaz de realizar inferência sobre novos dados provenientes dos sensores, classificando automaticamente os padrões de movimento detectados. Consequentemente, torna-se possível identificar eventos de queda em tempo real a partir da análise dos sinais gerados pela IMU (EDGE IMPULSE, 2025).

#### 3.2.4.3 Método de verificação baseado em janelas

Após a etapa de inferência realizada pelo modelo de ML, torna-se necessário aplicar um mecanismo adicional de verificação para reduzir a ocorrência de falsos positivos. Embora o modelo seja capaz de identificar padrões associados a quedas, movimentos cotidianos podem ocasionalmente gerar respostas semelhantes. Neste trabalho, implementou-se um método de validação baseado em persistência temporal, denominado neste trabalho como método K de M, cujo funcionamento está representado na Figura 7. Estratégias semelhantes de análise temporal de eventos são amplamente empregadas em sistemas de detecção de quedas baseados em sensores vestíveis, justamente para aumentar a confiabilidade das decisões do classificador (Mubashir; Shao; Seed, 2013). Conforme ilustrado na figura, o modelo de ML processa continuamente janelas de dados provenientes do sensor inercial e retorna uma probabilidade associada ao movimento realizado pelo indivíduo. Essa probabilidade então é comparada a um limiar previamente definido, que representa o nível mínimo de confiança necessário para considerar aquela janela como potencialmente associada a uma queda. No sistema desenvolvido, esse limiar foi definido como 0,85 após testes práticos de verificação com a pulseira, ou seja, somente as janelas cuja probabilidade de queda seja igual ou superior a 0,85 são consideradas indicativas de um possível evento.

Entretanto, a decisão não é tomada com base em uma única janela. Para aumentar a robustez da detecção, foi adotada uma estratégia de votação entre múltiplas janelas consecutivas. Essa estratégia é implementada por meio do método K de M, no qual são analisadas as M últimas janelas classificadas pelo modelo. Caso pelo menos K dessas janelas apresentem probabilidade de queda acima do limiar estabelecido, o sistema entra em um "estado de suspeita".

No protótipo desenvolvido foram utilizados os parâmetros  $M = 2$  e  $K = 1$ , o que significa que o sistema considera uma suspeita de queda quando pelo menos uma das duas últimas janelas analisadas apresenta probabilidade de queda superior ao limiar definido. Essa abordagem permite que o sistema responda rapidamente a possíveis eventos, ao mesmo tempo em que evita que oscilações momentâneas do classificador resultem em alarmes indevidos.

Uma vez que o sistema entra no estado de suspeita, inicia-se uma segunda etapa de validação baseada nas características físicas do movimento. Nessa fase, o sistema analisa continuamente as medições provenientes do acelerômetro e do giroscópio, buscando identificar padrões típicos de um evento de queda.

A primeira verificação consiste na detecção de impacto. Durante uma queda real, é esperado que ocorra um pico significativo na magnitude de aceleração no momento de contato com o solo, o que pode ser visto na Figura 19. Para representar essa condição, após testes práticos e visualização de gráficos que demonstram comportamento de quedas, os quais são mais aprofundados posteriormente, foi definido um limiar de 3,80 g para magnitude de aceleração. Caso esse valor seja ultrapassado, o evento é considerado compatível com um impacto. Além disso, também é considerada uma segunda condição de impacto moderado, na qual a magnitude da aceleração ultrapassa 3,40 g e é acompanhada por um pico significativo na velocidade angular medida pelo giroscópio. Nesse caso, o limiar adotado para a velocidade angular foi de 380 graus por segundo.

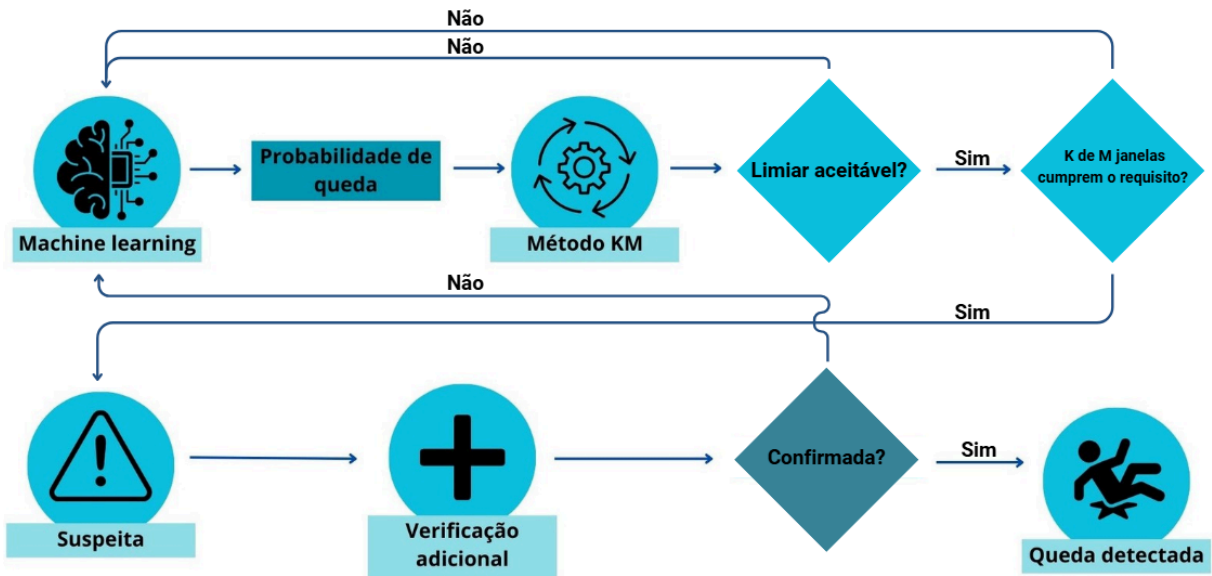
Caso uma dessas condições seja atendida dentro de uma janela temporal de aproximadamente 1,8 segundos após o estado de suspeita, o sistema avança para a etapa final de validação

A última etapa consiste na verificação de imobilidade após impacto. Para isso, o algoritmo monitora continuamente a magnitude da aceleração e da velocidade angular durante um intervalo mínimo de 1 segundo. Durante esse período, o sistema verifica se a velocidade angular permanece abaixo de 20 graus por segundo, indicando ausência de movimento rotacional significativo, e se a variação da magnitude da aceleração permanece dentro de um intervalo máximo de 0,12 g em relação ao valor de referência medido no início da verificação.

Caso essas condições sejam satisfeitas durante todo o intervalo de observação, o evento é finalmente classificado como queda confirmada.

Essa abordagem em múltiplos níveis permite combinar capacidade de reconhecimento de padrões do modelo de ML com critérios físicos diretamente relacionados à dinâmica de uma queda, resultando em um sistema mais robusto e menos suscetível a alarmes falsos.

Figura 7 – Fluxograma de funcionamento do sistema utilizando método K de M.



Fonte: Autoria própria.

### 3.2.5 Interface gráfica

A interação entre o portador do dispositivo e o cuidador deve ser simples e eficiente. Essa premissa exige uma plataforma *mobile* de fácil entendimento, garantindo a melhor experiência de usuário possível. Tais requisitos foram cruciais para a definição das tecnologias de *backend* (servidor) e *frontend* (aplicativo).

No contexto dessa arquitetura, foi desenvolvido um servidor intermediário em linguagem Python responsável por atuar como ponte entre o dispositivo vestível e a infraestrutura da plataforma de serviço Firebase do Google. Esse servidor se conecta a um *broker* MQTT e permanece escutando tópicos específicos publicados pelo dispositivo embarcado, como o estado de conexão do equipamento e eventos de detecção de queda. Quando uma mensagem é recebida, o *backend* interpreta seu conteúdo e executa diferentes ações conforme o tipo de evento. No caso de alteração de status de conexão, o servidor atualiza automaticamente o estado do dispositivo no banco de dados em tempo real do Firebase, permitindo que o aplicativo *Flutter* reflita instantaneamente se a pulseira está conectada ou desconectada. Já quando um evento de queda é detectado, o sistema registra a ocorrência no banco de dados e dispara uma notificação de emergência utilizando o FCM, que será entregue ao aplicativo do cuidador. Além disso, o servidor implementa mecanismos de controle de estado para evitar notificações duplicadas e ignora mensagens retidas do MQTT que poderiam gerar falsos alertas após reconexões. Dessa forma, o *backend* garante que as informações recebidas do dispositivo sejam processadas de maneira confiável e que os eventos críticos sejam comunicados ao cuidador de forma rápida e segura.

Para o *frontend* (a interface gráfica no dispositivo móvel do cuidador), foi escolhido o *Flutter*. Esta plataforma permite, a partir de um único código-fonte, gerar aplicativos nativos

tanto para Android quanto para iOS, o que exclui a necessidade de manter dois projetos distintos. Além disso, o *Flutter* é conhecido por sua facilidade em criar interfaces de usuário claras e fluidas. O aplicativo permite visualizar o status da pessoa monitorada e recebe as notificações imediatas em caso de qualquer distúrbio, permitindo que o cuidador possa agir o mais rápido possível (PROMETTEUR SOLUTIONS, 2025).

Nesta arquitetura, o *Firebase* (uma plataforma de serviços do Google) foi utilizado como a ponte de comunicação essencial entre o servidor *backend* e o aplicativo *frontend*. Sua principal função é através do *Firebase Cloud Messaging* (FCM), um serviço responsável por entregar as notificações *push*. Quando o servidor Python detectar um evento crítico, como uma queda ou saída da área segura recebida via MQTT, ele não enviará a notificação diretamente ao celular. Em vez disso, ele comanda o FCM para enviar o alerta. O FCM, por sua vez, é responsável por localizar e "acordar" o dispositivo do cuidador (seja Android ou iOS), garantindo que a notificação de emergência seja entregue de forma instantânea e confiável, mesmo que o aplicativo *Flutter* esteja fechado. Além disso, a plataforma possui um banco de dados em tempo real que possibilita o responsável visualizar no aplicativo o estado da pulseira, permitindo que haja um botão na interface para sinalizar que a queda foi verificada (GOOGLE, 2024).

Para o aplicativo em *Flutter*, quando em uso, foram criados diferentes estados de interface para sinalizar os eventos ao cuidador com facilidade. Estes estados incluem: uma tela de status normal (Figura 8a), indicando que o monitoramento está ativo e tudo está certo; uma tela de alerta (Figura 8b) para notificar quedas, junto com um botão para sinalizar que a queda foi averiguada pelo responsável, e uma tela que indica desconexão (Figura 8c), sinalizando a possibilidade do paciente ter saído da área de cobertura da rede.

**Figura 8 – Dashboard do Monitor de Segurança em três estados: a) operação normal, com hardware conectado e monitoramento ativo; b) queda detectada, exibindo alerta e opção de confirmação da ocorrência; e c) hardware desconectado, indicando perda de comunicação com a smartband.**



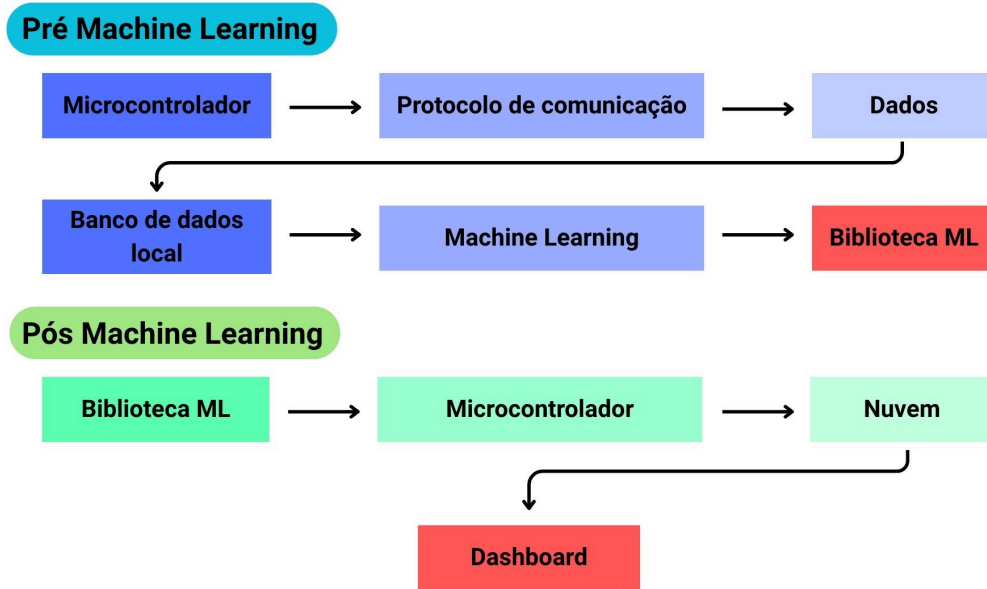
Fonte: Autoria própria.

### 3.3 METODOLOGIA EXPERIMENTAL

Nesta subseção, são apresentados os procedimentos adotados para a realização dos testes experimentais e a validação do sistema proposto. São detalhadas as etapas de coleta e tratamento dos dados, incluindo a simulação controlada de movimentos e quedas, bem como os métodos utilizados para comparação e análise dos resultados. Além disso, são descritos os recursos de *hardware* e *software* empregados, abrangendo desde a configuração dos sensores e módulos até o processamento dos sinais e aplicação dos algoritmos de ML.

O funcionamento do sistema proposto é dividido em duas etapas principais: uma fase prévia de treinamento e desenvolvimento do modelo de ML e uma fase posterior de operação embarcada, após a integração da biblioteca gerada. Na Figura 9 é apresentado o diagrama simplificado dessas duas etapas.

Figura 9 – Esquema de funcionamento do sistema antes do ML e depois.

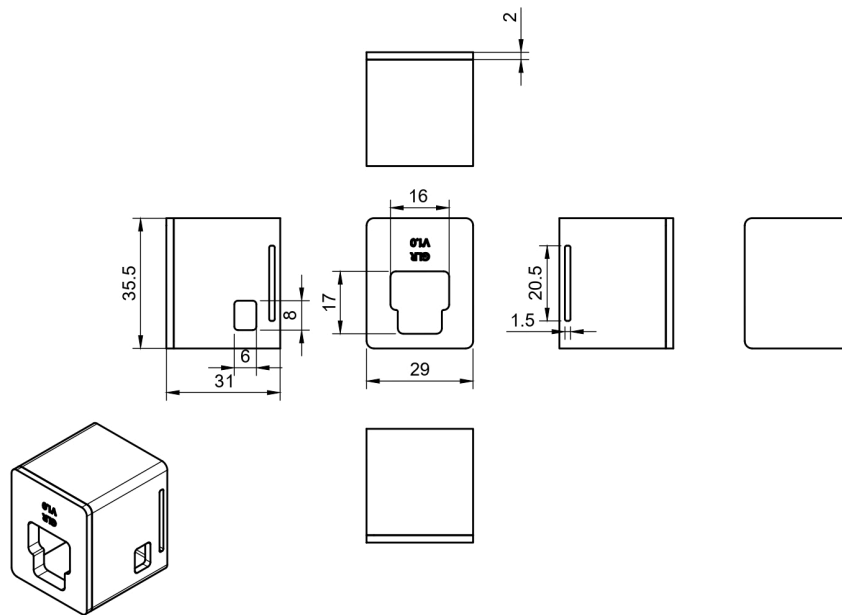


Fonte: Autoria própria.

### 3.3.1 PREPARAÇÃO PARA EXPERIMENTAÇÃO

Para a realização da coleta inicial de dados, foi necessário projetar e fabricar uma proteção específica (Figura 10), desenvolvida para acomodar o dispositivo utilizado nos testes, a M5Stack AtomS3R-EXT. Essa estrutura foi confeccionada por meio de impressão 3D, utilizando filamento PLA (Ácido Polilático) na cor branca (Figura 11 e 12), e teve como finalidade garantir a integridade física do *hardware* durante os experimentos, além de proporcionar condições adequadas para a aquisição precisa dos sinais. A utilização da proteção também se justifica pelo fato de que o dispositivo final foi implementado na forma de uma *smartband*. Para isso, a estrutura foi fixada ao pulso do usuário mediante a utilização de uma fita de velcro, o que pode ser visualizado nas Figuras 13 e 14, o que garante a estabilidade necessária sem comprometer a ergonomia durante os testes.

Figura 10 – Modelo 3D com medidas em milímetros da case para o microcontrolador.



Fonte: Autoria própria.

Figura 11 – Pulseira posicionada verticalmente.



Fonte: Autoria própria.

Figura 12 – Pulseira posicionada lateralmente.



Fonte: Autoria própria.

**Figura 13 – Foto lateral da pulseira em sua posição de operação.**



**Fonte: Autoria própria.**

**Figura 14 – Foto vertical da pulseira em sua posição de operação.**



**Fonte: Autoria própria.**

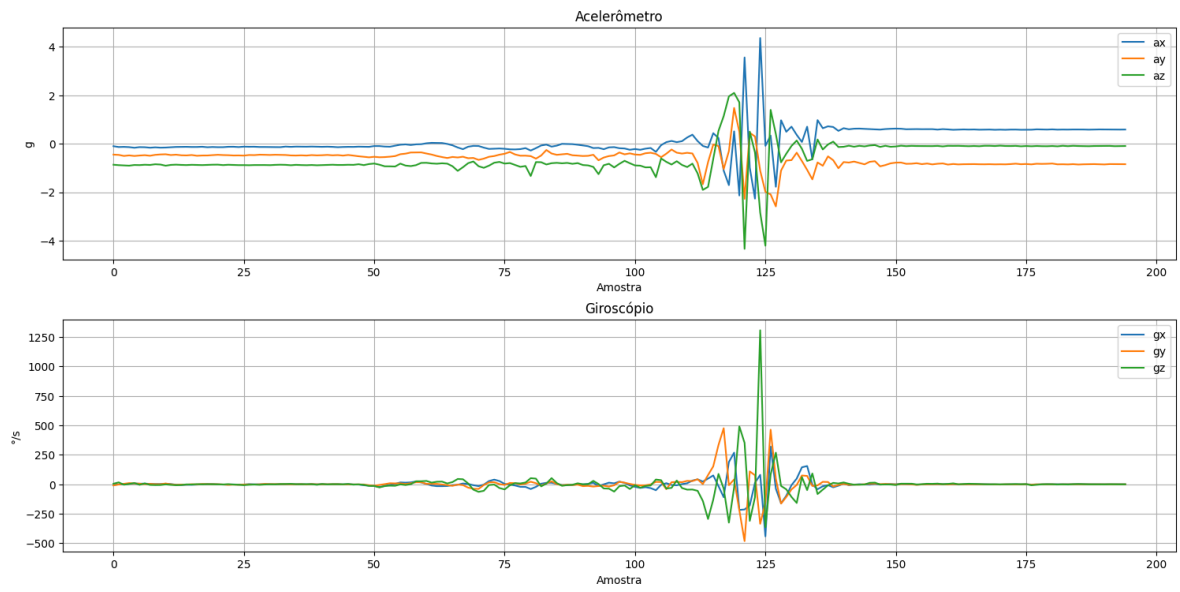
Após a montagem do dispositivo dentro da *case*, foram realizadas coletas de dados, com o objetivo de registrar os diferentes padrões de movimento que serviram de base para a etapa de treinamento do modelo de ML. A aquisição dos sinais foi conduzida de forma controlada, abrangendo situações específicas e variadas, de modo a permitir a posterior diferenciação entre atividades cotidianas e eventos anômalos, como quedas. Durante essa fase, buscou-se obter um conjunto de dados suficientemente representativo para abranger as principais variações de aceleração e rotação associadas aos movimentos naturais do corpo humano. Essa diversidade foi fundamental para garantir que o modelo embarcado fosse capaz de identificar corretamente comportamentos distintos em um ambiente real de operação. Assim, as coletas foram planejadas com cada sessão voltada à obtenção de um tipo específico de movimento, possibilitando a categorização e rotulagem adequadas dos sinais.

### 3.4 COLETA INICIAL DE DADOS

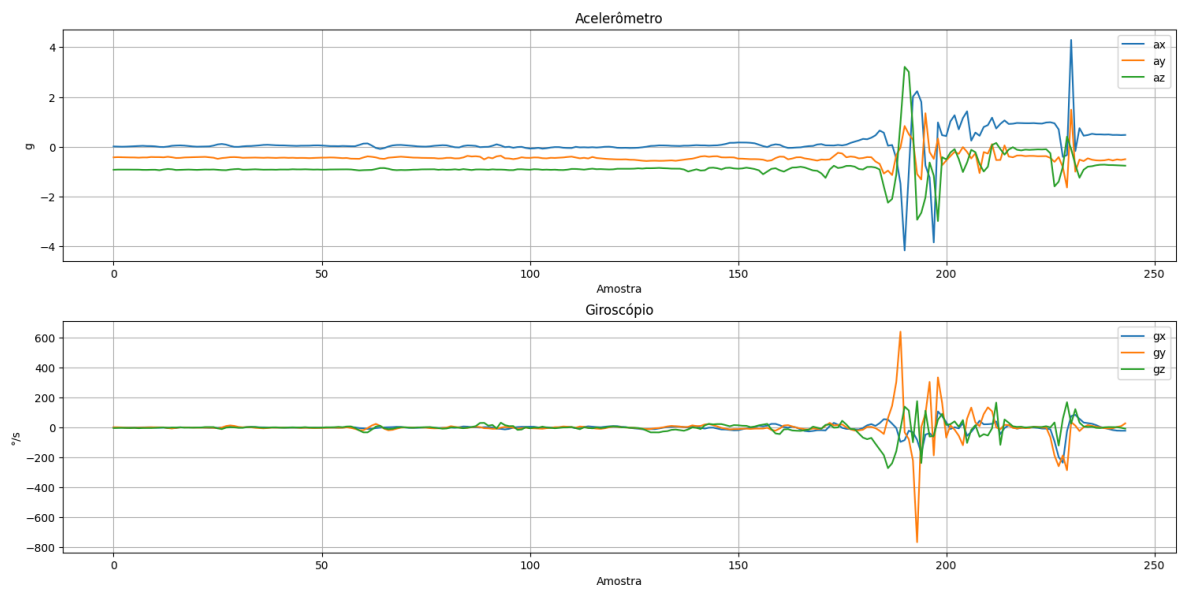
Antes da realização da coleta completa, buscou-se inicialmente visualizar o comportamento esperado dos sinais da IMU durante os diferentes tipos de movimento. Essa etapa preliminar permitiu compreender como as acelerações e velocidades angulares se manifestam ao longo do protocolo experimental, além de oferecer uma referência clara para comparar posteriormente os efeitos da filtragem por média móvel. Assim, as figuras apresentadas nesta seção ilustram tanto os dados brutos quanto os dados suavizados para diferentes valores de  $N$ .

A coleta dos dados foi conduzida de forma padronizada. O indivíduo vestiu a pulseira, protegida com o *case*, e o código foi iniciado, com um intervalo inicial de aproximadamente 2 segundos para estabilização do sensor. Em seguida, foi realizada uma caminhada curta, tanto para o protocolo de queda frontal quanto para o de queda lateral. Após o impacto, o indivíduo permaneceu imóvel no chão por cerca de 5 segundos, momento em que o código foi interrompido para finalizar a captura.

As Figuras 15, 16, 17 e 18 ilustram exemplos dos resultados obtidos para janelas de média móvel com  $N$  igual a 1, 5, 10 e 20 amostras, sendo que  $N$  unitário corresponde aos dados brutos não filtrados. Cada imagem apresentada a seguir mostra, na parte superior, as amostras dos sinais do acelerômetro e, na parte inferior, as amostras do giroscópio, ambos distribuídos em três dimensões. Em todas as figuras, os eixos seguem a mesma convenção visual: componente  $x$  representada em azul, componente  $y$  em laranja e componente  $z$  em verde. Essa padronização facilita comparações entre diferentes valores de  $N$  e permite observar o impacto da filtragem sobre a suavização dos sinais adquiridos.

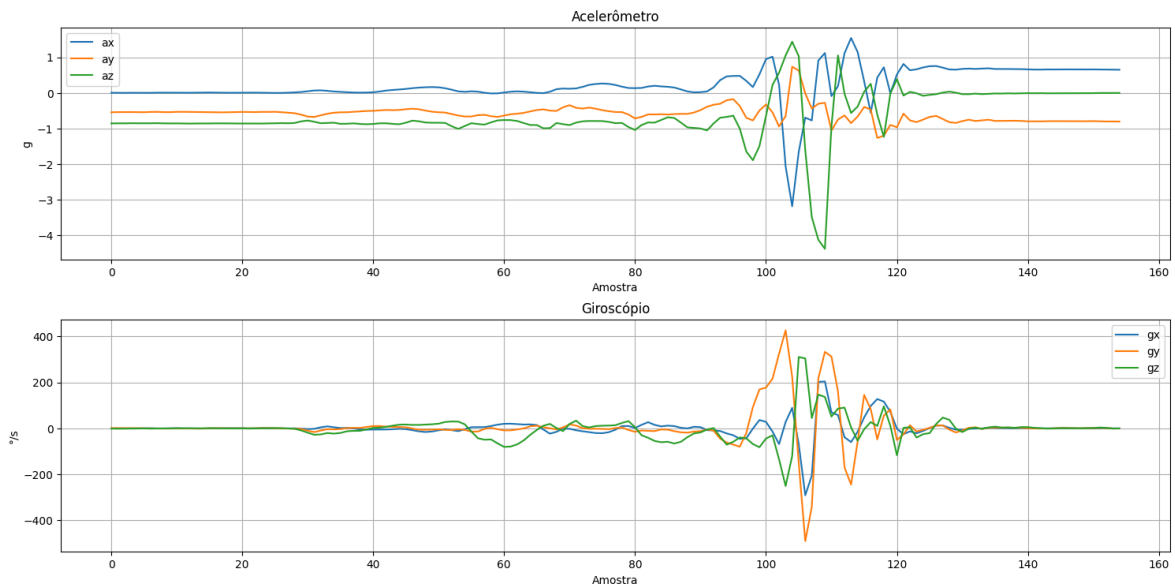
**Figura 15 – Sinais não filtrados do acelerômetro e do giroscópio ( $N = 1$ ).**

Fonte: Autoria própria.

**Figura 16 – Sinais filtrados do acelerômetro e do giroscópio com filtro de média móvel ( $N = 5$ ).**

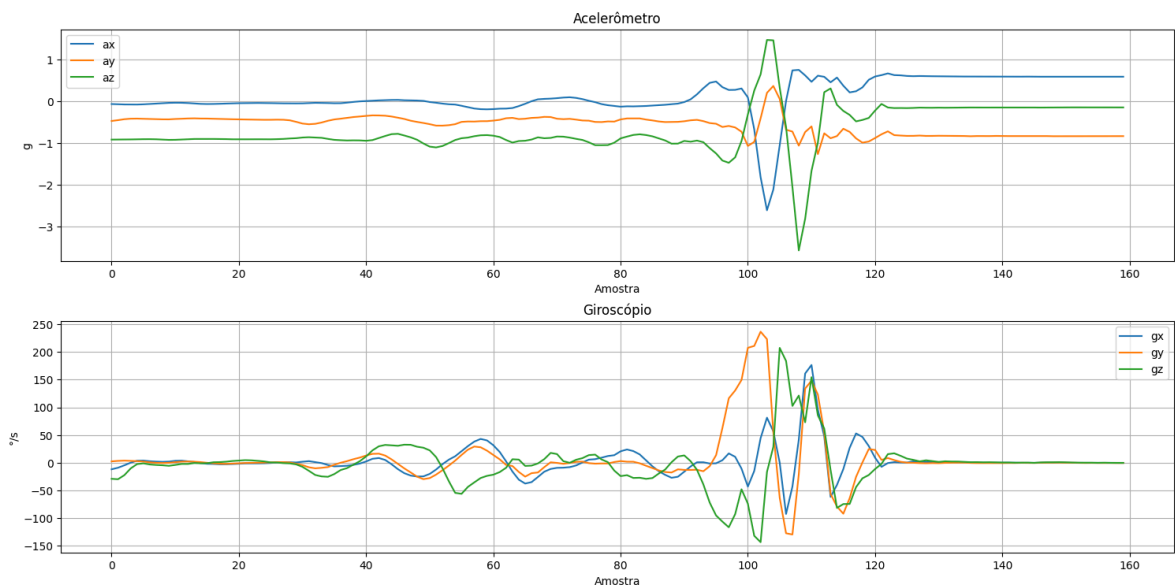
Fonte: Autoria própria.

**Figura 17 – Sinais filtrados do acelerômetro e do giroscópio com filtro de média móvel ( $N = 10$ ).**



**Fonte: Autoria própria.**

**Figura 18 – Sinais filtrados do acelerômetro e do giroscópio com filtro de média móvel ( $N = 20$ ).**



**Fonte: Autoria própria.**

Entretanto, com base nos resultados atualizados obtidos a partir das coletas realizadas, verificou-se que o aumento do valor de  $N$  impacta de forma mais perceptível o comportamento dos sinais. Conforme mostram os gráficos, para  $N$  unitário os dados apresentaram ruído acentuado, refletindo as oscilações naturais dos sensores. Com  $N$  igual a 5, observou-se uma atenuação significativa dessas flutuações, mantendo ainda boa fidelidade temporal — o que se mostrou ideal para capturar movimentos bruscos, como quedas, sem atrasar a resposta do sistema. Já para  $N$  igual a 10 ou  $N$  igual a 20, a suavização excessiva causou perda de detalhes

e atraso na identificação de variações rápidas. Diante disso, optou-se por manter  $N$  igual a 5 como parâmetro definitivo do filtro de média móvel, equilibrando precisão e estabilidade. Após a coleta e o tratamento dos dados filtrados, as seis variáveis provenientes do acelerômetro e do giroscópio foram organizadas em formato tabular para posterior utilização no modelo de ML.

### 3.4.1 ANÁLISE DOS DADOS OBTIDOS

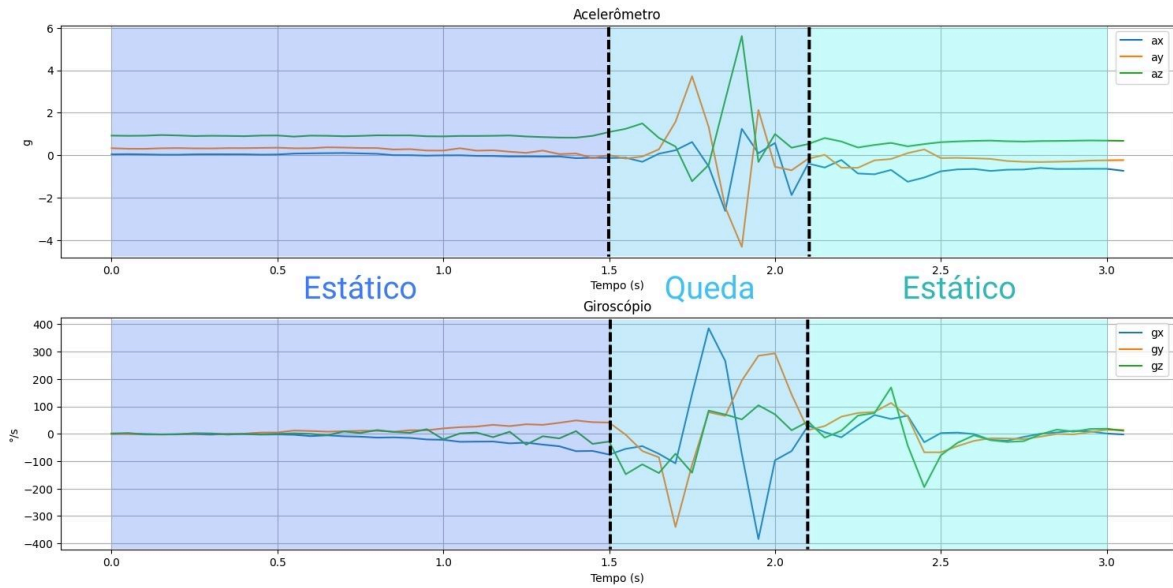
Após a etapa de coleta inicial, procedeu-se a uma análise gráfica dos sinais adquiridos com o objetivo de compreender o comportamento dinâmico dos dados provenientes da IMU. Essa etapa foi essencial para identificar padrões característicos presentes nas diferentes atividades registradas e, sobretudo, para observar como uma queda se manifesta nos eixos do acelerômetro e do giroscópio.

Os arquivos tabulares, os quais contêm dados correspondentes aos três eixos de ambos os sensores utilizados, foram processados e organizados em séries temporais, com uma taxa de amostragem de 20 Hz, permitindo a inspeção visual da evolução dos sinais ao longo do tempo. Nas primeiras sessões de gravação durante a etapa de Trabalho de Conclusão de Curso 1, as simulações envolveram dois voluntários e seguiram um protocolo padronizado, no qual cada registro incluiu um breve período estático inicial, seguido de alguns passos de caminhada, a realização da queda sobre um colchão e um intervalo final de imobilidade, repetindo 3 vezes por indivíduo apenas. Entretanto, a quantidade reduzida de dados obtida com essa abordagem resultou em desempenho insatisfatório do modelo, o que motivou a adoção de um protocolo mais estruturado e segmentado por evento na etapa definitiva de coleta.

O procedimento definitivo consistiu na realização de diferentes tipos de queda (frontal, lateral e de costas) feitas por 2 indivíduos diferentes, com início da gravação contendo 1 segundo de espera em posição estática, seguido da execução da queda e, posteriormente, 2 segundos adicionais de espera para finalização da gravação. Foram realizadas, ao todo, 200 quedas para a coleta dos dados, sendo 100 quedas por indivíduo. Além disso, foram registrados movimentos representativos de atividades da vida diária, como levantar os braços, lavar louça, pular, caminhar, correr e varrer o chão, sendo coletadas aproximadamente 10 amostras de cada atividade por indivíduo, com o objetivo de ampliar a variabilidade dos dados e melhorar a capacidade de generalização do modelo, resultando em 290 arquivos.

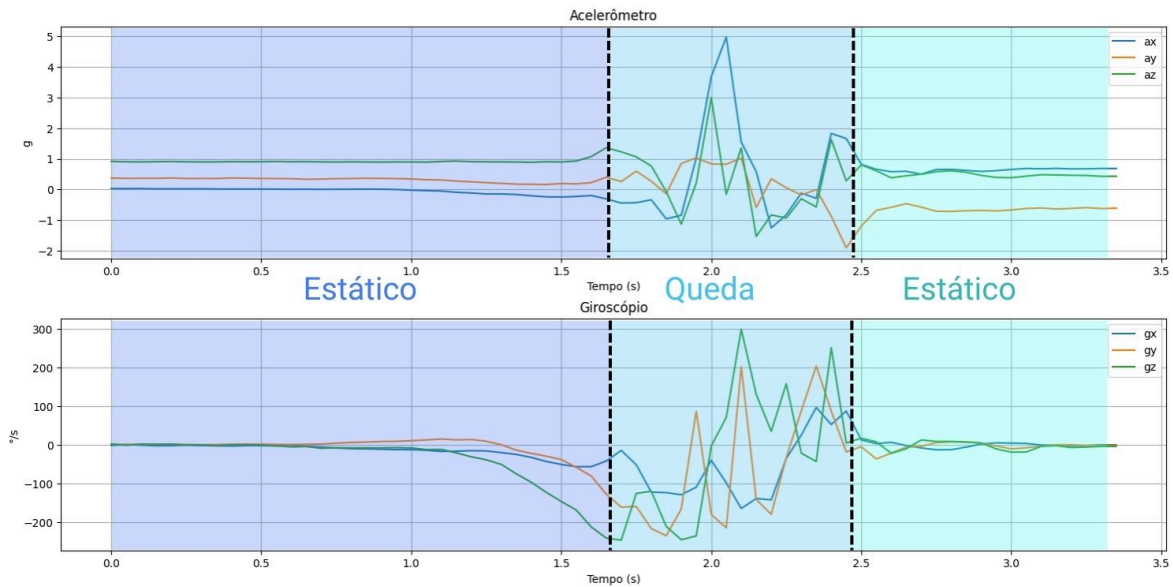
A Figura 19 apresenta um exemplo correspondente a uma queda frontal, evidenciando variações abruptas nos eixos do acelerômetro e do giroscópio imediatamente após o impacto. Já a Figura 20, referente a uma queda lateral, mostra um comportamento distinto, no qual os maiores picos ocorrem em eixos diferentes, especialmente aqueles associados ao movimento lateral do corpo. Essa comparação demonstra que cada tipo de queda produz assinaturas específicas e diferentes entre si na IMU.

**Figura 19 – Visualização dos sinais do acelerômetro e do giroscópio durante o protocolo experimental de queda frontal.**



Fonte: Autoria própria.

**Figura 20 – Visualização dos sinais do acelerômetro e do giroscópio durante o protocolo experimental de queda lateral.**



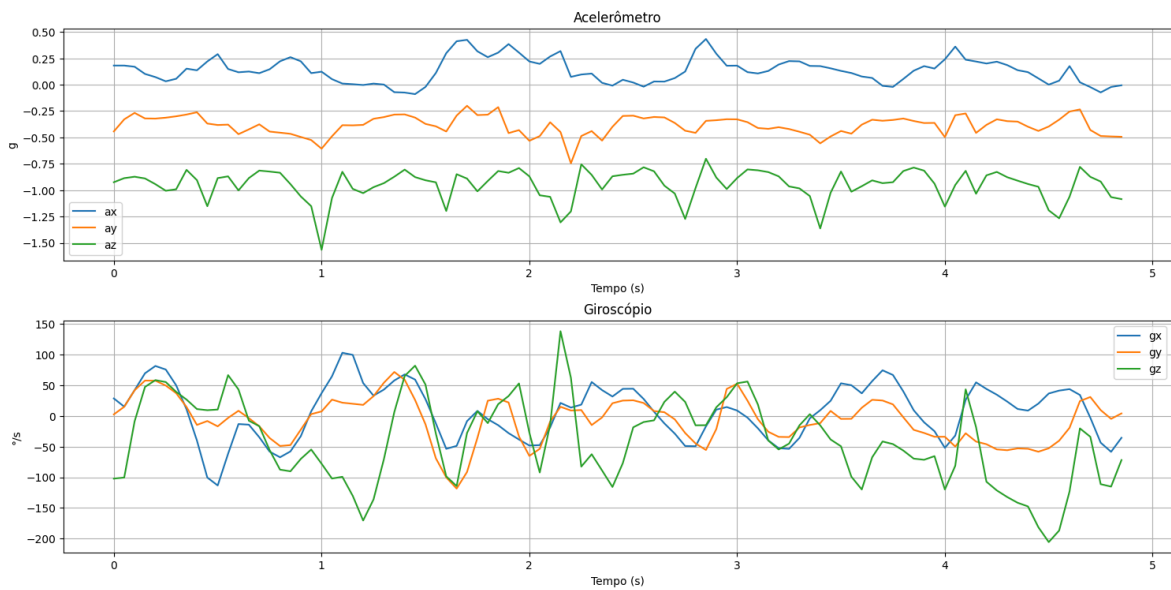
Fonte: Autoria própria.

### 3.4.1.1 CATEGORIZAÇÃO DE MOVIMENTOS EM DADOS TABULARES

Além das análises ilustradas nas Figuras 19 e 20, correspondentes aos exemplos de quedas frontais e laterais, foram realizadas coletas adicionais para compor um conjunto de dados ligeiramente mais abrangente e representativo. Para cada uma dessas novas coletas, também foram gerados e analisados gráficos, permitindo a visualização detalhada do compor-

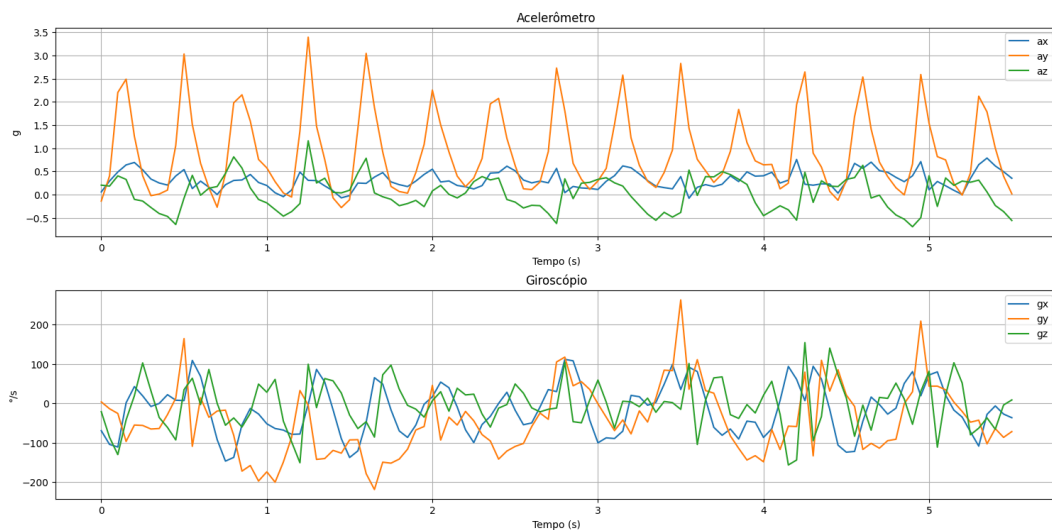
tamento dos sinais em diferentes cenários. A Figura 21 apresenta a pulseira submetida a um movimento de caminhada por aproximadamente 5 segundos, enquanto a Figura 22 exibe o movimento de corrida leve por aproximadamente 5 segundos. De forma complementar, a Figura 23 corresponde ao registro de um pulo realizado pelo indivíduo, ao passo que a Figura 24 representa o movimento de sentar-se. Todas essas representações gráficas contribuíram para a compreensão das assinaturas específicas de cada atividade, fornecendo uma base sólida para a fase subsequente de modelagem no *Edge Impulse*.

**Figura 21 – Visualização de movimento de caminhada contínua.**



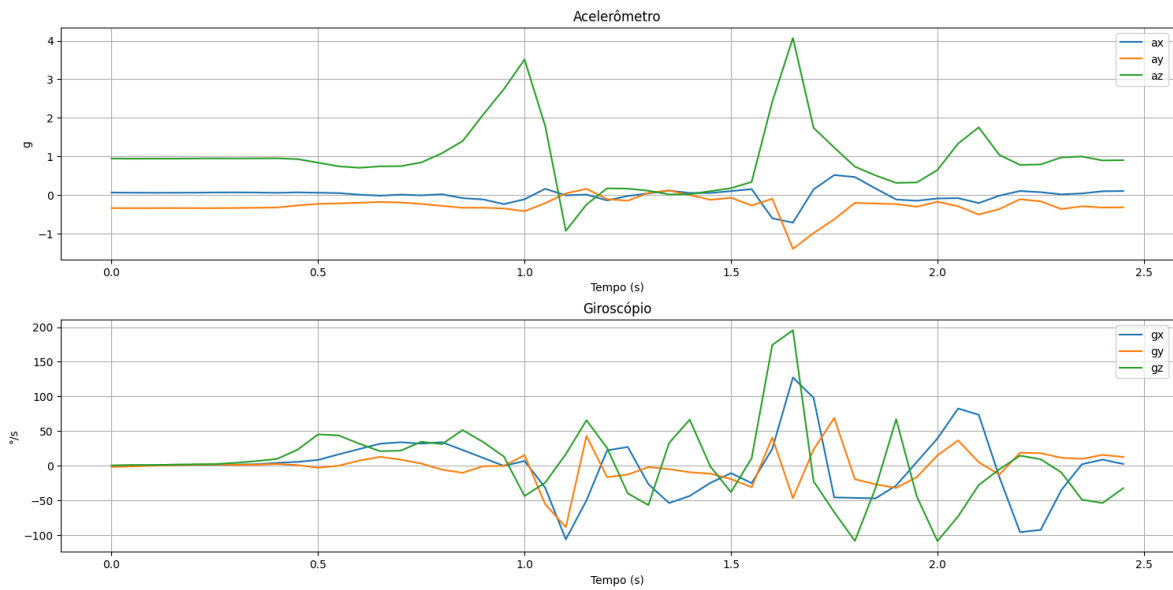
**Fonte: Autoria própria.**

**Figura 22 – Visualização de movimento de corrida leve contínua.**



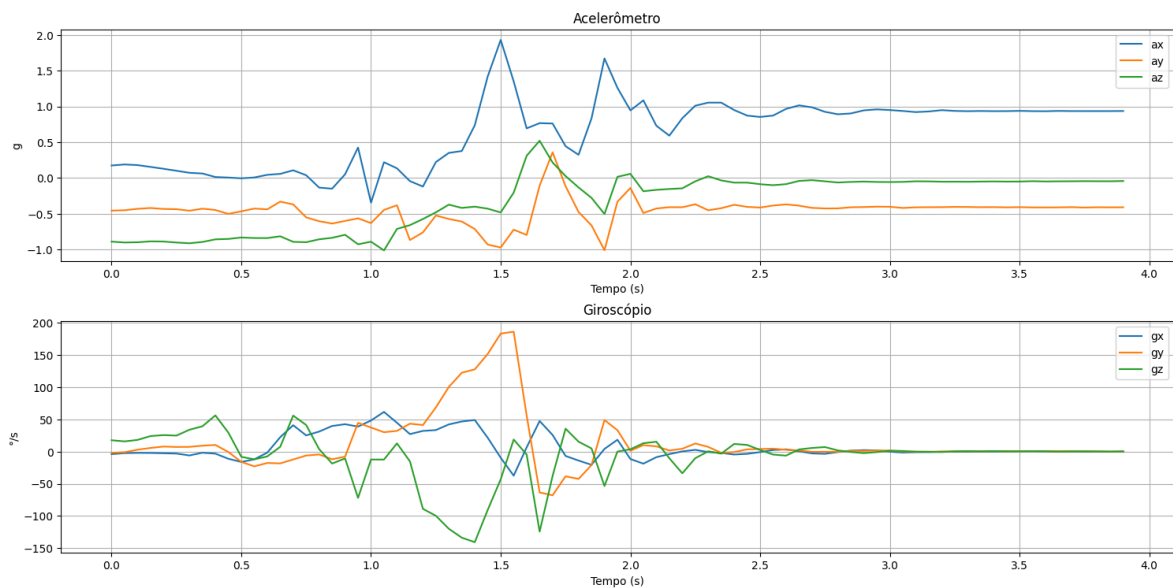
**Fonte: Autoria própria.**

**Figura 23 – Visualização de registro de um pulo realizado pelo indivíduo.**



**Fonte: Autoria própria.**

**Figura 24 – Visualização de registro do indivíduo sentado em uma cadeira.**



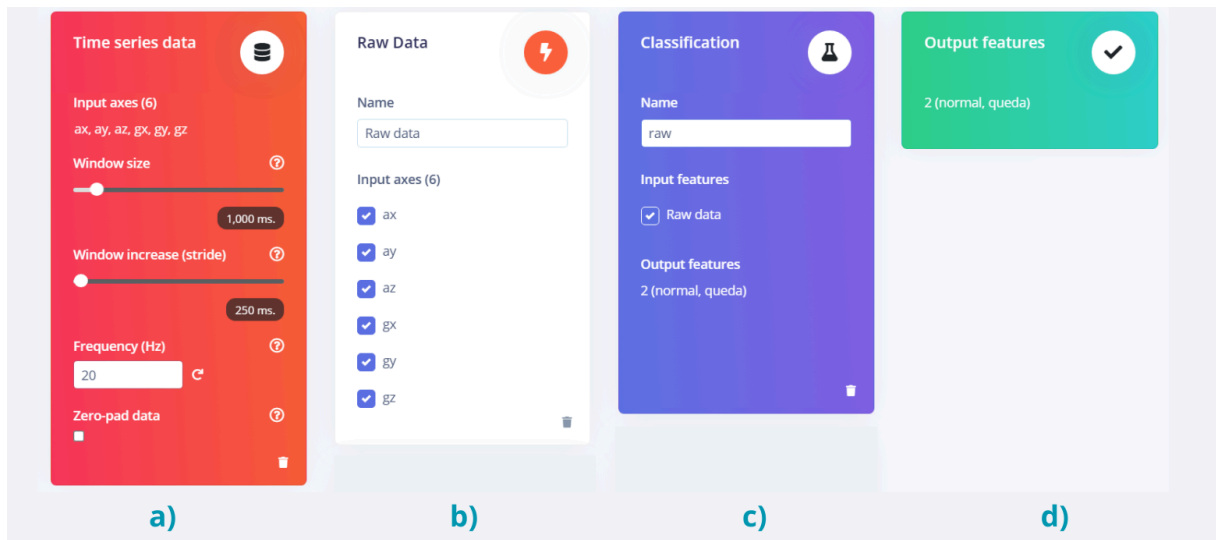
**Fonte: Autoria própria.**

#### 3.4.1.2 Configuração para Classificação Supervisionada

A configuração do classificador supervisionado na plataforma *Edge Impulse*, representada na Figura 25 foi realizada a partir do conjunto de dados previamente rotulados, composto por registros de eventos de queda e movimentos normais. O fluxo de processamento adotado foi estruturado em três etapas principais: segmentação temporal dos sinais, preparação dos dados de entrada e treinamento do modelo de rede neural responsável pela classificação.

Na primeira etapa, os dados brutos provenientes da IMU foram divididos em janelas temporais de 1000 ms com incremento (*stride*) de 250 ms, resultando em uma sobreposição de aproximadamente 75% entre os segmentos consecutivos. Essa estratégia permite representar diferentes partes de um mesmo movimento em múltiplas janelas, aumentando o número de amostras disponíveis e preservando as transições dinâmicas dos sinais. A frequência de amostragem foi configurada em 20 Hz, valor suficiente para capturar as variações relevantes dos movimentos humanos, ao mesmo tempo em que reduz a quantidade de dados processados e o custo computacional do sistema embarcado. Considerando esses parâmetros, cada segmento contém 20 amostras por eixo da IMU, totalizando 120 valores de entrada por janela, considerando os seis eixos (*ax, ay, az, gx, gy, gz*). Esses dados foram utilizados diretamente por meio do bloco *raw data*, mantendo os sinais em sua forma original. Essa escolha fundamenta-se na capacidade de modelos de *deep learning* aprenderem automaticamente representações relevantes a partir de séries temporais, reduzindo a necessidade de extração manual de características (Hammerla; Halloran; Plötz, 2016). Para padronização, foi aplicado um fator de escala unitário aos eixos, preservando a proporcionalidade dos sinais.

**Figura 25 – Interface de configuração do *pipeline* no *Edge Impulse*, em que: a) define os parâmetros de aquisição da IMU (6 eixos, janela de 1000 ms, incremento de 250 ms e frequência de 20 Hz); b) apresenta o bloco de dados brutos, com os 6 eixos selecionados como entrada; c) configura o bloco de classificação com 2 classes de saída (normal e queda); e d) resume as características de saída do modelo.**



**Fonte: Autoria própria.**

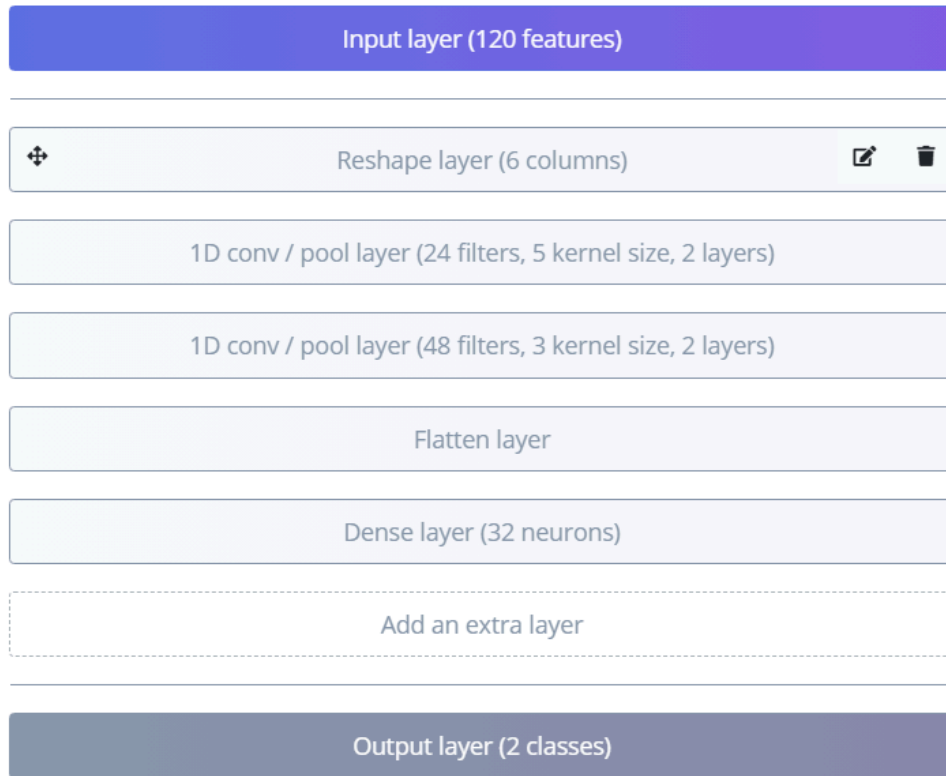
Na etapa de aprendizado, foi empregada uma rede neural convolucional unidimensional, adequada ao processamento de séries temporais provenientes de sensores. Diferentemente de redes totalmente conectadas, essa arquitetura explora dependências locais ao longo do tempo, identificando padrões diretamente nos sinais. A camada de entrada recebe 120 características correspondentes à janela analisada, sendo reorganizadas por meio de *reshape* em uma estrutura com seis colunas, representando os eixos da IMU ao longo do tempo (Edge Impulse, 2026).

Em seguida, são aplicadas duas camadas convolucionais com operações de *pooling*. O tamanho do *kernel* define a extensão temporal dos padrões analisados, enquanto os filtros atuam como detectores de características ao longo do sinal. Na primeira camada, são utilizados 24 filtros com *kernel* igual a 5, permitindo capturar padrões temporais mais amplos, como transições iniciais e variações abruptas associadas a possíveis quedas. Já na segunda camada, são empregados 48 filtros com *kernel* de tamanho 3, possibilitando a extração de padrões mais refinados a partir das representações anteriores. O aumento do número de filtros nessa etapa amplia a capacidade de combinação de características, contribuindo para uma melhor discriminação entre as classes. As operações de *pooling* reduzem a dimensionalidade das representações intermediárias, preservando as informações mais relevantes e contribuindo tanto para a redução do custo computacional quanto para a melhoria da generalização do modelo (Edge Impulse, 2026).

As características extraídas são então reorganizadas por meio de uma camada de *flatten* que converte a representação em um vetor unidimensional. Esse vetor é processado por uma camada densa com 32 neurônios, responsável por combinar as características aprendidas e gerar uma representação de alto nível dos padrões presentes nos sinais. A escolha desse número de neurônios visa evitar o aumento excessivo de parâmetros, reduzindo o risco de sobreajuste e mantendo a viabilidade de execução em sistemas embarcados. A camada de saída realiza a classificação entre movimentos normais e eventos de queda, sendo dimensionada de acordo com o número de classes do problema.

Conforme mostrado na Figura 26, o treinamento foi conduzido por 120 ciclos, utilizando taxa de aprendizado de 0,0005 e otimizador adaptativo configurado automaticamente pela plataforma, buscando um equilíbrio entre desempenho e custo computacional.

**Figura 26 – Arquitetura da rede neural configurada no *Edge Impulse*, composta por uma camada de entrada com 120 características, seguida de reorganização dos dados (*reshape*), camadas convolucionais 1D com operações de *pooling*, camada de *flatten*, camada densa com 32 neurônios e camada de saída para classificação em duas classes.**



**Fonte: Autoria própria.**

Não foram utilizadas características no domínio da frequência (*spectral features*), uma vez que eventos de queda apresentam natureza transitória e não periódica, sendo mais adequadamente representados no domínio do tempo. Conforme discutido por Wang *et al.* (2019), abordagens baseadas em *deep learning* tendem a obter melhores resultados quando operam diretamente sobre dados temporais em tarefas envolvendo atividades não periódicas. Além disso, a ausência de transformações espectrais reduz a complexidade computacional do sistema, fator relevante para implementação embarcada.

Essa decisão impacta diretamente a escolha da arquitetura do modelo. As configurações padrão da plataforma *Edge Impulse* frequentemente utilizam convoluções bidimensionais (2D), sobretudo em aplicações nas quais os dados são previamente transformados para o domínio da frequência ou representados como mapas bidimensionais, como espectrogramas. Entretanto, como discutido anteriormente, no presente trabalho os sinais foram mantidos no domínio temporal, preservando sua estrutura como séries temporais multicanais provenientes de sensores inerciais. Embora esses dados possam ser organizados em forma matricial, tal representação não caracteriza uma estrutura bidimensional espacial, mas sim uma única dimensão temporal associada a múltiplos canais de medição. Dessa forma, a utilização de convoluções unidimen-

sionais (1D) mostra-se mais adequada, pois permite a extração de padrões locais ao longo do tempo, considerando simultaneamente todos os eixos do sensor, sem impor relações espaciais artificiais entre eles. Além disso, essa abordagem resulta em menor complexidade computacional quando comparada às convoluções bidimensionais, aspecto fundamental para viabilizar a implementação em sistemas embarcados. A arquitetura final foi, portanto, definida com base nessas considerações teóricas e posteriormente validada por meio de testes experimentais, nos quais apresentou desempenho satisfatório e compatibilidade com as restrições do sistema.

Para realização do ML foram destinados 386 datasets para o treinamento e 104 datasets (apresentados anteriormente na Subseção 3.4.1) para testes dentro do *Edge Impulse*.

Após o treinamento, o modelo foi exportado como parte de uma biblioteca otimizada em C++, contendo os blocos de pré-processamento, extração de características e cálculo do score de anomalia. Essa biblioteca foi integrada ao *firmware* do microcontrolador, permitindo a execução embarcada em tempo real, sem necessidade de processamento externo. Ademais, o código completo em C++ e as bibliotecas utilizadas podem ser acessadas em <https://github.com/rodbcl/Edge-AI-Smartband>.

## 4 RESULTADOS

Neste capítulo são apresentados os resultados da coleta, visualização e categorização dos dados brutos gerados pela IMU. São descritos os procedimentos adotados para a organização dos sinais, a identificação de padrões característicos em diferentes tipos de movimento e a formação do conjunto de dados utilizado nas etapas subsequentes de modelagem. Em seguida, apresentam-se as configurações empregadas no modelos de classificação supervisionada no *Edge Impulse*, finalizando com uma discussão dos resultados observados durante os testes práticos.

### 4.1 Desempenho do Modelo durante Treinamento

Antes da exportação da biblioteca embarcada, o modelo foi avaliado na plataforma *Edge Impulse* utilizando o conjunto de validação, composto por 849 amostras rotuladas. Os resultados demonstraram desempenho satisfatório em ambas as versões do modelo: a versão em ponto flutuante (*float32*) atingiu acurácia de 95,5% e ROC-AUC de 0,956, enquanto a versão quantizada em inteiro de 8 bits (*int8*), utilizada na execução embarcada, apresentou acurácia de 95,2% e ROC-AUC de 0,951. A diferença mínima entre as duas versões indica que o processo de quantização não degradou de forma significativa o desempenho do modelo, aspecto relevante para sua viabilidade em sistemas com recursos computacionais restritos.

A análise da matriz de confusão, representada na Figura 27 referente à versão *int8* revelou que, das 528 amostras da classe normal, 504 foram classificadas corretamente, resultando em 24 falsos positivos. Para a classe queda, das 321 amostras avaliadas, 304 foram corretamente identificadas, com 17 falsos negativos. Considerando as métricas por classe, a classe normal apresentou precisão de 97,5%, *recall* de 95,3% e *F1-score* de 96,4%, enquanto a classe queda obteve precisão de 92,5%, *recall* de 96,0% e *F1-score* de 94,2%. Essas métricas são calculadas e fornecidas automaticamente pela plataforma *Edge Impulse* ao término do treinamento, a partir do conjunto de validação.

**Figura 27 – Matriz de confusão do modelo quantizado em *int8*, exibindo o desempenho de classificação para as classes normal e queda no conjunto de validação.**

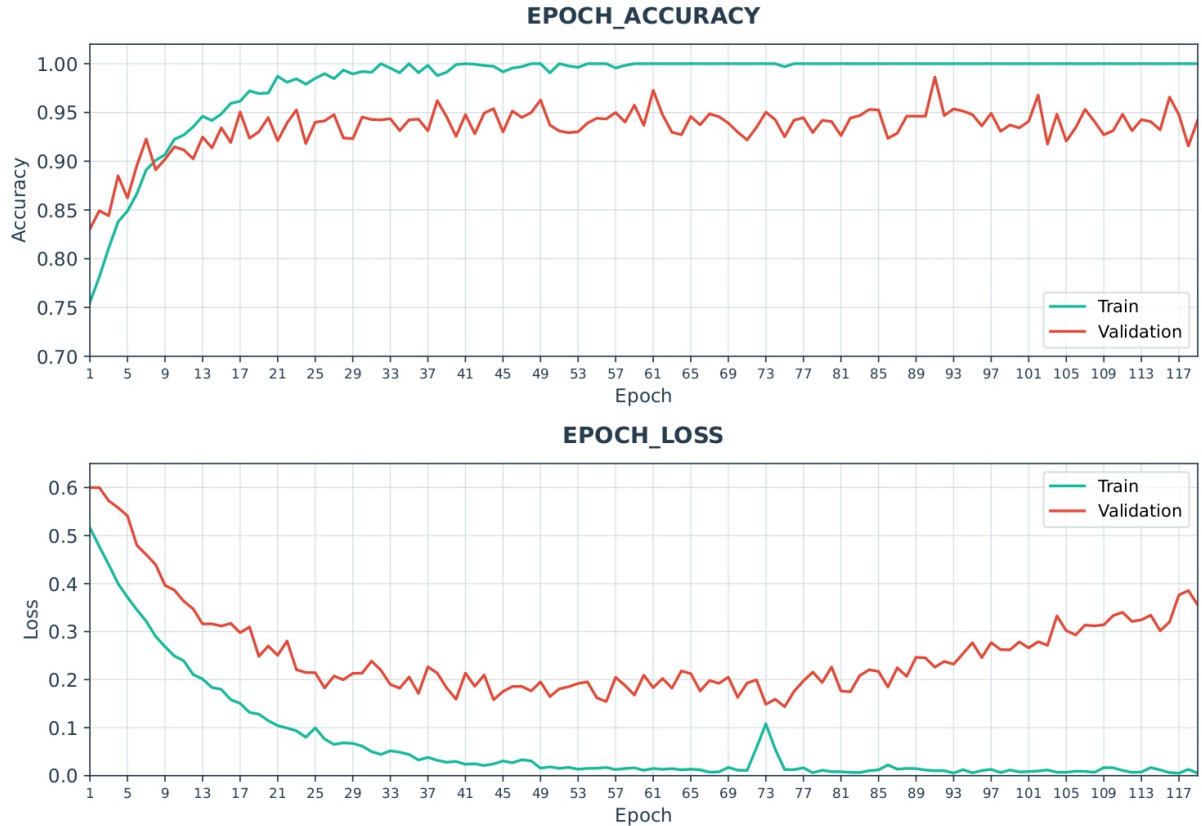
<i>Real ↓ / Predito →</i>	<b>NORMAL</b>	<b>QUEDAS</b>
<b>NORMAL</b>	<b>95,3%</b>	<b>4,7%</b>
<b>QUEDAS</b>	<b>4,0%</b>	<b>96,0%</b>
<b>F1 SCORE</b>	<b>0,96</b>	<b>0,94</b>

**Fonte: Autoria própria.**

A Figura 28 apresenta as curvas de acurácia e de *loss* ao longo das 120 épocas de treinamento, em que cada época corresponde a um ciclo completo de apresentação do conjunto de dados ao modelo durante o processo de ML. A curva de acurácia indica a proporção de amostras classificadas corretamente, enquanto a curva de *loss* representa o erro médio do modelo a cada época, sendo que valores menores indicam melhor ajuste. Observa-se que tanto a curva de treino quanto a de validação apresentaram crescimento rápido nas épocas iniciais, estabilizando-se acima de 95% a partir de aproximadamente a 20<sup>a</sup> época, o que indica convergência adequada do modelo dentro do número de ciclos configurado.

No gráfico de *loss*, a curva de treino apresentou queda consistente ao longo das épocas, aproximando-se de zero ao final do treinamento. A curva de validação reduziu-se de forma expressiva nas épocas iniciais, estabilizando-se em valores próximos a 0,3 com oscilações moderadas ao longo do restante do treinamento. A divergência observada entre as duas curvas é característica comum em modelos treinados com conjuntos de dados de tamanho limitado, indicando leve tendência ao sobreajuste (*overfitting*). Ainda assim, a acurácia de validação manteve-se estável e elevada ao longo de todo o treinamento, demonstrando que o modelo foi capaz de generalizar satisfatoriamente para amostras não vistas durante o treinamento.

**Figura 28 – Curvas de acurácia e de *loss* ao longo das épocas de treinamento do modelo supervisionado no *Edge Impulse*.**

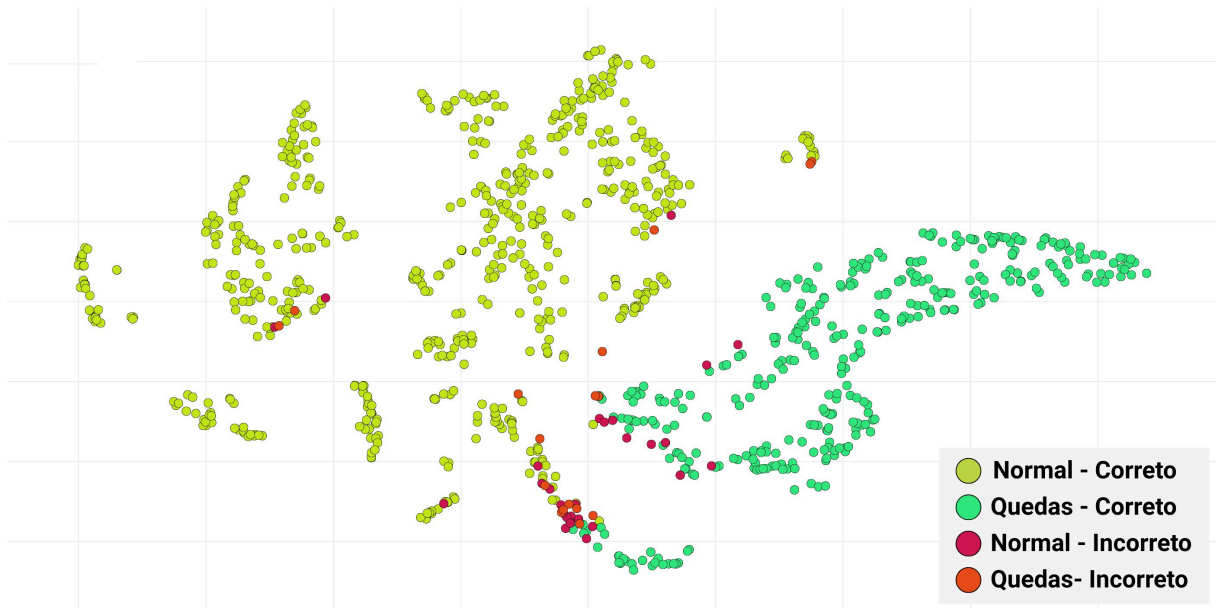


**Fonte: Autoria própria.**

A Figura 29 apresenta a projeção bidimensional das amostras do conjunto de validação gerada pela plataforma *Edge Impulse* por meio da técnica t-SNE (*t-Distributed Stochastic Neighbor Embedding*), a qual reduz os dados de alta dimensionalidade para duas dimensões, preservando as relações de similaridade entre as amostras. Pontos de mesma cor e agrupados indicam que o modelo aprendeu a representar essas amostras de forma semelhante internamente. No gráfico, os pontos verdes claros representam amostras da classe normal classificadas corretamente, os verdes escuros representam quedas classificadas corretamente, enquanto os pontos em rosa e vermelho indicam, respectivamente, amostras normais e quedas classificadas de forma incorreta.

Observa-se que as amostras das duas classes formam agrupamentos predominantemente separados, o que confirma que o modelo aprendeu representações internas distintas para cada classe. Os erros de classificação, representados pelos pontos em rosa e vermelho, concentram-se majoritariamente nas regiões de fronteira entre os agrupamentos, onde a sobreposição entre as classes é maior. Esse comportamento é esperado e reforça a análise da matriz de confusão, indicando que as classificações incorretas ocorrem principalmente em amostras cujas características se aproximam das duas classes simultaneamente.

**Figura 29 – Projeção bidimensional das amostras do conjunto de validação gerada pela técnica t-SNE, exibindo a separação entre as classes normal e queda e a distribuição dos erros de classificação.**



**Fonte: Autoria própria.**

É importante destacar que os resultados da etapa de validação refletem o desempenho do modelo sobre dados coletados em condições controladas e com protocolo padronizado. Como será discutido na seção seguinte, os resultados observados durante os testes práticos com o dispositivo embarcado apresentaram taxas de acerto inferiores, o que evidencia a diferença natural entre o desempenho em validação e a operação em ambiente real, onde a variabilidade dos movimentos é maior e menos previsível.

#### 4.1.1 Resultados dos Testes Práticos

Após a etapa de treinamento e exportação da biblioteca embarcada, foram realizados testes práticos com o dispositivo em operação, com o objetivo de verificar o comportamento do sistema em condições mais próximas do uso real. Essa etapa permitiu avaliar não apenas o desempenho dos modelos dentro da plataforma *Edge Impulse*, mas também sua resposta efetiva quando integrados ao *firmware* do microcontrolador.

##### 4.1.1.1 Desempenho na Detecção de Quedas

Nos testes realizados com o modelo supervisionado, observou-se que o sistema foi capaz de identificar corretamente uma parcela significativa dos eventos de queda simulados. De modo geral, as quedas frontais e de costas apresentaram melhor desempenho, com 10 e 8 detecções corretas de 10 ensaios para cada respectivamente. Esse resultado indica que esses

tipos de queda geraram padrões mais evidentes nos sinais da IMU, principalmente em função da intensidade do impacto e da variação brusca de orientação corporal.

A Tabela 1 apresenta os principais resultados durante os ensaios, incluindo a resposta do modelo supervisionado, as probabilidades atribuídas às classes e os valores máximos de aceleração e velocidade angular registrados em cada amostra. Com o objetivo de sintetizar o desempenho observado nos diferentes cenários de queda, a Tabela 2 apresenta um resumo da quantidade de ensaios realizados, do número de detecções corretas e da taxa de acerto correspondente para cada tipo de queda avaliado.

As quedas laterais apresentaram maior dificuldade de identificação, com 7 detecções corretas em 10 ensaios. A análise dos casos não detectados (amostras 4, 5, e 7) revelou que esses eventos não foram barrados pelo classificador de ML, mas sim pela etapa de verificação física subsequente, descrita na Seção 3.2.4.3. Especificamente, esses ensaios não atingiram os limiares de aceleração (3,80 g) nem a condição combinada de aceleração moderada (3,40 g) com velocidade angular elevada (380°/s), ambos necessários para confirmar o impacto. Esse comportamento é consistente com a natureza biomecânica das quedas laterais: ao cair para o lado, o indivíduo tende a amortecer o impacto com o braço de forma mais eficaz do que nas quedas frontais ou de costas, resultando em picos de aceleração menores registrados no pulso. Esses resultados indicam que os limiares físicos adotados, calibrados com base em quedas frontais e de costas, podem não ser igualmente adequados para quedas laterais, apontando para a necessidade de recalibração ou de limiares adaptativos por tipo de queda em trabalhos futuros.

Ao analisar os dados apresentados na Tabela 1, observa-se ainda que valores elevados de aceleração ou velocidade angular, isoladamente, não foram suficientes para garantir a detecção correta em todos os casos. Esse resultado confirma que o sistema não opera como um simples detector por limiar, mas depende do comportamento temporal completo dos sinais ao longo da janela analisada para tomar a decisão de classificação.

Para contextualizar a contribuição do modelo de ML, é relevante comparar o desempenho do sistema completo com o que seria obtido por uma abordagem de *baseline* baseada exclusivamente em limiares de aceleração, método amplamente utilizado em sistemas mais simples de detecção de quedas (Mubashir; Shao; Seed, 2013). Nessa abordagem hipotética, uma queda seria detectada sempre que o pico de aceleração ultrapassasse o limiar de 3,80 g, sem qualquer etapa de classificação por ML. Analisando os dados da Tabela 1, verifica-se que todas as quedas detectadas pelo sistema completo apresentaram picos de aceleração compatíveis com esse critério, sugerindo que o *baseline* alcançaria taxas de detecção similares para quedas frontais e de costas.

Entretanto, a limitação fundamental de uma abordagem por limiar simples reside na sua incapacidade de distinguir quedas de outros movimentos bruscos que gerem picos de aceleração comparáveis. Durante os testes práticos, verificou-se que atividades como corrida e pulo geraram picos de aceleração suficientes para ultrapassar o limiar de 3,80 g, o que resultaria em falsos positivos sistemáticos em um sistema baseado exclusivamente em limiares. O modelo

supervisionado, por sua vez, foi capaz de filtrar corretamente esses eventos, reconhecendo os padrões temporais característicos dessas atividades a partir do conjunto de dados de treinamento. Esse resultado evidencia que a principal contribuição do ML no sistema proposto não é o aumento da taxa de detecção de quedas em si, mas a redução significativa de falsos positivos em atividades normais de maior intensidade, aspecto crítico para a usabilidade de um dispositivo de monitoramento contínuo.

Foi registrado, no entanto, um caso de falso positivo mesmo com o sistema completo em operação. O evento ocorreu quando o usuário realizou um movimento brusco do braço seguido de imobilidade imediata ao apoiá-lo sobre uma superfície. Essa sequência — pico de aceleração elevado seguido de ausência de movimento — mimetiza estruturalmente a assinatura inercial de uma queda real, satisfazendo simultaneamente os critérios do classificador ML, os limiares de impacto e a verificação de imobilidade pós-evento. Diferentemente dos movimentos contínuos como caminhada e corrida, nos quais a ausência de imobilidade subsequente impede a confirmação do evento, esse padrão específico não pode ser distinguido de uma queda real com base apenas nos dados inerciais do pulso, representando uma limitação inerente à abordagem adotada. A inclusão de amostras dessa natureza no conjunto de treinamento, bem como a adição de sensores complementares como barômetro ou sensor de postura, constituem direções promissoras para mitigar essa limitação em trabalhos futuros.

**Tabela 1 – Resultados dos testes práticos de detecção de quedas.**

Evento	Amostra	Detectada?	Prob. queda	Prob. normal	Pico de aceleração (g)	Pico ang. (°/s)
Queda Frontal	1	Sim	0,999792	0,000208	7,422326	731,4671
	2	Sim	0,996221	0,003779	5,596677	364,1147
	3	Sim	0,998681	0,001319	5,055185	961,2969
	4	Sim	0,999197	0,000803	11,76507	794,8518
	5	Sim	0,864571	0,135429	3,856880	217,4731
	6	Sim	0,869050	0,130950	8,429150	1239,948
	7	Sim	1	≈0	7,262554	490,7361
	8	Sim	0,987958	0,012042	5,510440	1110,313
	9	Sim	1	≈0	7,540747	1812,317
	10	Sim	1	≈0	3,735645	784,3333
Queda Lateral	1	Sim	0,980792	0,019208	5,486835	772,3831
	2	Sim	0,999547	0,000453	5,904374	1158,734
	3	Sim	1	≈0	7,317915	778,2480
	4	Não	-	-	-	-
	5	Não	-	-	-	-
	6	Sim	1	≈0	7,746754	755,2205
	7	Não	-	-	-	-
	8	Sim	1	≈0	8,743870	956,5530
	9	Sim	0,999360	0,000640	7,252871	807,5056
	10	Sim	0,999780	0,000221	7,171060	1276,472
Queda de costas	1	Sim	0,976562	0,023438	8,304289	1196,576
	2	Sim	0,998016	0,001984	7,617810	793,5589
	3	Sim	1	≈0	9,063266	1134,159
	4	Sim	0,999956	0,000044	11,57661	1326,161
	5	Não	-	-	-	-
	6	Sim	0,885366	0,114634	5,808595	1282,329
	7	Sim	0,999868	0,000132	11,38922	2328,146
	8	Não	-	-	-	-
	9	Sim	0,998765	0,001235	11,36541	949,8740
	10	Sim	0,885366	0,114634	6,802395	1881,324

**Fonte: Autoria própria.**

**Tabela 2 – Resumo do desempenho do modelo supervisionado na detecção dos diferentes tipos de queda.**

Tipo de queda	Ensaio realizados	Deteções corretas	Taxa de acerto
Queda frontal	10	10	100%
Queda lateral	10	7	70%
Queda de costas	10	8	80%

**Fonte: Autoria própria.**

#### 4.1.1.2 Desempenho do *geofencing*

Nos testes realizados, o tempo total entre o afastamento do perímetro de cobertura Wi-Fi e o recebimento da notificação no aplicativo correspondeu a 21,78 segundos. Esse valor incorpora o intervalo de confirmação de 15 segundos configurado no *firmware* para validar a desconexão, acrescido da latência inerente à pilha de comunicação, composta pela publicação no *broker* MQTT, pelo encaminhamento da notificação *push* e pelo processamento no sistema operacional do dispositivo móvel. Os testes foram conduzidos com o aplicativo em segundo plano (*background*), condição que representa o cenário de uso real mais frequente.

O tempo de reconexão, medido a partir do retorno do indivíduo ao perímetro até a sinalização de restabelecimento no aplicativo, correspondeu a 4,45 segundos, indicando que o sistema reconhece a reentrada na área monitorada de forma rápida.

É importante destacar que ambos os tempos são influenciados por fatores externos ao *firmware*, como a qualidade da infraestrutura de rede local e o comportamento do sistema operacional do *smartphone* ao processar notificações em segundo plano, não representando, portanto, limitações intrínsecas ao sistema embarcado proposto.

#### 4.1.1.3 Tempo de Processamento e Carga Computacional

Além da avaliação da capacidade de detecção de quedas, foram analisados parâmetros relacionados ao desempenho computacional do sistema embarcado, com o objetivo de verificar sua viabilidade prática durante a operação em tempo real.

Durante os testes realizados, observou-se que o modelo supervisionado apresentou tempo de processamento reduzido e estável, com valores variando entre 7 e 8 ms por inferência. A média observada ao longo da execução permaneceu próxima de 7,03 ms, indicando que o sistema foi capaz de processar os dados de forma rápida e compatível com a aplicação proposta.

Verificou-se ainda que a maior parte do tempo de processamento esteve concentrada na etapa de classificação, enquanto as demais etapas associadas ao processamento do modelo não apresentaram impacto significativo no tempo total. Esse comportamento está relacionado à escolha de utilizar os sinais diretamente no domínio do tempo, sem a necessidade de etapas adicionais de transformação, o que contribuiu para a redução da complexidade computacional do sistema.

A Tabela 3 apresenta um resumo dos principais parâmetros observados durante os testes de execução embarcada. Com base na frequência de execução observada, a carga computacional estimada do modelo permaneceu em torno de 0,7% por inferência, valor considerado baixo para a plataforma utilizada. Esse resultado demonstra que a execução do modelo não comprometeu o funcionamento das demais rotinas do sistema.

**Tabela 3 – Resumo dos principais parâmetros de desempenho computacional observados durante os testes embarcados.**

<b>Parâmetro</b>	<b>Valor observado</b>
Tempo mínimo de inferência	7 ms
Tempo máximo de inferência	8 ms
Tempo médio de inferência	~7,03 ms
Carga computacional estimada (ML)	0,7%
SRAM interna total (ESP32-S3)	512 kB
<i>Heap</i> livre típico em operação	~243 kB
Menor <i>heap</i> livre registrado	~235 kB
Quedas confirmadas durante o ensaio	2 de 2
Estabilidade geral do sistema	Estável

**Fonte: Autoria própria.**

Em relação ao uso de memória, observou-se comportamento estável durante a execução contínua do sistema. A memória dinâmica livre, que representa a quantidade de memória disponível para alocação durante a execução do programa, permaneceu majoritariamente próxima de 243 kB ao longo dos testes, ou seja, 269 kB de memória foram utilizados pelo sistema do total de 512 kB disponibilizados pela M5Stack AtomS3R-Ext. Essa variação limitada indica que não houve consumo progressivo de memória ao longo do tempo. Complementarmente, a análise do binário gerado para o *firmware* revelou que a aplicação ocupa 1.370.207 bytes de memória Flash, o que representa 41% da partição destinada ao código no dispositivo. No que tange à memória estática, o sistema utiliza 56.292 bytes (17,2%) da RAM disponível para variáveis globais e estruturas de dados pré-alocadas.

Foram analisadas também as tarefas responsáveis pela aquisição dos sensores, processamento do modelo, comunicação com o servidor e lógica física complementar. Em todos os casos observou-se a manutenção de margem livre de memória, indicando que o dimensionamento adotado foi adequado para a carga de processamento exigida.

Durante os ensaios, o sistema operou de forma contínua e estável, sem ocorrência de reinicializações inesperadas ou falhas críticas perceptíveis. Além disso, foram observados dois eventos de queda confirmados (durante a observação de *logs* foram realizadas exatamente duas simulações de queda) durante a execução embarcada, evidenciando que a integração entre aquisição de dados, processamento e lógica de decisão ocorreu de maneira funcional no *hardware* utilizado.

De forma geral, os resultados demonstram que a solução proposta apresentou viabilidade computacional para aplicação embarcada, sendo capaz de executar as etapas de aquisição, processamento e tomada de decisão em tempo real, dentro das limitações do microcontrolador utilizado.

#### 4.1.2 Comparação com Trabalhos Relacionados

A fim de contextualizar os resultados deste trabalho, apresenta-se uma comparação com o estudo de Volochtchuk e Bassai (2023), que desenvolveu um sensor de monitoramento de queda em idosos utilizando o microcontrolador Arduino Nano 33 BLE Sense equipado com acelerômetro e giroscópio, classificando os eventos por meio de modelos de ML supervisionado na plataforma *Edge Impulse*.

##### 4.1.2.1 Hardware e Posicionamento do Sensor

Volochtchuk e Bassai (2023) avaliaram duas posições de fixação do dispositivo (pé e peito) e concluíram que o posicionamento no peito proporcionou resultados significativamente superiores, uma vez que os sinais coletados no pé apresentavam sobreposição entre os padrões de queda e atividades cotidianas como caminhar. O presente trabalho optou pelo posicionamento no pulso, por meio do M5Stack AtomS3R-Ext, o que favorece a aceitação pelo usuário

e permite monitoramento contínuo sem interferência nas atividades diárias. Essa escolha, contudo, introduz o desafio de movimentos bruscos do braço, que podem gerar assinaturas inerciais semelhantes às de uma queda real, conforme discutido na Seção 4.1.1.1.

#### 4.1.2.2 Desempenho dos Modelos de Classificação

Em termos de acurácia, o trabalho de Volochtchuk e Bassai (2023) reportou 92,31% no conjunto de teste utilizando a plataforma *Edge Impulse*. O presente trabalho obteve acurácia de 95,5% na versão em ponto flutuante (*float32*) e 95,2% na versão quantizada em inteiro de 8 bits (*int8*), ambas avaliadas sobre o conjunto de validação composto por 849 amostras. Adicionalmente, o presente trabalho reportou a métrica ROC-AUC, que atingiu 0,956 e 0,951 para as versões *float32* e *int8*, respectivamente (métrica não utilizada em Volochtchuk e Bassai (2023)), o que limita a comparação direta entre os trabalhos. A diferença mínima de desempenho entre as duas versões do modelo aqui desenvolvido indica que o processo de quantização não degradou de forma significativa a capacidade de classificação, aspecto relevante para a viabilidade embarcada do sistema.

#### 4.1.2.3 Falsos Positivos

Ambos os trabalhos identificaram limitações relacionadas à geração de falsos positivos. Em Volochtchuk e Bassai (2023), a principal dificuldade estava associada à semelhança entre os padrões de queda e atividades físicas de maior intensidade, como corrida. No presente trabalho, o único falso positivo registrado durante os testes práticos ocorreu em um cenário específico: um movimento brusco do braço seguido de imobilidade imediata ao apoiá-lo sobre uma superfície. Essa sequência mimetiza estruturalmente a assinatura inercial de uma queda real, satisfazendo simultaneamente os critérios do classificador, os limiares de impacto e a verificação de imobilidade pós-evento. Trata-se de uma limitação inerente ao posicionamento no pulso, não observada nos trabalhos que adotaram sensores no tronco, e que aponta para a necessidade de inclusão de amostras dessa natureza no conjunto de treinamento ou de sensores complementares em versões futuras do dispositivo.

#### 4.1.2.4 Implementação Embarcada

Uma distinção relevante entre os dois trabalhos reside no nível de integração alcançado. Volochtchuk e Bassai (2023) realizaram os testes finais por meio da função *Live Classification* do *Edge Impulse*, com o dispositivo conectado via USB a um computador, sem que o modelo fosse efetivamente executado de forma autônoma no microcontrolador. O presente trabalho avançou nessa direção ao exportar a biblioteca embarcada gerada pela plataforma e integrá-la diretamente ao *firmware* do microcontrolador, permitindo a execução completa do *pipeline* de aquisição, inferência e tomada de decisão em tempo real. O modelo embarcado apresentou tempo médio de inferência de aproximadamente 7,03 ms por ciclo e carga computacional estimada em torno de 0,7%, demonstrando viabilidade prática para operação contínua e autônoma do dispositivo.

## 5 CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de uma *smartband* com inteligência artificial embarcada para a detecção automática de quedas em indivíduos com comprometimento cognitivo. Ao longo do projeto, foram integradas com sucesso diferentes frentes de desenvolvimento: *hardware*, *firmware*, ML, comunicação e interface gráfica, resultando em um sistema funcional, compacto e com viabilidade computacional demonstrada em ambiente real.

A calibração dos sensores do protótipo foi realizada com sucesso por meio da biblioteca fornecida pelo fabricante do *hardware*, garantindo alta precisão nas leituras. Essa exatidão, combinada a um banco de dados bem estruturado com amostras dos próprios autores e a um modelo de ML treinado com êxito no *Edge Impulse*, reduziu significativamente a incidência de falsos positivos, consolidando a confiabilidade do ecossistema.

A otimização e o embarcamento do modelo de ML no *firmware* foram realizados com sucesso, permitindo que o processamento ocorresse localmente no dispositivo. Essa abordagem reduz a necessidade de transmissão contínua de dados ao servidor, contribuindo para a diminuição do consumo energético e melhorando a experiência de uso do dispositivo. Adicionalmente, foi implementado no *firmware* um mecanismo de *geofencing* baseado em varredura de redes Wi-Fi, capaz de identificar a ausência dos pontos de acesso da residência e sinalizar a saída do portador do perímetro configurado.

Por fim, foi desenvolvido com êxito um aplicativo móvel para o monitoramento do portador do dispositivo, capaz de transmitir perfeitamente o estado em que a pulseira se encontra. Além disso, a interface limpa e simples facilita a leitura para qualquer pessoa responsável pelo monitoramento. Como produto adicional deste trabalho, foi elaborado um artigo científico submetido ao Congresso Brasileiro de Engenharia Biomédica (CBEB) 2026, intitulado *Wrist-Worn Smartband with Embedded Artificial Intelligence for Fall Detection in Individuals with Cognitive Impairment*, reproduzido no Apêndice A. Portanto, conclui-se que este trabalho atingiu com sucesso os objetivos propostos, resultando no desenvolvimento de uma solução funcional, confiável e com potencial de aplicação real no monitoramento e na segurança de indivíduos com comprometimento cognitivo.

### 5.1 SUGESTÕES DE TRABALHOS FUTUROS

Apesar dos resultados positivos, o trabalho também identificou limitações que apontam direções para trabalhos futuros, como a expansão da base de dados com o objetivo de generalizar o modelo, diversificando os indivíduos presentes nas coletas. Adicionalmente, recomenda-se a realização de testes em ambientes reais com o público-alvo do trabalho. Outra sugestão é o desenvolvimento de um *hardware* próprio e específico para o projeto, com foco na ergonomia do dispositivo e no aprimoramento do sistema de *geofencing*, por meio da inclusão de um módulo de GPS, com o objetivo de tornar mais preciso o alerta de saída do perímetro.

## REFERÊNCIAS

- BAYILMIŞ, C. *et al.* A survey on communication protocols and performance evaluations for internet of things. **Digital Communications and Networks**, 2022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352864822000347>. Acesso em: mar. 2026.
- BLUETOOTH SIG . **Bluetooth® Core Specification v5.0**. [S./], 2016. Disponível em: [www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth\\_5-FINAL.pdf](http://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth_5-FINAL.pdf). Acesso em: set. 2025.
- BRUHN, J. *et al.* Falls and fall risk in community-dwelling people with dementia: A systematic review. **BMC Geriatrics**, 2021. Disponível em: [bmcgeriatr.biomedcentral.com/articles/10.1186/s12877-021-02641-9](http://bmcgeriatr.biomedcentral.com/articles/10.1186/s12877-021-02641-9). Acesso em: set. 2025.
- EDGE IMPULSE . **Edge Impulse Documentation**. [S./], 2025. Disponível em: <https://docs.edgeimpulse.com/>. Acesso em: jan. 2026.
- Edge Impulse . **Neural Networks**. [S./], 2026. Disponível em: <https://docs.edgeimpulse.com/knowledge/concepts/machine-learning/neural-networks>. Acesso em: mar. 2026.
- ESPRESSIF . **ESP32**. [S./], 2023. Disponível em: <https://www.espressif.com/>. Acesso em: jan. 2026.
- ESPRESSIF SYSTEMS . **ESP32-S3 Datasheet**. [S./], 2023. Disponível em: [www.espressif.com/sites/default/files/documentation/esp32-s3\\_datasheet\\_en.pdf](http://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf). Acesso em: set. 2025.
- FERREIRA, R. N.; RIBEIRO, N. F.; SANTOS, C. P. Fall risk assessment using wearable sensors: A narrative review. **Sensors**, 2022. Disponível em: [doi.org/10.3390/s22030984](https://doi.org/10.3390/s22030984). Acesso em: set. 2025.
- GONZAGA, M. R.; PEREIRA, R. C.; SANTOS, D. A. . **Uso de tecnologias vestíveis na assistência domiciliar de idosos: uma revisão integrativa**. [S./], 2022. Disponível em: [editorarealize.com.br/editora/anais/cieh/2022/TRABALHO\\_COMPLETO\\_EV179\\_MD1\\_ID470\\_TB169\\_11092022183306.pdf](http://editorarealize.com.br/editora/anais/cieh/2022/TRABALHO_COMPLETO_EV179_MD1_ID470_TB169_11092022183306.pdf). Acesso em: set. 2025.
- GOOGLE . **Google Firebase**. [S./], 2024. Disponível em: <https://firebase.google.com/?hl=pt-br>. Acesso em: jan. 2026.
- HAMMERLA, N. Y.; HALLORAN, S.; PLÖTZ, T. Deep, convolutional, and recurrent models for human activity recognition using wearables. **Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)**, 2016. Disponível em: <https://www.ijcai.org/Proceedings/16/Papers/255.pdf>. Acesso em: mar. 2026.
- IBM . **What is edge AI?** [S./], 2025. Disponível em: <https://www.ibm.com/think/topics/edge-ai>. Acesso em: jan. 2026.
- JANG, S. *et al.* An effective deep learning framework for fall detection. **Journal of Medical Internet Research**, 2024. Disponível em: <https://www.jmir.org/2024/1/e56750/>. Acesso em: set. 2025.
- JARA-OCHOA, H. J. *et al.* Comparative analysis of power consumption between MQTT and HTTP protocols in an IoT platform designed for real-time monitoring. **Sensors**, 2023. Disponível em: <https://www.mdpi.com/1424-8220/23/10/4896>. Acesso em: mar. 2026.
- LI, F. *et al.* A decade of progress in wearable sensors for fall detection. **Sensors**, 2025. Disponível em: <https://www.mdpi.com/1424-8220/25/7/2205>. Acesso em: set. 2025.

- LIN, J. *et al.* . **MCUNet: Tiny Deep Learning on IoT Devices**. [S.l.], 2020. Disponível em: [arxiv.org/abs/2007.10319](https://arxiv.org/abs/2007.10319). Acesso em: set. 2025.
- LORA ALLIANCE . **LoRaWAN Specification 1.0.4**. [S.l.], 2020. Acesso em: set. 2025.
- MARQUES, G. *et al.* Online fall detection using wrist devices. **Sensors**, 2023. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC9920426/>. Acesso em: set. 2025.
- Ministério da Saúde do Brasil . **Relatório nacional sobre a demência estima que cerca de 8,5% da população idosa convive com a doença**. [S.l.], 2024. Disponível em: <https://shre.ink/31nB>. Acesso em: set. 2025.
- MUBASHIR, M.; SHAO, L.; SEED, L. . **A survey on fall detection: Principles and approaches**. [S.l.], 2013. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0925231212003153>. Acesso em: set. 2025.
- NORDIC SEMICONDUCTOR . **nRF52840 Product Specification v1.1**. [S.l.], 2021. Disponível em: [docs.nordicsemi.com/bundle/ps\\_nrf52840/page/keyfeatures\\_html5.html](https://docs.nordicsemi.com/bundle/ps_nrf52840/page/keyfeatures_html5.html). Acesso em: set. 2025.
- NUNEZ, J. O. *et al.* Analysis of waist and wrist positioning wearable machine-learning models to detect falls. **Electronics Letters**, 2024. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ell2.13086>. Acesso em: set. 2025.
- OASIS . **MQTT Version 3.1.1**. [S.l.], 2014. Disponível em: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>. Acesso em: mar. 2026.
- OLMEDO-AGUIRRE, J. M. *et al.* Remote healthcare for elderly people using wearables: A review. **Sensors**, 2022. Disponível em: [pmc.ncbi.nlm.nih.gov/articles/PMC8869443](https://pmc.ncbi.nlm.nih.gov/articles/PMC8869443). Acesso em: set. 2025.
- PROMETTEUR SOLUTIONS . **Por que o Flutter é o Futuro do Desenvolvimento de Apps**. [S.l.], 2025. Disponível em: <https://prometteursolutions.com/blog/pt/por-que-o-flutter-e-o-futuro-do-desenvolvimento-de-apps/>. Acesso em: jan. 2026.
- SANTOS, L. C. *et al.* Identifying sensors-based parameters associated with fall risk in older adults. **BMC Geriatrics**, 2024. Disponível em: [doi.org/10.1186/s12877-024-04723-w](https://doi.org/10.1186/s12877-024-04723-w). Acesso em: set. 2025.
- SANTOS, R. F. *et al.* Prevalência e fatores associados às quedas em idosos com comprometimento cognitivo leve e demência. **Cadernos Saúde Coletiva**, 2017. Disponível em: [www.scielo.br/j/cadsc/a/kXH887GZMQVFZVKYy8Z5GGQ](https://www.scielo.br/j/cadsc/a/kXH887GZMQVFZVKYy8Z5GGQ). Acesso em: set. 2025.
- SEMTECH CORPORATION . **SX1276/77/78/79 Datasheet**. [S.l.], 2022. Disponível em: [semtech.com/products/wireless-rf/lora-connect/sx1276](https://semtech.com/products/wireless-rf/lora-connect/sx1276). Acesso em: set. 2025.
- SHELBY, Z. AND HARTKE, K. AND BORMANN, C. . **The Constrained Application Protocol (CoAP) – RFC 7252**. [S.l.], 2014. Disponível em: <https://datatracker.ietf.org/doc/html/rfc7252>. Acesso em: mar. 2026.
- STMICROELECTRONICS . **AN4872 – Wi-Fi Power Consumption Estimation**. [S.l.], 2020. Disponível em: [www.st.com/resource/en/application\\_note/an4872-wifi-power-consumption-estimation-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an4872-wifi-power-consumption-estimation-stmicroelectronics.pdf). Acesso em: set. 2025.
- VERMA, D. *et al.* Internet of things (iot) in nano-integrated wearable devices for healthcare: A survey. **Nano Communication Networks**, 2022. Disponível em: [sciencedirect.com/science/article/pii/S2590137022000486](https://www.sciencedirect.com/science/article/pii/S2590137022000486). Acesso em: set. 2025.

VOLOCHTCHUK, A. V. L.; BASSAI, S. W. **Desenvolvimento de sensor para monitoramento de queda em idosos**. 2023. Dissertação (Trabalho de Conclusão de Curso de Graduação) — Universidade Tecnológica Federal do Paraná Curitiba 2023. Disponível em: [https://github.com/sabrinawb/TCC\\_2.git](https://github.com/sabrinawb/TCC_2.git). Acesso em: mar. 2026.

WANG, J. *et al.* Deep learning for sensor-based activity recognition: A survey. **Pattern Recognition Letters**, 2019. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0167865518300450>. Acesso em: mar. 2026.

World Health Organization . **Falls**. [S.l.], 2021. Disponível em: [www.who.int/news-room/fact-sheets/detail/falls](http://www.who.int/news-room/fact-sheets/detail/falls). Acesso em: set. 2025.

WU, Y. *et al.* . **Monitoring wandering behaviors of dementia patients using BLE-based localization systems**. [S.l.], 2024. Disponível em: [arxiv.org/abs/2403.19704](https://arxiv.org/abs/2403.19704). Acesso em: set. 2025.

YU, R. *et al.* Falls in people with alzheimer's disease and mild cognitive impairment: A systematic review and meta-analysis. **Annals of Geriatric Medicine and Research**, 2022. Disponível em: [agmr.hapres.com/htmls/AGMR\\_1565\\_Detail.html](http://agmr.hapres.com/htmls/AGMR_1565_Detail.html). Acesso em: set. 2025.

**APÊNDICE A – Wrist-Worn Smartband with Embedded Artificial Intelligence for Fall  
Detection in Individuals with Cognitive Impairment**

# Wrist-Worn Smartband with Embedded Artificial Intelligence for Fall Detection in Individuals with Cognitive Impairment

Gabriel Cortes Pires Leopolski

*Dept. of Electrical Engineering  
(DAELT)  
Federal University of Technology  
– Paraná (UTFPR)  
Curitiba, Brazil  
gabrielleopolski@alunos.utfpr.edu.br*

Leonardo Zucco

*Dept. of Electrical Engineering  
(DAELT)  
Federal University of Technology  
– Paraná (UTFPR)  
Curitiba, Brazil  
zucco@alunos.utfpr.edu.br*

Rodrigo Baldissera Cerqueira Leite

*Dept. of Electrical Engineering  
(DAELT)  
Federal University of Technology  
– Paraná (UTFPR)  
Curitiba, Brazil  
rodrigobaldissera@alunos.utfpr.edu.br*

Renata Coelho Borges

*Graduate Program in  
Biomedical Engineering  
(PPGEB), DAELT  
Federal University of Technology  
– Paraná (UTFPR)  
Curitiba, Brazil  
renatacoelho@utfpr.edu.br*

Adriano Ruseler

*Dept. of Electrical Engineering  
(DAELT)  
Federal University of Technology  
– Paraná (UTFPR)  
Curitiba, Brazil  
ruseler@utfpr.edu.br*

Alexandre Francisco de Moraes Filho

*Federal Institute of  
Paraná (IFPR)  
Curitiba Campus  
Curitiba, Brazil  
alexandre.moraes@ifpr.edu.br*

Fábio Pires Itturriet

*Graduate Program in  
Biomedical Engineering  
(PPGEB), DAELT  
Federal University of Technology  
– Paraná (UTFPR)  
Curitiba, Brazil  
fabioitturriet@utfpr.edu.br*

Amauri Amorin Asséf

*Graduate Program in  
Electrical and Industrial Informatics  
Engineering (CPGEI), DAELT  
Federal University of Technology  
– Paraná (UTFPR)  
Curitiba, Brazil  
amauriassef@utfpr.edu.br*

**Abstract**—Population aging and the increasing prevalence of neuro-degenerative diseases have increased the need for technological solutions capable of supporting health monitoring and preventing adverse events, particularly falls among individuals with cognitive impairment. This paper presents the development of a wrist-worn smartband based on embedded artificial intelligence for automatic fall detection. The proposed system integrates an M5Stack AtomS3R-Ext microcontroller with an embedded Inertial Measurement Unit (IMU), a 1D convolutional neural network trained via Edge Impulse, and a multi-stage validation method to reduce false positives. Communication is handled through the Message Queuing Telemetry Transport (MQTT) protocol, enabling alerts via mobile application. The embedded model achieved 95.2% accuracy and a ROC-AUC of 0.951, with a mean inference time of 7 ms and a computational load of 0.7%. Experimental results demonstrated detection rates of 100% of forward falls, 80% of backward falls, and 70% of lateral falls. These findings demonstrate the feasibility of on-device AI inference for continuous, low-latency fall detection in resource-constrained wearable devices.

**Index Terms**—fall detection, embedded artificial intelligence, wearable devices, Edge Impulse, elderly monitoring, cognitive impairment

## I. INTRODUCTION

Population aging is a growing global trend, with the proportion of people aged 60 or older expected to double by 2050, according to the World Health Organization (WHO) [1]. In Brazil, approximately 8.5% of the elderly population live with some form of dementia, a figure projected to rise from 1.8 million to 5.7 million by 2050 [2]. Neurodegenerative diseases impair cognitive and functional abilities, substantially increasing the risk of falls. In this context, cognitive impairment refers to deficits in functions such as memory, attention, spatial orientation, judgment and autonomy, which may increase fall risk and hinder the ability to seek help after an adverse event. Studies report annual fallrates of 40-60% among elderly individuals with dementia [3], [4], compared to roughly one-third of the general older population [1].

Falls are a leading cause of morbidity and mortality among this population, making reliable, fast-response detection systems essential [1]. Wrist-worn wearables equipped with Inertial Measurement Units (IMUs) are particularly attractive due to their unobtrusiveness, continuous monitoring capability, and

high user acceptance [5], [6]. When combined with machine learning (ML), such devices can distinguish fall events from activities of daily living with high accuracy [7], [8]. However, several fall-detection systems still rely on offline processing or cloud-based inference [9], [11], which introduces latency and privacy concerns, and increases energy consumption. Edge Artificial Intelligence (AI), executing ML models directly on the microcontroller, addresses these limitations by enabling local, real-time inference without constant server dependency [10], [12].

### A. Related Work

Several studies have explored wearable-based fall detection using ML across different body placements, as summarized in Table I. Marques et al. [7] evaluated classical ML algorithms on a wrist-worn device, while Nunez et al. [8] compared waist and wrist placements, reporting that wrist sensors exhibit higher classification ambiguity than trunk-mounted alternatives. Jang et al. [9] employed a dual-stream CNN with self-attention for waist devices, and Li et al. [10] reviewed a decade of progress, highlighting trends toward embedded AI.

The most comparable study is that of Volochtchuk and Bassai [11], who reported 92.31% accuracy using Edge Impulse, but conducted their final tests with the device connected to a computer, without autonomous deployment. In contrast, this work prioritizes wrist placement for user acceptance and advances to full on-device inference, combining the CNN classifier with a multi-stage physical verification method to reduce false positives and a Wi-Fi-based geofencing module. Beyond falls, wandering behaviors [13] and IMU-derived fall-risk parameters [14] further reinforce the relevance of the proposed approach.

TABLE I  
SUMMARY OF RELATED WORK ON WEARABLE FALL DETECTION

Study	Placement	Algorithm	Key Contribution
Marques et al. [7]	Wrist	Classical ML	Offline evaluation of multiple algorithms
Nunez et al. [8]	Wrist/ Waist	Random Forest	Compares sensor placement impact
Jang et al. [9]	Waist	Dual-stream CNN	Self-attention for complex fall patterns
Li et al. [10]	Multiple	Classical ML / DL	Decade review; trends in embedded AI
Volochtchuk & Bassai [11]	Chest / Foot	SVM, Decision Tree, Logistic Regression	92.31% accuracy; no autonomous deployment
<b>This work</b>	<b>Wrist</b>	<b>1D-int8-CNN</b>	<b>Full on-device inference; Mobile App</b>

This paper presents a wrist-worn smartband prototype with fully embedded fall detection smartband built around the M5Stack AtomS3R-Ext microcontroller, a 1D convolutional neural network (CNN) quantized to 8-bit integer (int8) and

deployed via Edge Impulse platform [15], and a multi-level physical verification method that reduces false positives.

## II. MATERIALS AND METHODS

### A. System Architecture

The system comprises three main layers: (1) a wrist-worn embedded device responsible for data acquisition, on-device ML inference, and fall confirmation; (2) Message Queuing Telemetry Transport (MQTT)-based communication layer for event-driven messaging; and (3) a mobile application specially designed to generate and deliver alert notifications to caregivers following a confirmed fall event. Fig. 1 illustrates the overall system architecture, while Fig. 2 shows the mobile application interface.

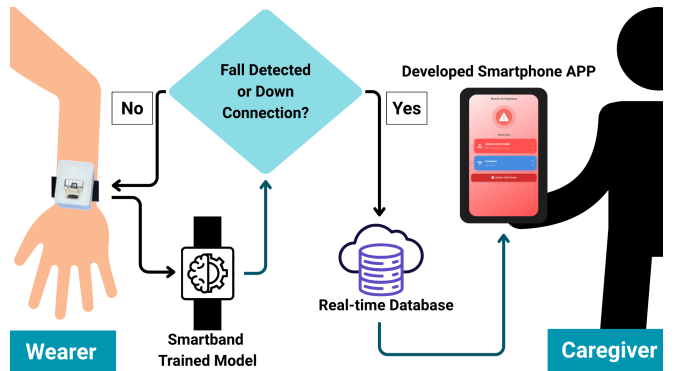


Fig. 1. System architecture: the embedded device acquires IMU data, runs on-device ML inference, and publishes fall or geofencing events via MQTT to a cloud backend, which triggers push notifications to the caregiver’s mobile application.

During normal activity, no data are published to the broker, minimizing network traffic and energy consumption. A fall event triggers a compact MQTT message to the backend, which updates the Firebase real-time database and dispatches a push notification via Firebase Cloud Messaging (FCM) to the caregiver’s Flutter application.

### B. Hardware Device

The M5Stack AtomS3R-Ext [16] was selected as the embedded platform due to its combination of compact form factor ( $24 \times 24 \times 13$  mm), built-in IMU (three-axis BMM150 geomagnetic sensor and a six-axis BMI270 attitude sensor), dual-core Xtensa<sup>®</sup> LX7 processor at up to 240 MHz, 8 MB of Flash, 8 MB of PSRAM, native Wi-Fi connectivity, and low cost (approximately US\$14.50). These characteristics make it well suited for a wrist-worn wearable requiring on-device ML inference and wireless communication. The device was also selected due to its native compatibility with Edge Impulse, an edge AI platform that provides an integrated workflow for the entire ML pipeline on embedded systems, including data acquisition, model training, optimization, and deployment. This integration enables the deployment of optimized TinyML models directly onto microcontrollers, reducing development

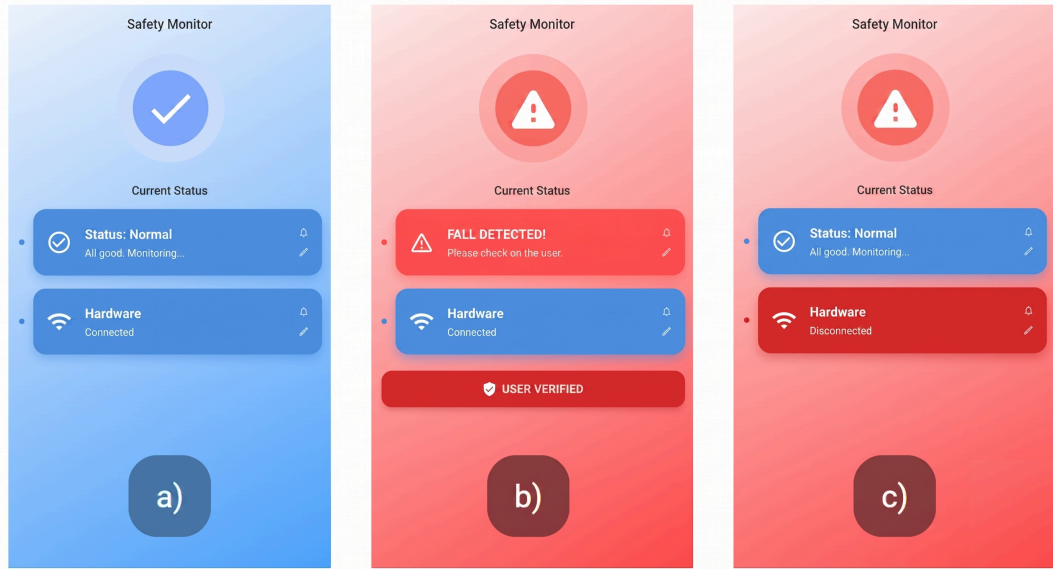


Fig. 2. Mobile application dashboard showing three operating states: (a) normal operation, with the wearable device connected and monitoring active; (b) fall detected, displaying an alert and confirmation button; (c) device disconnected, indicating loss of communication with the smartband.

complexity and computational overhead. The selected micro-controller and its battery module are shown in Fig. 3. An Atomic Battery Base (200 mAh,  $24 \times 24 \times 27$  mm) was used solely to provide autonomous power during data collection and testing; battery-life optimization was beyond the scope of this work and is left for future development. A custom 3D-printed PLA enclosure fixed the device to the wrist via a Velcro strap, yielding a total wearable mass of 35 g, as shown in Fig. 4 and 5.

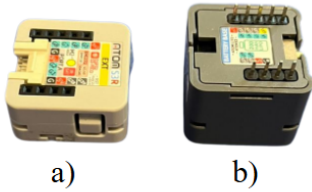


Fig. 3. Hardware used in the prototype: (a) the M5Stack AtomS3R-Ext microcontroller and (b) the Atomic Battery Base module.

### C. Signal Acquisition and Filtering

The IMU outputs accelerometer data in multiples of gravitational acceleration ( $g$ ) and gyroscope data in degrees per second ( $^\circ/s$ ). A moving-average filter with window size  $N$  was applied to attenuate high-frequency noise:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \quad (1)$$



Fig. 4. Prototype of the wrist-worn smartband: M5Stack AtomS3R-Ext enclosed in a 3D-printed PLA case and fixed to the wrist via a Velcro strap.



Fig. 5. Smartband worn on the wrist during operation.

where  $y[n]$  is the filtered output at sample  $n$ ,  $x[n]$  is the raw sensor signal, and  $N$  is the window size. Windows of  $N \in \{1, 5, 10, 20\}$  were evaluated.  $N = 5$  provided the best trade-off between noise suppression and dynamic response preservation, and was adopted as the definitive filter parameter.

#### D. Machine Learning Pipeline

1) *Dataset Collection*: A pilot experimental protocol was conducted with two healthy adult volunteers, both authors of this work, simulating fall events and activities of daily living (ADL). Fall events (forward, lateral, and backward) were recorded, resulting in 100 simulated fall trials per subject, totalling 200 labelled fall trials. ADL samples, walking, light running, jumping, sitting, dishwashing, and arm raising, were also recorded (approximately 10 samples per activity per subject). Raw IMU signals were then segmented into windowed samples of 1000 ms with a 250 ms stride (75% overlap) at a 20 Hz sampling frequency, yielding a total of 490 windowed samples distributed between two classes: *fall* and *non-fall*. The dataset was split randomly at the window level by the Edge Impulse platform, resulting in 386 training and 104 test samples, with both volunteers contributing to both subsets. Since windows were extracted with 75% overlap, adjacent windows from the same event may appear in both the training and test sets. This characterizes a potential data leakage, which may artificially inflate the reported accuracy and limit generalization to unseen subjects. Furthermore, falls in elderly individuals may exhibit kinematic signatures different from those of healthy young adults, due to factors such as reduced movement speed, impaired protective reflexes, and altered gait patterns. Consequently, the dataset collected in this study should be regarded as a proof-of-concept, and validation with the target population remains necessary. Fig. 6 illustrates a representative IMU signal sample recorded during a forward fall trial, showing the characteristic pattern of accelerometer and gyroscope responses across the three phases: motionless, fall, and post-fall motionless.

2) *Model Architecture*: The model was developed on the Edge Impulse platform. Raw six-axis IMU signals were segmented into 1000 ms windows with a 250 ms stride (75% overlap) at 20 Hz, resulting in 20 samples per axis and 120 input values per window. Time-domain raw data were used directly, without spectral transformation, since falls are transient and non-periodic events better represented in the temporal domain [17]. A 1D CNN was employed, as illustrated in Fig. 7. The architecture consists of an input layer (120 features), a reshape layer to organize the 120 input values according to the six IMU axes, two 1D conv/pool blocks (24 filters, kernel size of 5; followed by 48 filters, kernel size 3), a flatten layer, a dense layer (32 neurons), and a 2-class softmax output layer. Training was performed for 120 epochs with a learning rate of 0.0005, selected empirically to ensure stable convergence without overshooting the loss minimum.

After training, the model was quantized to int8 and exported as an optimized C++ library integrated directly into the device firmware. The int8 version was selected for embedded deployment to reduce memory footprint and inference cost on the microcontroller, enabling real-time operation within the resource constraints of the M5Stack AtomS3R-Ext.

#### E. Multi-Stage Fall Confirmation

To reduce false positives inherent to wrist-worn sensors, a three-stage verification pipeline was implemented after the CNN inference. All thresholds were determined empirically based on the analysis of IMU signal plots collected during the experimental protocol, selecting values that maximized fall detection while minimizing false positives in ADL scenarios.

**Stage 1 – K-of-M temporal voting**: a fall suspicion is raised when at least  $K = 1$  of the last  $M = 2$  inference windows exceeds a fall probability threshold of 0.85. These values were determined empirically: more conservative configurations (e.g.,  $K = 3$ ,  $M = 2$ ) caused genuine fall events to be missed, since requiring multiple high-probability windows within such a short sequence is overly restrictive given that a fall is a brief, transient event spanning few windows. While increasing  $M$  could potentially reduce false positives during high-intensity activities, it would also delay or suppress the detection of real falls, compromising the system’s primary safety objective. The selected  $K = 1$ ,  $M = 2$  configuration prioritizes detection sensitivity, with false-positive suppression delegated to the subsequent physical verification stages.

**Stage 2 – Impact verification**: within a 1.8 s window following suspicion, either (a) acceleration magnitude exceeds  $3.80 g$ , or (b) acceleration magnitude exceeds  $3.40 g$  simultaneously with angular velocity magnitude exceeding  $380^\circ/s$ .

**Stage 3 – Post-impact immobility**: over a minimum 1 s interval, angular velocity must remain below  $20^\circ/s$  and acceleration magnitude variation must stay within  $0.12 g$  of the reference value, confirming that the subject is motionless after impact.

Only events satisfying all three stages are classified as confirmed falls and trigger an MQTT alert.

### III. RESULTS

#### A. Model Validation Performance

The model was evaluated on a validation set of 849 windowed samples automatically held out by the Edge Impulse platform during training, where each sample corresponds to a 1000 ms window extracted from the collected event recordings. The float32 version achieved 95.5% accuracy and a ROC-AUC of 0.956. The int8 quantized version, used for embedded execution, achieved 95.2% accuracy and a ROC-AUC of 0.951, confirming that quantization introduced negligible performance degradation. Per-class metrics for the int8 model are summarized in Table II.

TABLE II  
PER-CLASS PERFORMANCE METRICS – INT8 MODEL (VALIDATION SET,  
 $n = 849$ )

Class	Precision	Recall	F1-score
Non-fall	97.5%	95.3%	0.964
Fall	92.5%	96.0%	0.942

The confusion matrix (Fig. 8) shows that 504 of 528 non-fall samples and 304 of 321 fall samples were correctly classified.

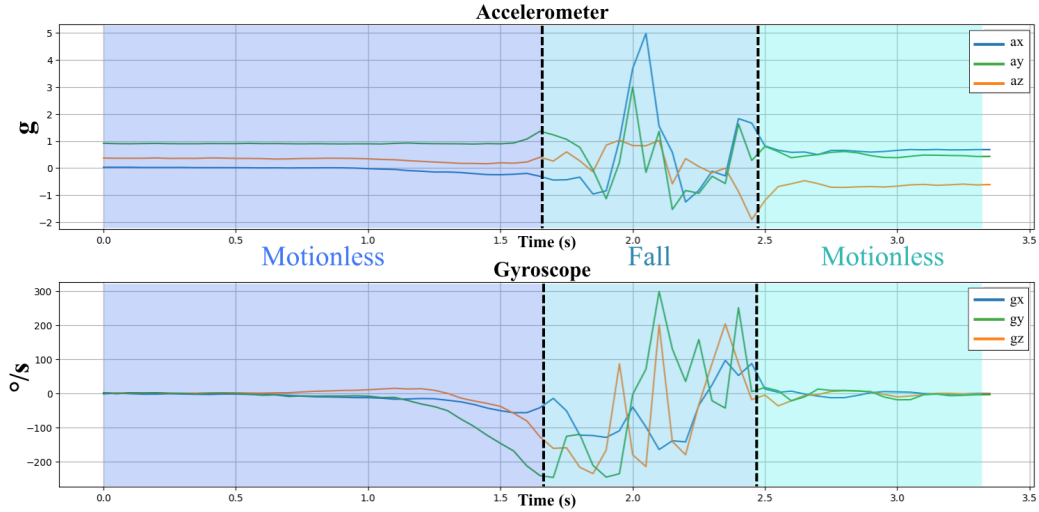


Fig. 6. Representative IMU signals during a experimental forward fall sample: accelerometer (top) and gyroscope (bottom) responses across three phases: motionless, fall, and post-fall motionless.



Fig. 7. 1D CNN architecture deployed on the M5Stack AtomS3R-Ext comprising an input layer (120 features), a reshape layer, two 1D convolution/pooling blocks, a flatten layer, a dense layer (32 neurons), and 2-class output layer.

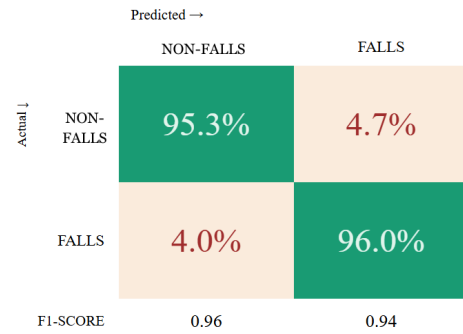


Fig. 8. Confusion matrix of the int8-quantized model evaluated on the validation set (849 samples).

t-distributed Stochastic Neighbor Embedding (t-SNE) was used to visualize the learned feature space and assess class separability in a 2D representation, confirming clear class separation, with misclassifications concentrated at the decision boundary, as shown in Fig. 9 Training and validation accuracy curves stabilized above 95% by epoch 20, indicating adequate convergence within 120 epochs, as shown in Fig. 10.

### B. Experimental Results – Fall Detection

Ten trials of each fall type were conducted with the device operating autonomously on the microcontroller, and the results are summarized in Table III.

Forward falls were detected in all ten trials. Lateral falls showed lower rates because wrist impact attenuation by the arm reduced peak acceleration below the Stage 2 threshold in three trials, a limitation inherent to wrist-worn placement.

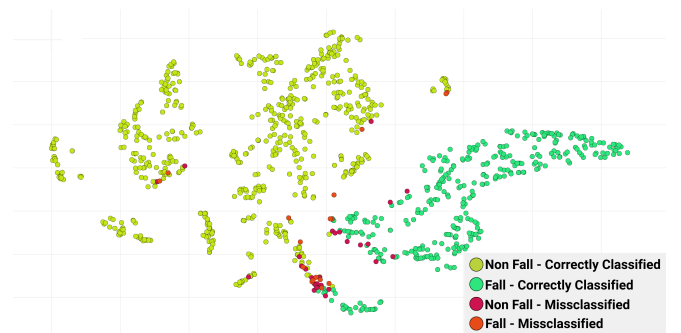


Fig. 9. t-SNE two-dimensional projection of the validation set showing class separability. Non-fall and fall samples are largely grouped in distinct clusters, with misclassified samples concentrated at the decision boundary.

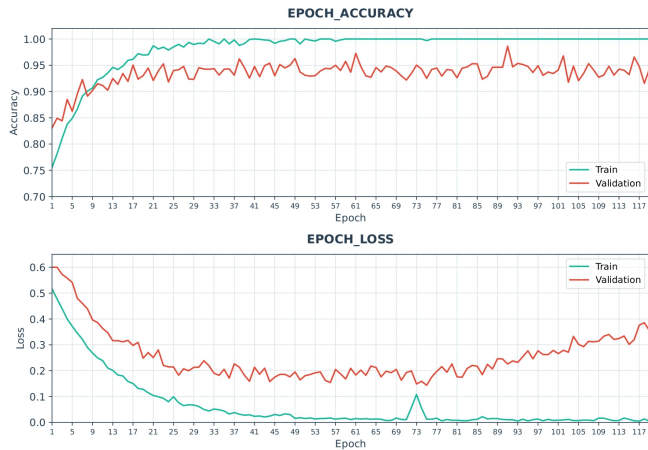


Fig. 10. Training and validation accuracy (top) and loss (bottom) curves over 120 epochs. Both metrics stabilize by epoch 20, with the divergence between training and validation loss indicating mild overfitting, expected given the limited dataset size.

TABLE III  
FALL DETECTION PERFORMANCE IN PRACTICAL ON-DEVICE TESTS

Fall Type	Trials	Detected	Detection Rate
Forward	10	10	100%
Backward	10	8	80%
Lateral	10	7	70%

One false positive was recorded during a rapid arm movement followed by immediate immobility on a surface, a motion that structurally mimics a fall signature and cannot be disambiguated from inertial data alone.

Compared to a baseline threshold-only approach (flag if acceleration  $> 3.80 g$ ), the full ML pipeline reduced false positives during high-intensity activities such as running and jumping, where peak accelerations exceeded the threshold but the temporal pattern did not match a fall event.

### C. Computational Performance

Table IV summarises the embedded computational metrics observed during continuous autonomous operation. The system operated continuously during the practical test sessions without resets or critical failures. The 0.7% CPU load confirms that inference does not compromise concurrent tasks (MQTT communication, geofencing, and sensor acquisition).

The int8-quantized model itself, as estimated by the Edge Impulse platform, requires only 6.7 kB of peak RAM and 48.6 kB of Flash memory, with an estimated inference time of 6 ms. This compact footprint represents a small fraction of the total resources available on the M5Stack AtomS3R-Ext, demonstrating the effectiveness of int8 quantization in enabling TinyML deployment on resource-constrained microcontrollers.

TABLE IV  
EMBEDDED COMPUTATIONAL PERFORMANCE – M5STACK ATOMS3R-EXT

Parameter	Value
Min. inference time	7 ms
Max. inference time	8 ms
Mean inference time	$\approx 7.03$ ms
Estimated ML CPU load	0.7%
Model peak RAM usage (int8)	6.7 kB
Model Flash usage (int8)	48.6 kB
Total SRAM (ESP32-S3)	512 kB
Typical free heap	$\approx 243$ kB
Firmware Flash usage	1,370,207 B (41%)
Static RAM usage	56,292 B (17.2%)

### D. Geofencing Performance

In addition to fall detection, the Wi-Fi-based geofencing module was evaluated under realistic conditions, with the mobile application running in the background. The total time between the wearer leaving the configured perimeter and the caregiver receiving the notification was 21.78 s. This interval includes a 15 s confirmation window intentionally configured in the firmware to validate the disconnection and avoid false alerts caused by momentary signal loss, with the remaining time corresponding to the latency of the communication stack (MQTT publication, Firebase Cloud Messaging push delivery, and smartphone processing). The reconnection time, measured from the wearer’s return to the perimeter until the status update in the application, was 4.45 s.

It is worth noting that both values are influenced by external factors, such as the quality of the local network infrastructure and the smartphone operating system’s handling of background notifications, and therefore do not represent intrinsic limitations of the proposed embedded system.

## IV. DISCUSSION

The experimental evaluation consisted of simulated fall tests performed by the authors under controlled conditions for the validation of the proposed IMU-based system, without the involvement of external participants or clinical trials. Although direct comparison is limited by differences in datasets and experimental protocols, the proposed system achieved 95.2% accuracy, compared with the 92.31% reported by Volochchuk and Bassai [11] using an Edge Impulse-based pipeline, while advancing to full autonomous embedded deployment. The 7 ms inference time and 0.7% CPU utilization demonstrate that 1D CNN inference is practical on a low-cost ESP32-S3 microcontroller, leaving ample headroom for other system tasks. The wrist placement, while preferred for user acceptance, introduces inherent challenges: lateral falls produce lower-amplitude IMU signatures due to natural arm bracing, and rapid arm gestures can occasionally mimic fall dynamics. These limitations are consistent with findings from Nunez et al. [8], who reported that wrist sensors exhibit higher classification ambiguity than chest-mounted alternatives. Adaptive

class thresholds or a complementary barometric pressure sensor could mitigate these issues in future work, since the altitude variation associated with a real fall would help distinguish it from isolated arm movements that do not involve a change in body height. The system also integrates a Wi-Fi-based geofencing module, which detects whether the device remains within the configured home network perimeter and notifies the caregiver in case of a boundary breach, providing an additional monitoring functionality for cognitively impaired individuals without requiring GPS hardware.

## V. CONCLUSION

This paper presented a wrist-worn smartband with fully embedded AI for fall detection targeting monitoring applications involving individuals with cognitive impairment. A 1D CNN quantized to int8 and deployed on an M5Stack AtomS3R-Ext achieved 95.2% accuracy and a ROC-AUC of 0.951, with a 7 ms mean inference time and 0.7% CPU load. Although the results are promising, the evaluation was limited to controlled simulations with a small number of volunteers. Future work includes expanding the dataset with more diverse subjects, refining lateral fall thresholds, and integrating GPS for outdoor geofencing. The source code and embedded libraries are available at <https://github.com/rodbcl/Edge-AI-Smartband>.

## ACKNOWLEDGMENT

The authors thank the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES), the National Council for Scientific and Technological Development (CNPq), the Araucária Foundation and the State Secretariat for Science, Technology and Higher Education of Paraná (SETI).

## REFERENCES

- [1] World Health Organization, "Falls," 2021. [Online]. Available: [www.who.int/news-room/fact-sheets/detail/falls](http://www.who.int/news-room/fact-sheets/detail/falls)
- [2] Brazilian Ministry of Health, "Relatório nacional sobre a demência," 2024. [Online]. Available: [www.gov.br/saude/pt-br/assuntos/noticias/2024/setembro/relatorio-nacional-sobre-a-demencia](http://www.gov.br/saude/pt-br/assuntos/noticias/2024/setembro/relatorio-nacional-sobre-a-demencia)
- [3] J. Bruhn et al., "Falls and fall risk in community-dwelling people with dementia: A systematic review," *BMC Geriatrics*, 2021.
- [4] R. Yu et al., "Falls in people with Alzheimer's disease and mild cognitive impairment: A systematic review and meta-analysis," *Ann. Geriatr. Med. Res.*, 2022.
- [5] J. M. Olmedo-Aguirre et al., "Remote healthcare for elderly people using wearables: A review," *Sensors*, 2022.
- [6] R. N. Ferreira, N. F. Ribeiro, and C. P. Santos, "Fall risk assessment using wearable sensors: A narrative review," *Sensors*, vol. 22, no. 3, 2022.
- [7] G. Marques et al., "Online fall detection using wrist devices," *Sensors*, 2023.
- [8] J. O. Nunez et al., "Analysis of waist and wrist positioning wearable machine-learning models to detect falls," *Electronics Letters*, 2024.
- [9] S. Jang et al., "An effective deep learning framework for fall detection," *J. Med. Internet Res.*, 2024.
- [10] F. Li et al., "A decade of progress in wearable sensors for fall detection," *Sensors*, vol. 25, no. 7, 2025.
- [11] A. V. L. Volochchuk and S. W. Bassai, "Development of a sensor for fall monitoring in the elderly," B.Sc. Thesis, UTFPR, 2023.
- [12] J. Lin et al., "MCUNet: Tiny deep learning on IoT devices," *arXiv*, 2020.
- [13] Y. Wu et al., "Monitoring wandering behaviors of dementia patients using BLE-based localization systems," *arXiv*, 2024.
- [14] L. C. Santos et al., "Identifying sensors-based parameters associated with fall risk in older adults," *BMC Geriatrics*, 2024.
- [15] Edge Impulse, "Edge AI Platform," 2026. [Online]. Available: <https://www.edgeimpulse.com/>
- [16] M5Stack, "AtomS3R-Ext," 2023. [Online]. Available: <https://docs.m5stack.com/en/core/AtomS3R>
- [17] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proc. IJCAI*, 2016.