

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA ELÉTRICA**

EDUARDO SPERAFICO SCHNEIDER PINTO

**RECONHECIMENTO DOS MOVIMENTOS HUMANOS UTILIZANDO
INTELIGÊNCIA COMPUTACIONAL**

**São Leopoldo
2022**

EDUARDO SPERAFICO SCHNEIDER PINTO

**RECONHECIMENTO DOS MOVIMENTOS HUMANOS UTILIZANDO
INTELIGÊNCIA COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica, pelo Curso de Engenharia Elétrica da Universidade do Vale do Rio dos Sinos (UNISINOS).

Orientador(a): Prof. Dr. João Olegário de Oliveira de Souza

São Leopoldo

RESUMO

Este trabalho possui como objetivo, o desenvolvimento de um algoritmo que se utilize de inteligência artificial para classificar as atividades diárias do corpo humano. Será utilizado um banco de dados já existente de sinais de acelerômetro, os dados deste acelerômetro após processados foram classificados por uma rede neural MLP. As atividades humanas diárias que foram foco do escopo deste trabalho são: andar, correr, ficar parado, sentado e a subida e descida de escadas.

Após a seleção do banco de dados, foram elaboradas as rotinas de reconhecimento das atividades humanas, incluindo etapas como pré-processamento, janelamento, cálculo de *features* e classificação. Foram também analisados diferentes valores para o janelamento e parâmetros de modo a encontrar uma melhor performance. Com relação aos classificadores, foi testado o classificador do tipo redes neurais artificiais.

Ao fim do trabalho, foi obtida uma acurácia média de 80,74% para as 6 classes. Foram encontrados problemas principalmente na identificação de movimentos de subida e descida de escadas, evidenciados pela grande melhora de acurácia média quando estes movimentos foram excluídos do treinamento e teste da rede neural. No geral, a acurácia média do classificador desenvolvido foi inferior a acurácia média de classificadores verificados em outros artigos acadêmicos onde a acurácia média alcançava patamares superiores a 90%.

Palavras-chave: classificadores; inteligência artificial; *machine learning*; redes neurais; atividades humanas; acelerômetro.

LISTA DE FIGURAS

Figura 1 - Movimento de caminhada	15
Figura 2 - Movimento de caminhada	16
Figura 3 - Atividade de subida de escada (A) e atividade de descida de escada (B)	16
Figura 4 - Ilustração de uma pessoa parada em pé	17
Figura 5 - Ilustração de um ser humano sentado em cadeira	17
Figura 6 - Diagrama de blocos do processo de classificação dos dados	18
Figura 7 - Neurônio e suas 4 partes.	19
Figura 8 - Diagrama de uma rede neural artificial com uma camada oculta	20
Figura 9 – Ilustração gráfica da ideia de um algoritmo KNN	21
Figura 10 - Etapas do Algoritmo KNN	22
Figura 11 – a) Distribuição dos dados entre duas <i>features</i> X1 e X2; b) projeção dos pontos das features sobre um novo eixo.....	24
Figura 12 - Exemplo de janelamento onde TL é o tempo de largura da janela, Tinc o espaço de tempo para a medida e Td o tempo que o sistema de identificação levaria para reconhecer o movimento.....	25
Figura 13 - – a) computador utilizado para realizar os experimentos. b) relógio eZ430-Chronos. c) Movimento de corrida para captação de movimentos d) Ação de ficar parado. e) movimento de caminhada. f) Ação de ficar sentado.....	32
Figura 14 - Classificação dos movimentos em tempo real comparando PC e App desenvolvido para iOS	34
Figura 15 - Eixos de captura do acelerômetro no experimento	37
Figura 16 - Sinal capturado do acelerômetro para movimento de caminhada	37
Figura 17 – Número de medidas por atividade.....	41
Figura 18 – Número de medidas por usuário	41
Figura 19 – Eixos X, Y e Z para o primeiro trecho da atividade 1	43
Figura 20 – Dados para Treinamento.....	44
Figura 21 – Dados para Teste.....	45
Figura 22 – Matriz de Confusão	49
Figura 23 – Matriz de confusão com a inclusão do LDA	51
Figura 24 – Matriz de Confusão para testes com janela	53
Figura 25 – Matriz de confusão para o classificador com normalização de dados....	55
Figura 26 – Matriz de confusão para testes com remoção de offset.....	56

Figura 27 – Matriz de Confusão com filtro implementado	59
Figura 28 – Matriz de confusão para classificador de 5 classes	61
Figura 29 – Matriz de confusão para teste com apenas 4 classes.....	63

LISTA DE TABELAS

Tabela 1 – Distribuição de medidas das atividades pelo <i>database</i>	36
Tabela 2 - Dados capturados por usuário	38
Tabela 3 – Acurácias médias por classe por atividade para o classificador do artigo	39
Tabela 4 - Acurácias médias por classe por atividade para o classificador do artigo utilizando apenas 5 classes.....	39
Tabela 5 – <i>Dataframe</i> com dados do <i>database</i>	40
Tabela 6 – Labels atribuídos as atividades	44
Tabela 7 – Camadas da Rede Neural	47
Tabela 8 – Acurácia média obtida por classe na primeira execução.....	49
Tabela 9 – Acurácia média por classe após implementação de redução com LDA ..	51
Tabela 10 – Testes com diferentes janelas e overlaps	52
Tabela 11 – Acurácia média por classe	53
Tabela 12 – Acurácia por classe com normalização de dados.....	54
Tabela 13 – Acurácia por classe para o teste de remoção do offset.....	56
Tabela 14 – Testes do classificador com filtro passa faixas.....	58
Tabela 15 – Acurácia por classe com filtro implementado	59
Tabela 16 – Acurácia do classificador em combinações de métodos	60
Tabela 17 – Labels para testes com 5 classes.....	60
Tabela 18 – Acurácias por classe com 5 classes.....	61
Tabela 19 – Labels para testes com 4 classes.....	62
Tabela 20 – Acurácia por classe para teste com 4 classes.....	62
Tabela 21 – Comparativo de acurácias por movimento com o classificador do artigo	64
Tabela 22 - Comparativo de acurácias por movimento com o classificador do artigo quando utilizadas 5 classes.....	65

LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
BT	<i>Bluetooth</i>
IA	Inteligência Artificial
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
K-NN	<i>K-Nearest Neighbors</i> , K-Vizinhos Próximos
MAV	Média Absoluta dos Valores
MCU	Microcontrolador
NBR	Normas Brasileiras de Regulação
OMS	Organização Mundial de Saúde
PWR	<i>Power</i> , Alimentação do Sistema
RMS	<i>Root Mean Square</i> , Valor Eficaz
SSC	<i>Slope Signal Changes</i> , Inversão do Sinal
WL	<i>Waveform Length</i> , Comprimento de forma de Onda
MLP	<i>Multi-Layer Perceptron</i> , Perceptron Multi-Camadas
LDA	<i>Linear Discriminant Analysis</i> , <i>Análise Linear Discriminante</i>

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Tema	11
1.2 Delimitação do tema	11
1.3 Problema	11
1.4 Objetivos	12
1.4.1 Objetivo geral	12
1.4.2 Objetivos específicos.....	12
1.5 Justificativa.....	12
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 Atividades Humanas Diárias	14
2.2 Classificação de Movimentos Humanos	18
2.2.1 Classificadores.....	18
2.2.1.1 Redes Neurais Artificiais	19
2.2.1.2 Algoritmo K-NN (<i>K Nearest Neighbor</i> , K Vizinhos Próximos em português) 21	
2.2.1.3 Análise Discriminante Linear e redução de dimensionalidade	23
2.2.2 Janelamento.....	24
2.2.3 <i>Features</i>	25
2.2.3.1 Média Absoluta dos valores	25
2.2.3.2 Valor Eficaz	26
2.2.3.3 Comprimento de forma de onda.....	27
2.2.3.4 Assimetria.....	27
2.2.3.5 Variância	27
2.2.3.6 Logaritmo do Valor eficaz.....	28
2.2.3.7 Curtose.....	28
2.2.4 Acurácia	28
2.3 Acelerômetros	29
2.3.1 Acelerômetros piezoelétricos	29
2.3.2 Acelerômetros piezorresistivos.....	30
2.3.3 Acelerômetros capacitivos.....	30
2.3.4 Acelerômetros Wireless.....	30
2.3 Linguagem Python	31

2.4 Trabalhos Relacionados	31
2.4.1 Classificação da Atividade Humana Utilizando um Único Acelerômetro fixado no Pulso	31
2.4.2 Reconhecimento de Atividades Humanas em Tempo Real Baseado em <i>Deep-Learning</i> e sua Implementação em um Smartphone	33
3 METODOLOGIA	35
3.1 Banco de dados	35
3.1.1 Resultados do estudo do database no artigo <i>Activity Recognition using Cell Phone Accelerometers</i>	38
3.2 Algoritmo desenvolvido para identificação	40
3.2.1 Estudo do <i>Database</i>	40
3.2.2 Identificação de Início e Fim de trechos	42
3.2.4 Janelamento	45
3.2.5 Cálculo de <i>Features</i>	46
3.2.6 Parametrização da Rede Neural utilizada para classificar os dados	46
3.2.7 Métricas de avaliação dos resultados	47
4. ANÁLISE DOS RESULTADOS	48
4.1 Resultados iniciais da Rede Neural	48
4.2 Redução com LDA	50
4.2 Ajustes em valores de janela e <i>overlap</i>	52
4.3 Normalização dos dados	53
4.4 Remoção do offset (nível DC do sinal)	55
4.5 Implementação de um filtro passa faixas	56
4.6 Testes do sistema com apenas 5 classes	60
4.7 Testes do sistema sem os movimentos de subida e descida de escadas ...	62
4.8 Comparação com outros trabalhos relacionados	63
5 CONSIDERAÇÕES FINAIS	66
5.1 Sugestões para Trabalhos Futuros	67
REFERÊNCIAS	68
APÊNDICE A – TRECHO DE CÓDIGO PARA IDENTIFICAÇÃO DE INÍCIO E FIM DE ATIVIDADE	71
APÊNDICE B – TRECHO DE CÓDIGO PARA AS FUNÇÕES DE CÁLCULOS DE FEATURES	72

ANEXO C – TRECHO DE CÓDIGO QUE CHAMA AS FUNÇÕES PARA CÁLCULO DE FEATURES73
ANEXO D - TRECHO DE CÓDIGO DA PARAMETRIZAÇÃO DA REDE NEURAL75

1 INTRODUÇÃO

O reconhecimento de atividades humanas vem se tornando um tópico bastante estudado em artigos e trabalhos acadêmicos. Trata-se, principalmente, da identificação de um movimento específico do corpo humano, como caminhar, sentar-se ou ficar parado em pé de acordo com os dados de entrada (ALEMAYOH, LEE e OKAMOTO, 2019). O crescente interesse se dá em parte pela facilidade de acesso a sensores que capturem movimentação, incluídos hoje em praticamente qualquer *smartphone*.

Junto com os sensores, também temos um interesse crescente nas áreas relacionadas a inteligência artificial, mais especificamente, os classificadores baseados em *machine learning*. Estes, possibilitam que um sistema computacional aprenda a classificar os dados proporcionados, possibilitando assim que haja uma classificação em tempo real de uma base de dados de grande proporção, algo trabalhoso caso fosse feito manualmente.

Este trabalho, visa aliar estas duas tecnologias, de modo a identificar atividades do cotidiano do ser humano. Serão verificadas atividades simples: caminhar, correr, sentar-se, ficar parado subir e descer escadas. Estas atividades, já capturadas por um sensor (acelerômetro) em um banco de dados já existente serão classificadas através de um classificador que se utilize de inteligência computacional.

1.1 Tema

A identificação das atividades humanas é atrativa para a várias áreas diferentes. Dentre estas áreas destacam-se a saúde, segurança, esportes e entretenimento. Por exemplo, na área de esportes é possível saber se uma pessoa está se exercitando, informações de consumo de energia pelo corpo, nível de intensidade das atividades poderia ser usados para disponibilizar orientação ao usuário. Na área da saúde no caso de pessoas idosas, onde quedas são um grande perigo, a classificação de atividades pode ser utilizada para a detecção e prevenção de quedas (CHERNBUMROONG, AITKINS E YU, 2011).

Como aplicações no ambiente hospitalar, os acelerômetros não substituem um eletroencefalograma, mas podem servir como dispositivos auxiliares na avaliação dos movimentos para alguns casos de epilepsia (GIACOMOSSI, 2011).

1.2 Delimitação do tema

Com o intuito de identificar as atividades humanas reconhecidas por sensores instalados no corpo, foi tomada a decisão de construir um algoritmo de identificação de atividades humanas se utilizando de aprendizado de máquina.

1.3 Problema

Monitoração contínua e o reconhecimento dos movimentos possibilitam uma grande oferta de serviços personalizados em diferentes áreas como esportes, segurança, cuidados médicos, rastreamento, residências inteligentes, interfaces para jogos etc. (GINEZ DA SILVA, 2013). O desenvolvimento de uma ferramenta que se utilize do poder da inteligência artificial para identificação e reconhecimento de movimentos se provaria muito útil para as aplicações utilizadas nas áreas mencionadas acima.

1.4 Objetivos

1.4.1 Objetivo geral

Este trabalho tem como objetivo o desenvolvimento de um algoritmo que se utilize da inteligência artificial para reconhecer os movimentos do corpo humano. Foi utilizado um banco de dados (já existente), com movimentos capturados, e após isto a classificação destes movimentos pelo algoritmo de aprendizado de máquina. Os movimentos a serem identificados primariamente serão os movimentos de caminhada, sentar-se e estar parado, subir e descer escada e correr.

1.4.2 Objetivos específicos

Dentre os objetivos específicos podemos destacar os seguintes:

- a) Selecionar o banco de dados de movimentos já capturados a serem utilizados.
- b) Elaborar as rotinas de reconhecimento de atividades humanas, contemplando as etapas de pré-processamentos dos sinais dos bancos de dados, janelamento, cálculo de *features* e classificação dos sinais.
- c) Definir os intervalos de tempo para captura dos movimentos (janelamento).
- d) Validar o sistema e analisar parâmetros com o objetivo de aumentar o desempenho da classificação dos sistemas.
- e) Comparar os resultados com a literatura existente

1.5 Justificativa

A identificação automática das atividades do corpo humano utilizando-se de sensores fixados no corpo ganhou grande visibilidade de pesquisa, pois permite uma identificação não intrusiva e que não obstrua as de atividades do corpo (CHERNBUMROONG, AITKINS E YU, 2011).

Uma ferramenta (algoritmo) que seja capaz de identificar os movimentos do corpo humano, poderia ser utilizada em qualquer uma das aplicações já citadas anteriormente neste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

De modo a melhor entender os pontos apresentados neste trabalho é inicialmente necessário conhecer alguns tópicos, conceitos e tecnologias necessárias para o desenvolvimento. Foram consultadas fontes bibliográficas e outros trabalhos e artigos similares onde algumas destas ideias foram desenvolvidas.

Dentre os tópicos apresentados, estão uma descrição de atividades humanas diárias, bem como uma pequena definição das atividades que serão analisadas neste trabalho. Depois disto haverá uma etapa de classificação das atividades humanas, onde estarão descritos os classificadores utilizados, o janelamento e as *features*. Então serão revistos tópicos sobre a linguagem Python utilizada neste trabalho, bem como sobre acelerômetros. Por fim, serão descritos alguns artigos sobre o mesmo assunto e seus resultados alcançados de modo a melhor entender o escopo deste trabalho e seu processo.

2.1 Atividades Humanas Diárias

Em uma visão simplista, o movimento pode ser definido como a mudança de posição, porém, a interpretação da movimentação humana como uma simples mudança de posição do corpo ou suas partes é incapaz de caracterizar as qualidades e elementos relacionados à movimentação (BROOKE e WHITING, 1973).

O estudo do movimento é uma área com uma certa complexidade envolvida e objeto de vários estudos teóricos. Existe, por exemplo uma separação entre movimentos inconscientes como tremores ou movimentos por reflexo e movimentos conscientes onde se assume que houve uma formulação de raciocínio para execução de certa ação.

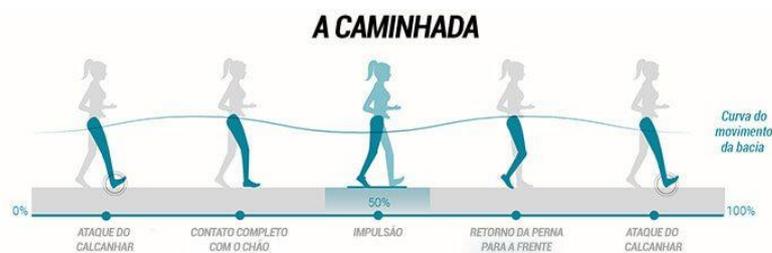
Também é importante ressaltar a importância do movimento com relação à saúde, segundo à Organização Mundial de Saúde (OMS), a inatividade física e o sedentarismo são um dos fatores de risco responsáveis por 6 % da mortalidade global (GINEZ, 2013).

Tendo em vista as definições de movimento já apresentadas, devemos definir algumas das atividades humanas diárias que serão abordadas neste trabalho. As atividades a serem definidas, são atividades corriqueiras que indivíduos executam diariamente, em especial, caminhar, correr, subir ou descer uma escada, sentar e ficar

parado em pé. É importante destacar o fato que a maioria das atividades humanas diárias consistem em uma sequência de repetição de um movimento único (ALEMAYOH et al., 2019).

Podemos então definir as atividades humanas diárias. Inicialmente, o ato de caminhar, pode ser definido simplesmente, como uma combinação do movimento de dar um passo, sucessivamente (ALEMAYOH et al., 2019), o ato do passo, por sua vez, é o movimento de mover sua perna para frente e posicionando primeiramente o calcanhar no chão e posteriormente abaixando o resto do pé, após isto, a outra perna repete o mesmo movimento. A figura 1 ilustra o movimento de caminhada.

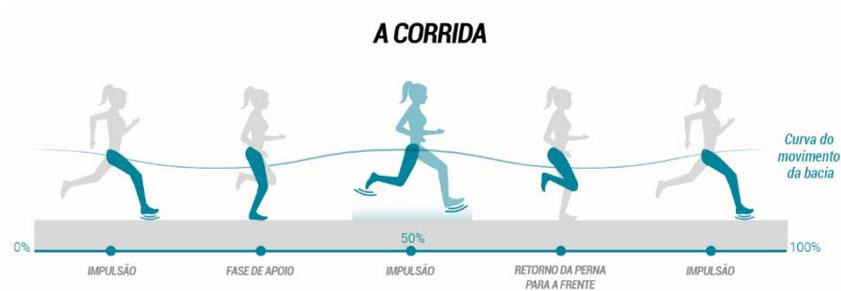
Figura 1 - Movimento de caminhada



Fonte: Decathlon

Com o movimento de caminhada definido, podemos partir para a definição do movimento de corrida. Este, possui certa similaridade com o movimento de caminhada, uma vez que consiste em movimentar as pernas para se locomover por alguma distância. Assim, move-se o pé de trás para a frente, o contato inicial com o chão se dando a partir do meio do calcanhar e meio do pé, após isto apoiando o pé completamente de modo a buscar impulsão para a próxima passada. Na próxima passada então, assim como na corrida o outro pé é trazido para a frente. Importante ressaltar que neste caso os movimentos são realizados mais rápida e bruscamente que na caminhada e que o movimento dos braços geralmente é oposto aos movimentos das pernas (quando a perna direita é trazida para frente o braço esquerdo também é movimentado para frente) de modo a manter o equilíbrio. A seguir está apresentada na figura 2 o movimento de corrida.

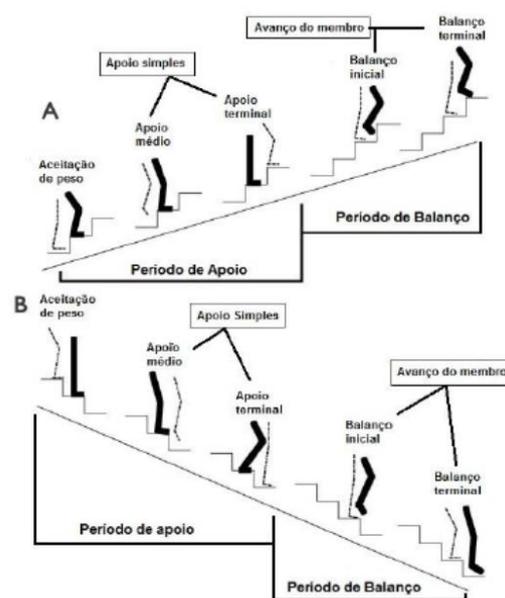
Figura 2 - Movimento de caminhada



Fonte: Decathlon

O movimento de subida e descida de escada também consiste na movimentação das pernas, porém com o intuito de mover-se para cima ou para baixo em uma escada. O movimento de subida, é constituído por três tarefas básicas: aceitação de peso, apoio simples e avanço do membro. A aceitação do peso se define pelo contato do pé com o degrau e resposta a carga. Já o apoio simples, é compreendido pelas fases de apoio médio, apoio terminal e pré-balanço (o apoio médio sendo o início do apoio da perna no degrau, apoio terminal a finalização deste processo de apoio e o pré-balanço a preparação para o movimento da perna). Por fim, o avanço do membro é a movimentação da perna de trás para o próximo degrau (LESSA e GOUVÊA, 2018). Os movimentos estão apresentados na figura 3.

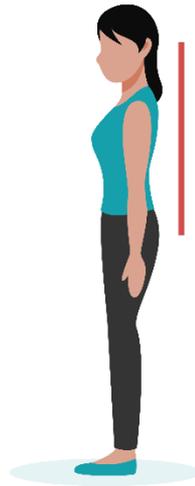
Figura 3 - Atividade de subida de escada (A) e atividade de descida de escada (B)



Fonte: LESSA e GOUVÊA, 2018

Outra das atividades humanas diárias citadas acima é o ato de ficar em pé parado. Este, por sua vez se caracteriza pelo indivíduo com os dois pés no chão sem movimento e em sua maioria do tempo posicionados paralelamente. A figura 4 ilustra esta atividade.

Figura 4 - Ilustração de uma pessoa parada em pé



Fonte: Elaborado pelo autor

Por fim existe a atividade humana diária de estar sentado que consiste em apoiar as nádegas em uma superfície horizontal, podendo ser uma cadeira ou também o chão. Os pés geralmente ficam apoiados no chão. A figura 5 apresenta uma ilustração do posicionamento de um ser humano sentado em uma cadeira.

Figura 5 - Ilustração de um ser humano sentado em cadeira



Fonte: Elaborado pelo autor

2.2 Classificação de Movimentos Humanos

De acordo com o descrito no objetivo deste estudo, o foco dele é o desenvolvimento de um algoritmo que classifique os sinais recebidos por um acelerômetro de modo a classificar a atividade humana executada através de inteligência artificial. Abaixo está apresentado o diagrama de blocos para a classificação na figura 6.

Figura 6 - Diagrama de blocos do processo de classificação dos dados



Fonte: Elaborado pelo autor

Na etapa do pré-processamento os dados obtidos foram condicionados para serem utilizados na classificação. Na etapa de janelamento, foram definidos os intervalos (janelas) de tempo das amostras coletadas que serão utilizadas. Após isto foi realizado o cálculo das *features* onde extraiu-se algumas características dos dados obtidos. Tendo de posse todos estes dados realizou-se a classificação dos dados de modo a definir qual a classe e por fim o pós-processamento aonde foram analisados e verificados os resultados da classificação.

Nesta seção serão apresentados alguns conceitos necessários para a classificação destes sinais, como os classificadores de IA, o janelamento do sinal, *features*, dentre outros.

2.2.1 Classificadores

Por início deste subcapítulo, precisamos definir o conceito de um classificador, um classificador é um algoritmo que mapeia os dados de entrada recebidos para uma categoria específica pré-definida se utilizando de inteligência computacional (GARG, 2018). No caso deste trabalho, por exemplo, teremos os sinais de saída de um acelerômetro e através das características (*features*) deste sinal, identificaremos a

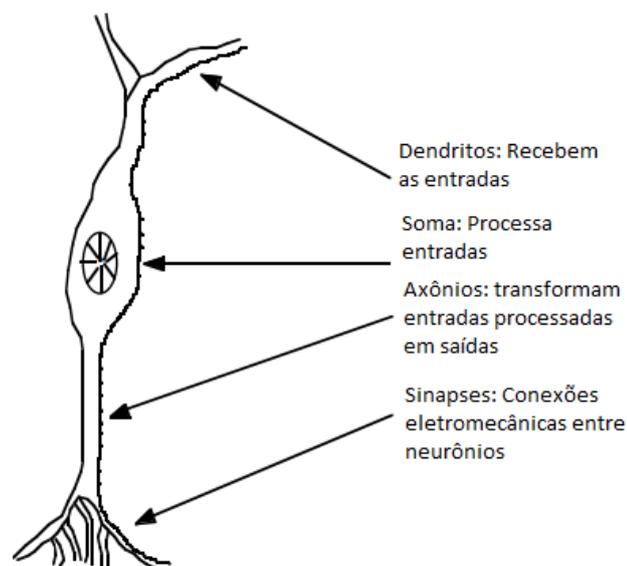
atividade humana que está sendo realizada dentre as categorias pré-definidas: caminhar, correr, subir ou descer escadas, estar parado em pé e estar sentado.

No geral, estes algoritmos precisam ser treinados, de modo a “aprender” como classificar os dados de entrada, assim se enviam os dados onde a classificação já é conhecida, de modo que o algoritmo “aprenda” como identificar a categoria de acordo com suas características. A seguir serão descritos alguns destes algoritmos de classificação.

2.2.1.1 Redes Neurais Artificiais

Um dos algoritmos de classificação mais utilizados, as redes neurais artificiais simulam, como seu nome diz, o funcionamento do sistema nervoso humano. Muitos especialistas em neurociência, buscam entender o funcionamento do cérebro, porém na engenharia, o interesse em redes neurais se dá por acreditarmos que estudando redes neurais poderemos construir melhores sistemas computacionais (ALPAYDIN, 2010). O cérebro humano, composto de neurônios é uma unidade processadora que recebe informação como entrada, a processa e emite uma saída. A figura 7 ilustra um neurônio real.

Figura 7 - Neurônio e suas 4 partes.

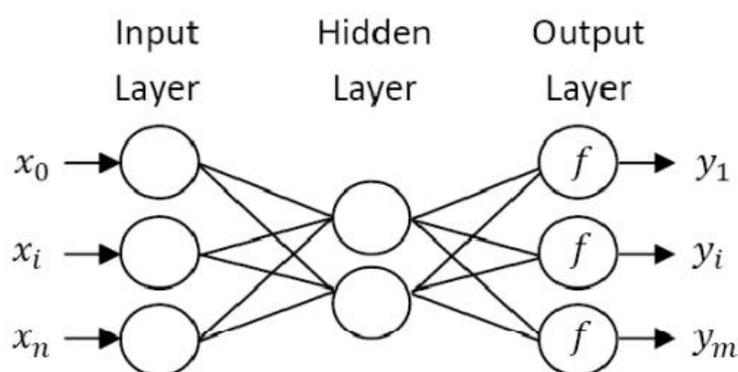


Fonte: Adaptado de (ANDERSON, 1996)

Como apresentado na imagem, os dados de entrada são recebidos nos dendritos, processados pela Soma, convertidos em saídas nos axônios e por fim tem sua saída de dados nas sinapses nervosas que são as conexões entre os neurônios.

Uma rede neural artificial, por sua vez, consiste em vários nós de entrada (dendritos), que são conectados a vários nós de saída (axônios) (CHERNBUMROONG, AITKINS E YU, 2011). Este conceito está ilustrado na figura 8.

Figura 8 - Diagrama de uma rede neural artificial com uma camada oculta



Fonte: Adaptado de (CHERNBUMROONG e AITKINS, 2011)

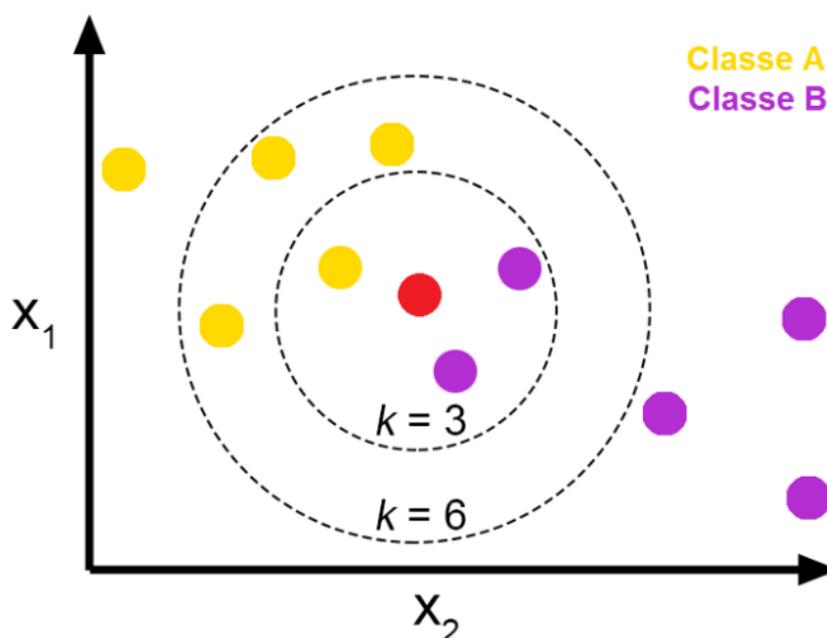
Pode-se visualizar com esta figura, o que acontece no caso do algoritmo de uma rede neural artificial, são inseridos os dados por diferentes nós de entrada na chamada camada de entrada (na figura apresentada como *input layer*, em inglês). Após isto os dados são transferidos para a camada oculta (apresentada na figura como *hidden layer*, em inglês). No caso apresentado na figura, existe apenas uma subcamada para o processamento de dados, porém podem ser especificadas mais subcamadas de acordo com o projetista, de modo a adequar a rede neural a suas necessidades. Nessa transmissão, a informação em um neurônio é transmitida para todos os neurônios da camada seguinte, porém há um “peso” para cada par. Esse peso é automaticamente definido pela rede conforme são feitos os treinamentos (BOELTER, 2021). Ao final os dados são transferidos para a camada de saída (na imagem exemplificada por *output layer*, em inglês).

A saída de uma rede neural, permite que as respostas sejam por sua vez, categorizadas entre uma das categorias já definidas, ou um resultado inconclusivo, onde a rede não consegue “encaixar” os dados de entrada em nenhuma categoria.

2.2.1.2 Algoritmo K-NN (*K Nearest Neighbor*, K Vizinhos Próximos em português)

O algoritmo KNN, é um dos algoritmos mais simples e com ampla usabilidade entre os algoritmos classificadores. Seu princípio de funcionamento consiste em identificar o quanto o dado de entrada é similar a outros dados de alguma determinada categoria, assim identificando a categoria do dado. O K-NN em si não compara o dado não classificado com todos os outros dados, mas sim a distância entre os dados para realizar a classificação (KHIMRAJ et al., 2020). No algoritmo é utilizada uma variável K (daí o nome, K vizinhos próximos) que define o número de vizinhos próximos que serão utilizados para comparar os dados. Tomando como exemplo que tenhamos duas classes, e um valor a ser classificado, o algoritmo selecionará os K valores mais similares ao dado de entrada e dentre estes classificará o dado de entrada com a classe de maior incidência dos valores já conhecidos. Na figura 9 temos uma representação gráfica deste processo com delimitações para $K=3$ e $K=6$.

Figura 9 – Ilustração gráfica da ideia de um algoritmo KNN



Fonte: Adaptado de (JOSÉ, 2018)

Nestes casos o sistema classificaria o dado de entrada (em vermelho) como sendo da classe A para $K=3$ e como sendo da classe B para $K=6$. Logo, podemos ver a importância deste parâmetro no algoritmo e sua precisão na classificação de movimentos.

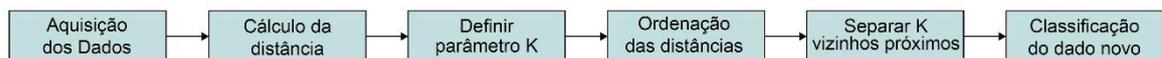
É um algoritmo de aprendizado “lento” (*lazy*) ou seja, não é necessário dados de treinamento para gerar seu modelo inicial, o que diminui o processo inicial, porém acarreta uma necessidade de análise posterior mais apurada de modo a validar seus resultados (NAVLANI, 2018).

De acordo com (KHIMRAJ et al., 2020), podemos definir as etapas que este algoritmo realiza para a classificação em 6 passos principais:

1. Aquisição dos dados não classificados.
2. Cálculo da distância entre os dados não classificados para todos os dados já classificados.
3. Definição do parâmetro K.
4. Ordenação das distâncias.
5. Separação dos k vizinhos próximos.
6. Classificação do dado novo para a classe com maior incidência entre os k vizinhos próximos.

Estas etapas estão apresentadas na figura 10.

Figura 10 - Etapas do Algoritmo KNN



Fonte: Elaborado pelo autor

A distância entre dos dados pode ser calculada de diversas maneiras como por exemplo a distância Euclidiana, de Manhattan, de Minkowski (KHIMRAJ et al., 2020). Dentre estas a mais usual de ser utilizada é a distância Euclidiana. A equação para a distância Euclidiana está apresentada abaixo em (1).

$$\begin{aligned}
 d(p, q) &= \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \\
 &= \sqrt{\sum_{i=1}^n (p_i - q_i)^2}
 \end{aligned} \tag{1}$$

Onde:

p_i é a coordenada do dado 1.

q_i é a coordenada do dado 2.

$d(p, q)$ é a distância entre estes dois dados.

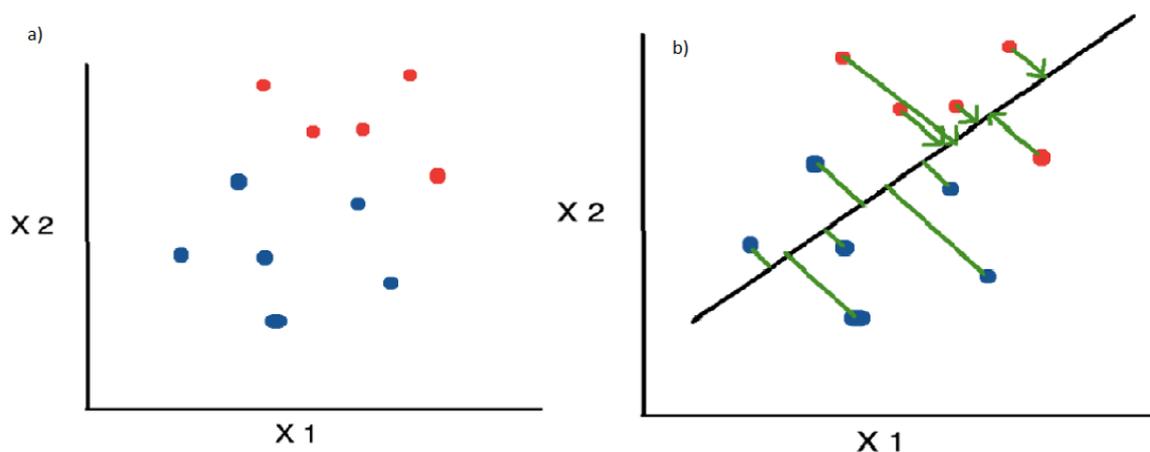
2.2.1.3 Análise Discriminante Linear e redução de dimensionalidade

Uma técnica bastante utilizada na classificação de dados e estatística, a análise discriminante linear (referida também pela sigla LDA, *Linear Discriminant Analysis*) apesar de simples, ela frequentemente produz resultados de classificação robustos e satisfatórios. A LDA pode ser utilizada tanto como um classificador de dados, quanto uma ferramenta para a redução da dimensionalidade e filtragem de dados. O LDA reduz a dimensionalidade dos dados de um *dataset* com o mínimo de perda de informação possível.

A LDA consiste em reduzir um conjunto de dados com n dimensões e reduzi-lo, para geralmente 2 dimensões, separando os dados em conjuntos de dados similares (BOELTER, 2021). O LDA utiliza as informações das *features* dos dados, de modo a criar um outro eixo e projetar os dados neste novo eixo diminuindo a variância e maximizando a distância entre os centros das diferentes classes.

Na figura 11, temos um exemplo da projeção de dados em LDA.

Figura 11 – a) Distribuição dos dados entre duas *features* X1 e X2; b) projeção dos pontos das *features* sobre um novo eixo



Fonte: Adaptado de (MAKLIN, 2019)

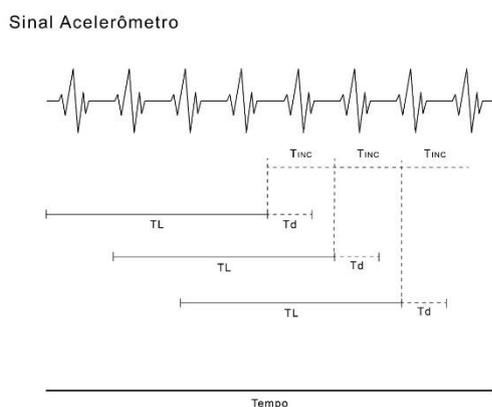
Analisando a figura 11 podemos melhor explicar a projeção dos dados. Em a) possuímos uma dispersão de dados com duas classes diferentes diferenciados pela cor azul e vermelha. Os dados estão posicionados de acordo com os valores de suas duas *features*, X1 e X2. Em b) vemos a projeção dos dados sobre um novo eixo linear entre os pontos, mantendo sem perdermos as características das *features* X1 e X2.

2.2.2 Janelamento

Com o sinal do movimento capturado, se faz necessário a definição do intervalo de tempo no qual as medidas serão analisadas, de modo a individualizar cada movimento. De modo a tornar o processo de classificação dos movimentos, é necessário dividir o sinal em parcelas menores.

O conceito de diminuir o sinal em parcelas menores que se deslocam pelo sinal, realizando classificações é chamado de janelamento. Utilizam-se dois parâmetros principais a serem definidos para o janelamento: a largura da janela, ou o tamanho da parcela que será tomado para análise e o deslocamento entre as “janelas”, que é o intervalo de tempo entre o início de uma janela e o início da próxima. Na figura 12 está apresentado um exemplo de janelamento.

Figura 12 - Exemplo de janelamento onde TL é o tempo de largura da janela, Tinc o espaço de tempo para a medida e Td o tempo que o sistema de identificação levaria para reconhecer o movimento.



Fonte: Elaborado pelo autor

Tendo definidos as janelas do sinal, será prosseguido com a classificação dos movimentos. Os dados brutos, porém, não são submetidos diretamente ao algoritmo de reconhecimento, mas sim alguns parâmetros (GINEZ, 2013) ou como também apresentados neste trabalho, algumas *features*.

2.2.3 Features

Como descrito no tópico acima na seção 2.2.2, o sinal dentro da janela não é submetido a uma análise ponto a ponto. Assim, definem-se alguns parâmetros, ou como citado no título deste capítulo, algumas *features*.

Abaixo estão apresentadas algumas das *features* encontradas em outras bibliografias consultadas para o desenvolvimento deste trabalho.

2.2.3.1 Média Absoluta dos valores

A primeira *feature* identificada é a média absoluta dos valores (*Mean Absolute Value, abreviada para MAV*), obtida através da soma dos valores de módulo divididos

pelo número de pontos (ABBASPOUR, et al., 2019). A equação para a média absoluta está apresentada em (2).

$$MAV = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (2)$$

Onde:

MAV é a Média absoluta dos valores.

N é o número total de valores.

x_i é o valor de x para o determinado valor de i .

2.2.3.2 Valor Eficaz

O valor eficaz, em inglês *Root Mean Square*, se obtém pela raiz quadrada da soma quadrática de todos os pontos dividida pelo número de pontos (ABBASPOUR et al., 2019). A equação é apresentada em (3).

$$RMS = \sqrt{\sum_{i=1}^N |x_i|^2} \quad (3)$$

Onde:

RMS é o valor eficaz.

$|x_i|$ é o modulo dos valores a ser tirado a média.

2.2.3.3 Comprimento de forma de onda

Em inglês descrito como *Waveform Length (WL)* é a soma das diferenças entre todos os pontos da onda e indica a complexidade do sinal (ABBASPOUR et al., 2019). A equação esta apresentada em (4).

$$WL = \sum_{i=1}^N (|x_i - x_{i-1}|) \quad (4)$$

Onde:

WL é o comprimento da forma de onda.

x_i são os valores para os quais será calculado o comprimento da forma de onda.

2.2.3.4 Assimetria

A assimetria pode ser calculada através da seguinte equação apresentada em (5).

$$Skew = \frac{3(MAV - Mediana)}{Desvio\ Padr\~{o}} \quad (5)$$

Onde:

MAV é a média absoluta dos valores.

$Mediana$ é a mediana dos valores.

$Desvio\ Padr\~{o}$ é o desvio padrão dos valores.

2.2.3.5 Variância

A equação para a variância esta apresentada em (6).

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{2} \quad (6)$$

Onde :

σ^2 é a variância.

x_i é o valor analisado.

\bar{x} é a média aritmética do conjunto.

n é o número de dados do conjunto.

2.2.3.6 Logaritmo do Valor eficaz

O logaritmo do valor eficaz, consiste em apenas calcular o logaritmo do valor eficaz (RMS, citado anteriormente no item 2.2.3.2).

2.2.3.7 Curtose

A equação da curtose está apresentada em (7).

$$Kur(x) = \frac{Q3 - Q1}{(2 * P90) - P10} \quad (7)$$

Onde:

$Q3$ é o primeiro quartil.

$Q1$ é o valor da posição central entre o primeiro valor e a mediana.

$P90$ é o percentil 90.

$P10$ é o percentil 10.

2.2.4 Acurácia

Uma das principais métricas que foram utilizadas para o desenvolvimento deste trabalho, a acurácia é a média global do acerto do modelo ao classificar as classes e pode ser calculada através da equação (8).

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \quad (8)$$

Onde:

VP é o número de verdadeiros positivos, classificação correta da classe positiva
VN é o número de verdadeiros negativos, quando o modelo classifica a classe como negativa corretamente para o exemplo.

FP é o número de falsos positivos, erro quando o modelo previu uma classe incorretamente

FN é o número de falsos negativos, quando o modelo prevê a classe como negativa, porém era positiva

2.3 Acelerômetros

Um acelerômetro é um identificador que estima as mudanças de velocidade, inclinação, revolução, vibração e impacto (KHIMRAJ et al., 2020). Estes acelerômetros podem ser encontrados em diversos tamanhos e diferentes tipos, destacando-se os piezoelétricos, piezorresistivos, capacitivos e os MEMs (*microelectromechanical systems*, em português, sistemas micro eletromecânicos).

2.3.1 Acelerômetros piezoelétricos

Estes acelerômetros são produzidos através de materiais piezoelétricos, ou seja, materiais que sobre o efeito de algum movimento ou pressão, geram um campo magnético como os cristais de quartzo. Os sensores piezoelétricos oferecem alta sensibilidade e usualmente apresentam baixo custo. Respondem a deformações menores do que 1 µm e são adequados para medição de esforços variáveis, tais como força, pressão e aceleração. (BALBINOT, 2019). O acelerômetro piezoelétrico apresentará uma carga elétrica proporcional ao movimento realizado pelo corpo.

O pequeno tamanho deste tipo de transdutor, e a possibilidade de fabricação de dispositivos para sensibilidade unidirecional o tornam atraente para muitas aplicações em especial o monitoramento de vibrações.

2.3.2 Acelerômetros piezorresistivos

Estes são acelerômetros que se utilizam do efeito piezorresistivo, que consiste na variação da resistência de semicondutores de acordo com o alongamento e pressão que estes estão sobre efeito. Essa família de acelerômetros é indicada para frequências baixas, por exemplo, inferiores a 1 Hz, e pode ser utilizada na caracterização de sistemas estáticos ao contrário dos acelerômetros piezoelétricos (BALBINOT, 2019).

2.3.3 Acelerômetros capacitivos

Estes acelerômetros são mais indicados para sistemas estáticos e medição de inclinação com aceleração constante. A utilização destes acelerômetros, principalmente com a fabricação baseada no silício apresenta diversas vantagens como: maior sensibilidade, redução de peso, redução nos custos de fabricação, maior número de aplicações e redução dos problemas derivados da temperatura e da umidade (BALBINOT, 2019).

2.3.4 Acelerômetros Wireless

Em uma situação como a abordada neste trabalho, é necessário que o acelerômetro não possua cabos o conectando a outros sistemas, de modo a facilitar a movimentação do indivíduo que está realizando os movimentos. Na atualidade, a tecnologia de integração de sensores com sistemas de comunicação e eletrônica digital tem permitido o desenvolvimento de sensores pequenos com condicionadores internos (BALBINOT, 2019).

Assim em um sistema de medidas que se utilize de um acelerômetro Wireless é composto pelos seguintes elementos:

- Sensor.
- Condicionador de sinais.
- Conversor Analógico/Digital.
- Unidade Microprocessada.
- Sistema de alimentação.

- Sistema de comunicação, geralmente por radiofrequência.
- Sistema de armazenamento de informações (geralmente algum tipo de memória).

2.3 Linguagem Python

Amplamente utilizada, o Python é uma linguagem de programação orientada a objetos de alto nível, poderosa e com muitas funcionalidades disponíveis, especialmente no campo de *Data Science* e *Machine Learning*. Pode ser considerada um *software* livre de dois modos, uma vez que não é necessário comprar uma licença para utilizá-la e incluí-la em qualquer aplicação sua, bem como também é uma linguagem *open source*, o que permite que seja livremente modificada e distribuída.

Uma das mais apreciadas características do *Python* é sua facilidade. Ler um código em *Python* quase dá a sensação de estarmos lendo em inglês (SWAROOP, 2021). Indo nesta mesma direção, também é uma linguagem fácil de aprender, focando assim na resolução do problema ao invés na linguagem em si.

Outra qualidade a ser destacada é que pode ser utilizada em variadas plataformas, como por exemplo Windows, Mac, Linux, entre outras.

2.4 Trabalhos Relacionados

Nesta seção, abordamos outros sistemas similares e já existentes em outros artigos ou no mercado. Através do estudo de sistemas que executem uma atividade similar se buscará um maior entendimento do campo de estudo deste trabalho e de soluções que possam ser utilizadas.

2.4.1 Classificação da Atividade Humana Utilizando um Único Acelerômetro fixado no Pulso

Neste trabalho foi proposta a utilização de um único acelerômetro fixado no pulso do usuário. O objetivo do trabalho foi identificar as *features* utilizadas em sistemas de classificação de atividades e depois comparar a eficiência de dois algoritmos classificadores, Redes neurais artificiais e Árvores de Decisão para a

separação de 5 tipos de movimentos diferentes capturados: sentar-se, caminhar, correr, deitar de face para cima e ficar parado.

O dado dos sensores foi capturado com um relógio eZ430-Chronos, um relógio de esporte que é construído sobre um microcontrolador MSP430 e possui um transmissor *wireless* da Texas Instruments. O relógio se comunicava com o computador através de um receptor *wireless* conectado por USB. Para o pré-processamento, foram desenvolvidas funções no MATLAB. Na figura 13 estão apresentados os equipamentos utilizados para o desenvolvimento do artigo, bem como algumas das ações executadas para a captação dos movimentos.

Figura 13 - – a) computador utilizado para realizar os experimentos. b) relógio eZ430-Chronos. c) Movimento de corrida para captação de movimentos d) Ação de ficar parado. e) movimento de caminhada. f) Ação de ficar sentado



Fonte: (CHERNBUMROONG, AITKINS E YU, 2011)

Neste experimento o participante utilizava o relógio na sua mão não dominante. Os dados foram capturados de 7 participantes em dias diferentes. A classificação foi feita utilizando o *software* Weka. Foram escolhidos os modelos com melhor performance de cada método de classificação. No geral a árvore de decisão performou um pouco melhor que a rede neural. Os resultados indicaram também que a atividade de estar sentado é frequentemente confundida com a atividade de estar deitado.

Ao final foi concluído que com apenas um acelerômetro no pulso se alcançou uma alta eficácia na identificação das atividades humanas. Os resultados sugerem

que utilizando apenas 13 *features* simples no domínio de tempo e frequência foi possível alcançar uma precisão de 94,13 % (CHERNBUMROONG, AITKINS E YU, 2011). Outra pontuação levantada no estudo é que apesar de apenas um acelerômetro ser possível a identificação de atividades, estas precisam ser simples e não podem envolver um grande grau de complexidade.

2.4.2 Reconhecimento de Atividades Humanas em Tempo Real Baseado em *Deep-Learning* e sua Implementação em um Smartphone

Neste artigo, foi desenvolvido um método onde os dados da movimentação humana foram coletados por um smartphone e estruturados de modo a compor uma imagem virtual. Também foi desenvolvido uma aplicação em iOS para gravar, transmitir e classificar os movimentos em tempo real.

O smartphone utilizado foi afixado na cintura dos participantes do estudo. Os sensores utilizados foram o acelerômetro de 3 eixos e o giroscópio de 3 eixos do smartphone. Através dos dados obtidos por estes sensores, foi gerada uma imagem virtual do movimento que posteriormente foi utilizada na identificação do movimento.

O aplicativo desenvolvido e instalado no smartphone tem três funções: gravar os dados em formato CSV, transmitir os dados para um computador para o reconhecimento instantâneo e reconhecimento das atividades humanas no próprio smartphone (ALEMAYOH, et al., 2019). As atividades definidas como escopo deste trabalho foram andar, pular, correr, andar de bicicleta, subir e descer escadas, se deitar e ficar parado.

Para a identificação dos movimentos, foi utilizada uma rede neural de tipo CNN, devido a sua popularidade entre aplicações de reconhecimento de imagem (ALEMAYOH, et al., 2019).

Para a aplicação em iOS foi utilizado o framework CoreML, de modo a integrar o *machine learning* com o *app*. Foi adaptada a biblioteca TensorFlow para este framework. Oito cobaias participaram do experimento, e os resultados indicaram uma precisão na identificação das atividades entre 99 % e 99,5 %. Está apresentado na figura 14 o movimento de caminhada e os resultados da classificação do computador e do app comparados.

Figura 14 - Classificação dos movimentos em tempo real comparando PC e App desenvolvido para iOS



Fonte: (ALEMAYOH, et al., 2019).

Foi concluído que a utilização de sinais no domínio de tempo e frequência tiveram bons resultados, mas o domínio do tempo se provou melhor em termos de representar o movimento.

3 METODOLOGIA

Neste capítulo são apresentadas algumas técnicas e tecnologias que foram utilizadas no desenvolvimento do trabalho. Entre as tecnologias apresentadas está o banco de dados com dados já adquiridos que foi utilizado para treinamento e teste do trabalho.

3.1 Banco de dados

Para o desenvolvimento deste trabalho, foi necessário um banco de dados, de modo a treinar e a validar os dados e classificadores utilizados. O banco de dados escolhido foi o banco desenvolvido pela *WISDM Labs* durante uma pesquisa em 2010. O banco foi utilizado em um artigo, que serviu como comparativo com o classificador desenvolvido neste trabalho, logo posteriormente será apresentado em um subcapítulo os resultados do estudo.

Este banco baseia-se na captação dos mesmos movimentos definidos previamente na fundamentação teórica deste trabalho, são estes movimentos: caminhar, correr, subir e descer escadas, ficar parado e se sentar. Os dados foram capturados através do acelerômetro de um telefone *Android*. Em todos os casos foram coletados dados de acelerômetros a cada 50 ms, totalizando 20 amostras por segundo (KWAPISZ, WEISS & MOORE, 2010). Os dados do acelerômetro foram capturados em 3 eixos X, Y e Z. Os valores nestes eixos representam a aceleração dada na unidade g que representa a aceleração gravitacional ($1\text{ g} = 9,81\text{ m/s}^2$). Ao total o banco de dados possui 1098207 medidas de acelerômetro distribuídas entre as 6 atividades. A tabela 1 indica a atividade, o número de medidas capturadas pelo acelerômetro e a proporção do *database* para cada atividade.

Tabela 1 – Distribuição de medidas das atividades pelo *database*

Atividade	Número de medidas	Proporção (%)
Caminhar	424400	38,6
Correr	342177	31,2
Sentar	122869	11,2
De pé	100427	9,1
Subir Escadas	59939	5,5
Descer Escadas	48395	4,4

Fonte: Elaborado pelo autor

O *database* também já está preparado para ser utilizado em algoritmos de classificação, uma vez que existe um arquivo onde seus dados já estão definidos em determinadas *features*. Os dados foram divididos em janelas de 10 segundos com 200 medidas (uma vez que, como descrito acima, o banco de dados possui 20 medidas por segundo).

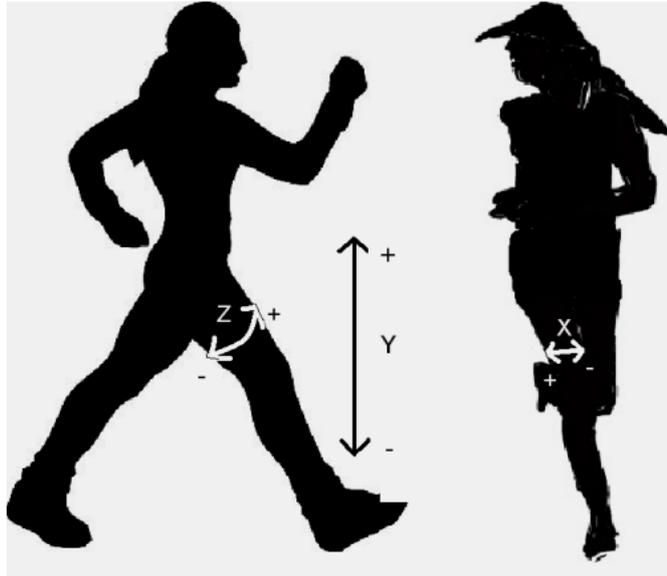
As *features* definidas foram 6 tipos base, que em alguns casos foram desdobrados para cada eixo, assim totalizando 43 *features* a serem utilizadas para classificar os dados. Os tipos de *features* estão apresentados abaixo, com o número utilizado entre parênteses.

- Média (3): Aceleração média em cada eixo.
- Desvio Padrão (3): Desvio padrão em cada eixo.
- Média Absoluta (3): Média absoluta entre o valor das 200 medidas para cada eixo.
- Aceleração resultante média (1): Obtida pela raiz quadrada da soma do quadrado dos valores de aceleração de cada eixo.
- Tempo entre Picos (3): Tempo entre os picos das formas de onda.
- Distribuição em categorias (30): São determinados um intervalo de valores para cada eixo (XYZ), e este intervalo é então dividido em 10 categorias iguais. Então os valores foram distribuídos entre as categorias.

Os dados foram capturados em 3 eixos onde o eixo Z captura os movimentos para frente da perna, o eixo Y os movimentos para cima e para baixo e o eixo X captura

os movimentos horizontais da perna. Os eixos de captura estão apresentados na figura 15.

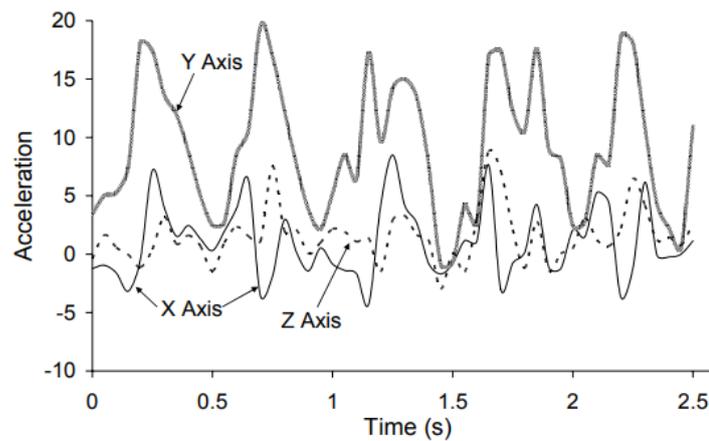
Figura 15 - Eixos de captura do acelerômetro no experimento



Fonte : Adaptado de (KWAPISZ, WEISS & MOORE, 2010).

Na figura 16 por sua vez está apresentado o sinal de caminhada obtido pelo acelerômetro e os eixos.

Figura 16 - Sinal capturado do acelerômetro para movimento de caminhada



Fonte: Adaptado de (KWAPISZ, WEISS & MOORE, 2010).

O banco de dado foi capturado com base nos resultados da medição de 29 usuários executando as atividades já mencionadas.

A tabela 2 a seguir mostra o número de exemplos capturados no banco de dados e com relação a atividade. Considera-se um exemplo, uma janela de 10 segundos (200 medidas do acelerômetro), com as *features* calculadas.

Tabela 2 - Dados capturados por usuário

Usuário	Caminhar (nº de exemplos)	Correr (nº de exemplos)	Subir escada (nº de exemplos)	Descer Escada (nº de exemplos)	Sentar-se (nº de exemplos)	Ficar parado (nº de exemplos)	Total
1	74	15	13	25	17	7	151
2	48	15	30	20	0	0	113
3	62	58	25	23	13	9	190
4	65	57	25	22	6	8	183
5	65	54	25	25	77	27	273
6	62	54	16	19	11	8	170
7	61	55	13	11	9	4	153
8	57	54	12	13	0	0	136
9	31	59	27	23	13	10	163
10	62	52	20	12	16	9	171
11	64	55	13	12	8	9	161
12	36	63	0	0	8	6	113
13	60	62	24	15	0	0	161
14	62	0	7	8	15	10	102
15	61	32	18	18	9	8	146
16	65	61	24	20	0	8	178
17	70	0	15	15	7	7	114
18	66	59	20	20	0	0	165
19	69	66	41	15	0	0	191
20	31	62	16	15	4	3	131
21	54	62	15	16	12	9	168
22	33	61	25	10	0	0	129
23	30	5	8	10	7	0	60
24	62	0	23	21	8	15	129
25	67	64	21	16	8	7	183
26	85	52	0	0	14	17	168
27	84	70	24	21	11	13	223
28	32	19	26	22	8	15	122
29	65	55	19	18	8	14	179
Soma	1683	1321	545	465	289	223	4526
%	37,2	29,2	12,0	10,2	6,4	5,0	100

Fonte: Adaptado de (KWAPISZ, WEISS & MOORE, 2010).

3.1.1 Resultados do estudo do database no artigo *Activity Recognition using Cell Phone Accelerometers*

Para compararmos os resultados deste artigo com os resultados do classificador desenvolvido, é necessário apresentá-los neste trabalho. O classificador utilizado foi

uma rede neural MLP, e janela de 10 segundos. Foi encontrada uma acurácia média de 91,7 %. As acurácias médias por classe estão apresentadas na tabela 3.

Tabela 3 – Acurácias médias por classe por atividade para o classificador do artigo

Movimento	Acurácia média (%)
Descer escadas	44,3
Correr	98,3
Sentado	95,0
De pé	91,9
Subir escadas	61,5
Caminhar	91,7

Fonte: Adaptado de (KWAPISZ, WEISS & MOORE, 2010).

Foi identificado que havia uma acurácia menor e problemas em especial com as atividades de subida e descida de escadas, assim tentou-se unir estas duas classes de modo a verificar a performance do classificador neste caso. A janela utilizada foi de 10 segundos e um classificador para este caso do tipo J48. A nova acurácia média foi de 93,7 %. As acurácias por movimento estão apresentadas na tabela 4.

Tabela 4 - Acurácias médias por classe por atividade para o classificador do artigo utilizando apenas 5 classes

Movimento	Acurácia média (%)
Caminhar	90,6
Correr	96,9
Sentado	96,5
De pé	93,7
Subir/Descer escadas	77,6

Fonte: Adaptado de (KWAPISZ, WEISS & MOORE, 2010).

3.2 Algoritmo desenvolvido para identificação

Nesta subseção estão descritas as etapas envolvidas no desenvolvimento do algoritmo de classificação dos movimentos, bem como as etapas necessárias para adequar o banco de dados e separação entre dados de treinamento e teste.

3.2.1 Estudo do *Database*

Primeiramente, foi necessário estudar o *database* escolhido para a realização deste trabalho. Logo foram conduzidos alguns experimentos com este *database* com o intuito de identificar algumas de suas características. De início atribuímos o *database* no python para um *dataframe*. A visualização do *dataframe* está apresentada abaixo na tabela 5:

Tabela 5 – *Dataframe* com dados do *database*

	user-id	activity	timestamp	x-axis	y-axis	z-axis
0	33	Jogging	49105962326000	-0.694638	12.680544	0.503953
1	33	Jogging	49106062271000	5.012288	11.264028	0.953424
2	33	Jogging	49106112167000	4.903325	10.882658	-0.081722
3	33	Jogging	49106222305000	-0.612916	18.496431	3.023717
4	33	Jogging	49106332290000	-1.184970	12.108489	7.205164
...
1098199	19	Sitting	131623331483000	9.000000	-1.570000	1.690000
1098200	19	Sitting	131623371431000	9.040000	-1.460000	1.730000
1098201	19	Sitting	131623411592000	9.080000	-1.380000	1.690000
1098202	19	Sitting	131623491487000	9.000000	-1.460000	1.730000
1098203	19	Sitting	131623531465000	8.880000	-1.330000	1.610000

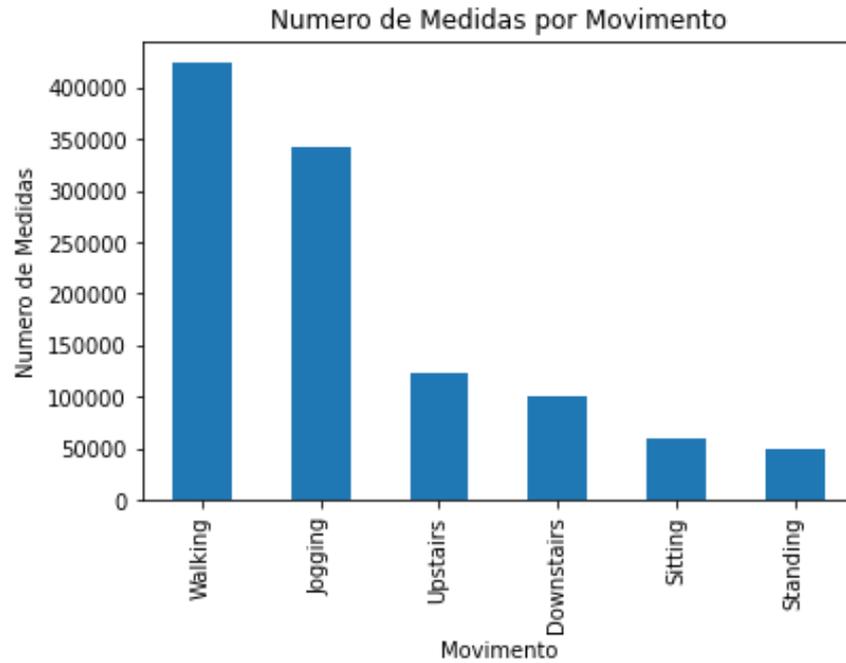
1098203 rows x 6 columns

Fonte: Elaborado pelo autor

Observando este *dataframe* podemos verificar que ele possui colunas, o user-id, atividade, marcação do tempo e os três eixos do acelerômetro. De posse destes dados, foi possível utilizar o *dataframe* para plotar um gráfico que identificasse os números de medidas do acelerômetro que possuímos por atividade, o gráfico está representado na Figura 17. Cada medida é o valor dos eixos X, Y e Z do acelerômetro,

junto com o movimento que estava sendo realizado no momento, ID do usuário e uma marcação do tempo no qual a medida foi capturada.

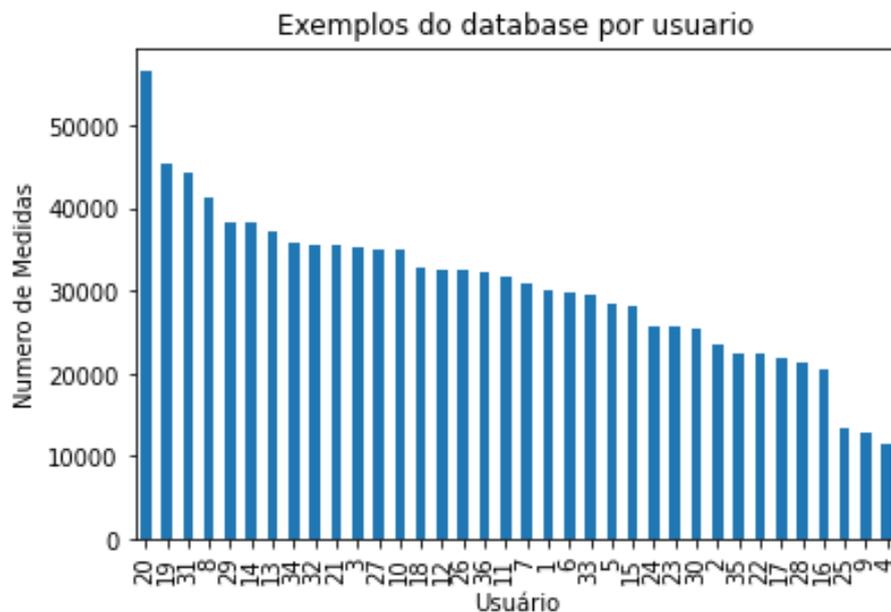
Figura 17 – Número de medidas por atividade



Fonte: Elaborado pelo autor

Também foram plotados os números de medidas de acordo com o usuário. O gráfico abaixo na figura 18 apresenta o número de medidas por usuário:

Figura 18 – Número de medidas por usuário



Fonte: Elaborado pelo autor

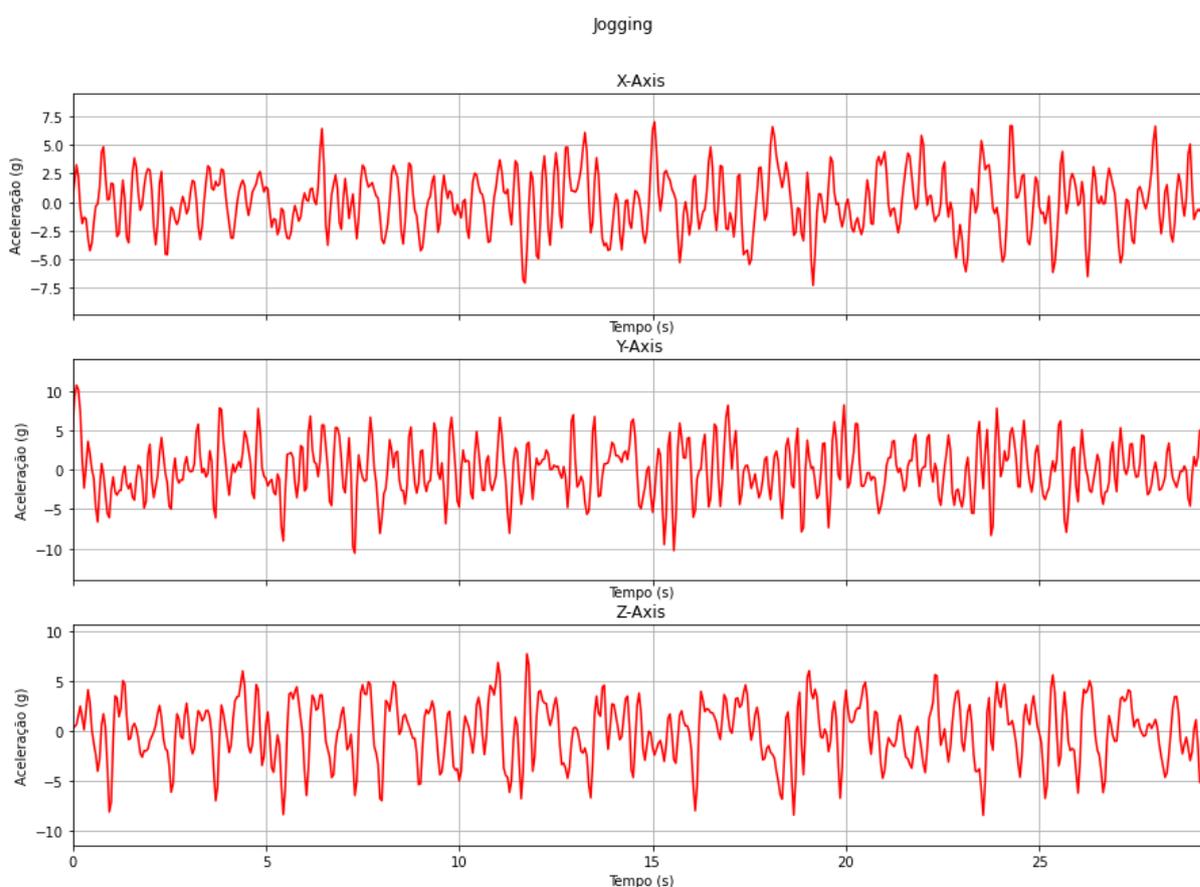
3.2.2 Identificação de Início e Fim de trechos

Um dos primeiros desafios enfrentados no desenvolvimento do programa para adequar os dados do *database* para serem utilizados no treinamento e teste do algoritmo de *machine learning* é a própria maneira que o *database* é apresentado. Os dados estão agrupados por usuário e não seguem uma sequência determinada, tanto no caso do usuário como na ordem que as atividades aparecem. Deste modo a saída encontrada, foi identificar em listas os inícios e fim de cada trecho de acordo com as atividades.

Assim, foi desenvolvida uma lógica em que o *dataframe* inteiro era percorrido de modo a identificar inícios e fim de atividade. A lógica de identificação para o início consistia em verificar se a atividade anterior era diferente da atividade atual, neste caso tinha-se um início. Do mesmo modo, caso o próximo valor fosse diferente do valor atual possuíamos um fim de atividade. Isto foi realizado para cada uma das atividades, assim por fim obtivemos 6 pares de lista, cada uma identificando o início e o fim e os índices destes pares de lista sendo correspondentes (`inicio_act_1[1]`, por exemplo, é o início do trecho para a atividade número 1 e este termina em `fim_act_1[1]`). O trecho de código para a identificação dos inícios e fim das atividades está contido no anexo A.

Com destes dados é possível plotar o sinal dos eixos X, Y e Z de um determinado trecho por exemplo, como demonstrado na figura 19 onde foi plotado o segundo trecho da atividade 1 (corrida). O tempo é dado em segundos, enquanto a aceleração é representada pela unidade g, onde 1 g é equivalente a aceleração da gravidade, ou seja $9,81 \text{ m/s}^2$.

Figura 19 – Eixos X, Y e Z para o primeiro trecho da atividade 1



Fonte: Elaborado pelo autor

3.2.3 Separação dos dados entre teste e treinamento

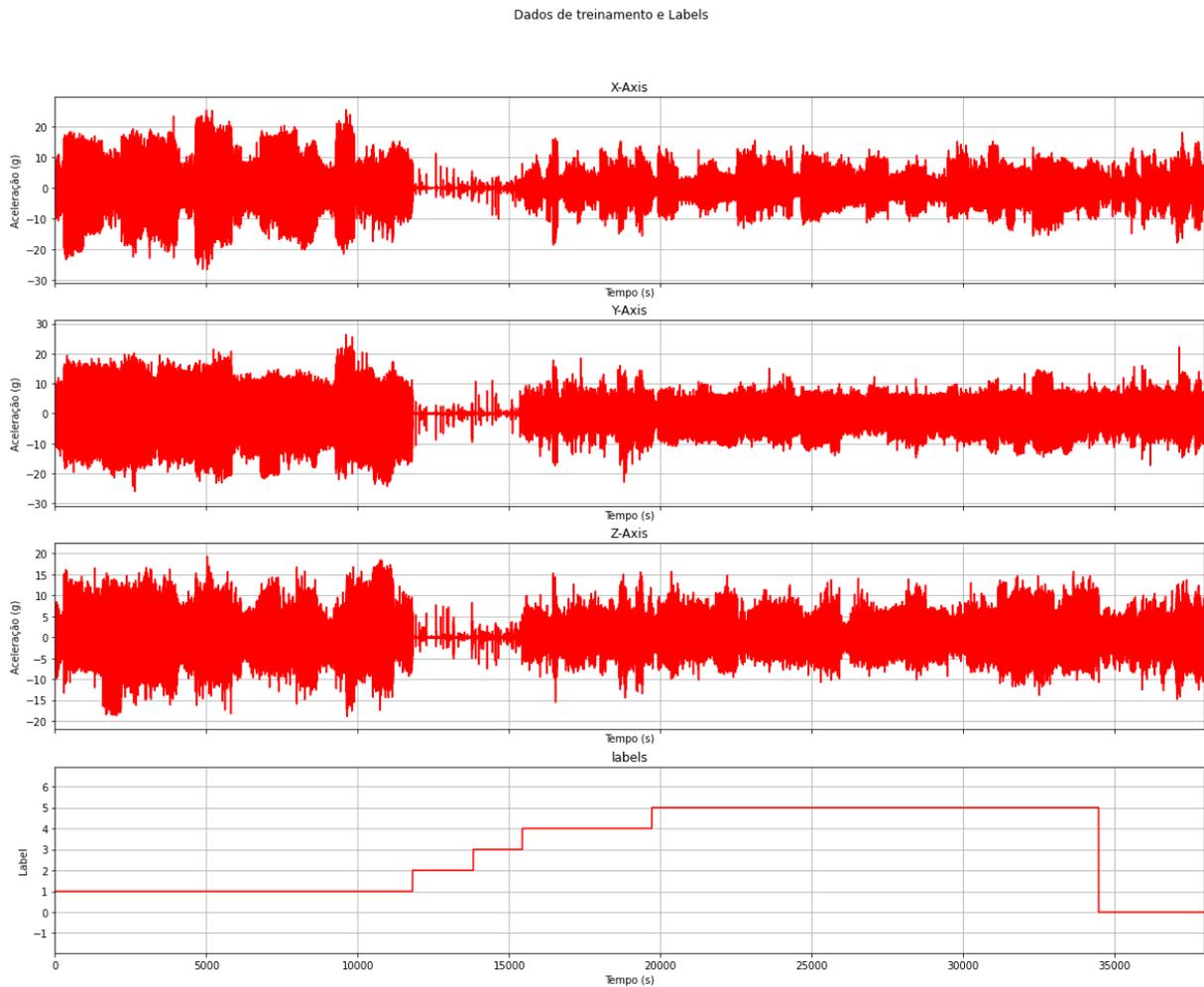
Outra etapa importante de nosso algoritmo era a separação dos dados entre dados de treinamento do algoritmo classificador e dados utilizados posteriormente para o teste. Foi utilizada uma proporção de 70 % do *database* utilizado para treinamento do algoritmo enquanto os 30 % restantes foram destinados ao teste do algoritmo. De modo a dividir, foi implementada uma lógica que através dos valores contidos nos trechos, se identificava o número total de *records* para cada atividade e se identificava qual valor correspondia a 70 % deste total. Após isto, foi utilizada uma lógica para identificar em qual índice dos valores conseguiríamos atingir este valor de 70 % do total. Abaixo temos a figura 20 ilustrando a parcela dividida para teste, os três eixos do acelerômetro e a atividade equivalente. O *label* atribuído às atividades no gráfico se dá de acordo com a tabela 6.

Tabela 6 – Labels atribuídos as atividades

Atividade	Valor
Correr	1
Sentar	2
De pé	3
Subir escada	4
Caminhar	5
Descer escada	0

Fonte: Elaborado pelo autor

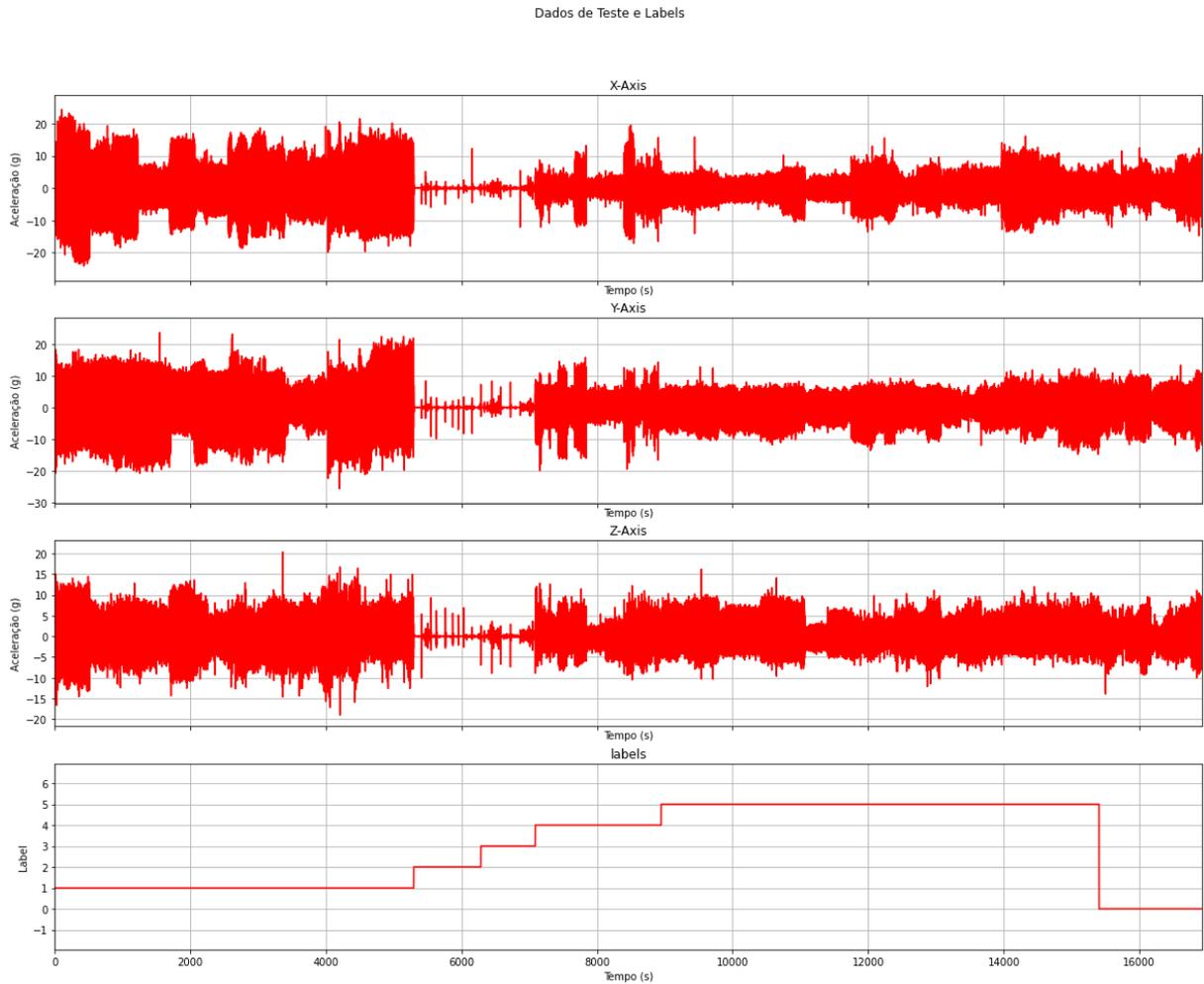
Figura 20 – Dados para Treinamento



Fonte: Elaborado pelo autor

Do mesmo modo, na figura 21 abaixo está apresentada a parcela separada para teste do sistema e seu *label* equivalente:

Figura 21 – Dados para Teste



Fonte: Elaborado pelo autor

3.2.4 Janelamento

Como descrito na fundamentação teórica, utilizou-se uma técnica de janelamento neste trabalho. Esta técnica consiste em definir janelas de tempo em que o sinal será analisado e as *features* calculadas. Também se emprega um *overlap* nestas janelas de modo que a próxima janela cubra parte da janela anterior.

No algoritmo deste trabalho foi inicialmente utilizada uma janela de 10 segundos com um *overlap* de 5 segundo.

3.2.5 Cálculo de *Features*

Após termos as janelas definidas, foi necessário calcular as *features* que serão utilizadas na classificação. Para isto foram utilizadas de funções prontas, incluídas na biblioteca *numpy* do Python, uma biblioteca focada em cálculos e funções matemáticas. Foram utilizadas 7 *features* diferentes, sendo cada uma destas 7 *features* calculadas para cada um dos três eixos do acelerômetro, assim totalizando 21 *features*.

As *features* utilizadas foram: a média absoluta dos valores, o valor eficaz (RMS), o valor eficaz logarítmico, o comprimento de onda, curtose, a assimetria e a variância do sinal. No anexo B deste relatório está demonstrado o trecho das funções definidas para o cálculo das *features*, enquanto o anexo C demonstra o código que realiza a chamada destas funções para calcularem o valor.

3.2.6 Parametrização da Rede Neural utilizada para classificar os dados

Com o *database* de movimentos capturados, dividido entre etapa de treinamento e teste e *features* calculadas foi iniciado a etapa de parametrização da rede neural que será utilizada para a classificação dos dados.

Foi utilizada a API Keras, para redes neurais que nos permite trabalhar com bibliotecas de *deep learning* como a TensorFlow (utilizada neste trabalho). O modelo escolhido foi o modelo Sequencial utilizando 6 camadas, sendo uma destas a camada de entrada.

A primeira camada foi ajustada para um formato de entrada de 21 colunas uma vez que nosso dado possui 21 *features*. A segunda e quarta camadas foram parametrizadas como camadas do tipo “*Dense*” para conter 100 neurons e utilizando a função de ativação *Rectified Linear Unit* (“*relu*”). A terceira camada é uma camada do tipo dropout. A quinta camada é uma camada do tipo “*Flatten*”. Por fim, a sexta e última camada é uma camada do tipo “*Dense*” parametrizada para 6 classes e função de ativação “*softmax*”.

A tabela 7 abaixo apresenta as camadas incluídas e seus parâmetros:

Tabela 7 – Camadas da Rede Neural

Camada	Tipo de Camada	Função de Ativação	Número de neurons
1	Camada de Entrada	N/A	N/A
2	Dense	Relu	100
3	Dropout	N/A	N/A
4	Dense	Relu	100
5	Flatten	N/A	N/A
6	Dense	softmax	N/A

Fonte: Elaborado pelo autor

O anexo D deste relatório demonstra a parametrização da rede neural nas linhas de código escritas em Python.

3.2.7 Métricas de avaliação dos resultados

Com o *database* adequado para o classificador e a rede neural MLP parametrizada, o próximo passo de desenvolvimento foi a avaliação de resultados. Foram utilizadas, principalmente duas métricas para a avaliação dos resultados do classificador desenvolvido: a acurácia e matrizes de confusão.

A acurácia, tem sua equação descrita na seção 2.2.4 e consiste em avaliar a proporção de acertos do classificador. Uma vez que a acurácia do classificador pode variar se o classificador for treinado e testado em diferentes vezes, cada teste do classificador foi executado 10 vezes e foi extraída uma acurácia média destas execuções. Também foram verificadas as acurácias médias por movimento.

As matrizes de confusão, são matrizes que apresentam, por tipo de classe, a proporção para qual o classificador classificou os exemplos para cada classe. Estas matrizes foram utilizadas de modo a avaliar quais movimentos estavam sendo confundidos assim podendo desenvolver conclusões e buscar soluções para problemas encontrados.

4. ANÁLISE DOS RESULTADOS

Após realizado o desenvolvimento do algoritmo para adequação do banco de dados de movimentos humanos para treinamento, foi necessário testar os resultados de modo a verificar se a solução de utilizar uma rede neural para classificar estava adequada para esta aplicação. Este capítulo apresentará os resultados obtidos, bem como o processo de aprimoramento do algoritmo de modo a melhorar a eficácia do classificador em reconhecer os tipos de movimentos envolvidos. O principal indicador de eficácia utilizado para avaliar o classificador será a acurácia média total, com relação a quantidade de exemplos classificados corretamente com relação ao total de exemplos dos dados de teste.

Todos os classificadores foram rodados pelo menos 10 vezes (ciclo treinamento e teste), de modo a extrair a eficácia média dentre estes ciclos, uma vez que as diferentes execuções de treinamento e teste de um mesmo *dataset* podem resultar em variações na acurácia.

4.1 Resultados iniciais da Rede Neural

Foi realizada a primeira execução do classificador desenvolvido de modo a verificarmos os resultados iniciais obtidos e a partir daí foram tomadas diferentes abordagens de modo a melhorar a acurácia final do classificador. Para a primeira testagem, foi escolhido o valor da janela correspondente a 10 segundos, enquanto o valor do *overlap* corresponde a 5 segundos. Estes valores foram determinados de modo a serem os mesmos valores de outros artigos estudados de modo a ser possível uma comparação entre os classificadores desenvolvidos.

O tipo de classificador utilizado foi uma rede neural MLP (Perceptron Multi-Camadas), com a com a parametrização conforme descrito no apêndice D.

Após a execução do algoritmo 10 vezes de modo a obtermos a média das acurácias, foi obtida uma acurácia média de 77,03 %, o que não pode ser considerada uma boa acurácia e indicava que o classificador necessitava de aprimoramento. A tabela 8 abaixo, demonstra cada classe (o movimento associado a classe está indicado entre parênteses) e sua acurácia média obtida.

Tabela 8 – Acurácia média obtida por classe na primeira execução

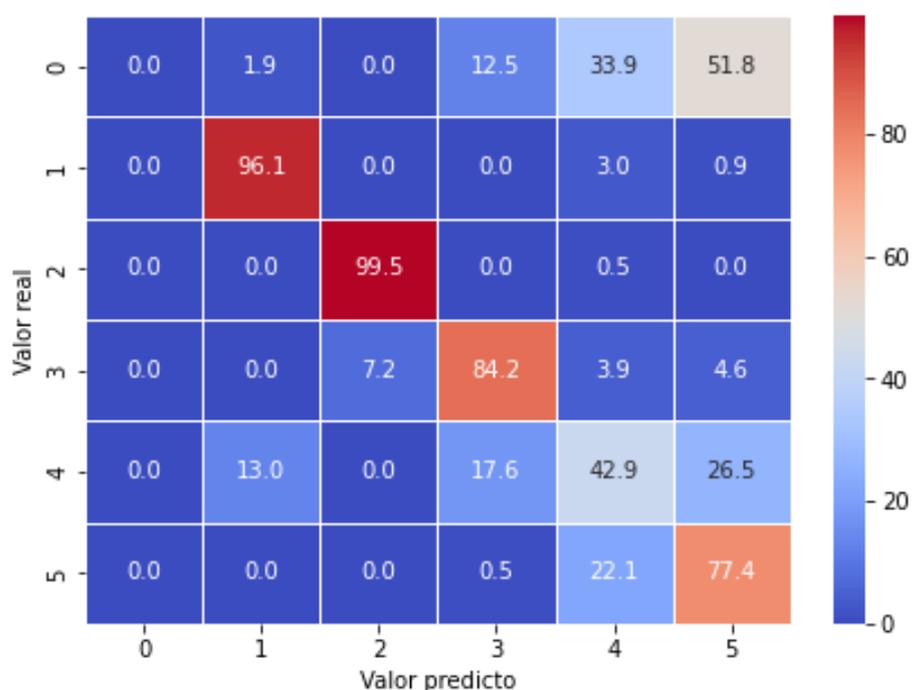
Classe	Acurácia (%)
Classe 0 (Descer escadas)	0,00
Classe 1 (Correr)	96,82
Classe 2 (Sentado)	98,63
Classe 3 (De pé)	73,09
Classe 4 (Subir escadas)	44,78
Classe 5 (Caminhar)	82,06

Fonte: Elaborado pelo autor

A uma análise inicial, podemos verificar que o classificador tem uma boa eficácia em reconhecer o movimento de estar sentado, bem como de correr. Nota-se também que o classificador não consegue identificar o movimento de descer escadas.

Com o intuito de ter uma melhor noção dos resultados obtidos, foi também gerada a matriz de confusão dos resultados do classificador, de modo a verificar como os dados estão sendo classificados. A figura 22 demonstra a matriz de confusão para uma das 10 execuções deste primeiro teste realizado.

Figura 22 – Matriz de Confusão



Fonte: Elaborado pelo autor

Ao verificarmos a matriz de confusão, é possível tirar algumas conclusões sobre o teste do algoritmo. Com relação a classe 0 (descer escadas), esta classe não está sendo identificada, devido ao algoritmo confundi-la principalmente com o movimento 4 (subir escadas) e o movimento 5 (caminhar). Pode-se identificar uma lógica, uma vez que o movimento de subir escadas e descer é similar. Este mesmo problema também foi identificado em outros artigos bibliográficos que utilizaram este mesmo *database* e estão contidos na bibliografia deste trabalho (KWAPISZ, WEISS & MOORE, 2010).

Possuindo estes primeiros resultados iniciais como base, foi dado início a testagem de diferentes métodos para o aprimoramento da capacidade de classificação, dentre eles, a testagem de diferentes valores para janela e *overlap*, uma redução dos dados através de LDA, a implementação de um filtro passa faixas nos dados, a remoção do offset do sinal e a normalização dos dados. Estas iniciativas, bem como seus resultados, serão descritas nos próximos subcapítulos deste trabalho.

4.2 Redução com LDA

A primeira iniciativa realizada de modo a tentar aprimorar o desempenho do classificador foi a realizar uma redução de dimensionalidade com LDA (*Linear Discriminant Analysis* ou Análise Discriminante Linear) nos dados utilizados para teste e treinamento. Foram utilizadas uma janela de 10 segundos e um *overlap* de 5 segundos. O tipo de classificador é uma rede neural MLP conforme configurado no apêndice D, apenas com uma modificação no formato de entrada uma vez que o LDA requer uma mudança.

Após a inclusão da redução com LDA, o classificador foi rodado 10 vezes e foi verificado que a sua Acurácia média piorou, resultando em um valor de 73,7 %. A tabela 9 abaixo apresenta cada classe e sua acurácia obtida.

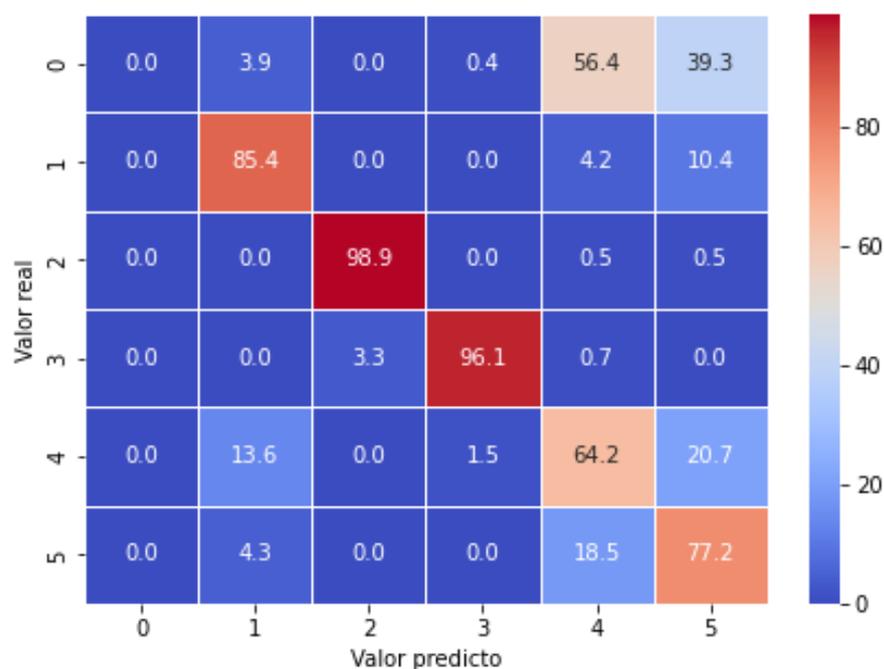
Tabela 9 – Acurácia média por classe após implementação de redução com LDA

Classe	Acurácia (%)
Classe 0 (Descer escadas)	0,00
Classe 1 (Correr)	86,16
Classe 2 (Sentado)	98,36
Classe 3 (De pé)	93,55
Classe 4 (Subir escadas)	61,33
Classe 5 (Caminhar)	75,66

Fonte: Elaborado pelo autor

Foi possível perceber que houve um aumento na acurácia para as classes 3 e 4, enquanto houve um decréscimo de acurácia para as demais classes. A classe 0 continua não sendo identificada neste caso. A figura 23 demonstra a matriz de confusão para uma das 10 execuções.

Figura 23 – Matriz de confusão com a inclusão do LDA



Fonte: Elaborado pelo autor

4.2 Ajustes em valores de janela e *overlap*

Outro passo necessário era ajustar o tamanho das janelas e *overlap* de modo a verificar os efeitos no classificador. Foram testados alguns valores experimentalmente de modo a determinar qual obteria a melhor acurácia do sistema.

Os testes foram realizados sem a redução LDA inclusa no código desenvolvido. As janelas foram variadas de 2 segundos (20 medidas) a 50 segundos (1000 medidas). Os *overlaps* foram variados de 1 segundo (20 medidas) a 25 segundos (500 medidas). O tipo de classificador utilizado foi uma rede neural MLP configurada conforme o subcapítulo 3.2.6. Todos os testes foram executados 10 vezes, de modo a verificar sua acurácia total média.

A tabela 10 ilustra os valores testados de janela e *overlap* e a acurácia média obtida após as 10 execuções.

Tabela 10 – Testes com diferentes janelas e *overlaps*

Janela (s)	<i>Overlap</i> (s)	Acurácia (%)
2	1	72,15
10	1	77,04
10	2,5	75,74
10	5	77,03
25	1	77,53
25	2,5	77,89
25	5	79,39
50	5	83,63
50	25	84,38

Fonte: Elaborado pelo autor

Através da inspeção da tabela, podemos verificar que a combinação de janela de 50 segundos com *overlap* de 25 segundos apresentou a melhor acurácia média (84,38%). Foram testados valores maiores de janela e *overlap*, porém os resultados não foram melhores. A tabela 11 abaixo apresenta a acurácia média por classe com os parâmetros ajustados para a janela de 50 segundos e *overlap* de 25 segundos.

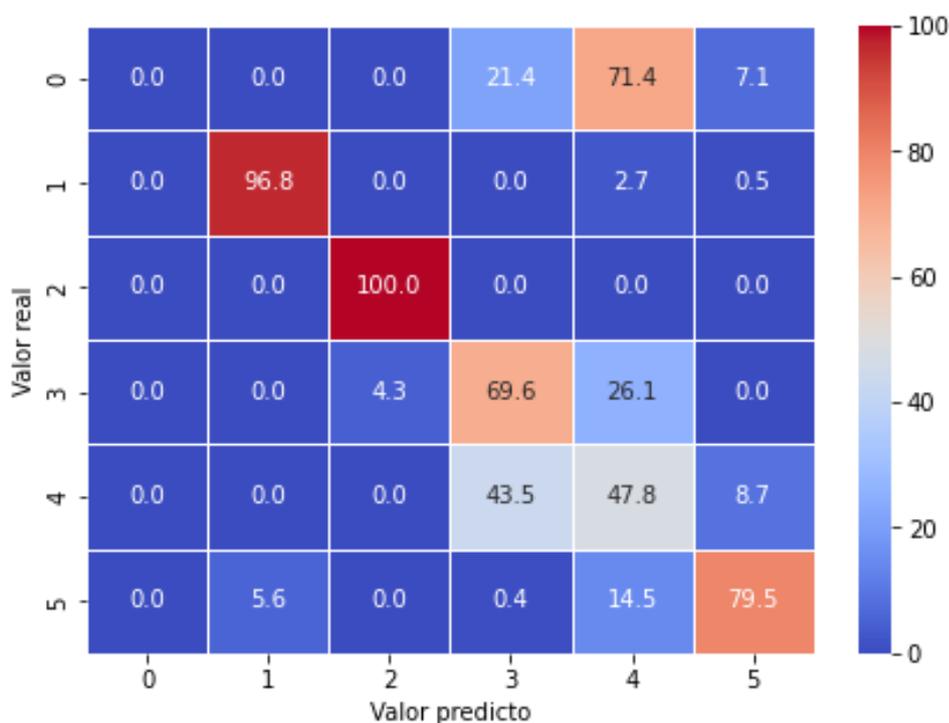
Tabela 11 – Acurácia média por classe

Classe	Acurácia (%)
Classe 0 (Descer escadas)	0
Classe 1 (Correr)	95,91
Classe 2 (Sentado)	100
Classe 3 (De pé)	81,74
Classe 4 (Subir escadas)	47,83
Classe 5 (Caminhar)	78,59

Fonte: Elaborado pelo autor

É possível notar que o classificador ainda não consegue reconhecer os valores de descida de escadas e possui dificuldades em reconhecer a subida também. Na figura 24 está apresentada a matriz de confusão gerada para estas condições.

Figura 24 – Matriz de Confusão para testes com janela



Fonte: Elaborado pelo autor

4.3 Normalização dos dados

Outra prática muito utilizada em classificadores é a normalização dos dados. A normalização, busca transformar todas os valores do banco de dados na mesma

ordem de grandeza. Para o teste do algoritmo com a normalização foi utilizada a janela de 10 segundos e *overlap* de 5 segundos.

O classificador utilizado foi uma rede neural MLP configurada conforme o subcapítulo apêndice D. O classificador foi executado 10 vezes, de modo a obtermos a acurácia média uma vez que o comportamento pode variar com os treinamentos. Ao final dos testes com a normalização foi obtida uma acurácia geral de 76,13 %. A tabela 12 demonstra a acurácia por classe obtida.

Tabela 12 – Acurácia por classe com normalização de dados

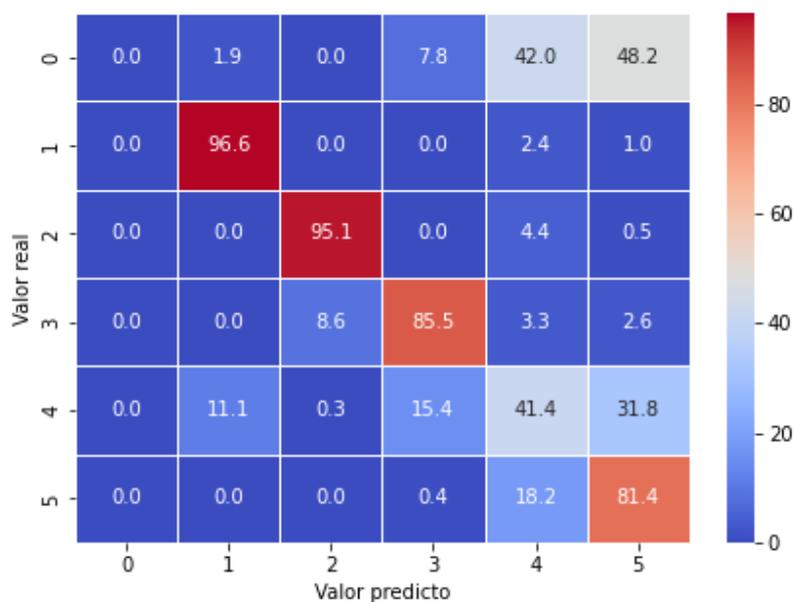
Classe	Acurácia (%)
Classe 0 (Descer escadas)	0,00
Classe 1 (Correr)	96,06
Classe 2 (Sentado)	94,96
Classe 3 (De pé)	84,34
Classe 4 (Subir escadas)	42,56
Classe 5 (Caminhar)	80,16

Fonte: Elaborado pelo autor

Podemos notar uma redução na acurácia geral, bem como uma redução de acurácia no reconhecimento da atividade de classe 5 (caminhar).

A figura 25 apresenta uma das matrizes de confusão para o comportamento do classificador com a normalização de dados inclusa.

Figura 25 – Matriz de confusão para o classificador com normalização de dados



Fonte: Elaborado pelo autor

4.4 Remoção do offset (nível DC do sinal)

Por muitas vezes o sinal capturado por um sensor pode incluir um nível DC devido ao seu valor de alimentação. De modo a verificar se este valor DC estava prejudicando a acurácia do classificador foi implementada uma lógica no código para remover este valor DC dos dados obtidos pelo acelerômetro.

Os testes foram executados com as janelas de 10s e *overlap* de 5s e utilizando como classificador uma rede neural MLP.

A acurácia média alcançada com a remoção do valor de offset do sinal foi de 75,46 %. A tabela 13 apresenta os valores de acurácia para cada classe de movimentos.

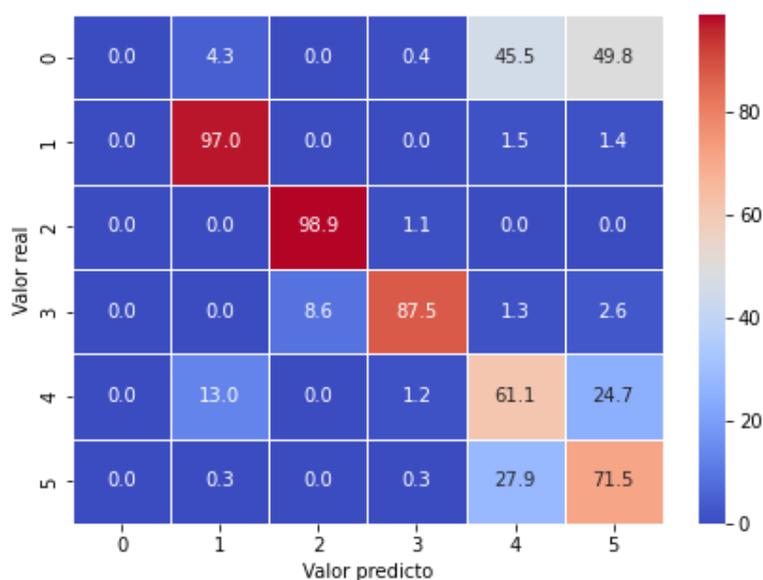
Tabela 13 – Acurácia por classe para o teste de remoção do offset

Classe	Acurácia (%)
Classe 0 (Descer escadas)	0
Classe 1 (Correr)	96,77
Classe 2 (Sentado)	98,52
Classe 3 (De pé)	85,99
Classe 4 (Subir escadas)	59,85
Classe 5 (Caminhar)	72,74

Fonte: Elaborado pelo autor

Abaixo está apresentada na figura 26 a matriz de confusão para uma das execuções deste teste.

Figura 26 – Matriz de confusão para testes com remoção de offset



Fonte: Elaborado pelo autor

4.5 Implementação de um filtro passa faixas

Outra iniciativa tomada para tentar melhorar a eficácia do sistema, foi a implementação de um filtro passa faixas para a frequência de amostragem do acelerômetro. Assim filtrar-se-ia outras frequências inclusas no sinal. Os parâmetros de ajuste do filtro são as frequências de corte (alta e baixa) e a frequência de sintonia.

Foram testadas experimentalmente várias frequências, de modo a encontrar qual parametrização do filtro alcançaria a melhor acurácia do sistema. Foi percebido também, que a utilização da redução por LDA e o offset afetavam os valores de acurácia final, logo os testes das frequências foram realizados sem estas funcionalidades.

Posteriormente foram realizados testes de modo a determinar a eficácia da combinação da redução LDA e offset com o filtro e encontrar o melhor valor de acurácia possível.

Foi utilizada uma janela de 10 segundos combinada com um *overlap* de 5 segundos. O tipo de classificador utilizado foi uma rede neural MLP. A tabela 14 mostra os valores testados e a acurácia obtida.

Tabela 14 – Testes do classificador com filtro passa faixas

Teste n°	Low (Hz)	High (Hz)	Fs (Hz)	Ordem	Acurácia (%)
1	5	100	1000	1	78,77
2	5	100	1000	2	78,15
3	5	100	1000	3	77,67
4	5	100	500	1	78,46
5	5	100	500	2	78,67
6	5	100	500	3	80,74
7	5	80	200	1	75,63
8	5	80	200	2	77,36
9	5	80	200	3	76,95
10	5	50	110	1	77,01
11	5	50	110	2	74,00
12	5	50	110	3	72,62
13	10	50	110	1	75,52
14	10	50	110	2	75,96
15	10	50	110	3	77,02
16	5	40	100	1	74,04
17	5	40	100	2	72,79
18	5	40	100	3	74,17
19	5	40	85	1	74,34
20	5	40	85	2	77,54
21	5	40	85	3	75,91

Fonte: Elaborado pelo autor

De acordo com o que foi possível verificar nos testes, o melhor resultado foi obtido quando utilizado o filtro com os parâmetros listados no teste de número 6, frequência de corte baixa de 5 Hz, frequência de corte alta de 100 Hz, frequência de sintonia de 500 Hz e filtro de ordem 3.

Com esta configuração, foi alcançada uma acurácia de 80,74 %. Os valores de acurácia por classe estão apresentados na tabela 15.

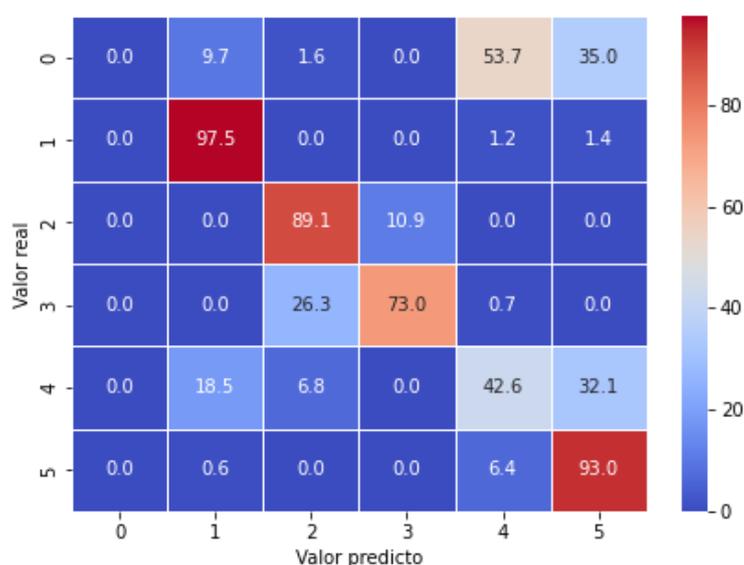
Tabela 15 – Acurácia por classe com filtro implementado

Classe	Acurácia (%)
Classe 0 (Descer escadas)	0
Classe 1 (Correr)	95,73
Classe 2 (Sentado)	87,16
Classe 3 (De pé)	70,46
Classe 4 (Subir escadas)	44,13
Classe 5 (Caminhar)	90,55

Fonte: Elaborado pelo autor

Houve uma melhora na identificação de todos os movimentos, porém o movimento de descer escadas ainda não é identificado, sendo principalmente confundido com o movimento de subir escadas, como podemos ver na matriz de confusão para uma das execuções do sistema apresentada na figura 27.

Figura 27 – Matriz de Confusão com filtro implementado



Fonte: Elaborado pelo autor

Conforme mencionado anteriormente no capítulo, também foram testados em conjunto com o filtro a implementação da LDA e offset de modo a verificar se a acurácia do classificador era melhorada.

A tabela 16 demonstra as combinações realizadas e a acurácia atingida.

Tabela 16 – Acurácia do classificador em combinações de métodos

Teste	Acurácia (%)
Filtro + LDA e Offset	71,56
Filtro + Offset	65,55
Filtro + LDA	75,09
Apenas Filtro	80,74

Fonte: Elaborado pelo autor

4.6 Testes do sistema com apenas 5 classes

Dada a ineficiência do sistema de reconhecer a atividade de descer escadas e com o intuito de ampliar a acurácia geral do sistema, esta classe foi removida, e os exemplos desta classe foram unidos com os exemplos de subir escadas, criando assim uma classe referente ao movimento de subir ou descer uma escada. Este processo também foi testado em artigos semelhantes ao utilizados neste trabalho (KWAPISZ, WEISS & MOORE, 2010).

A tabela 17 apresenta os novos valores de *labels* das classes para esta tentativa.

Tabela 17 – Labels para testes com 5 classes

<i>Label</i>	Atividade
0	Caminhar
1	Correr
2	Sentar
3	De pé parado
4	Subir/Descer Escadas

Fonte: Elaborado pelo autor

Os movimentos de subir e descer escadas foram concentrados na atividade de *label* 4, enquanto a atividade de caminhar, anteriormente *label* 5 foi movida para o *label* 0. A janela utilizada foi de 10 s combinada com um *overlap* de 5 segundos. O tipo de classificador foi uma rede MLP.

Após rodarmos o classificador com 5 classes, foi alcançada uma acurácia de 85,96 %. As acurácias por classe estão listadas na tabela 18.

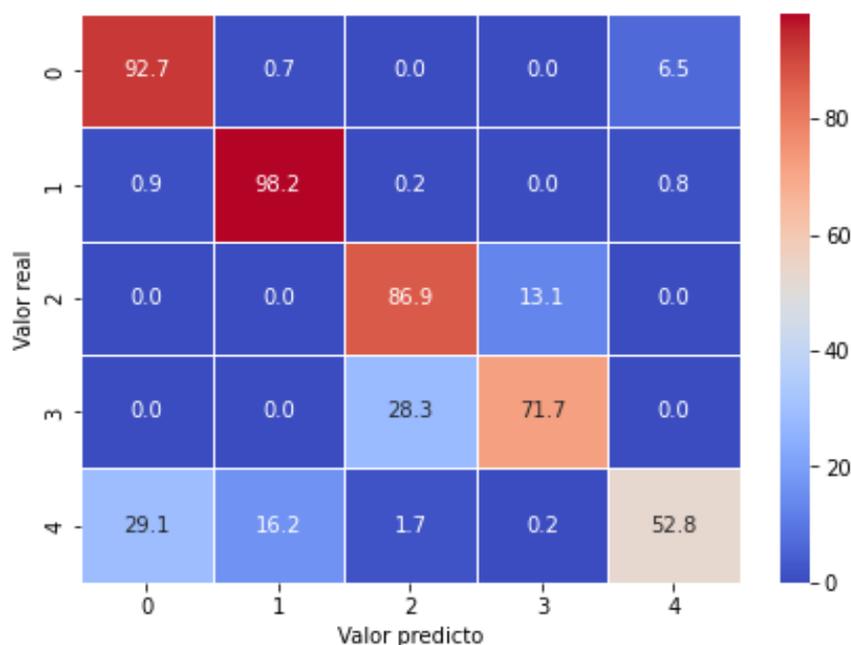
Tabela 18 – Acurácias por classe com 5 classes

Classe	Acurácia (%)
Classe 0 (Caminhar)	89,85
Classe 1 (Correr)	97,48
Classe 2 (Sentado)	83,66
Classe 3 (De pé)	73,02
Classe 4 (Subir/Descer escadas)	50,71

Fonte: Elaborado pelo autor

A matriz de confusão para uma das execuções do sistema está apresentada na figura 32.

Figura 28 – Matriz de confusão para classificador de 5 classes



Fonte: Elaborado pelo autor

Analisando a matriz de confusão e acurácias por classe, percebe-se que além do ganho de acurácia média de 5% no classificador, também se tem um ganho de

acurácia em algumas classes assim como uma diminuição em outras. A classe que apresenta mais problemas ainda é a classe com relação a subida e descida de escadas que é confundida com os movimentos de caminhar e correr uma vez que estes movimentos são semelhantes.

4.7 Testes do sistema sem os movimentos de subida e descida de escadas

Tomando em vista os testes anteriores, foi realizado um teste de modo a verificar qual seria a acurácia do sistema sem os movimentos de subida e descida de escadas, visto que estes além de terem baixa acurácia, interferem na acurácia de outros movimentos.

Foi realizada uma nova disposição de classes para os movimentos, a tabela 19 apresenta a nova distribuição de *labels* para os movimentos.

Tabela 19 – Labels para testes com 4 classes

<i>Label</i>	Atividade
0	Caminhar
1	Correr
2	Sentar
3	De pé parado

Fonte: Elaborado pelo autor

Após as modificações necessárias no código o algoritmo foi rodado e verificou-se uma acurácia de 95,92 %. A tabela 20 apresenta a acurácia por classe deste teste.

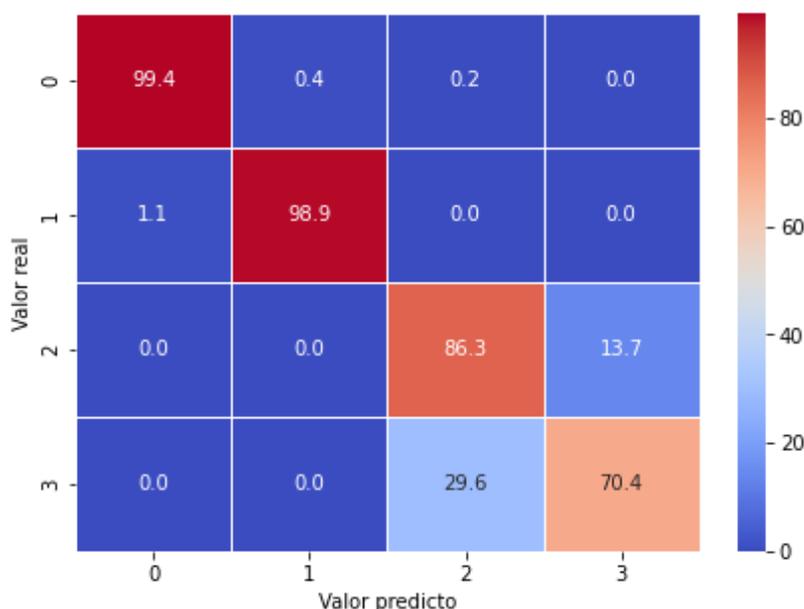
Tabela 20 – Acurácia por classe para teste com 4 classes

Classe	Acurácia (%)
Classe 0 (Caminhar)	98,00
Classe 1 (Correr)	99,03
Classe 2 (Sentado)	86,39
Classe 3 (De pé)	68,75

Fonte: Elaborado pelo autor

Por fim, podemos demonstrar a matriz de confusão deste teste na figura 29.

Figura 29 – Matriz de confusão para teste com apenas 4 classes



Fonte: Elaborado pelo autor

Através da inspeção da tabela podemos avaliar os resultados e concluir que existe uma ótima precisão, com a maioria das classes acima ou próxima de 90%. O movimento com menor acurácia é o de estar parado de pé, que em alguns casos é confundido com o movimento de estar sentado, uma vez que os dois são movimentos estacionários, sem grandes variações bruscas.

4.8 Comparação com outros trabalhos relacionados

Ao fim de todos os testes, por ser um dos objetivos específicos, era necessário comparar os resultados com outros estudos relacionados onde classificadores foram desenvolvidos de modo a identificar movimentos humanos através da captação do sinal de um acelerômetro.

O principal artigo a ser usado como comparação, é o artigo *Activity Recognition using Cell Phone Accelerometers* (KWAPISZ, WEISS & MOORE, 2010). Este estudo utilizou o mesmo banco de dados deste trabalho, os testes deste trabalho foram realizados de modo a utilizar o mesmo valor de tamanho de janela que o artigo, de modo a fazer as comparações entre as acurácias dos classificadores serem coerentes.

Para as 6 classes (movimentos) a acurácia encontrada no artigo foi de 91,7% com o classificador MLP, enquanto no nosso trabalho foi de 80,74% quando utilizado

o filtro. A tabela 21 apresenta a comparação de acurácias por classe dos dois classificadores.

Tabela 21 – Comparativo de acurácias por movimento com o classificador do artigo

Movimento	Classificador desenvolvido (%)	Classificador artigo (%)
Descer escadas	0	44,30
Correr	95,73	98,30
Sentado	87,16	95,0
De pé	70,46	91,90
Subir escadas	44,13	61,50
Caminhar	90,55	91,70

Fonte: Elaborado pelo autor

No geral, a acurácia média total e dos movimentos foi menor no classificador desenvolvido do que no classificador do artigo. Movimentos como correr e caminhar tiveram resultados bastante próximos. Também se percebe que no artigo existiu o mesmo problema relacionado aos movimentos de subida e descida de escadas.

Devido aos problemas com os movimentos de subida e descida também foi realizado no artigo a junção destes movimentos. Neste caso o artigo se utilizou de um classificador do tipo árvore de decisão J48, enquanto no classificador desenvolvido o tipo de classificador foi uma rede neural MLP.

A tabela 22 apresenta o comparativo dos resultados por atividade com apenas 5 classes entre o classificador desenvolvido e o classificador do artigo.

Tabela 22 - Comparativo de acurácias por movimento com o classificador do artigo quando utilizadas 5 classes

Movimento	Classificador desenvolvido (%)	Classificador artigo (%)
Caminhar	89,85	90,6
Correr	97,48	96,9
Sentado	83,66	96,5
De pé	73,02	93,7
Subir/Descer escadas	50,71	77,6

Fonte: Elaborado pelo autor

Avaliando a tabela nota-se que a atividade de caminhar do classificador desenvolvido tem uma acurácia próxima ao do artigo, enquanto a de correr tem uma acurácia superior. As demais acurácias são inferiores ao classificador do artigo. A acurácia média do artigo foi de 93,7 % enquanto a acurácia média do classificador desenvolvido foi de 85,96 %.

5 CONSIDERAÇÕES FINAIS

Pode-se dizer que todos os objetivos específicos do trabalho foram alcançados, uma vez que foi selecionado um banco existente, houve o desenvolvimento das rotinas para o reconhecimento dos movimentos e a definição de intervalos de janelamento. Ao fim os resultados foram validados e foram testadas várias mudanças de parâmetros de modo a incrementar a acurácia dos sistemas. Também foi possível, ao fim comparar os resultados obtidos com outros trabalhos similares que se utilizaram de inteligência artificial para classificar e identificar movimentos humanos, apesar de perceber-se ao fim do estudo que a acurácia do classificador desenvolvido é inferior aos classificadores desenvolvidos em trabalhos semelhantes.

Um dos principais pontos a serem destacados neste trabalho, é a dificuldade da identificação de movimentos relacionados a subida e descida de escadas. O movimento de descida de escadas não pode ser identificado com o classificador desenvolvido, sendo na maioria dos testes confundido com o movimento de subida. Quando os dois movimentos foram unidos em apenas uma classe, percebeu-se que ainda assim, existe uma certa dificuldade na identificação destes movimentos.

Ao remover os movimentos de subida e descida de escadas do treinamento e teste do classificador se percebeu o aumento da acurácia, alcançando patamares acima de 90 % de acurácia.

Com relação aos outros movimentos, de acordo com os resultados é possível dizer que o classificador consegue identificar os movimentos de caminhada, corrida e sentar-se com uma boa acurácia.

Outro comentário a ser realizado com relação ao classificador é o tamanho da janela. A janela de 10s foi utilizada de modo a comparar o classificador desenvolvido com o classificador do artigo *Activity Recognition using Cell Phone Accelerometers* (KWAPISZ, WEISS & MOORE, 2010). Seria interessante testar e aprimorar um classificador com menores janelas de modo a viabilizar o classificador a ser utilizado em mais aplicações.

Para futuras oportunidades, pode-se pensar em algumas possibilidades de iniciativas de modo a melhorar a performance do classificador. Poderiam ser testadas outras *features* e uma diferente configuração da rede neural. Outra possibilidade seria testar outros tipos de classificadores de modo a verificar se a acurácia média e de certos movimentos pode ser melhorada.

5.1 Sugestões para Trabalhos Futuros

Dadas as experiências adquiridas durante o desenvolvimento deste trabalho, é possível deixarmos alguma sugestão de trabalhos a serem desenvolvidos em futuras oportunidades.

- Testar o desempenho de outras *features* no classificador.
- Realizar a testagem de outros tipos de classificadores com relação ao banco de dados estudado.
- Capturar os movimentos com um acelerômetro de modo a construir o próprio banco de dados de movimentos.
- Realizar uma testagem e aprimoração do algoritmo utilizando menores valores de janela.

REFERÊNCIAS

ALEMAYOH, Tsige Tadesse; LEE, Jae Hoon; OKAMOTO, Shingo. **Deep Learning Based Real-time Daily Human Activity Recognition and Its Implementation in a Smartphone**, 16th International Conference on Ubiquitous Robots (UR), 2019, pp. 179-182, 2019.

ALPAYDIN, Ethem. **Introduction to Machine Learning**. 2. Ed. Londres: The MIT Press, 2010.

ANDERSON, Dave; MCNEILL, George. **Artificial Neural Networks Technology**. 1. Ed. Nova York: Kaman Sciences Corporation, 1992.

BITALINO. **Accelerometer (ACC) Assembled Sensor Data Sheet**. 2020.

Disponível em: <<https://bitalino.com/storage/uploads/media/assembled-acc-sensor-datasheet.pdf>> . Acesso em: 4 nov. 2021.

BITALINO. **Accelerometer (ACC) Sensor Data Sheet**. 2020. Disponível em:

<<https://bitalino.com/storage/uploads/media/revolution-acc-sensor-datasheet-revb.pdf>>. Acesso em: 5 nov. 2021.

BITALINO. **BITalino (r)evolution MuscleBIT Bundle Data Sheet**. 2020. Disponível

em: <<https://bitalino.com/storage/uploads/media/bitalino-musclebit-bundle-datasheet-revb-1.pdf>> . Acesso em: 3 nov. 2021.

BITALINO. **OpenSignals (r)evolution (v.2021) Software Manual**, 2021. Disponível

em: <<https://bitalino.com/downloads/software>>. Acesso em: 2 nov. 2021.

BOELTER, Josué da Silva. **Classificação de sinais eletromiográficos utilizando redes neurais artificiais, análise discriminante linear e floresta aleatória**.

Trabalho de Conclusão (Trabalho de conclusão – Engenharia Eletrônica).

Engenharia Eletrônica. Universidade do Vale do Rio dos Sinos. São Leopoldo, 2021.

CHERNBUMROONG, Saisakul; AITKINS, Anthony S; YU, Hongnian. **Activity classification using a single wrist-worn accelerometer**, 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings, pp. 1-6, 2011.

DA SILVA, Fernando G; **Reconhecimento de movimentos humanos utilizando um acelerômetros e inteligência computacional**. 2013. Dissertação (Mestrado em Engenharia). Mestrado em Engenharia. Universidade de São Paulo. São Paulo, 2013.

GIACOMOSSI, Luiz Carlos; **Método não invasivo utilizando acelerômetro para classificar movimentos normais e anormais de humanos**. 2011. Dissertação (Mestrado em Ciências – Área de Concentração: Engenharia Biomédica). Programa de pós-graduação em engenharia elétrica e informática industrial. Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

GODFREY, A; BOURKE, A.K; ÓLAIGHIN, G.M; VAN DER VEN, P; NELSON, J. **Activity classification using a single chest mounted tri-axial accelerometer**. Medical Engineering & Physics journal, New Castle, n. 33, p. 1127- 1135, mai. 2011.

JOSÉ, Italo. **KNN (K-Nearest Neighbors) #1**. 2018. Disponível em: <<https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e>>. Acesso em: 26 nov. 2021.

KHIMRAJ; SHUKLA, Praveen Kumar; VIJAYVARGIYA, Ankit; KUMAR, Rajesh. **Human Activity Recognition using Accelerometer and Gyroscope Data from Smartphones**, International Conference on Emerging Trends in Communication, Control and Computing (ICONC3), pp. 1-6, 2020.

KWAPISZ, Jennifer R; WEISS, Gary M; MOORE, Samuel A. **Activity Recognition using Cell Phone Accelerometers**, ACM SIGKDD Explorations Newsletter, New York, n. 12, Issue 2, p. 74-82, dez. 2010.

LEE, Song-Mi; YOON, Sang Min; CHO, Heeryon. **Human activity recognition from accelerometer data using Convolutional Neural Network**. 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 131-134, 2017.

LESSA, Luciana Moreira; GOUVÊA, Jayme Pereira. **Análise biomecânica da marcha humana durante o subir e descer escadas.** Cadernos UniFOA, Volta Redonda, n. 38, p. 21-36, dez. 2018.

MACHADO, Simone Christiane. **Posição em pé e o trabalho de professores da universidade presbiterana Mackenzie.** Trabalho de Conclusão (Trabalho de conclusão – Engenharia de Segurança do Trabalho). Engenharia de Segurança do trabalho. Universidade Presbiterana Mackenzie. São Paulo, 2009.

NAVLANI, Avinash. **KNN Classification using Scikit-learn.** 2018. Disponível em: <<https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>>. Acesso em: 7 nov 2021.

SWAROOP, C. H. **A Byte of Python.** 2004. Disponível em: <<https://python.swaroopch.com/>>. Acesso em: 4 nov. 2021.

APÊNDICE A – TRECHO DE CÓDIGO PARA IDENTIFICAÇÃO DE INÍCIO E FIM DE ATIVIDADE

```
inicio_act_1.append(1)
E1_Activity = Activity
for i in range (0,len(E1_Activity)):
    if (i != 343416) and (i != 0) and (i != 343415) and (i != 343417):
        if (E1_Activity[i] == 1) and (E1_Activity[i] != E1_Activity[i-1]):
            inicio_act_1.append(i+1)
        if (E1_Activity[i] == 1) and (E1_Activity[i] != E1_Activity[i+1]):
            fim_act_1.append(i+1)
        if (E1_Activity[i] == 2) and (E1_Activity[i] != E1_Activity[i-1]):
            inicio_act_2.append(i+1)
        if (E1_Activity[i] == 2) and (E1_Activity[i] != E1_Activity[i+1]):
            fim_act_2.append(i+1)
        if (E1_Activity[i] == 3) and (E1_Activity[i] != E1_Activity[i-1]):
            inicio_act_3.append(i+1)
        if (E1_Activity[i] == 3) and (E1_Activity[i] != E1_Activity[i+1]):
            fim_act_3.append(i+1)
        if (E1_Activity[i] == 4) and (E1_Activity[i] != E1_Activity[i-1]):
            inicio_act_4.append(i+1)
        if (E1_Activity[i] == 4) and (E1_Activity[i] != E1_Activity[i+1]):
            fim_act_4.append(i+1)
        if (E1_Activity[i] == 5) and (E1_Activity[i] != E1_Activity[i-1]):
            inicio_act_5.append(i+1)
        if (E1_Activity[i] == 5) and (E1_Activity[i] != E1_Activity[i+1]):
            fim_act_5.append(i+1)
        if (E1_Activity[i] == 0) and (E1_Activity[i] != E1_Activity[i-1]):
            inicio_act_0.append(i+1)
        if (E1_Activity[i] == 0) and (E1_Activity[i] != E1_Activity[i+1]):
            fim_act_0.append(i+1)
```

APÊNDICE B – TRECHO DE CÓDIGO PARA AS FUNÇÕES DE CÁLCULOS DE FEATURES

```
# Mean Absolute Value (MAV)
def featureMAV(data):
    return np.mean(np.abs(data), axis=0)

# Root Mean Square (RMS)
def featureRMS(data):
    return np.sqrt(np.mean(data**2, axis=0))

# Log Root Mean Square (logRMS)
def featureLRMS(data):
    return np.log(np.sqrt(np.mean(data**2, axis=0)))

# Waveform length (WL)
def featureWL(data):
    return np.sum(np.abs(np.diff(data, axis=0)),axis=0)/data.shape[0]

# Kurtosis (KUR)
def featureKUR(data):
    #data = norm.rvs(size=1000, random_state=3)
    return kurtosis(data)

# Skewness (SKEW)
def featureSKEW(data):
    return skew(data)

# Variance (VAR)
def featureVAR(data):
    variance = np.var(data)
    return variance
```

ANEXO C – TRECHO DE CÓDIGO QUE CHAMA AS FUNÇÕES PARA CÁLCULO DE FEATURES

```
def calcular_features (segment_x,segment_y,segment_z):  
    labels_total = []  
    total_features = []  
    for z in range (0, len(segment_x)):  
        r0 = featureRMS(segment_x[z])  
        r1 = featureRMS(segment_y[z])  
        r2 = featureRMS(segment_z[z])  
        w0 = featureWL(segment_x[z])  
        w1 = featureWL(segment_y[z])  
        w2 = featureWL(segment_z[z])  
        v0 = featureMAV(segment_x[z])  
        v1 = featureMAV(segment_y[z])  
        v2 = featureMAV(segment_z[z])  
        f0 = featureKUR(segment_x[z])  
        f1 = featureKUR(segment_y[z])  
        f2 = featureKUR(segment_z[z])  
        g0 = featureSKEW(segment_x[z])  
        g1 = featureSKEW(segment_y[z])  
        g2 = featureSKEW(segment_z[z])  
        o0 = featureLRMS(segment_x[z])  
        o1 = featureLRMS(segment_y[z])  
        o2 = featureLRMS(segment_z[z])  
        b0 = featureVAR(segment_x[z])  
        b1 = featureVAR(segment_y[z])  
        b2 = featureVAR(segment_z[z])  
        total_features.append([r0,r1,r2,  
                               b0,b1,b2,  
                               v0,v1,v2,  
                               w0,w1,w2,  
                               f0,f1,f2,  
                               g0,g1,g2,
```

```
        o0,o1,o2])  
return total_features
```

ANEXO D - TRECHO DE CÓDIGO DA PARAMETRIZAÇÃO DA REDE NEURAL

```
clf = Sequential()
clf.add(Reshape((1,21), input_shape = (input_shape,)))
clf.add(Dense(neuron, activation = 'relu'))
clf.add(Dropout(dropout))
clf.add(Dense(100, activation = 'relu'))
clf.add(Dropout(dropout))
clf.add(Dense(neuron, activation = 'relu'))
clf.add(Flatten())
clf.add(Dense(num_classes, activation='softmax'))
#print(model.summary())
clf.compile(loss='categorical_crossentropy',
            optimizer='adam', metrics=['accuracy'])
```