



**DETECÇÃO DE MODO DE TRANSPORTE UTILIZANDO  
*SMARTPHONES* E REDES NEURAIAS ARTIFICIAIS  
PARA PREVENÇÃO DE ACIDENTES**

**MATEUS MENDELSON ESTEVES DA SILVA**

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**FACULDADE DE TECNOLOGIA  
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**DETECÇÃO DE MODO DE TRANSPORTE UTILIZANDO  
*SMARTPHONES* E REDES NEURAIAS ARTIFICIAIS  
PARA PREVENÇÃO DE ACIDENTES**

**MATEUS MENDELSON ESTEVES DA SILVA**

**DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO  
DE ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNI-  
VERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS  
PARA A OBTENÇÃO DO GRAU DE MESTRE EM SISTEMAS MECATRÔNICOS.**

**APROVADA POR:**

---

**Prof. Alexandre Zaghetto, DSc., CIC/UnB  
(Orientador)**

---

**Prof. Adson Ferreira da Rocha, Dr., FGA/UnB  
Membro Externo**

---

**Prof. Li Weigang, Dr., CIC/UnB  
Membro Interno**

---

**Prof. Marcus Vinicius Lamar, Dr., CIC/UnB  
Membro Suplente**

**BRASÍLIA, 09 DE ABRIL DE 2018.**

## FICHA CATALOGRÁFICA

SILVA, MATEUS MENDELSON ESTEVES DA  
DETECÇÃO DE MODO DE TRANSPORTE UTILIZANDO *SMARTPHONES* E REDES NEURAIIS ARTIFICIAIS PARA PREVENÇÃO DE ACIDENTES [Distrito Federal] 2018. xiv,69p., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2018).  
Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia.  
Departamento de Engenharia Mecânica

1. Redes Neurais Artificiais	2. Smartphones
3. Prevenção de Acidentes	4. Acelerômetro
I. ENM/FT/UnB	II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

MENDELSON, M. (2018). DETECÇÃO DE MODO DE TRANSPORTE UTILIZANDO *SMARTPHONES* E REDES NEURAIIS ARTIFICIAIS PARA PREVENÇÃO DE ACIDENTES, Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM140, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 69p.

## CESSÃO DE DIREITOS

AUTOR: Mateus Mendelson Esteves da Silva

TÍTULO: DETECÇÃO DE MODO DE TRANSPORTE UTILIZANDO *SMARTPHONES* E REDES NEURAIIS ARTIFICIAIS PARA PREVENÇÃO DE ACIDENTES.

GRAU: Mestre                      ANO: 2018

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

---

Mateus Mendelson Esteves da Silva  
Departamento de Eng. Mecânica (ENM) - FT  
Universidade de Brasília (UnB)  
Campus Darcy Ribeiro  
CEP 70919-970 - Brasília - DF - Brasil

*Dedicado ao único e soberano Deus,  
que mandou Seu filho unigênito para  
que eu pudesse ter vida: Jesus  
Cristo.*

## **AGRADECIMENTOS**

*Ao professor Alexandre Zaghetto pelas muitas horas e energia dedicadas à orientação.*

*Ao verdadeiro Deus, que com Sua misericórdia me deu a vida.*

*Aos meus pais, Romualdo Silveira da Silva e Márcia Esteves Silveira, pelo inabalável apoio e por me ensinarem constantemente que não há obstáculos intransponíveis.*

*À minha namorada, Maria Dara, pela paciência e ajuda.*

*Aos meus colegas do BiTGroup por anos de momentos inesquecíveis e inúmeros cafés.*

## **RESUMO**

### **DETECÇÃO DE MODO DE TRANSPORTE UTILIZANDO *SMARTPHONES* E REDES NEURAIAS ARTIFICIAIS PARA PREVENÇÃO DE ACIDENTES**

**Autor: Mateus Mendelson Esteves da Silva**

**Orientador: Prof. Alexandre Zaghetto, DSc., CIC/UnB**

**Programa de Pós-graduação em Sistemas Mecatrônicos**

**Brasília, Abril de 2018**

Aparelhos celulares estão se tornando cada dia mais sofisticados. As gerações mais recentes são denominadas *smartphones*, deixando de ser apenas dispositivos de telecomunicação verbal e tornando-se computadores portáteis com considerável capacidade de processamento e memória. Atualmente, eles são compostos por diversos e poderosos sensores, dentre eles o acelerômetro. Este trabalho apresenta um método baseado em Redes Neurais Artificiais e no descritor de texturas GLCM que, utilizando sinais capturados apenas a partir de acelerômetros de telefones celulares convencionais, é capaz de identificar situações de risco de acidentes. O método é capaz de identificar 3 modos de transporte: carro, bicicleta e a pé. Dependendo do modo de transporte, é possível identificar se o *smartphone* está sendo transportado em mãos, em um painel ou no bolso do usuário, identificando se ele está em movimento ou parado. Por fim, também identifica o modo de interação com o dispositivo: em ligação, olhando/digitando ou sem interação. Os testes mostram que o método proposto é capaz de identificar uma entre 18 possíveis classes de transporte com uma taxa de acerto de 94,31% com uma janela de 14 segundos.



## **ABSTRACT**

### **USING ARTIFICIAL NEURAL NETWORKS AND SMARTPHONE ACCELEROMETERS FOR ACCIDENT PREVENTION**

**Author: Mateus Mendelson Esteves da Silva**

**Supervisor: Prof. Alexandre Zaghetto, DSc., CIC/UnB**

**Programa de Pós-graduação em Sistemas Mecatrônicos**

Mobile phones are increasingly sophisticated. The most recent generations are called smartphones. They are no longer merely communication devices; they became portable computers with significant processing power and memory. Nowadays, they are composed of several powerful sensors, among them the accelerometer. This work presents a method based in Artificial Neural Networks and in GLCM texture descriptor that, using signals captured only from the accelerometer of conventional mobile phones, is able to identify accident risk situations. This method is able to identify 3 transport modes: car, bicycle and foot. Depending on the mode of transportation, it is possible to identify whether the smartphone is being transported in the hand, in a dashboard or in the user's pocket, and whether they are moving or stopped. Finally, it also identifies the mode of interaction with the device: in a phone call, looking/typing or with no interaction. Tests showed that the proposed method is able to identify one among 18 transport classes with a success rate of 94.31% with a window of 14 seconds.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>4</b>
2.1	2.1 FUNDAMENTAÇÃO TEÓRICA	4
2.1.1	2.1.1 ACELERÔMETROS	4
2.1.2	2.1.2 NORMAS E NORMALIZAÇÃO	4
2.1.3	2.1.3 ESPAÇOS DE COR RGB E YCbCr	6
2.1.4	2.1.4 IMAGENS EM NÍVEIS DE CINZA	7
2.1.5	2.1.5 MATRIZ DE CO-OCORRÊNCIA DE NÍVEIS DE CINZA	7
2.1.5.1	2.1.5.1 CONTRASTE	12
2.1.5.2	2.1.5.2 CORRELAÇÃO	12
2.1.5.3	2.1.5.3 ENERGIA	12
2.1.5.4	2.1.5.4 HOMOGENEIDADE	12
2.1.6	2.1.6 RECONHECIMENTO DE PADRÕES	12
2.1.7	2.1.7 REDES NEURAIS ARTIFICIAIS	15
2.1.7.1	2.1.7.1 NEURÔNIOS ARTIFICIAIS	16
2.1.7.1.1	2.1.7.1.1 FUNÇÕES DE ATIVAÇÃO	17
2.1.7.2	2.1.7.2 REDES NEURAIS ARTIFICIAIS <i>Feed-forward</i>	20
2.1.7.3	2.1.7.3 TREINAMENTO	22
2.1.7.3.1	2.1.7.3.1 ALGORITMO DE TREINAMENTO <i>Backpropagation</i>	23
2.1.7.3.2	2.1.7.3.2 ALGORITMO DE TREINAMENTO DE LEVENBERG-MARQUARDT	27
2.1.7.4	2.1.7.4 TÉCNICA DE MONTE CARLO	28
2.2	2.2 TRABALHOS CORRELATOS	30
<b>3</b>	<b>MÉTODO PROPOSTO</b>	<b>36</b>
3.1	3.1 EXTRAÇÃO DE CARACTERÍSTICAS	37
3.2	3.2 CLASSES ANALISADAS	41
3.2.1	3.2.1 MODO DE TRANSPORTE: CARRO	43
3.2.2	3.2.2 MODO DE TRANSPORTE: BICICLETA	44
3.2.3	3.2.3 MODO DE TRANSPORTE: A PÉ	45
3.3	3.3 CLASSIFICADOR HIERÁRQUICO	46
3.4	3.4 CONSTRUÇÃO DA BASE DE DADOS	47
<b>4</b>	<b>RESULTADOS EXPERIMENTAIS</b>	<b>49</b>
4.1	4.1 TREINAMENTO DO SISTEMA	49

4.1.1	TREINAMENTO COM MONTE CARLO .....	52
4.2	4.2 RESULTADOS .....	53
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>65</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>67</b>

## LISTA DE FIGURAS

2.1	Sistema de coordenadas de referência utilizado. Os vetores $\hat{i}$ , $\hat{j}$ e $\hat{k}$ são os vetores unitários dos eixos $x_1$ , $x_2$ e $x_3$ , respectivamente, e $\hat{n}$ é o vetor unitário normal à tela do <i>smartphone</i> . A aceleração é dada por $\vec{a} = a_{x_1}\hat{i} + a_{x_2}\hat{j} + a_{x_3}\hat{k}$	5
2.2	Conversão de uma imagem colorida para imagens com diferentes quantidades de níveis de cinza. É importante notar que, conforme a quantidade de níveis de cinza cai, o detalhamento da imagem também é perdido. (a) Imagem colorida original (RGB); (b) imagem com 8 bits e 256 níveis de cinza; (c) imagem com 4 bits e 16 níveis de cinza; (d) imagem com 3 bits e 8 níveis de cinza; (e) imagem com 2 bits e 4 níveis de cinza; e (f) imagem com 1 bit e 2 níveis de cinza. ....	8
2.3	Imagem de exemplo em níveis de cinza que será utilizada para melhor entendimento do funcionamento da GLCM. Cada quadrado representa um píxel e há apenas quatro possíveis valores de intensidade: 0 (o mais escuro), 1, 2 e 3 (o mais claro).....	9
2.4	Valores relativos à Figura 2.3. Cada quadrado representa um píxel e há apenas quatro possíveis valores de intensidade: 0 (o mais escuro), 1, 2 e 3 (o mais claro).....	9
2.5	Valores relativos à Figura 2.3 com algumas posições em destaque, considerando apenas $p_{v(0,1)}$ . Os píxeis em azul exemplificam píxeis de referência; píxeis em vermelho, de vizinhança. É importante notar que, ressaltados, há duas ocorrências de $p_{v(0,1)} = 2$ para $p_r = 1$ e uma ocorrência de $p_{v(0,1)} = 0$ para $p_r = 0$ . ....	10
2.6	Matriz GLCM construída a partir da Figura 2.3 considerando $p_{v(0,1)}$ , cujos valores estão na Figura 2.4. Os índices nas abscissas representam os valores de $p_{v(0,1)}$ , enquanto os índices nas ordenadas representam os valores de $p_r$ . ....	11
2.7	As 8 matrizes GLCM resultantes da vizinhança de 8 da Figura 2.3. As matrizes são obtidas considerando o píxel vizinho indicado no canto superior esquerdo de cada uma. ....	11
2.8	Valores de contraste, correlação, energia e homogeneidade obtidos a partir das GLCMs da Figura 2.7. ....	13
2.9	Histogramas comparativos entre 3 variações da flor Iris: setosa, versicolor e virgínica. Cada histograma compara uma característica diferente: (a) histograma comparativo do comprimento das sépalas; (b) histograma comparativo da largura das sépalas; (c) histograma comparativo do comprimento das pétalas; e (d) histograma comparativo da largura das pétalas. ....	14

2.10	Uso das características "comprimento da sépala" e "largura da pétala" em conjunto. É importante notar que, apesar da classe "setosa" ser completamente distinguível, há sobreposição dos conjuntos "versicolor" e "virgínica" em alguns pontos. Dessa forma, um simples classificador linear não é suficiente.....	15
2.11	Estrutura básica de um neurônio artificial. ....	17
2.12	Funções de ativação (a) degrau e (b) logística. É importante notar que a função logística pode ser tratada como uma versão suavizada da função degrau.	19
2.13	Duas funções de ativação bastante utilizadas: (a) tangente hiperbólica e (b) função linear. ....	20
2.14	Arquitetura de uma rede neural artificial <i>feed-forward</i> de $n$ camadas ocultas. $p_1, p_2, \dots, p_m$ representam as $m$ entradas da rede neural, enquanto que cada $W_{i,j}$ indica o $j$ -ésimo neurônio da $i$ -ésima camada. É importante notar que as quantidades de neurônios $\alpha, \beta, \dots, \omega$ não são necessariamente iguais.....	21
2.15	Modelo de componentes de uma camada $n$ de uma rede neural. O campo local induzido é representado por $u_j^n(m)$ para cada neurônio $j$ da camada $n$ , $v_i^{n-1}(m)$ é o elemento $i$ do sinal de saída da camada $n - 1$ na iteração $m$ , $w_{ji}^n(m)$ é o valor da sinapse que vem do elemento $i$ da camada $n - 1$ para o neurônio $j$ da camada $n$ , $R_n$ é o número de elementos do vetor de saída da camada $n$ e $b_j^n(m)$ é o viés do neurônio $j$ da camada $n$ . ....	22
2.16	Rede neural com sinapses inicializadas em valores aleatórios. $x_i$ indica a $i$ -ésima entrada da rede, $\tilde{y}_i$ indica a $i$ -ésima saída da rede e $N_i^n$ indica o $i$ -ésimo neurônio da camada $n$ . A função de ativação dos neurônios $N_1^1, N_2^1, N_3^1 e N_2^2$ é a logística sigmoide; do neurônio $N_1^2$ , uma função linear. ....	25
3.1	Fluxograma do método proposto. O sinal original é dividido em janelas de $s$ segundos de aquisições, que são reorganizadas em uma matriz. Essa matriz, por sua vez, é dividida em 3 faixas de frequências. Após a escolha das faixas de interesse, a extração de características é realizada e, por fim, as características são apresentadas ao classificador hierárquico, que decidirá se a situação apresentada é de risco ou não. É importante notar que o fluxograma demonstra o funcionamento do método para apenas um único eixo $x_i$ , sendo necessário concatenar as características dos demais eixos conforme as Equações 3.4, 3.5 e 3.6. ....	36
3.2	Demarcação das faixas de frequência utilizadas em cada área da matriz. A área demarcada por Lo indica a faixa de baixa frequência (raio $\lfloor \frac{k}{6} \rfloor$ ); Me, a faixa intermediária (raio $\lfloor \frac{2 \times k}{6} \rfloor$ ); Hi, a área de alta frequência (raio $\frac{k}{2}$ ). A área não hachurada, por sua vez, é descartada e, portanto, não é utilizada na extração de características. O cálculo da GLCM é aplicado sobre cada uma das faixas e são extraídos contraste, correlação, energia e homogeneidade conforme as Equações 2.15, 2.16, 2.17 e 2.18, respectivamente. ....	38

3.3	Representação das características extraídas de uma janela de dados para um único eixo a partir das faixas de frequência da Figura 3.2. Como resultado, cada faixa de cada eixo fornece 4 valores (contraste, correlação, energia e homogeneidade médios). Caso haja outros eixos, os valores de cada faixa são concatenados com seus respectivos valores dos demais eixos.....	39
3.4	Esquema de nomenclatura de classes de transporte. Cada classe é dividida em quatro divisões: modo de transporte, local de transporte, situação e modo de interação. Cada divisão é identificada por uma letra e seu valor é identificado por um número. Por exemplo, a classe $a_p b_q c_1 d_r$ indica que o <i>smartphone</i> está sendo transportado no modo de transporte $p$ , no local $q$ , o usuário está em movimento e o modo de interação é o $r$ . .....	41
3.5	Classificações realizadas pelo sistema. Cada classe é representada através da união dos índices de cada divisão. A classe $a_1 b_1 c_1 d_1$ , por exemplo, indica que o <i>smartphone</i> está sendo transportado em um carro e levado em mãos enquanto o carro está em movimento e o usuário está realizando uma ligação. A classe $a_3 b_1 c_1 d_2$ , por sua vez, indica que o usuário está a pé com o <i>smartphone</i> em mãos enquanto caminha e olha ou digita algo, e assim por diante. ....	42
3.6	Classificações realizadas quando o usuário está utilizando um carro ( $a_1$ ). As setas em vermelho formam as possíveis classes para a situação em que o dispositivo é transportado em mãos (classes $a_1 b_1 c_1 d_1$ , $a_1 b_1 c_1 d_2$ , $a_1 b_1 c_2 d_1$ e $a_1 b_1 c_2 d_2$ ); a seta verde, as situações em que o <i>smartphone</i> é transportado no painel do carro (classes $a_1 b_2 c_1 d_3$ e $a_1 b_2 c_2 d_3$ ). ....	43
3.7	Classificações realizadas quando o usuário está utilizando uma bicicleta ( $a_2$ ). As setas em vermelho formam as possíveis classes para a situação em que o dispositivo é transportado em mãos (classes $a_2 b_1 c_1 d_2$ e $a_2 b_1 c_2 d_2$ ); a seta verde, as situações em que o <i>smartphone</i> é transportado no painel da bicicleta (classes $a_2 b_2 c_1 d_3$ e $a_2 b_2 c_2 d_3$ ); as pretas, quando no bolso (classes $a_2 b_3 c_1 d_3$ e $a_2 b_3 c_2 d_3$ ). ....	44
3.8	Classificações realizadas quando o usuário está a pé ( $a_3$ ). As setas em vermelho formam as possíveis classes para a situação em que o dispositivo é transportado em mãos (classes $a_3 b_1 c_1 d_1$ , $a_3 b_1 c_1 d_2$ , $a_3 b_1 c_2 d_1$ e $a_3 b_1 c_2 d_2$ ); a seta preta, as situações em que o <i>smartphone</i> é transportado no bolso do usuário (classes $a_3 b_3 c_1 d_3$ e $a_3 b_3 c_2 d_3$ ). ....	45
3.9	Representação do classificador hierárquico. Dado um conjunto de características, o sistema o classifica como "situação de risco" ou "não é situação de risco" .....	46

3.10	Posicionamento do <i>smartphone</i> em cada modo de transporte: (a) <i>smartphone</i> posicionado em suporte no painel do carro, em frente à saída do ar-condicionado. O dispositivo é colocado em pé com a tela voltada para o motorista. O ar-condicionado e o som do carro permaneceram desligados o tempo todo; (b) <i>smartphone</i> colocado dentro de um bolso posicionado no painel da bicicleta com a tela virada para cima; e (c) <i>smartphone</i> sendo colocado no bolso. A parte de trás do dispositivo encontra-se virada para a perna do usuário e o aparelho é inserido de cabeça para baixo. ....	48
4.1	Fluxograma do processo de treinamento.....	49
4.2	Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica se o celular está em movimento ou não. Resultados obtidos a partir da Tabela 4.1. ....	53
4.3	Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica qual o modo de transporte sendo utilizado. Resultados obtidos a partir da Tabela 4.2. ....	55
4.4	Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica se o celular está sendo utilizado em situação de risco ou não, dado que o modo de transporte é um carro. Resultados obtidos a partir da Tabela 4.3.....	58
4.5	Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica se o celular está sendo utilizado em situação de risco ou não, dado que o modo de transporte é uma bicicleta. Resultados obtidos a partir da Tabela 4.4. ....	60
4.6	Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica se o celular está sendo utilizado em situação de risco ou não, dado que o modo de transporte é a pé. Resultados obtidos a partir da Tabela 4.5. ....	62
4.7	Curva que relaciona o tamanho das janelas com as taxas de acerto para o classificador hierárquico, dado que foram utilizadas as faixas de frequência do conjunto <i>LoMeHi</i> . Resultados obtidos a partir da Tabela 4.6. ....	64

## LISTA DE TABELAS

2.1	Resumo das abordagens e resultados obtidos em trabalhos correlatos. ....	35
4.1	Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem se a janela em questão indica movimento ou não. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente. ....	54
4.2	Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem qual o modo de transporte sendo utilizado, dado que o usuário está em movimento. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente. ....	56
4.3	Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem se o <i>smartphone</i> está sendo utilizado em uma situação de risco, dado que o usuário está em um carro em movimento. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente. ....	57
4.4	Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem se o <i>smartphone</i> está sendo utilizado em uma situação de risco, dado que o usuário está em uma bicicleta e em movimento. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente. ....	59
4.5	Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem se o <i>smartphone</i> está sendo utilizado em uma situação de risco, dado que o usuário está a pé e em movimento. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente.....	61
4.6	Tabela referente aos resultados obtidos pelo classificador hierárquico tendo como entrada os dados do conjunto de teste geral. O classificador utiliza todas as faixas e as redes neurais adotadas são aquelas apresentadas em suas respectivas tabelas. O tamanho da janela está em segundos. ....	63

## LISTA DE SÍMBOLOS

<i>GLCM</i>	Gray-Level Co-occurrence Matrix
<i>RNA</i>	Rede Neural Artificial
<i>FFT</i>	Fast Fourier Transform
<i>SMO</i>	Sequential Minimal Optimization
<i>KNN</i>	K-Nearest Neighbor
<i>GPS</i>	Global Positioning System
<i>SVM</i>	Support Vector Machine

# 1 INTRODUÇÃO

Aparelhos celulares estão se tornando cada vez mais sofisticados. As gerações mais recentes recebem o nome de *smartphones*, deixando de ser apenas dispositivos de telecomunicação verbal e tornando-se computadores portáteis com considerável capacidade de processamento e memória. Atualmente, eles são compostos por diversos e poderosos sensores. Entre os sensores mais comuns, é importante mencionar: GPS (do inglês, *Global Positioning System*), sensores de imagens (câmeras), de cor, de frequência cardíaca, de áudio (microfones), de temperatura e até mesmo de aceleração (acelerômetro). Esses sensores podem ser utilizados nas mais diversas aplicações, conforme as necessidades dos usuários. Seu uso, entretanto, tem sido cada vez mais irresponsável. O uso de *smartphones* enquanto se conduz um veículo, por exemplo, é um grande fator de risco de acidentes.

O monitoramento e a análise de motoristas é o processo de, automaticamente, coletar dados de condução e aplicar de um modelo computacional capaz de gerar um índice de risco de acidentes [1]. Surgem, então, técnicas capazes de detectar o modo de transporte (carro, ônibus, bicicleta, a pé etc) utilizado pelo portador do aparelho celular e gerar índices capazes de indicar se a situação de uso do dispositivo gera riscos àqueles ao seu redor. Em especial, o uso destes dispositivos ao dirigir e andar na rua tem se tornado cada vez mais comum, o que oferece grande risco de acidentes. Em um estudo realizado em seis capitais europeias (Amsterdã, Berlim, Bruxelas, Paris, Roma e Estocolmo), foi constatado que cerca de 17% dos 14 mil pedestres monitorados utilizam seus *smartphones* enquanto se locomovem na rua. Ainda no mesmo estudo, há relatos de situações de altíssimo risco ou até mesmo de acidentes [2]:

"Algo que foi observado repetidas vezes foram grupos de jovens olhando para um *smartphone* juntos enquanto atravessavam a rua. Em um dos casos, o grupo inteiro chegou a colidir com um ciclista."

"Uma mulher empurrando um carrinho de bebê por uma faixa de pedestres em uma área repleta de sinais de trânsito, enquanto enviava mensagens em seu celular sem olhar para a cor do sinal enquanto atravessava."

"Um incidente em Estocolmo chamou a atenção de forma especial: uma jovem parou no meio da pista, pegou seu *smartphone* e começou a mandar mensagens. Ela somente percebeu que estava no meio da rua quando um motorista de um ônibus que vinha em sua direção buzinou e, então, ela saiu."

De acordo com outro estudo, que realizou mineração de dados no Twitter, foi constatado que 18% dos *tweets* monitorados indicavam que o usuário estava jogando *PokémonGO* e dirigindo ao mesmo tempo [3].

Nota-se, então, que o uso de *smartphones* gera riscos de vida não somente ao dirigir um veículo, mas também no simples caminhar na rua, tanto para o portador do dispositivo quanto para aqueles ao seu redor.

O presente trabalho apresenta um método capaz de detectar padrões de deslocamento a partir do acelerômetro de um telefone celular, possibilitando, assim, a identificação de situações de risco de acidentes. O método proposto é capaz de identificar 3 modos de transporte: carro, bicicleta e a pé. Dependendo do modo de transporte, é possível identificar se o *smartphone* está sendo transportado em mãos, em um painel ou no bolso do usuário, identificando se ele está em movimento ou parado. Por fim, a abordagem aqui apresentada também identifica o modo de interação com o dispositivo: em ligação, olhando/digitando ou sem interação. É montado um classificador hierárquico que decide se a situação é de risco ou não. As situações que indicam risco são:

- usuário no carro com o celular em mãos, em movimento e em ligação;
- usuário no carro com o celular em mãos, em movimento e olhando/digitando;
- usuário em uma bicicleta com o celular em mãos, em movimento e olhando/digitando;
- usuário a pé com o celular em mãos, em movimento e em ligação e
- usuário a pé com o celular em mãos, em movimento e olhando/digitando.

Para que o processo de identificação de cada classe possa ser realizado, o acelerômetro do *smartphone* é o único sensor utilizado, não dependendo de nenhum outro sinal. Janelas de  $s$  segundos de cada um dos 3 eixos do acelerômetro são tomadas e, então, uma matriz é organizada e processada de forma a ser representada como uma imagem em níveis de cinza. A transformada de Fourier é aplicada sobre essa imagem e o resultado é organizado de forma a posicionar as menores frequências no centro da nova imagem. A partir dessa imagem e da faixa de frequência de interesse, o descritor de texturas GLCM é aplicado e são calculadas 4 características: o contraste, a correlação, a energia e a homogeneidade. Sendo assim, é formado um vetor de 12 valores para cada faixa de frequência, que é apresentado ao classificador hierárquico que, por sua vez, classifica os dados em "situação de risco" ou "não situação de risco" de acordo com a classe identificada.

O presente documento está organizado da seguinte forma: o Capítulo 2 apresenta os conceitos necessários ao entendimento desse documento, tais como as ideias básicas sobre acelerômetros, normalização, imagens em níveis de cinza e do descritor de texturas GLCM, bem como redes neurais artificiais. Ainda, para fins de comparação posterior, são apresentados trabalhos correlatos.

O Capítulo 3 apresenta as classes analisadas, a metodologia da construção da base de dados e o método de extração de características.

O Capítulo 4, por sua vez, apresenta a arquitetura do classificador hierárquico, o método de treinamento do sistema, a escolha das redes neurais e os resultados obtidos.

Finalmente, o Capítulo 5 realiza a comparação dos resultados aqui obtidos com outros trabalhos que representam o estado da arte. Também realiza uma breve discussão e descreve trabalhos futuros.

## 2 REVISÃO BIBLIOGRÁFICA

Antes da apresentação do método proposto, é necessário apresentar alguns conceitos. Nesta seção, será apresentada a fundamentação teórica: os conceitos sobre acelerômetros, normalização, as ideias básicas por trás das imagens em níveis de cinza e do descritor de texturas GLCM, bem como das redes neurais artificiais. Por fim, serão apresentados trabalhos correlatos.

### 2.1 FUNDAMENTAÇÃO TEÓRICA

#### 2.1.1 Acelerômetros

Há algum tempo, telefones celulares deixaram de ser apenas dispositivos de comunicação verbal. Na verdade, tais dispositivos evoluíram para os chamados *smartphones* e tornaram-se computadores portáteis com considerável capacidade de processamento e memória. Esses dispositivos, além de serem pequenos computadores, possuem diversos sensores embarcados, dentre os quais destacam-se os acelerômetros, que são capazes de estimar a aceleração do telefone em todas as direções no espaço tridimensional.

O valor da aceleração  $\vec{a}$  é expresso como um vetor que apresenta as componentes,  $a_{x_1}$ ,  $a_{x_2}$  e  $a_{x_3}$ , nos eixos  $x_1$ ,  $x_2$  e  $x_3$ , em unidades gravitacionais ( $g = 9,780327 \text{ m/s}^2$ ). Para efeito de esclarecimento, a Figura 2.1 apresenta o sistema de coordenadas adotado pelo sensor, sendo  $\hat{i}$ ,  $\hat{j}$  e  $\hat{k}$  os vetores unitários dos eixos  $x_1$ ,  $x_2$  e  $x_3$ , respectivamente. Em outras palavras, a aceleração é dada por  $\vec{a} = a_{x_1}\hat{i} + a_{x_2}\hat{j} + a_{x_3}\hat{k}$ . A orientação da aceleração em relação ao dispositivo é definida de tal forma que  $-1g$  é aplicada ao dispositivo no eixo  $x_3$  quando este encontra-se no estado *FaceUp* ou, de maneira análoga, quando o vetor unitário  $\hat{n}$  normal à tela do celular aponta na direção e sentido de  $\hat{k}$ .

#### 2.1.2 Normas e Normalização

Antes de apresentar métodos de normalização, é importante definir o cálculo das normas de um vetor.

Para qualquer  $\rho > 0$ , define-se  $\|\cdot\|_\rho$  como a norma de um vetor  $X$ . As normas  $\rho$  seguem a forma geral da Equação 2.1.

$$\|X\|_\rho = (|x_1|^\rho + |x_2|^\rho + \dots + |x_n|^\rho)^{\frac{1}{\rho}} \quad (2.1)$$

Sendo assim, as normas  $\rho$  de maior interesse são apresentadas nas equações 2.2, 2.3 e

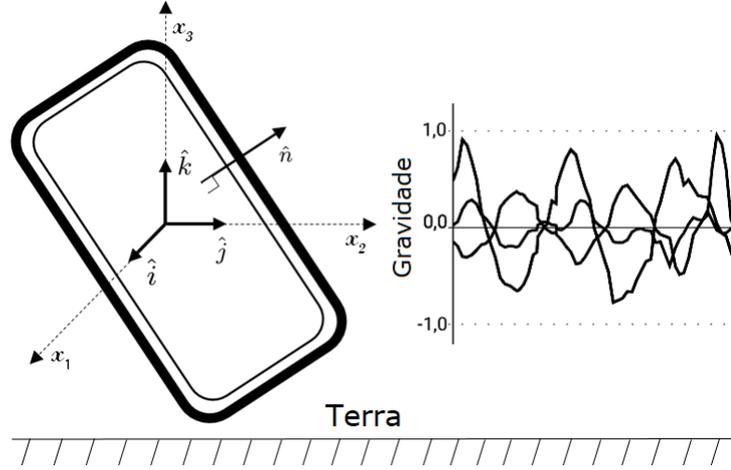


Figura 2.1: Sistema de coordenadas de referência utilizado. Os vetores  $\hat{i}$ ,  $\hat{j}$  e  $\hat{k}$  são os vetores unitários dos eixos  $x_1$ ,  $x_2$  e  $x_3$ , respectivamente, e  $\hat{n}$  é o vetor unitário normal à tela do *smartphone*. A aceleração é dada por  $\vec{a} = a_{x_1}\hat{i} + a_{x_2}\hat{j} + a_{x_3}\hat{k}$

2.4.

$$\|X\|_1 = |x_1| + |x_2| + \dots + |x_n| \quad (2.2)$$

$$\|X\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \dots + |x_n|^2} \quad (2.3)$$

$$\|X\|_\infty = \max_{1 \leq i \leq n} |x_i| \quad (2.4)$$

Para matrizes, há 2 normas que merecem atenção, apresentadas nas equações 2.5 e 2.6. Assume-se que  $M$  indica uma matriz de dimensões  $m \times n$  e  $\|M\|_\rho$  sua norma  $\rho$ .

$$\|M\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |M_{i,j}| \quad (2.5)$$

$$\|M\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |M_{i,j}| \quad (2.6)$$

Normalizar uma matriz significa remapear seus valores para outro domínio que pode ser generalizado. Há diferentes abordagens que realizam essa transformação. Para explicar o funcionamento de cada técnica, assume-se que  $M$  indica a matriz original e que  $\hat{M}$  indica a matriz normalizada.

A primeira abordagem consiste em dividir cada elemento de  $M$  pelo maior valor existente na matriz, conforme mostrado na Equação 2.7. Essa abordagem garante que o maior valor

existente na matriz normalizada será 1.

$$\hat{M}_{i,j} = \frac{M_{i,j}}{M_{max}}, \quad \forall i, j, \quad (2.7)$$

sendo  $M_{max}$  o maior valor encontrado em  $M$ .

Uma segunda abordagem consiste em dividir cada um dos elementos de uma matriz pela soma total de todos os seus elementos, conforme a Equação 2.8. Dessa forma, a soma de todos os elementos de  $\hat{M}$  resulta em 1.

$$\hat{M}_{i,j} = \frac{M_{i,j}}{\sum_{i,j} M_{i,j}}, \quad \forall i, j \quad (2.8)$$

Outra técnica consiste em dividir cada valor de  $M$  pela sua norma  $\|M\|$ , como descrito pela Equação 2.9.

$$\hat{M}_{i,j} = \frac{M_{i,j}}{\|M\|}, \quad \forall i, j \quad (2.9)$$

Uma quarta abordagem consiste em normalizar  $M$  de acordo com a Equação 2.10.

$$\hat{M}_{i,j} = \frac{M_{i,j} - M_{min}}{M_{max} - M_{min}}, \quad \forall i, j, \quad (2.10)$$

sendo  $M_{min}$  e  $M_{max}$  o menor e o maior valores encontrados em  $M$ . Dessa forma, garante-se que  $\hat{M}_{min} = 0$  e  $\hat{M}_{max} = 1$ .

### 2.1.3 Espaços de Cor RGB e YCbCr

Imagens coloridas são aquelas que carregam informação de cor. No espaço de cores *RGB* (do inglês, *Red, Green, Blue*), por exemplo, a imagem é definida a partir de três imagens componentes, cada qual responsável por registrar as intensidades de vermelho, verde e azul refletidas por uma cena. O *RGB*, porém, não é necessariamente o espaço mais eficiente para se representar cores. Sabe-se que o sistema visual humano é mais sensível à luminância (brilho) do que à cor. No entanto, o *RGB* não tira proveito desse fato. Nesse espaço de cores, a luminância está espalhada pelas componentes *R*, *G* e *B*, tornando-as igualmente relevantes. Uma outra representação muito popular, denominada *YCbCr*, separa a componente de luminância *Y* das componentes de cor, *C<sub>b</sub>* e *C<sub>r</sub>*, também chamadas de crominância. A componente *Y* pode ser vista como uma versão monocromática da imagem colorida. A conversão de uma imagem *RGB* em *YCbCr*, e vice-versa, pode ser realizada por meio das Equações 2.11 e 2.12, respectivamente. A principal vantagem da representação *YCbCr* é que as componentes *C<sub>b</sub>* e *C<sub>r</sub>* podem ser amostradas a uma resolução espacial mais baixa que a

componente  $Y$ , sem que haja uma queda significativa na qualidade visual da imagem.

$$\begin{aligned} Y &= 0,299 \times R + 0,587 \times G + 0,114 \times B \\ C_b &= 0,564 \times (B - Y) \\ C_r &= 0,713 \times (R - Y) \end{aligned} \quad (2.11)$$

$$\begin{aligned} R &= Y + 1,402 \times C_r \\ G &= Y - 0,344 \times C_b - 0,714 \times C_r \\ B &= Y + 1,772 \times C_b \end{aligned} \quad (2.12)$$

#### 2.1.4 Imagens em Níveis de Cinza

A representação de uma imagem pode ser feita por meio de duas abordagens: imagens coloridas ou imagens em níveis de cinza. Esta seção trata de imagens em níveis de cinza.

Cada píxel assume um valor que representa uma luminosidade, sendo o preto a menor intensidade e o branco, a maior intensidade. A quantidade de níveis de cinza representáveis depende da quantidade de bits disponíveis para representá-los. Quanto maior for a quantidade de bits, maior é a tendência à sensação de qualidade da imagem (conforme Figura 2.2), pois mais valores podem ser representados e, assim, aumenta-se o nível de detalhamento permitido. A Equação 2.13 mostra como o número de bits disponíveis e a quantidade de níveis de cinza  $L$  se relacionam.

$$L(b) = 2^b \quad (2.13)$$

, sendo  $b$  a quantidade de bits utilizada para representar um nível de cinza e  $L(b)$  a quantidade de níveis de cinza representáveis com  $b$  bits.

A relação entre uma imagem colorida e uma imagem em níveis de cinza se dá a partir de seus valores  $R$ ,  $G$  e  $B$  (componentes vermelho, verde e azul) da imagem colorida. Um píxel na posição  $(i, j)$  é transformado para uma intensidade luminosa  $NC(i, j)$  a partir de uma imagem  $RGB$  de acordo com a Equação 2.14. É importante notar que a escala de cinza também pode ser obtida a partir da componente  $Y$  do espaço de cor  $YC_bC_r$ , conforme Equação 2.11.

$$NC(i, j) = 0,299 \times R(i, j) + 0,587 \times G(i, j) + 0,114 \times B(i, j) \quad (2.14)$$

#### 2.1.5 Matriz de Co-ocorrência de Níveis de Cinza

A Matriz de Co-ocorrência de Níveis de Cinza [4] (do inglês, *Gray Level Co-occurrence Matrix* - GLCM) é um algoritmo descritor de texturas. Descritores de texturas são algoritmos



(a)



(b)



(c)



(d)



(e)



(f)

Figura 2.2: Conversão de uma imagem colorida para imagens com diferentes quantidades de níveis de cinza. É importante notar que, conforme a quantidade de níveis de cinza cai, o detalhamento da imagem também é perdido. (a) Imagem colorida original (RGB); (b) imagem com 8 bits e 256 níveis de cinza; (c) imagem com 4 bits e 16 níveis de cinza; (d) imagem com 3 bits e 8 níveis de cinza; (e) imagem com 2 bits e 4 níveis de cinza; e (f) imagem com 1 bit e 2 níveis de cinza.

que recebem uma imagem como entrada e extraem dela características capazes de distinguir aquela imagem de outras.

Por definição, a GLCM é uma matriz que indica a frequência com a qual diferentes combinações de valores de píxeis ocorrem em uma imagem, assumindo-se que os píxeis estão em níveis de cinza [5]. Esta matriz possui dimensões  $L \times L$ , onde  $L$  é a quantidade de níveis de cinza existentes na imagem de entrada.

Para melhor entendimento do funcionamento da técnica, a Figura 2.3, cujos valores encontram-se na Figura 2.4, será tomada como exemplo. É importante notar que há apenas 4 possíveis valores que cada píxel pode assumir.

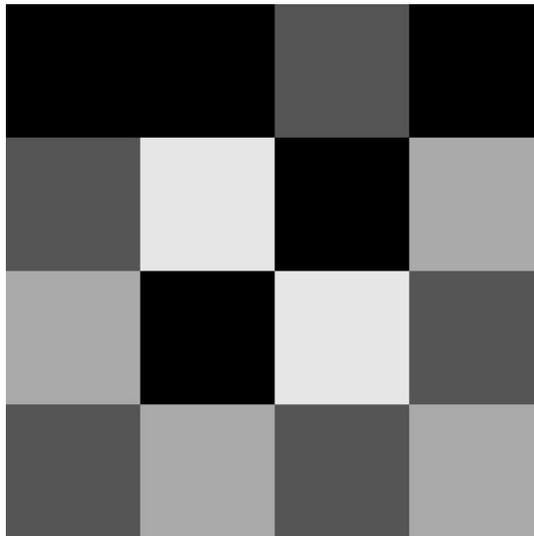


Figura 2.3: Imagem de exemplo em níveis de cinza que será utilizada para melhor entendimento do funcionamento da GLCM. Cada quadrado representa um píxel e há apenas quatro possíveis valores de intensidade: 0 (o mais escuro), 1, 2 e 3 (o mais claro).

	0	1	2	3
0	0	0	1	0
1	1	3	0	2
2	2	0	3	1
3	1	2	1	2

Figura 2.4: Valores relativos à Figura 2.3. Cada quadrado representa um píxel e há apenas quatro possíveis valores de intensidade: 0 (o mais escuro), 1, 2 e 3 (o mais claro).

A GLCM é utilizada para cálculos de segunda ordem [5], ou seja, são consideradas relações entre o valor de um píxel de referência ( $p_r$ ) e o valor de um píxel vizinho ( $p_{v(i,j)}$ ), onde  $(i, j)$  indica o deslocamento do píxel vizinho em relação ao píxel de referência. Por exemplo, tomando a posição  $(2, 2)$  na Figura 2.4 como referência, tem-se as seguintes relações:

$$\begin{aligned}
p_r &= 3 \\
p_{v(-1,-1)} &= 3 \\
p_{v(-1,0)} &= 0 \\
p_{v(-1,1)} &= 2 \\
p_{v(0,-1)} &= 0 \\
p_{v(0,1)} &= 1 \\
p_{v(1,-1)} &= 2 \\
p_{v(1,0)} &= 1 \\
p_{v(1,1)} &= 2,
\end{aligned}$$

sendo que o conjunto destes píxeis vizinhos forma a chamada *vizinhança de 8*, pois há 8 píxeis formando sua vizinhança.

O método consiste em gerar uma matriz que conta a quantidade de vezes em que o valor de um píxel vizinho ocorre em relação a um píxel de referência em determinada direção. Para exemplificar a construção de uma matriz, será considerado o píxel à direita do píxel de referência, ou seja,  $p_{v(0,1)}$ , conforme Figura 2.5.

A Figura 2.6 mostra a matriz GLCM gerada a partir da Figura 2.4, considerando  $p_{v(0,1)}$ . Os índices nas colunas representam os valores de  $p_{v(0,1)}$ , enquanto que os índices nas linhas representam os valores de  $p_r$ . Dessa maneira, a posição (1, 2) da GLCM indica a quantidade de vezes em que um píxel de valor 2 aparece à direita de um píxel de valor 1, o que ocorre duas vezes; a posição (0, 0) indica quantas vezes em que ocorre um 0 à direita de outro 0, ou seja, uma única vez; a posição (2, 1), por sua vez, possui valor 0, o que indica que não há nenhum píxel vizinho de valor 1 à direita de um píxel de referência de valor 2, e assim por diante.

	0	1	2	3
0	0	0	1	0
1	1	3	0	2
2	2	0	3	1
3	1	2	1	2

Figura 2.5: Valores relativos à Figura 2.3 com algumas posições em destaque, considerando apenas  $p_{v(0,1)}$ . Os píxeis em azul exemplificam píxeis de referência; píxeis em vermelho, de vizinhança. É importante notar que, ressaltados, há duas ocorrências de  $p_{v(0,1)} = 2$  para  $p_r = 1$  e uma ocorrência de  $p_{v(0,1)} = 0$  para  $p_r = 0$ .

A Figura 2.7 mostra todas as GLCMs formadas a partir da vizinhança de 8, tendo como base a Figura 2.3. As matrizes são obtidas considerando o píxel vizinho indicado no canto superior esquerdo de cada uma delas.

	0	1	2	3
0	1	1	1	1
1	1	0	2	1
2	1	1	0	0
3	1	1	0	0

Figura 2.6: Matriz GLCM construída a partir da Figura 2.3 considerando  $p_{v(0,1)}$ , cujos valores estão na Figura 2.4. Os índices nas abscissas representam os valores de  $p_{v(0,1)}$ , enquanto os índices nas ordenadas representam os valores de  $p_r$ .

$p_{v(-1,-1)}$	0	1	2	3	$p_{v(-1,0)}$	0	1	2	3	$p_{v(-1,1)}$	0	1	2	3	$p_{v(0,-1)}$	0	1	2	3
0	1	1	0	0	0	0	1	0	1	0	2	0	0	0	0	1	1	1	1
1	2	0	0	0	1	1	0	2	1	1	2	1	0	0	1	1	0	1	1
2	0	1	1	1	2	2	2	0	0	2	0	0	0	2	2	1	2	0	0
3	1	0	0	1	3	2	0	0	0	3	0	1	1	0	3	1	1	0	0
$p_{v(0,1)}$	0	1	2	3	$p_{v(1,-1)}$	0	1	2	3	$p_{v(1,0)}$	0	1	2	3	$p_{v(1,1)}$	0	1	2	3
0	1	1	1	1	0	2	2	0	0	0	0	1	2	2	0	1	2	0	1
1	1	0	2	1	1	0	1	0	1	1	1	0	2	0	1	1	0	1	0
2	1	1	0	0	2	0	0	0	1	2	0	2	0	0	2	0	0	1	0
3	1	1	0	0	3	0	0	2	0	3	1	1	0	0	3	0	0	1	1

Figura 2.7: As 8 matrizes GLCM resultantes da vizinhança de 8 da Figura 2.3. As matrizes são obtidas considerando o pixel vizinho indicado no canto superior esquerdo de cada uma.

A GLCM, por si só, não costuma ser utilizada diretamente por sistemas de classificação. Tais sistemas utilizam as métricas que são derivadas dessa matriz, dentre as quais se destacam o contraste, a correlação, a energia e a homogeneidade. Para os cálculos apresentados a seguir, assume-se que a GLCM está normalizada de acordo com a Equação 2.8.

#### 2.1.5.1 Contraste

Medição do contraste entre um píxel e seus vizinhos. É calculada por meio da Equação 2.15.

$$Con = \sum_{i,j} |i - j|^2 glcm(i, j) \quad (2.15)$$

#### 2.1.5.2 Correlação

Medição do quão correlacionado está um píxel em relação à sua vizinhança, conforme a Equação 2.16.

$$Cor = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j) glcm(i, j)}{\sigma_i \sigma_j} \quad (2.16)$$

#### 2.1.5.3 Energia

É a soma do quadrado de cada elemento da GLCM, conforme a Equação 2.17.

$$Ene = \sum_{i,j} glcm(i, j)^2 \quad (2.17)$$

#### 2.1.5.4 Homogeneidade

Medição da proximidade da distribuição dos elementos na GLCM em relação à sua própria diagonal, de acordo com a Equação 2.18.

$$Hom = \sum_{i,j} \frac{glcm(i, j)}{1 + |i - j|} \quad (2.18)$$

Utilizando a Figura 2.7, obtêm-se os respectivos valores da Figura 2.8.

### 2.1.6 Reconhecimento de Padrões

O ser humano é capaz de distinguir padrões utilizando-se de seus 5 sentidos ou combinações deles. É possível distinguir pessoas pela sua aparência física (visão), materiais pela sua textura (tato), perfumes de diversas marcas pelo seu cheiro (olfato), alimentos pelo seu

$P_{e(-1,-1)}$	Contraste: 1,5556    Correlação: 0,4743 Energia: 0,1358    Homogeneidade: 0,6389	$P_{e(-1,0)}$	Contraste: 3,75    Correlação: -0,6636 Energia: 0,1389    Homogeneidade: 0,3958	$P_{e(-1,1)}$	Contraste: 1    Correlação: 0,6582 Energia: 0,1852    Homogeneidade: 0,6481
$P_{e(0,-1)}$	Contraste: 3,25    Correlação: -0,3941 Energia: 0,0972    Homogeneidade: 0,4444	$P_e$		$P_{e(0,1)}$	Contraste: 3,25    Correlação: -0,3941 Energia: 0,0972    Homogeneidade: 0,4444
$P_{e(1,-1)}$	Contraste: 1    Correlação: 0,6582 Energia: 0,1852    Homogeneidade: 0,6481	$P_{e(1,0)}$	Contraste: 3,75    Correlação: 0,6636 Energia: 0,1389    Homogeneidade: 0,3958	$P_{e(1,1)}$	Contraste: 1,5556    Correlação: 0,4743 Energia: 0,1358    Homogeneidade: 0,6389

Figura 2.8: Valores de contraste, correlação, energia e homogeneidade obtidos a partir das GLCMs da Figura 2.7.

gosto (paladar) e até mesmo distinguir o tipo de motor em um carro pelo barulho que faz (audição).

Em cada um dos casos citados, o cérebro (um sistema classificador) realiza uma classificação de acordo com as características a ele fornecidas.

É natural que o ser humano deseje projetar e desenvolver máquinas capazes de reconhecer padrões. Dessa forma, surge a área de reconhecimento de padrões.

Reconhecimento de padrões é o ato de tomar dados como entrada e tomar uma ação baseada na categoria em que aquele padrão se encaixa [6].

Para melhor explicar o processo de reconhecimento de padrões, será utilizado o problema da flor Iris.

O problema consiste em classificar uma amostra de uma flor Iris em uma de 3 subcategorias, utilizando-se uma câmera: setosa, versicolor e virgínica.

Busca-se encontrar um modelo matemático capaz de realizar a distinção entre as 3 classes. Aquele que o fizer é tomado como uma solução para o problema.

Primeiramente, é necessário realizar uma etapa de pré-processamento, para retirar ruídos indesejáveis da imagem adquirida. Faz-se, ainda, a segmentação da imagem, ou seja, a área que contém a flor é separada do plano de fundo. Dessa forma, a entrada está pronta para a etapa de extração de características.

A etapa de extração de características consiste na realização de medições e cálculo de propriedades da flor em análise. As características obtidas são apresentadas a um classificador, que será responsável por dizer a qual classe aquela flor pertence.

Para o exemplo aqui apresentado, é utilizada a base de dados de [7], composta por 50 flores de cada classe. As características extraídas são:

1. comprimento da sépala;
2. largura da sépala;
3. comprimento da pétala e

#### 4. largura da pétala.

A Figura 2.9 mostra os histogramas gerados a partir de cada uma das características em separado, comparando as classes.

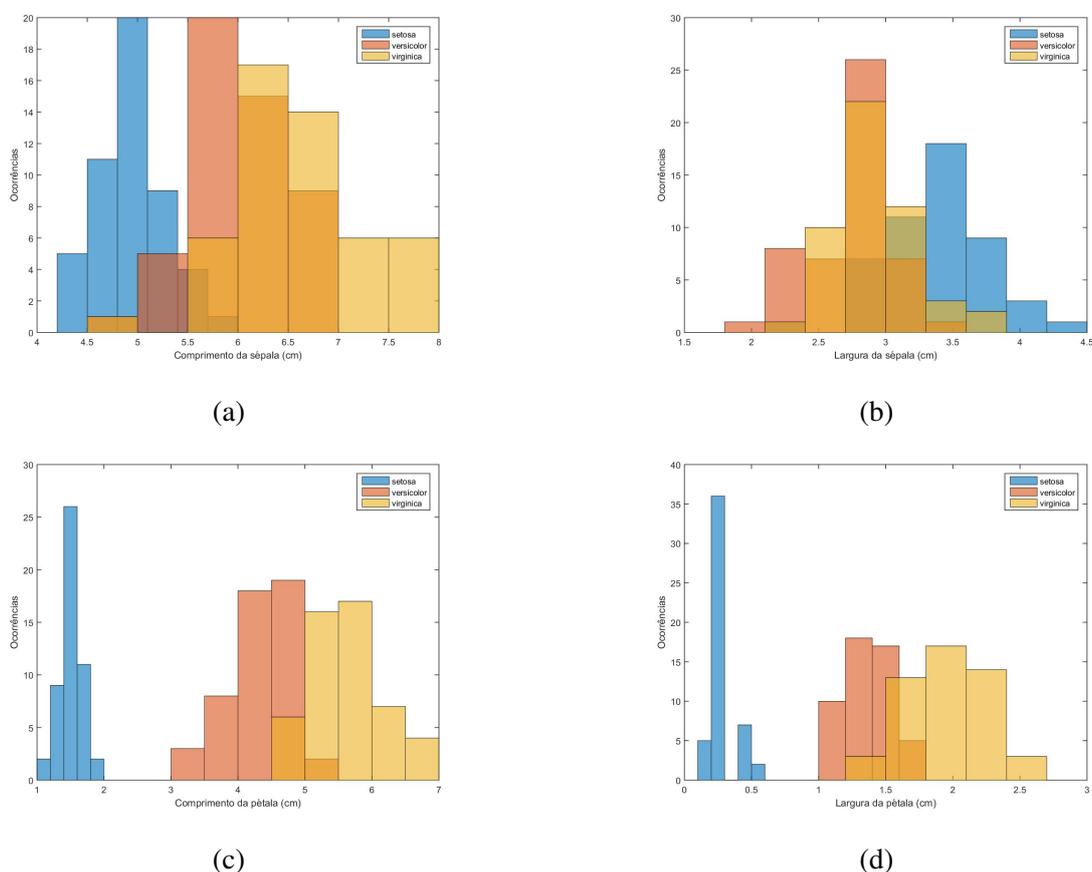


Figura 2.9: Histogramas comparativos entre 3 variações da flor Iris: setosa, versicolor e virgínica. Cada histograma compara uma característica diferente: (a) histograma comparativo do comprimento das sépalas; (b) histograma comparativo da largura das sépalas; (c) histograma comparativo do comprimento das pétalas; e (d) histograma comparativo da largura das pétalas.

É importante notar que, nas Figuras 2.9a e 2.9b, as características não são parâmetros com alta capacidade de distinção entre as classes, pois não é possível traçar linhas que separem as classes.

A Figura 2.9d torna possível a distinção da classe "setosa" bem clara. A Figura 2.9c, por sua vez, torna as distinções entre as 3 classes razoavelmente claras, porém não separa totalmente as classes "versicolor" e "virgínica".

Também é possível realizar a classificação a partir da combinação de 2 ou mais características ao mesmo tempo.

A Figura 2.10 exemplifica o uso de duas características plotadas em conjunto no mesmo

gráfico. As duas características são "comprimento da sépala" e "largura da pétala". Um classificador linear é capaz de reconhecer as situações em que a amostra é "setosa" ou "não setosa". Para distinguir entre as classes "versicolor" e "virgínica", entretanto, um classificador linear não é suficiente.

Para resolver esse tipo de problema, utilizam-se classificadores mais potentes, por exemplo, redes neurais artificiais, o que é descrito na Seção 2.1.7.

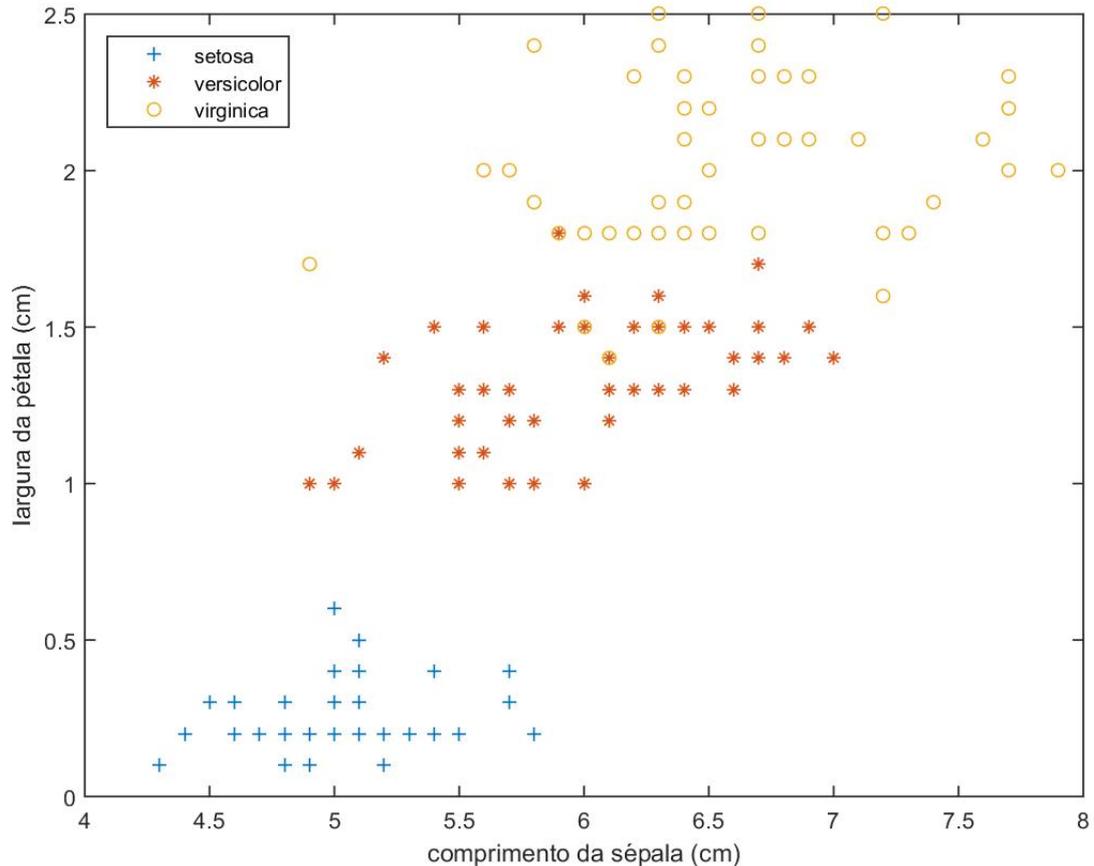


Figura 2.10: Uso das características "comprimento da sépala" e "largura da pétala" em conjunto. É importante notar que, apesar da classe "setosa" ser completamente distinguível, há sobreposição dos conjuntos "versicolor" e "virgínica" em alguns pontos. Dessa forma, um simples classificador linear não é suficiente.

### 2.1.7 Redes Neurais Artificiais

Um dos grandes desafios na computação é a elaboração de métodos capazes de classificar elementos de acordo com certos padrões. Na maioria das vezes, essa é uma atividade que não oferece grandes dificuldades quando realizada por seres humanos. Entretanto, há situações que são mais complexas e que envolvem uma grande quantidade de elementos, podendo

tornar o trabalho de classificação manual extremamente árduo. Uma solução para essas situações é o uso de computadores.

Há diversas abordagens voltadas à classificação. Dentre elas, destacam-se as redes neurais artificiais [8].

Redes neurais artificiais surgiram como uma tentativa de se imitar computacionalmente as redes neurais biológicas presentes no sistema nervoso central humano, estas consistem em um arranjo de neurônios interconectados que trocam informação entre si. Dessa forma, permitem a detecção, o aprendizado e a aplicação de padrões. Ademais, as redes neurais artificiais são muito utilizadas em áreas ligadas à inteligência artificial, principalmente para resolver problemas de classificação, de ajuste de função (regressão), de robótica, de controle etc.

Para compreender melhor o funcionamento de uma rede neural artificial, é preciso, primeiro, entender sua estrutura mais básica: o neurônio.

#### 2.1.7.1 Neurônios Artificiais

Assim como em uma rede neural biológica, as redes neurais artificiais são compostas por neurônios, que visam modelar o funcionamento de um neurônio real até certo ponto. Um neurônio possui três estruturas básicas: o corpo celular, os dendritos e o axônio. O corpo celular contém o núcleo da célula e é responsável por praticamente toda a produção de proteínas e membranas, tal como a respiração celular. Em suma, o corpo celular é responsável pelas funções vitais do neurônio. Os dendritos e os axônios são responsáveis por realizar a conexão entre neurônios. Dendritos se conectam apenas a axônios; axônios, apenas a dendritos. Os dendritos são responsáveis por receber os impulsos provenientes de outros neurônios e transmiti-los para o corpo celular, que processa essas informações. Após o processamento das informações, o corpo celular excita o axônio, que transmite novas informações aos dendritos do neurônio conectado à ele. É assim que os impulsos nervosos são transmitidos e processados no cérebro humano.

Analogamente, um neurônio artificial tem seu comportamento dividido em três partes: núcleo, entrada e saída. Entrada e saída são, respectivamente, dendritos e axônio, tendo a função de apenas conectar um neurônio a outro. O núcleo é o responsável por realizar os cálculos. A Figura 2.11 ilustra a estrutura de um neurônio artificial.

A forma com a qual um neurônio define sua saída em função de sua entrada será melhor detalhada mais adiante. Neste ponto, entretanto, é necessário esclarecer quais tipos de cálculos são realizados. A entrada  $p$  de um neurônio pode ser composta por um único valor escalar ou por um vetor de  $R$  elementos. Então, cada elemento de  $p$  é multiplicado por um peso  $e$ , depois, um deslocamento  $b$  (sempre um valor escalar) é adicionado. O conjunto contendo também  $R$  pesos é denotado por  $w$ . Nesse ponto, o resultado é submetido a uma

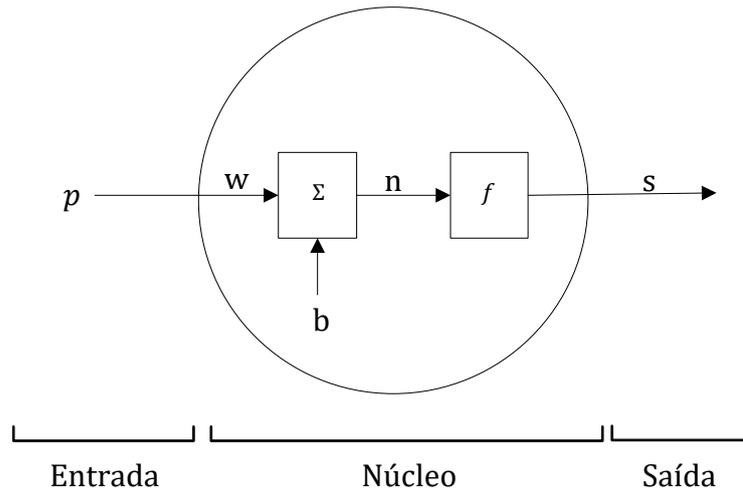


Figura 2.11: Estrutura básica de um neurônio artificial.

função (que será chamada de função de ativação) que determina sua saída  $s$  (também um escalar). Tomando a Figura 2.11 como referência, têm-se que o valor intermediário  $n$  é definido conforme a Equação 2.19.

$$n = \left( \sum_{i=1}^R p_i \times w_i \right) + b \quad (2.19)$$

A partir de  $n$ , é definida a saída  $s$  conforme a Equação 2.20.

$$s = f(p \cdot w + b) = f(n), \quad (2.20)$$

onde  $f(n)$  é a função de ativação, conceito que ainda será apresentado.

Pode-se adotar diversas funções para a função de ativação  $f(\cdot)$ , que serão apresentadas a seguir [9, 10].

### 2.1.7.1.1 Funções de Ativação

Independente da função de ativação escolhida, o neurônio segue o mesmo modelo básico descrito pela Figura 2.11 e pelas equações 2.19 e 2.20, recebendo uma entrada  $p$  e fornecendo uma saída  $s$ .

A função degrau é a função de ativação mais simples utilizada. A saída  $s$  é composta por apenas um único valor, 0 ou 1. O neurônio precisa ser configurado com um parâmetro, indicado por  $T$ . Dessa forma, define-se a função de ativação  $f(\cdot)$  do neurônio conforme a

Equação 2.21.

$$f(n) = u(n - T) = \begin{cases} 1, & \text{se } n - T \geq 0 \\ 0, & \text{se } n - T < 0 \end{cases} \quad (2.21)$$

, sendo  $u(n)$  a função degrau.

Por exemplo, dado o conjunto de entradas  $p = [1, 0, 1]$ , o conjunto de pesos  $w = [6, 7, 5]$ , um deslocamento  $b = 0$  e um limiar  $T = 15$ , a saída do neurônio que utiliza a função de ativação degrau seria  $s = 0$ , como demonstrado na Equação 2.22.

$$\begin{aligned} n &= \left( \sum_{i=1}^3 p_i \times w_i \right) + b = 1 \times 6 + 0 \times 7 + 1 \times 5 + 0 = 11 \\ \implies u(n - T) &= u(11 - 15) \\ \implies s &= 0 \end{aligned} \quad (2.22)$$

Sendo assim, diz-se que o neurônio não foi excitado.

Apesar da simplicidade da função degrau, o mais comum é que outros modelos sejam utilizados. Saídas puramente binárias podem prejudicar a precisão do sistema. Pequenas alterações nos pesos poderiam causar uma mudança brusca na saída do neurônio. Assim, deseja-se uma solução na qual pequenas variações nos valores dos pesos e do deslocamento causem pequenas variações na saída.

Ao adotar uma função que permita saídas entre 0 e 1, conseqüentemente o neurônio também fornece saídas entre 0 e 1. Finalmente, tem-se um modelo que permite inserir valores reais e que respondem com valores reais.

Define-se a função de ativação logística como na forma descrita na Equação 2.23.

$$\sigma(p) = f(p) = \frac{1}{1 + e^{-(\sum_{i=1}^R p_i \times w_i) + b}} \quad (2.23)$$

Para tornar a notação mais compacta, a Equação 2.23 pode ser reescrita na forma da Equação 2.24.

$$\sigma(n) = f(n) = \frac{1}{1 + e^{-n}} \quad (2.24)$$

É interessante realizar uma análise rápida do comportamento de  $\sigma(n)$  em alguns pontos. Quando  $n$  tende ao infinito, tem-se o limite da Equação 2.25.

$$\lim_{n \rightarrow \infty} \sigma(n) = \lim_{n \rightarrow \infty} \frac{1}{1 + e^{-n}} = 1 \quad (2.25)$$

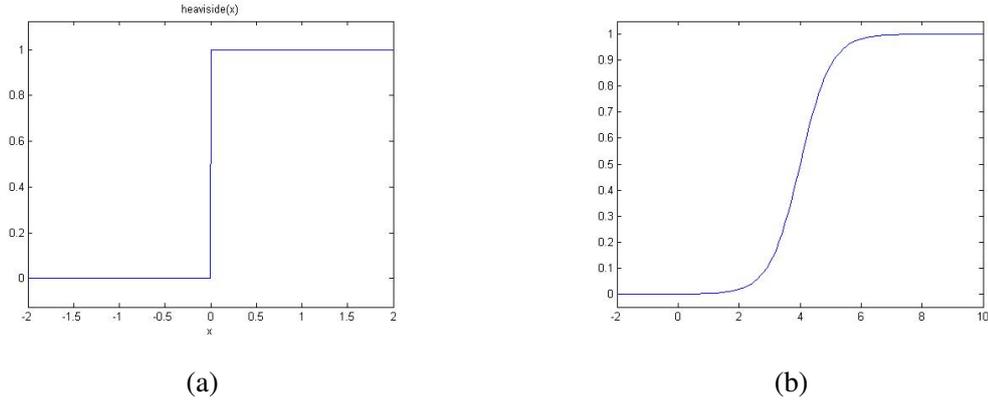


Figura 2.12: Funções de ativação (a) degrau e (b) logística. É importante notar que a função logística pode ser tratada como uma versão suavizada da função degrau.

Quando  $n$  tende ao menos infinito, tem-se o limite da Equação 2.26.

$$\lim_{n \rightarrow -\infty} \sigma(n) = \lim_{n \rightarrow -\infty} \frac{1}{1 + e^{-n}} = 0 \quad (2.26)$$

Nos extremos, o comportamento de  $\sigma(n)$  e de  $u(n)$  é praticamente o mesmo. É importante notar, então, que a função de ativação logística (também conhecida como log-sigmoide) consiste em uma versão suavizada da função de ativação degrau. A Figura 2.12 ilustra as duas funções de ativação.

Retomando o exemplo anterior, a Equação 2.27 demonstra a saída de um neurônio cuja função de ativação utilizada é a logística.

$$\begin{aligned} n &= \left( \sum_{i=1}^3 p_i \times w_i \right) + b = 1 \times 6 + 0 \times 7 + 1 \times 5 + 0 = 11 \\ \implies \sigma(n) &= \sigma(11) = \frac{1}{1 + e^{-11}} \\ \implies s &= 0.99 \end{aligned} \quad (2.27)$$

Como é possível notar, as funções de ativação são responsáveis por definir a maneira com a qual o neurônio irá tratar as entradas. Se de forma binária, se com variações mais suaves ou mais abruptas. Há diversas funções que podem ser utilizadas como funções de ativação para neurônios artificiais, entretanto, além das já citadas, convém citar pelo menos mais duas delas, geralmente utilizadas em redes de múltiplas camadas [11]. A Figura 2.13 ilustra outras duas funções de transferência muito utilizadas: a tangente hiperbólica e a linear.

A tangente hiperbólica apresenta o formato descrito na Equação 2.28.

$$\tanh(n) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \quad (2.28)$$

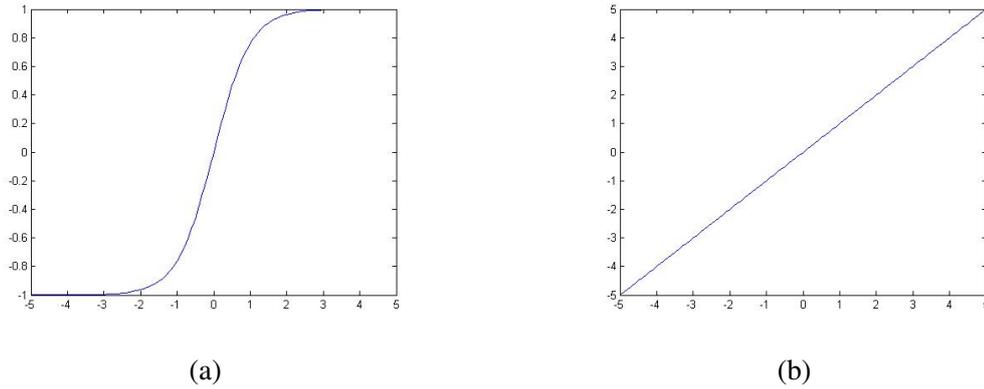


Figura 2.13: Duas funções de ativação bastante utilizadas: (a) tangente hiperbólica e (b) função linear.

Dada sua simplicidade, a função de ativação linear não será apresentada, porém vale a pena mencionar que a tangente hiperbólica, caso seu declive seja adequadamente ajustado, pode vir a se assemelhar à uma função linear.

É importante notar que a saída de um neurônio pode ser configurada para abranger o intervalo de saída desejado, não se limitando apenas ao intervalo  $[0, 1]$ . Isso é feito por meio da calibração dos parâmetros do neurônio e pela escolha da função de ativação.

Apesar de tudo, é fácil perceber que os neurônios artificiais ainda não reproduzem o comportamento complexo de um neurônio real. Não obstante, a modelagem computacional atingida até o presente momento tem demonstrado grandes avanços.

#### 2.1.7.2 Redes Neurais Artificiais *Feed-forward*

Tendo definido o modelo de neurônio a ser utilizado, uma rede neural é formada por meio da ligação entre neurônios. Na Figura 2.14, notam-se  $n + 1$  colunas de neurônios. Cada uma dessas colunas forma o que é chamado de camada. Cada camada, com exceção da última, fornece novos valores de entrada para a seguinte. Diz-se que o conjunto de entradas inicial para a rede forma uma camada, a camada de entrada. Ela não realiza qualquer tipo de computação, sendo formada simplesmente pelos valores de entrada para a rede. A última camada é chamada de camada de saída; as demais, de camadas ocultas. Redes neurais podem ter uma ou múltiplas camadas.

Por se tratar do escopo deste trabalho, apenas um tipo de rede neural será abordado neste texto.

Uma rede neural artificial *feed-forward* é um tipo de rede neural com as seguintes características:

- possui pelo menos duas camadas;

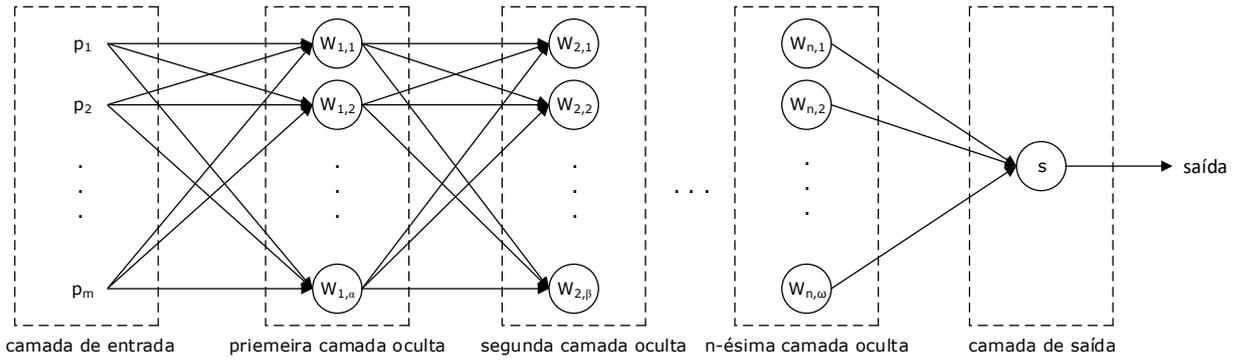


Figura 2.14: Arquitetura de uma rede neural artificial *feed-forward* de  $n$  camadas ocultas.  $p_1, p_2, \dots, p_m$  representam as  $m$  entradas da rede neural, enquanto que cada  $W_{i,j}$  indica o  $j$ -ésimo neurônio da  $i$ -ésima camada. É importante notar que as quantidades de neurônios  $\alpha, \beta, \dots, \omega$  não são necessariamente iguais.

- cada neurônio de uma camada se conecta com todos os neurônios da camada seguinte e não se comunica com camadas anteriores e
- não há conexão entre neurônios de uma mesma camada.

Assim, uma rede *feed-forward* não possui *loops* nem realimentação de neurônios. Os dados fluem a partir da entrada da rede sempre em direção à saída. Este tipo de rede é muito utilizado na resolução de problemas de reconhecimento de padrões.

Há um tipo de rede *feed-forward* que possui *loops*, conhecidas como redes neurais recorrentes, porém seus algoritmos de aprendizagem não são tão robustos [9] e, por isso, essas redes não são tão utilizadas.

Para a rede da Figura 2.15, a propagação se inicia com o cálculo do chamado campo local induzido  $u_j^n(m)$  para cada neurônio  $j$  da camada  $n$  segundo a Equação 2.29, onde  $v_i^{n-1}(m)$  é o elemento  $i$  do sinal de saída da camada  $n - 1$  na iteração  $m$ ,  $w_{ji}^n(m)$  é o valor da sinapse que vem do elemento  $i$  da camada  $n - 1$  para o neurônio  $j$  da camada  $n$ ,  $R_{n-1}$  é o número de elementos do vetor de saída da camada  $n - 1$ .

$$u_j^n(m) = \sum_{i=0}^{R_{n-1}} w_{ji}^n(m) v_i^{n-1}(m) \quad (2.29)$$

Se  $i = 0$ , então  $v_0^{n-1}(m) = +1$  e  $w_{j0}^n = b_j^n(m)$ , onde  $b_j^n(m)$  é o viés do neurônio  $j$  da camada  $n$ , cuja saída é dada pela Equação 2.30, onde  $f_j^n$  é a função de transferência do neurônio  $j$  da camada  $n$ .

$$v_j^n(m) = f_j^n(u_j^n(m)) \quad (2.30)$$

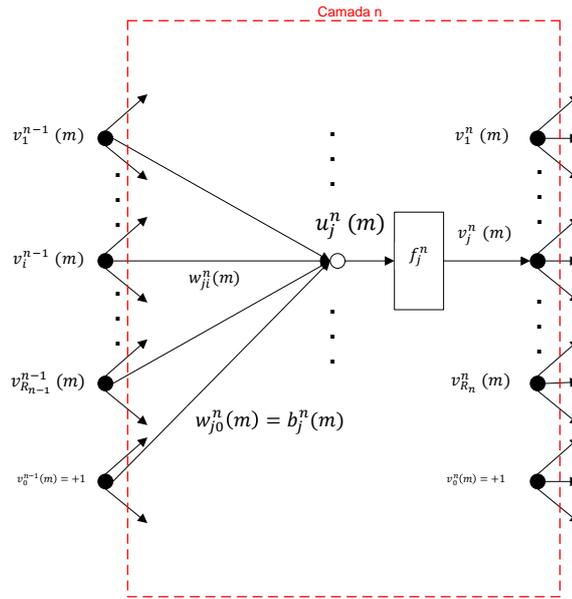


Figura 2.15: Modelo de componentes de uma camada  $n$  de uma rede neural. O campo local induzido é representado por  $u_j^n(m)$  para cada neurônio  $j$  da camada  $n$ ,  $v_i^{n-1}(m)$  é o elemento  $i$  do sinal de saída da camada  $n - 1$  na iteração  $m$ ,  $w_{ji}^n(m)$  é o valor da sinapse que vem do elemento  $i$  da camada  $n - 1$  para o neurônio  $j$  da camada  $n$ ,  $R_n$  é o número de elementos do vetor de saída da camada  $n$  e  $b_j^n(m)$  é o viés do neurônio  $j$  da camada  $n$ .

Com o cálculo das saídas  $v$  dos neurônios da última camada, a primeira etapa foi concluída, que é processo de propagação do sinal, restando agora realizar a segunda etapa, que é a retropropagação do erro e, conseqüentemente, realizar os ajustes das sinapses e vieses.

### 2.1.7.3 Treinamento

Para fazer uso de uma rede neural, é recomendado definir 3 conjuntos de dados mutuamente excludentes, ou seja, os elementos de cada conjunto devem pertencer somente a este conjunto, não estando presentes em nenhum outro. Definem-se:

- **Treinamento:** esse conjunto deve ser utilizado para calibrar/treinar a rede, definindo os pesos  $w$  e deslocamentos  $b$ .
- **Validação:** esse conjunto é utilizado para evitar que a rede fique viciada (ocorrência de *overfitting*) no conjunto de treinamento. Uma vez que a rede está sendo treinada, o conjunto de validação é submetido à rede a cada treinamento. Quando a taxa de erro sobre o conjunto de treinamento e a taxa de erro sobre o conjunto de validação se distanciam acima de um certo limiar, o treinamento é interrompido, pois houve indicação de que a rede iria começar a ficar viciada no conjunto de treinamento e tendendo a errar entradas não pertencentes a esse conjunto.
- **Teste:** é o conjunto de elementos que realmente serão colocados à prova no sistema,

ou seja, é o conjunto de dados que gera as métricas finais da rede.

Isto é, o primeiro conjunto treina a rede, o segundo conjunto evita que ela fique viciada e o terceiro é utilizado para verificar o desempenho da rede.

Dois paradigmas de treinamento merecem atenção:

- **Supervisionado:** esse paradigma consiste em fornecer os elementos em conjunto com um gabarito. Durante o treinamento, a rede neural tem acesso tanto aos elementos quanto à classe na qual eles devem ser classificados.
- **Não-supervisionado:** o paradigma não-supervisionado, por sua vez, somente tem acesso aos dados de entrada e visa separá-los em diferentes classes de forma a distingui-los.

Por fim, é necessário definir um método para calcular os valores ótimos para os pesos  $w$  e deslocamentos  $b$ .

Dentro de cada paradigma, há diferentes técnicas de treinamento. Um dos algoritmos mais utilizados para treinamento supervisionado é o *backpropagation*. Ele consiste em tentar reduzir o erro entre as saídas obtidas e as desejadas. Para simplificar, pode-se dizer que isso é feito por meio do cálculo da variação do erro (gradiente) ao aumentar ou diminuir os pesos. Portanto, ao perceber que a taxa está aumentando demais, o treinamento é terminado e os pesos que geraram os menores erros são adotados.

Ainda, há outras técnicas que podem resultar em redes mais precisas. As técnicas de Levenberg-Marquardt, que consiste em uma variação do *backpropagation* mais eficiente [12, 13], e *BFGS Quasi-Newton* são as mais rápidas, apesar de tenderem a consumir mais memória. Para treinamentos com grandes bases de dados, *Scaled Conjugate Gradient* e *Resilient Backpropagation* costumam ser mais eficientes e consomem menos memória [11]. A escolha de qual método utilizar depende de cada caso e deve ser pensada com cautela.

Por não se tratar do escopo do presente trabalho, o paradigma não-supervisionado não será abordado.

### 2.1.7.3.1 Algoritmo de Treinamento *Backpropagation*

O erro  $e_j^n(m)$  de um neurônio  $j$  de uma camada  $n$  é calculado através da subtração do resultado esperado  $t_j^n(m)$  pelo resultado obtido pelo processo de propagação do sinal  $v_j^n(m)$ , conforme Equação 2.31.

$$e_j^n(m) = t_j^n(m) - v_j^n(m) \quad (2.31)$$

Calculado o erro  $e_j^n(m)$ , é necessário retropropagá-lo ajustando as sinapses e vieses da rede. O primeiro da retropropagação dos erros consiste em calcular os gradientes locais para cada neurônio.

Para cada neurônio da camada  $n$ , os gradientes são dados pela Equação 2.33.

$$\delta_j^n(m) = f'_j(v_j^n(m)) \sum_{k=0}^{R_{n+1}} \delta_k^{n+1}(m) w_{kj}^{n+1}(m) \quad (2.32)$$

Para o caso em que a camada  $n$  é a camada de saída, se segue a Equação 2.32.

$$\delta_j^n(m) = f'_j(v_j^n(m)) e_j^n(m) \quad (2.33)$$

Uma vez calculados os gradientes locais, estes são utilizados para calcular a atualização das sinapses e vieses. A atualização dos valores das sinapses que ligam os neurônios da camada  $n - 1$  aos neurônios da camada  $n$  é realizada segundo a Equação 2.34.

$$w_{ji}^n(m+1) = w_{ji}^n(m) + \eta \cdot v_i^{n-1}(m) \delta_j^n(m) \quad (2.34)$$

Finalmente, os vieses são atualizados conforme a Equação 2.35.

$$b_j^n(m+1) = b_j^n(m) + \eta \cdot \delta_j^n(m) \quad (2.35)$$

O termo  $\eta$  é a taxa de aprendizagem e  $f'(*)$  é a diferenciação em relação ao argumento. Para que a rede neural melhore sua performance é necessário que vários conjuntos sejam inseridos na camada de entrada e que a propagação e a retropropagação ocorram diversas vezes. Por fim, um critério de parada deve ser criado, visto que o resultado ideal pode não ser encontrado. Um exemplo de critério de parada que pode ser utilizado se baseia na taxa absoluta da variação do erro médio quadrático para um conjunto de entrada, indicando que a parada deve ocorrer quando um valor mínimo desejado for alcançado.

Para melhor entendimento, será utilizado um exemplo numérico de uma iteração. É tomada a inicialização da rede neural apresentada na Figura 2.16, sendo que  $x_i$  indica a  $i$ -ésima entrada da rede,  $\tilde{y}_i$  indica a  $i$ -ésima saída calculada pela rede e  $N_i^n$  indica o  $i$ -ésimo neurônio da camada  $n$ . A função de ativação dos neurônios  $N_1^1, N_2^1, N_3^1$  e  $N_2^2$  é a logística sigmoide; do neurônio  $N_1^2$ , uma função linear. O objetivo é treinar a rede de tal forma que, para as entradas da Equação 2.36, as saídas  $\tilde{y}_1$  e  $\tilde{y}_2$  assumam os valores desejados  $y_1$  e  $y_2$ , apresentados na Equação 2.37.

$$\begin{aligned} x_1 &= 0,1 \\ x_2 &= 0,7 \end{aligned} \quad (2.36)$$

$$\begin{aligned} y_1 &= 0,2 \\ y_2 &= 1 \end{aligned} \quad (2.37)$$

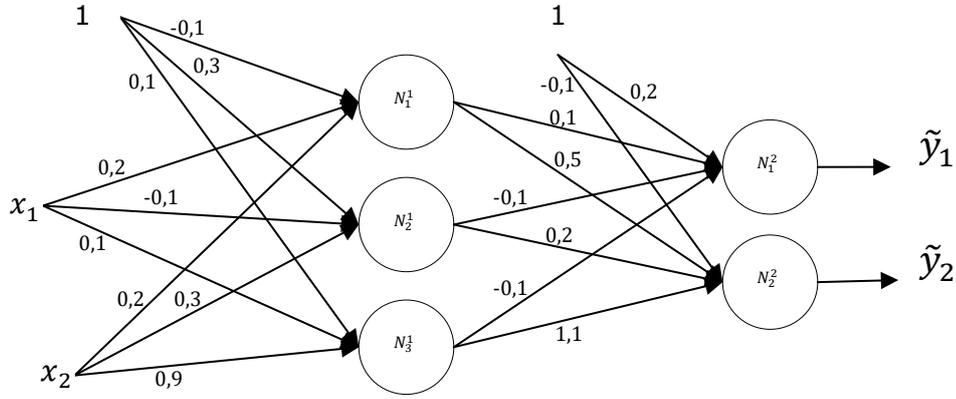


Figura 2.16: Rede neural com sinapses inicializadas em valores aleatórios.  $x_i$  indica a  $i$ -ésima entrada da rede,  $\tilde{y}_i$  indica a  $i$ -ésima saída da rede e  $N_i^n$  indica o  $i$ -ésimo neurônio da camada  $n$ . A função de ativação dos neurônios  $N_1^1, N_2^1, N_3^1$  e  $N_2^2$  é a logística sigmoide; do neurônio  $N_1^2$ , uma função linear.

Calculando os valores propagados na primeira camada, obtém-se os resultados descritos na Equação 2.38.

$$\begin{aligned} u_1^1 &= -0,1 + (0,2) \cdot (0,1) + (0,2) \cdot (0,7) = 0,06 \implies v_1^1 = \tanh(0,06) = 0,06 \\ u_2^1 &= 0,3 + (-0,1) \cdot (0,1) + (0,3) \cdot (0,7) = 0,5 \implies v_2^1 = \tanh(0,5) = 0,46 \\ u_3^1 &= 0,1 + (0,1) \cdot (0,1) + (0,9) \cdot (0,7) = 0,74 \implies v_3^1 = \tanh(0,74) = 0,63 \end{aligned} \quad (2.38)$$

Calculando os valores propagados na segunda camada, a Equação 2.39 demonstra os resultados obtidos.

$$\begin{aligned} u_1^2 &= 0,2 + (0,1) \cdot (0,06) + (-0,1) \cdot (0,46) + (-0,1) \cdot (0,63) = 0,097 \implies v_1^2 = 0,097 \\ u_2^2 &= -0,1 + (0,5) \cdot (0,06) + (0,2) \cdot (0,46) + (1,1) \cdot (0,63) = 0,715 \implies v_2^2 = \tanh(0,715) = 0,614 \end{aligned} \quad (2.39)$$

Dessa forma, as saídas da rede neural são obtidas por meio da Equação 2.40.

$$\begin{aligned} \tilde{y}_1 &= v_1^2 = 0,097 \\ \tilde{y}_2 &= v_2^2 = 0,614 \end{aligned} \quad (2.40)$$

Comparando os valores calculados pela rede com os valores desejados, são obtidos os erros conforme Equação 2.41, onde  $\epsilon_i$  indica o erro alcançado na  $i$ -ésima saída da rede.

$$\begin{aligned}\epsilon_1 &= y_1 - \tilde{y}_1 = 0,2 - 0,097 = 0,103 \\ \epsilon_2 &= y_2 - \tilde{y}_2 = 1 - 0,614 = 0,386\end{aligned}\quad (2.41)$$

Antes da próxima etapa, é necessário calcular as derivadas das funções de ativação nos pontos apropriados. Sabendo que  $\frac{d}{dv}v = 1$  e que  $\frac{d}{dv}\tanh(v) = 1 - v^2$ , temos os valores da Equação 2.42.

$$\begin{aligned}f'_{1^1}(v_1^1) &= 1 - 0,06^2 = 1 \\ f'_{2^1}(v_2^1) &= 1 - 0,46^2 = 0,79 \\ f'_{3^1}(v_3^1) &= 1 - 0,63^2 = 0,6 \\ f'_{1^2}(v_1^2) &= 1 \\ f'_{2^2}(v_2^2) &= 1 - 0,61^2 = 0,63\end{aligned}\quad (2.42)$$

Após o cálculo das derivadas das funções de ativação, é possível calcular cada  $\delta_i^n$ , como mostra a Equação 2.43.

$$\begin{aligned}\delta_1^2 &= f'_{1^2} \cdot \epsilon_1^2 = 1 \cdot 0,103 = 0,103 \\ \delta_2^2 &= f'_{2^2} \cdot \epsilon_2^2 = 0,63 \cdot 0,386 = 0,24 \\ \delta_1^1 &= f'_{1^1} \cdot [(\delta_1^2) \cdot (w_{11}^2) + (\delta_2^2) \cdot (w_{21}^2)] = 1 \cdot [(0,103) \cdot (0,1) + (0,24) \cdot (0,5)] = 0,130 \\ \delta_2^1 &= f'_{2^1} \cdot [(\delta_1^2) \cdot (w_{12}^2) + (\delta_2^2) \cdot (w_{22}^2)] = 0,79 \cdot [(0,103) \cdot (-0,1) + (0,24) \cdot (0,2)] = 0,03 \\ \delta_3^1 &= f'_{3^1} \cdot [(\delta_1^2) \cdot (w_{13}^2) + (\delta_2^2) \cdot (w_{23}^2)] = 0,6 \cdot [(0,103) \cdot (-0,1) + (0,24) \cdot (1,1)] = 0,152\end{aligned}\quad (2.43)$$

Para a atualização das sinapses, segue-se a Equação 2.44.

$$\Delta w_{ji}^n = 2 \cdot \alpha \cdot v_j^{n-1} \cdot \delta_i^n \quad (2.44)$$

Adotando o valor  $\alpha = 0,1$ , serão calculados os novos valores de duas sinapses ( $w_{32}^1$  e  $w_{11}^2$ ) e dois vieses ( $b_1^2$  e  $b_1^1$ ), fazendo uso das Equações 2.35 e 2.44, conforme Equação 2.45.

$$\begin{aligned}\Delta b_1^2 &= 2 \cdot \alpha \cdot v_0^2 \cdot \delta_1^2 = 2 \cdot (0,1) \cdot 1 \cdot (0,103) = 0,021 & \implies b_1^2 &= 0,2 + 0,021 = 0,221 \\ \Delta b_1^1 &= 2 \cdot \alpha \cdot v_0^1 \cdot \delta_1^1 = 2 \cdot (0,1) \cdot 1 \cdot (0,130) = 0,026 & \implies b_1^1 &= -0,1 + 0,026 = -0,074 \\ \Delta w_{11}^2 &= 2 \cdot \alpha \cdot v_1^1 \cdot \delta_1^2 = 2 \cdot (0,1) \cdot (0,06) \cdot (0,103) = 0,001 & \implies w_{11}^2 &= 0,1 + 0,001 = 0,101 \\ \Delta w_{32}^1 &= 2 \cdot \alpha \cdot v_2^0 \cdot \delta_3^1 = 2 \cdot (0,1) \cdot (0,7) \cdot (0,152) = 0,021 & \implies w_{32}^1 &= 0,9 + 0,021 = 0,921\end{aligned}\quad (2.45)$$

### 2.1.7.3.2 Algoritmo de Treinamento de Levenberg-Marquardt

Há diversos algoritmos utilizados para o treinamento de redes neurais. Entre eles, o algoritmo *Error Backpropagation* pode ser considerado um dos avanços mais significativos no treinamento de redes neurais. Apesar das diversas e pequenas alterações ao longo do tempo, seu maior problema está na lentidão da convergência. Isso ocorre devido a duas razões [14]:

1. Ao tamanho do passo do algoritmo, que deveria se adaptar adequadamente aos gradientes. Quando mais ingreme for o gradiente, menores deveriam ser os passos e
2. À curvatura da superfície de erro, que pode não ser a mesma em todas as direções, o que pode resultar no problema do vale de erro, resultando em um lento processo de convergência.

O algoritmo de Gauss-Newton, por sua vez, apresenta uma grande melhoria na velocidade de convergência. Esse algoritmo é capaz de encontrar passos de tamanhos apropriados para cada direção e, assim, convergir mais rapidamente. Quando a função a ser minimizada se trata de uma superfície quadrática, ele pode convergir até mesmo na primeira iteração. Tal melhoria só ocorre em situações nas quais a função de erro é moderada, caso contrário o algoritmo tende a divergir [14].

Surge, então, o método de Levenberg-Marquardt [15, 16], nome atribuído devido a dois pesquisadores que desenvolveram o mesmo método independentemente (Levenberg, em 1944, e Marquardt, em 1963). Esse algoritmo é uma fusão dos métodos de Backpropagation e de Gauss-Newton, herdando a velocidade de Gauss-Newton e a estabilidade do Backpropagation. Quando comparado ao Gauss-Newton, Levenberg-Marquardt é um pouco mais lento, porém mais robusto, pois é capaz de convergir mesmo em situações nas quais a superfície de erro é muito mais complexa do que o caso quadrático.

A ideia principal do algoritmo de Levenberg-Marquardt é a realização do treinamento em duas etapas: quando o treinamento está próximo de uma área de curvatura complexa, ele funciona como o Backpropagation até que a curvatura local se encontre apropriada para uma aproximação quadrática. Então, ele se torna o algoritmo de Gauss-Newton, que possui a capacidade de acelerar significativamente a velocidade de convergência. Esse método é adequado para o treinamento de problemas pequenos e médios [14].

O método de Levenberg-Marquardt busca encontrar uma solução para o problema conhecido como "minimização de mínimos quadrados não-lineares".

Sejam a função  $f(\cdot)$  e o vetor paramétrico  $\mathbf{p} \in \mathbb{R}^m$  tais que definem um vetor de estimação  $\tilde{\chi} = f(\mathbf{p})$ ,  $\tilde{\chi} \in \mathbb{R}^n$ . Assume-se, ainda, que um parâmetro inicial  $\mathbf{p}_0$  e uma medição  $\chi$  são fornecidos. O objetivo é encontrar um vetor paramétrico  $\mathbf{p}^+$  tal que minimize a distância quadrática  $\epsilon^T \epsilon$ , sendo  $\epsilon = \chi - \tilde{\chi}$ .

A ideia básica é encontrar uma aproximação linear para  $f(\cdot)$  na vizinhança de  $\mathbf{p}$ . Para um  $|\delta_{\mathbf{p}}|$  pequeno, a expansão da série de Taylor resulta na aproximação da Equação 2.46 [17].

$$f(\mathbf{p} + \delta_{\mathbf{p}}) \approx f(\mathbf{p}) + \mathbf{J}\delta_{\mathbf{p}}, \quad (2.46)$$

onde  $\mathbf{J}$  é a matriz Jacobiana  $\frac{\partial f(\mathbf{p})}{\partial(\mathbf{p})}$ . Partindo do vetor inicial  $\mathbf{p}_0$ , é produzida uma série de vetores  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots$  que convergem em direção ao minimizador local  $\mathbf{p}^+$  de  $f(\cdot)$ .

É necessário, então, encontrar o valor de  $\delta_{\mathbf{p}}$  que minimiza a Equação 2.47.

$$\|\chi - f(\mathbf{p} + \delta_{\mathbf{p}})\| \approx \|\chi - f(\mathbf{p}) - \mathbf{J}\delta_{\mathbf{p}}\| = \|\epsilon - \mathbf{J}\delta_{\mathbf{p}}\| \quad (2.47)$$

O valor de  $\delta_{\mathbf{p}}$  que se deseja encontrar é a solução para um problema de mínimos quadrados lineares: o mínimo é obtido quando  $\mathbf{J}\delta_{\mathbf{p}} - \epsilon$  é ortogonal ao espaço coluna de  $\mathbf{J}$ . Isso resulta na Equação 2.48, que implica em  $\delta_{\mathbf{p}}$  como uma solução da Equação 2.49 [17].

$$\mathbf{J}^T(\mathbf{J}\delta_{\mathbf{p}} - \epsilon) = 0 \quad (2.48)$$

$$\mathbf{J}^T \mathbf{J}\delta_{\mathbf{p}} = \mathbf{J}^T \epsilon \quad (2.49)$$

Define-se uma matriz  $N$  tal que  $N_{ii} = \mu + [\mathbf{J}^T \mathbf{J}]_{ii}$  e os elementos fora da diagonal principal são idênticos aos seus correspondentes.  $\mu$  é chamado de termo amortecedor. A Equação 2.49 é resolvida repetidas vezes até que o termo  $\mathbf{p} + \delta_{\mathbf{p}}$  resulte em uma redução do erro  $\epsilon$ . Neste caso, o cálculo é aceito e  $\mu$  é diminuído. Para cada solução obtida em que o erro  $\epsilon$  não é diminuído, aumenta-se o valor de  $\mu$  e calcula-se uma nova solução.

As iterações são realizadas sucessivamente até que pelo menos uma das condições abaixo seja atendida [17]:

- a magnitude do gradiente de  $\epsilon^T \epsilon$ , ou seja,  $\mathbf{J}^T \epsilon$  na Equação 2.49 ficar abaixo de um limiar  $\epsilon_1$ ;
- a variação da magnitude de  $\delta_{\mathbf{p}}$  ficar abaixo de um limiar  $\epsilon_2$ ;
- o erro  $\epsilon^T \epsilon$  ficar abaixo de um limiar  $\epsilon_3$  e
- um número máximo de iterações  $\kappa_{max}$  seja atingido.

#### 2.1.7.4 Técnica de Monte Carlo

A definição da técnica de Monte Carlo, também conhecida pelo nome de amostragem estatística, não é um consenso. [18] diz que modelos probabilísticos são simulações estocásticas, sendo que Monte Carlo se resume aos processos de Monte Carlo de integração e testes

estatísticos de Monte Carlo. Em [19], Monte Carlo se resume no uso de técnicas estocásticas para resolver problemas determinísticos. Ou seja, Monte Carlo se refere ao uso de repetições como um método para descobrir, a longo prazo, o resultado de um evento.

Dentre os primeiros relatos dessa técnica, é importante mencionar seu uso por cientistas que trabalhavam na bomba atômica (projeto Manhattan). O nome Monte Carlo refere-se a uma cidade em Mônaco conhecida por seus cassinos. Desde seu surgimento durante a Segunda Guerra Mundial, técnicas de Monte Carlo vêm sendo utilizadas para modelar uma grande variedade de sistemas físicos e conceituais [20]. Apesar de sua primeira aplicação em larga escala remeter ao projeto Manhattan, há relatos de que seu primeiro uso data de 1934, quando Enrico Fermi estudava o fenômeno da difusão de nêutrons em Roma [20, 21].

É necessário realizar a distinção entre 3 nomenclaturas [19]:

- Simulação: representação fictícia da realidade;
- Método de Monte Carlo: técnica que pode ser utilizada para resolver problemas matemáticos ou estatísticos e
- Simulação de Monte Carlo: realização de diversas amostragens para determinar as propriedades de um fenômeno.

Para melhor ilustrar as diferenças entre as nomenclaturas, será utilizado o exemplo da altura de pessoas em uma população, que se aproxima de uma distribuição Normal [22]. Portanto, tem-se:

- Simulação: monta-se uma variável Normal pseudoaleatória. O valor obtido indica a altura de um membro da população. O cenário montado é de uma simulação, mas não de uma simulação de Monte Carlo.
- Método de Monte Carlo: dado um conjunto de pessoas, contar a frequência com que cada altura ocorre. Esse cenário representa um método de Monte Carlo, pois determina o comportamento de várias amostras, mas não é uma simulação.
- Simulação de Monte Carlo: montar um grande número de variáveis Normais pseudoaleatórias e obter um valor de cada uma. Esse cenário descreve uma simulação de Monte Carlo do comportamento de repetidas amostragens da população.

É importante notar que, dependendo da aplicação, é difícil ou até mesmo impossível separar claramente os passos que compõem cada uma dessas etapas.

## 2.2 TRABALHOS CORRELATOS

A utilização de *smartphones* como ferramentas de predição das situações em que seus portadores encontram-se vem sendo cada vez mais explorada. Pode-se desejar detectar o modo de transporte do usuários, doenças e até mesmo emoções.

Em [23] os autores propõem o reconhecimento de emoções utilizando o padrão de uso do teclado através do acelerômetro do celular. Enquanto o usuário digita, ele também deve selecionar, na tela do celular, qual o seu estado emocional atual. Dessa forma, os dados são coletados e separados em grupos de acordo com o estado indicado pelo usuário.

O objetivo é testar diversos classificadores e escolher aquele que apresenta a maior taxa de acerto. Os classificadores utilizados nos testes foram: Naïve Bayes, J48, IBK, Regressão Linear Multi-Resposta e SVM. A partir dos sinais originais, são extraídas características que são apresentadas aos classificadores, que estimam o estado emocional do usuário. As características extraídas são:

1. velocidade;
2. número de *backspaces* e
3. tempo decorrido entre cada caractere digitado.

Como resultado, o classificador J48 alcançou os melhores resultados nos cenários apresentados, obtendo taxas de acerto acima de 90%.

Também cabe mencionar a utilização de acelerômetros na detecção de doenças como o Parkinson. O trabalho apresentado em [24] realiza a detecção de sintomas que indicam a presença da doença de Parkinson. Essa doença afeta principalmente pessoas entre 40 e 60 anos de idade. Entretanto, também há relatos de vítimas com idade entre 20 e 30 anos. Já que um método para sua cura ainda não foi desenvolvido, quanto mais próximo do início do desenvolvimento da doença o diagnóstico for realizado, mais eficientes são os tratamentos existentes.

Atualmente, a medicina não possui métodos objetivos para a detecção da doença de Parkinson, limitando-se à análise clínica do médico. Seus sintomas, muitas vezes, se assemelham aos sintomas de outras doenças neurológicas e, por isso, aumenta-se a taxa de diagnósticos errôneos, prejudicando o tratamento do paciente.

A ferramenta proposta é composta por duas etapas: uma objetiva (aplicação para Android que realiza análises por meio de toques e testes de movimento) e uma subjetiva (aplicação de questionários).

A abordagem objetiva se vale de um *smartphone* e um *smartwatch*. São realizados os seguintes testes:

- de tremor de mão;
- de toque de dedo e
- de desenho de espiral

Os testes de toque e de desenho são utilizados para avaliar a consistência dos movimentos do paciente, enquanto que o teste de tremor de mão avalia a estabilidade do paciente quando está parado.

Combinadas as duas etapas, é possível fornecer um diagnóstico mais preciso do que o atual.

Aplicações em *e-Health* que utilizam-se de acelerômetros estão igualmente se tornando comuns. Dentre elas, há abordagens de rastreadores de atividades físicas em exercícios, que buscam identificar o tipo de atividade que o usuário está praticando.

Foi realizado um estudo sobre o rastreamento de atividades físicas para que o usuário tenha um estilo de vida mais saudável e, para tal, é realizado o cálculo de seu gasto calórico [25]. Para que os dados pudessem ser coletados, foram utilizados os seguintes sensores: acelerômetro, giroscópio e GPS. A taxa de aquisição dos dados é realizada na frequência de 15 Hz.

A cada 5 segundos de dados, são formadas janelas e, a partir delas, são extraídas diversas características, tais como média, desvio padrão, magnitude da aceleração, energia espectral, entropia no domínio da frequência etc. Após a extração dessas características, foi realizado um ranqueamento de acordo com o ganho de informações que cada uma fornecia com o fim de montar a base de dados.

O objetivo desse trabalho é testar o desempenho de diversos classificadores nos cenários propostos. Os classificadores analisados são: Árvore de Decisão, KNN, Naïve Bayes e SVM.

Para a realização dos testes, a base de dados foi dividida em 10 conjuntos de dados para a realização de validação cruzada, permitindo obter métricas mais próximas ao verdadeiro desempenho de cada classificador.

As atividades físicas rastreadas são: inativo, caminhando, correndo, subindo escadas, descendo escadas, dirigindo e andando de bicicleta. O classificador que apresentou o melhor desempenho foi a Árvore de Decisão, alcançando uma taxa de 94,4%. Esse classificador foi, então, escolhido para ser implementado em um aplicativo Android.

O trabalho apresentado em [26] se interessa pela determinação dos hábitos do usuário. A base de dados é composta por aquisições realizadas em 29 usuários diferentes, que se submetiam às seguintes classes: caminhando, correndo, subindo escadas, descendo escadas, ficando parado em pé e ficando sentado.

O único sensor utilizado nas aquisições é o acelerômetro do próprio *smartphone*, que foi configurado para funcionar em uma frequência de 20 Hz. O método realiza a classificação

utilizando janelas de 10 segundos de dados. Para cada janela, são extraídas 43 características, que são apresentadas a um classificador:

- média: é extraída a média de cada um dos 3 eixos do acelerômetros (3 características);
- desvio padrão: é calculado o desvio padrão de cada um dos 3 eixos do acelerômetro (3 características);
- diferença absoluta média: é calculada a diferença absoluta média entre cada uma das aquisições que compõem a janela de 10 segundos para cada eixo (3 características);
- aceleração resultante média: é calculada a média das normas 2 (Equação 2.3) de cada tripla  $(x_i, y_i, z_i)$  contida na janela de 10 segundos (1 característica);
- tempo entre picos: é calculado o tempo, em milissegundos, entre os picos das senoides associadas a cada atividade em cada eixo (3 características) e
- quantização: é determinada a amplitude dos valores de cada eixo (*máximo–mínimo*), divide-se essa amplitude em 10 limiares igualmente espaçados e depois calcula-se qual fração de cada elemento da janela se encaixa em cada limiar (30 características).

Para a realização dos testes, a base de dados foi dividida em 10 conjuntos de dados para a realização de validação cruzada.

O estudo consiste em testar 3 classificadores e determinar aquele que apresenta a melhor performance. Os classificadores analisados são: Árvore de Decisão J48, Regressão Logística e Redes Neurais Artificiais Multicamada. As Redes Neurais Artificiais Multicamada foram o classificador que apresentou a maior taxa de acerto média, de 91,7%.

Por sua vez, em [27] o foco é detectar posturas de inatividade e de atividades físicas, incluindo o cálculo do gasto energético do usuário. A base de dados é composta por 25 horas de dados, adquiridos a partir de 31 indivíduos (21 homens e 10 mulheres).

Os dados são adquiridos através de um acelerômetro funcionando a 33,33 Hz e são divididos em janelas de 5 segundos. A partir dos 3 eixos do acelerômetro, foram derivadas diversas componentes do sinal, tais como: magnitude da aceleração, ângulos entre o vetor aceleração e cada um dos eixos do celular, diversas métricas no domínio da frequência. No total, são extraídas 28 componentes do sinal. De cada componente, são extraídas 12 características: média, mediana, máximo, mínimo, raiz quadrática média, desvio padrão, desvio médio, intervalo intercuartil, energia, entropia, skewness e kurtosis. No total, cada janela de 5 segundos fornece 336 características, que são apresentadas ao classificador.

O classificador utilizado é uma Árvore de Decisão, sendo treinada para classificar as seguintes classes: caminhando, correndo, ficando parado em pé, ficando sentado, ficando deitado e atividades não listadas (tratadas como "fora do escopo").

O *smartphone* somente é transportado em dois possíveis lugares: no bolso ou no cinto do usuário. Para as situações em que o dispositivo é transportado no bolso, foi obtida uma taxa de acerto de 99,5%; quando no cinto, de 99,4%.

Algumas aplicações, ainda, focam na detecção do modo de transporte para que o planejamento de empresas e órgãos possam realizar melhor seus planejamentos. Em [28], foram realizadas gravações de 15 indivíduos por 2 meses (totalizando 355 horas de dados), fazendo o uso de um acelerômetro e do GPS. Quando o GPS não estava disponível, a rede do celular foi utilizada para calcular a posição aproximada do dispositivo.

Os tamanhos das janelas de dados são definidos dinamicamente em tempo de execução, variando em momentos em que o usuário está em estado estacionário, movendo-se lentamente ou movendo-se rapidamente. Para isso, são definidos limiares que caracterizam cada uma dessas situações. O tamanho máximo da janela, entretanto, é definido em 2 minutos. O classificador utilizado é o Modelo Oculto de Markov Discreto e há 8 classes analisadas: ônibus, carro, bicicleta, bonde, trem, metrô, caminhada e moto. A taxa média de acerto obtida é de 75,88%.

A detecção do modo de transporte para o cuidado de idosos tem sido outro problema abordado, pois assim é possível realizar a estimacão do risco que uma atividade fora de casa representa para uma pessoa idosa. Esse é um contexto específico no qual a detecção do modo de transporte se aplica.

Em [29], foi utilizada uma técnica que coleta dados a partir de um acelerômetro operando a uma frequência de 33,33 Hz e de um Wi-Fi. Para a formação da base de dados, foram realizadas aquisições com um grupo de 15 indivíduos, totalizando 24 horas de dados gravados.

Cada indivíduo foi submetido a situações em que se encontrava nos seguintes modos de transporte: inativo, caminhando, ônibus, carro, trem e metrô.

Os dados são, então, organizados em janelas de 5 segundos. A partir do sinal original  $(x, y, z, magnitude)$ , são derivadas outras componentes. Cada componente fornece gravidade e aceleração linear, além de seus valores absolutos. Também são calculados os ângulos entre cada eixo do *smartphone* e o vetor aceleração. Uma componente adicional é calculada a partir de uma filtragem passa-baixa na magnitude. No total, são derivadas 21 componentes do sinal.

Para cada componente, são extraídas 17 características, sendo elas: média, mediana, máximo, mínimo, raiz quadrática média, desvio padrão, desvio médio, intervalo intercuartil, média mínima e média máxima, altura de pico, altura média de pico, contagem cruzada média, entropia, energia, skewness e kurtosis.

Sendo assim, cada janela de 5 segundos resulta em 357 características, que são apresentadas ao classificador.

Os testes são realizados avaliando 3 diferentes classificadores: Árvore de Decisão J48, SMO e Naïve Bayes. O resultado médio geral obtido é de mais de 90%.

O trabalho em [30], por sua vez, utiliza GPS para identificar se o idoso está realizando alguma atividade fora do comum e, assim, calcular o risco de acidentes do idoso naquela situação.

Em [31], foram utilizados um acelerômetro, com taxa de aquisição de 30 Hz, e um barômetro, com taxa de aquisição de 5 Hz. Foram coletados dados de 25 indivíduos (7 homens e 18 mulheres), totalizando mais de 7 horas de gravações de dados.

Os dados são separados em janelas de 5 segundos e, então, são extraídas 29 características de cada janela. Essas características são classificadas em 3 grandes grupos, conforme mostrado a seguir.

- Domínio espectral: frequência máxima, frequência média, frequência fundamental, espectro de máxima potência, energia total, centroide espectral, propagação espectral, skewness espectral, kurtosis espectral, declive espectral, diminuição espectral, rolagem espectral, deslocamento espectral, distância da curva e variação espectral.
- Domínio estatístico: skewness, kurtosis, histograma, média, desvio padrão, intervalo intercuartil, variância, raiz quadrática média e desvio absoluto médio.
- Domínio temporal: correlação, centroide temporal, autocorrelação, taxa de passagem zero e regressão linear.

Após a extração das características, elas são remapeadas no intervalo  $[-1, 1]$ , ou seja, são normalizadas de acordo com a Equação 2.10.

Essas características são apresentadas a uma Árvore de Decisão, que realiza a classificação dentro das seguintes classes: caminhando, correndo, subindo escadas, descendo escadas, ficando parado em pé e ficando sentado. A precisão média obtida foi de  $94,5 \pm 6,8\%$ .

A Tabela 2.1 apresenta um compilado dos trabalhos aqui mencionados referentes a detecção do modo de transporte.

Trabalho	Aplicação	Classes	Sensores	Classificador(es)	Tamanho da Janela	Taxa de Acerto
[25]	Rastreamento de atividades físicas.	Inativo, caminhando, correndo, subindo escadas, descendo escadas, dirigindo e andando de bicicleta.	Acelerômetro, giroscópio e GPS	Árvore de Decisão, K-Nearest Neighbor, Naïve Bayes e SVM	5 segundos	94,4%
[26]	Rastreamento de atividades físicas.	Caminhando, correndo, subindo escadas, descendo escadas, ficando sentado e ficando parado em pé.	Acelerômetro	Árvore de Decisão J48, Regressão Logística e Redes Neurais Artificiais Multicamada	10 segundos	91,7%
[27]	Rastreamento de atividades físicas.	Caminhando, correndo, ficando parado em pé, ficando sentado, ficando deitado e atividades não listadas (tratadas como "fora do escopo").	Acelerômetro	Árvore de Decisão	5 segundos	99,5%
[28]	Deteção de modo de transporte para planejamento de cidades.	Ônibus, carro, bicicleta, bonde, trem, metrô, caminhada e moto.	Acelerômetro, GPS e rede celular	Modelo Oculto de Markov Discreto	Variável de acordo com a localidade	75,88%
[29]	Deteção de modo de transporte para o cuidado de idosos.	Inativo, caminhando, ônibus, carro, trem e metrô.	Acelerômetro e Wi-Fi	Árvore de Decisão J48, Naïve Bayes e SMO	5 segundos	Acima de 90%
[30]	Cálculo do risco de acidentes com idosos.	Estar em casa, ir ao centro e ir ao mercado.	GPS	Modelo Probabilístico Próprio		
[31]	Rastreamento de atividades físicas para fisioterapia em idosos.	Caminhando, correndo, subindo escadas, descendo escadas, ficando parado em pé e ficando sentado.	Acelerômetro e barômetro	Árvore de Decisão	5 segundos	94,5 ± 6,8%

Tabela 2.1: Resumo das abordagens e resultados obtidos em trabalhos correlatos.

### 3 MÉTODO PROPOSTO

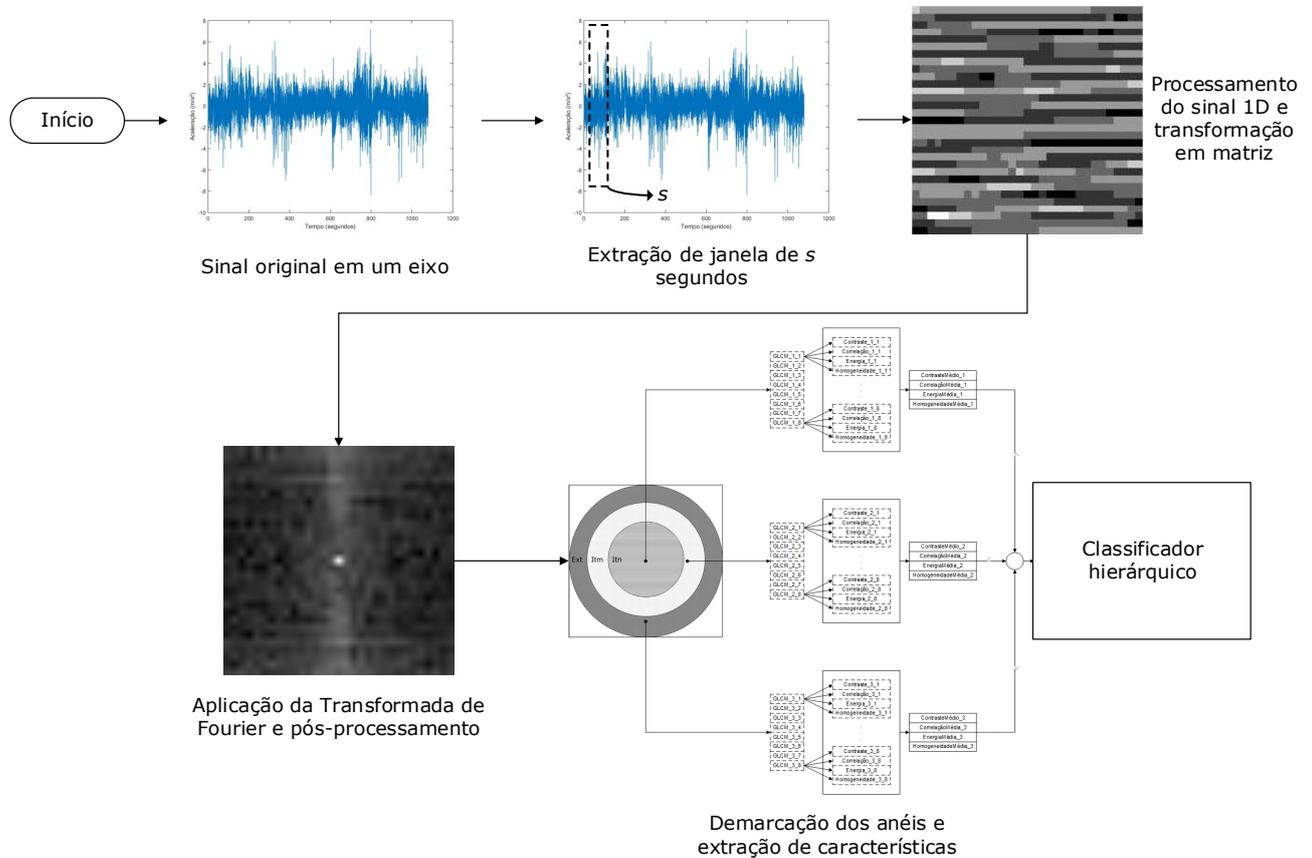


Figura 3.1: Fluxograma do método proposto. O sinal original é dividido em janelas de  $s$  segundos de aquisições, que são reorganizadas em uma matriz. Essa matriz, por sua vez, é dividida em 3 faixas de frequências. Após a escolha das faixas de interesse, a extração de características é realizada e, por fim, as características são apresentadas ao classificador hierárquico, que decidirá se a situação apresentada é de risco ou não. É importante notar que o fluxograma demonstra o funcionamento do método para apenas um único eixo  $x_i$ , sendo necessário concatenar as características dos demais eixos conforme as Equações 3.4, 3.5 e 3.6.

O uso de *smartphones* tem se tornado cada vez mais perigoso para seus usuários e para aqueles ao seu redor. Faz-se necessário, então, o desenvolvimento de um método capaz de identificar situações de uso que geram riscos de saúde aos seres humanos. Tal método deve ser utilizável independente do ambiente, seja ele ao céu aberto ou em ambientes fechados sem quaisquer sinais externos ao dispositivo.

Para que esse problema seja resolvido, o método aqui proposto faz uso apenas do acelerômetro do celular, resultando na independência de sinais externos ao dispositivo e permitindo

a obtenção de valiosos indicativos de perigo.

Neste capítulo, serão apresentados o método de extração de características, as classes analisadas, o classificador hierárquico e a construção da base de dados. O fluxograma do método proposto é apresentado na Figura 3.1.

### 3.1 EXTRAÇÃO DE CARACTERÍSTICAS

Dadas uma janela de  $s$  segundos, a frequência  $f$  de aquisições e a classe desejada, a quantidade de aquisições  $w$  por janela é dada pela Equação 3.1.

$$w = s \times f \quad (3.1)$$

Antes de realizar a divisão em janelas, entretanto, os primeiros 5 segundos de aquisições são descartados. Esse período é considerado como tempo de preparação para que o *smartphone* possa ser posicionado conforme desejado pelo usuário.

Até esse momento, a janela é composta por um sinal que possui três eixos, cada um com  $w$  aquisições. Para cada eixo, o sinal é normalizado conforme a Equação 2.10. Com o eixo normalizado, é formada uma matriz  $i$  de dimensões  $k \times k$ , onde  $k = \lfloor \sqrt{w} \rfloor$ . A cada  $k$  elementos da janela, uma linha em  $i$  é formada. Sendo assim, a matriz é composta pelos valores da janela dispostos de tal forma que, se todas as linhas de  $i$  fossem concatenadas, a janela original seria obtida.

Aplica-se a FFT sobre a matriz  $i$ . A matriz resultante,  $I$ , é deslocada de tal forma que a componente de frequência zero esteja localizada no centro. Para que  $I$  possa ser tratada como uma imagem em níveis de cinza, faz-se  $I = 255 \times I$ .

Por fim, o espectro obtido é normalizado a partir da Equação 2.7, conforme Equação 3.2.

$$I_N(i, j) = \frac{255 \times I_A(i, j)}{I_{A_{max}}}, \quad \forall i, j, \quad (3.2)$$

sendo  $I_A$  uma matriz definida conforme a Equação 3.3 e  $I_{A_{max}}$  o maior valor encontrado nesta mesma matriz.

$$I_A(i, j) = \ln(|I(i, j)|), \quad \forall i, j \quad (3.3)$$

A partir da matriz  $I_N$ , são determinadas três faixas de frequência distintas: uma faixa de baixa frequência, uma de frequência intermediária e outra de alta frequência. Essas faixas são delimitadas por áreas que indicam quais píxeis fazem parte de cada uma. Já que a matriz é quadrada, é importante ressaltar que alguns píxeis são descartados: os mais próximos aos vértices da matriz. O raio de cada faixa, por sua vez, é determinado de acordo com  $k$ . A faixa de baixa frequência é determinada por um círculo de raio  $\lfloor \frac{k}{6} \rfloor$ . Para determinar a faixa

intermediária, é definido um círculo de raio  $\lfloor \frac{2 \times k}{6} \rfloor$ . Desse círculo, é subtraída a área do círculo interno, resultando na faixa intermediária. Para determinar a faixa externa, define-se uma área circular de raio  $\frac{k}{2}$  e subtrai-se a área do círculo intermediário. Dessa forma, as três áreas são definidas, conforme Figura 3.2.

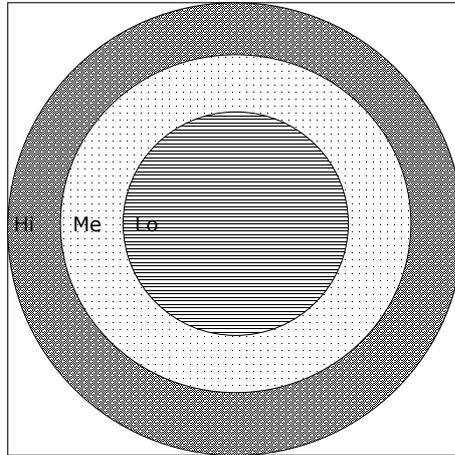


Figura 3.2: Demarcação das faixas de frequência utilizadas em cada área da matriz. A área demarcada por Lo indica a faixa de baixa frequência (raio  $\lfloor \frac{k}{6} \rfloor$ ); Me, a faixa intermediária (raio  $\lfloor \frac{2 \times k}{6} \rfloor$ ); Hi, a área de alta frequência (raio  $\frac{k}{2}$ ). A área não hachurada, por sua vez, é descartada e, portanto, não é utilizada na extração de características. O cálculo da GLCM é aplicado sobre cada uma das faixas e são extraídos contraste, correlação, energia e homogeneidade conforme as Equações 2.15, 2.16, 2.17 e 2.18, respectivamente.

Em cada faixa de frequência, o seguinte processo é realizado: uma GLCM é calculada em cada direção referente à vizinhança de 8 ( $p_{v(-1,-1)}$ ,  $p_{v(-1,0)}$ ,  $p_{v(-1,1)}$ ,  $p_{v(0,-1)}$ ,  $p_{v(0,1)}$ ,  $p_{v(1,-1)}$ ,  $p_{v(1,0)}$  e  $p_{v(1,1)}$ ). Como resultado, cada faixa fornece 8 GLCMs, uma para cada direção.

Para cada GLCM, calcula-se seus respectivos valores de contraste, correlação, energia e homogeneidade conforme as Equações 2.15, 2.16, 2.17 e 2.18, respectivamente.

Obtêm-se, então, 8 valores de contraste, 8 de correlação, 8 de energia e 8 de homogeneidade. É calculada a média de cada um desses conjuntos de 8 valores, ou seja, o conjunto de características de cada faixa de frequência de cada eixo de uma janela é composto por 4 valores. Já que cada janela possui 3 eixos, o resultado é composto por 12 valores, que serão apresentados a um classificador, e é composto por 3 valores de contraste, 3 de correlação, 3 de energia e 3 de homogeneidade, conforme diagrama da Figura 3.3.

Dessa forma, o vetor na Equação 3.4 ilustra como o conjunto de características é formado

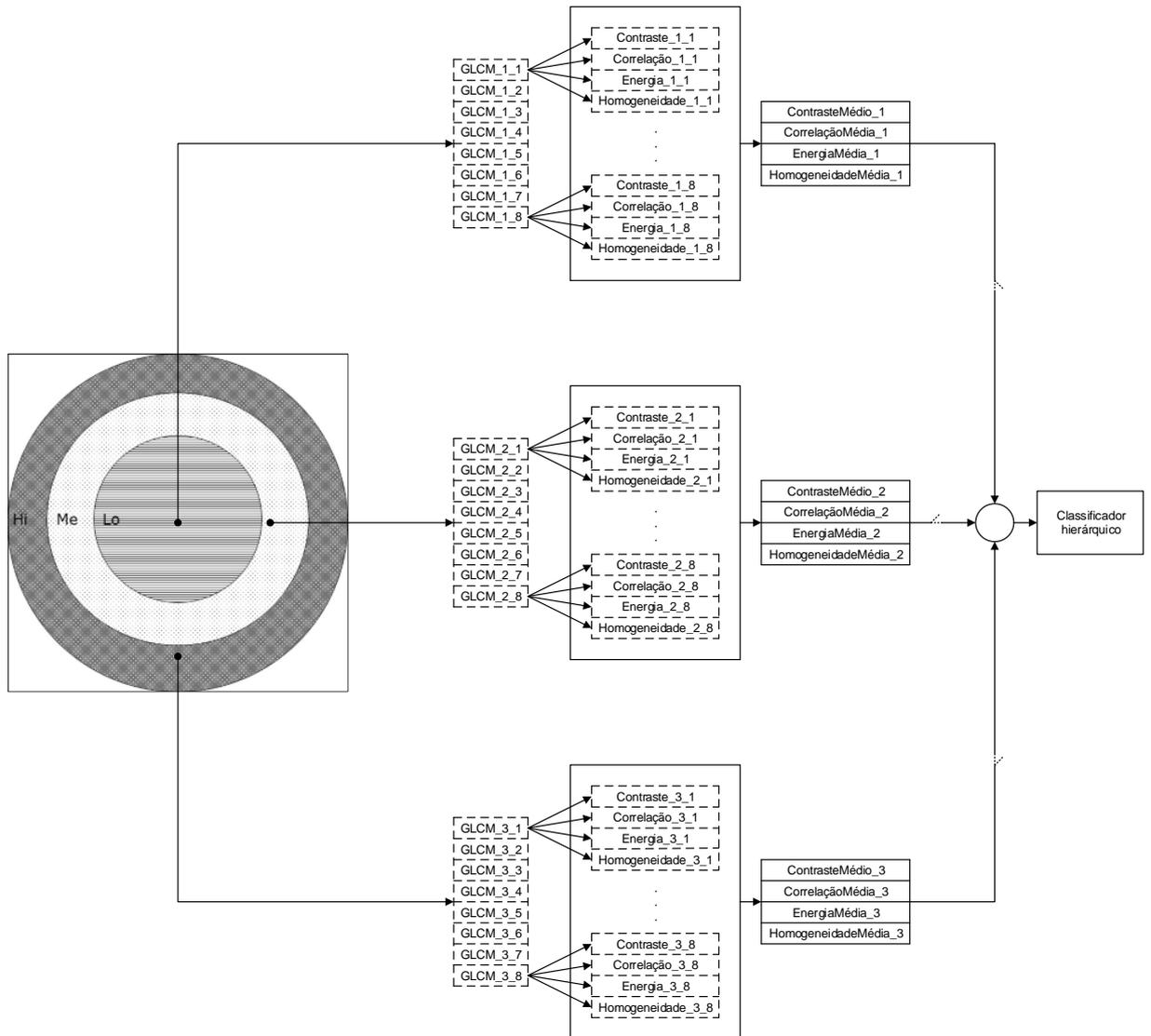


Figura 3.3: Representação das características extraídas de uma janela de dados para um único eixo a partir das faixas de frequência da Figura 3.2. Como resultado, cada faixa de cada eixo fornece 4 valores (contraste, correlação, energia e homogeneidade médios). Caso haja outros eixos, os valores de cada faixa são concatenados com seus respectivos valores dos demais eixos.

a partir da faixa de baixa frequência utilizando todos os eixos.

$$\begin{aligned}
caracLo = [ & conMLo_{x1}, conMLo_{x2}, conMLo_{x3}, \\
& corMLo_{x1}, corMLo_{x2}, corMLo_{x3}, \\
& enMLo_{x1}, enMLo_{x2}, enMLo_{x3}, \\
& homMLo_{x1}, homMLo_{x2}, homMLo_{x3} ],
\end{aligned} \tag{3.4}$$

sendo  $conMLo_{xi}$  o contraste médio da faixa de baixa frequência no eixo  $xi$ ,  $corMLo_{xi}$  a correlação média da faixa de baixa frequência no eixo  $xi$ ,  $enMLo_{xi}$  a energia média da faixa de baixa frequência no eixo  $xi$  e  $homMLo_{xi}$  a homogeneidade média da faixa de baixa frequência no eixo  $xi$ .

Esse vetor pode ser concatenado com outros vetores de diferentes faixas para formar um conjunto de características ainda maior. O vetor na Equação 3.5 exemplifica a estrutura formada pelo uso da faixa de baixa frequência em conjunto com a faixa de frequência intermediária.

$$\begin{aligned}
caracLoMe = [ & caracLo, conMMe_{x1}, conMMe_{x2}, conMMe_{x3}, \\
& corMMe_{x1}, corMMe_{x2}, corMMe_{x3}, \\
& enMMe_{x1}, enMMe_{x2}, enMMe_{x3}, \\
& homMMe_{x1}, homMMe_{x2}, homMMe_{x3} ],
\end{aligned} \tag{3.5}$$

sendo  $conMMe_{xi}$  o contraste médio da faixa de frequência intermediária no eixo  $xi$ ,  $corMMe_{xi}$  a correlação média da faixa de frequência intermediária no eixo  $xi$ ,  $enMMe_{xi}$  a energia média da faixa de frequência intermediária no eixo  $xi$  e  $homMMe_{xi}$  a homogeneidade média da faixa de frequência intermediária no eixo  $xi$ .

Além disso, também pode haver a concatenação dos vetores de todas as faixas, formando o vetor da Equação 3.6.

$$\begin{aligned}
caracLoMeHi = [ & caracLoMe, conMHi_{x1}, conMHi_{x2}, conMHi_{x3}, \\
& corMHi_{x1}, corMHi_{x2}, corMHi_{x3}, \\
& enMHi_{x1}, enMHi_{x2}, enMHi_{x3}, \\
& homMHi_{x1}, homMHi_{x2}, homMHi_{x3} ]
\end{aligned} \tag{3.6}$$

sendo  $conMHi_{xi}$  o contraste médio da faixa de alta frequência no eixo  $xi$ ,  $corMHi_{xi}$  a correlação média da faixa de alta frequência no eixo  $xi$ ,  $enMHi_{xi}$  a energia média da faixa de alta frequência no eixo  $xi$  e  $homMHi_{xi}$  a homogeneidade média da faixa de alta frequência no eixo  $xi$ .

### 3.2 CLASSES ANALISADAS

A Figura 3.4 apresenta o esquema de nomenclatura de classes de transporte proposto no presente trabalho. Cada classe possui quatro divisões: modo de transporte, local de transporte, situação e modo de interação. Cada classe é identificada por uma letra (de  $a$  até  $d$ , indicando a divisão) seguida por um número (indicando o valor naquela divisão). Por meio da combinação das letras e números, é possível nomear cada classe.

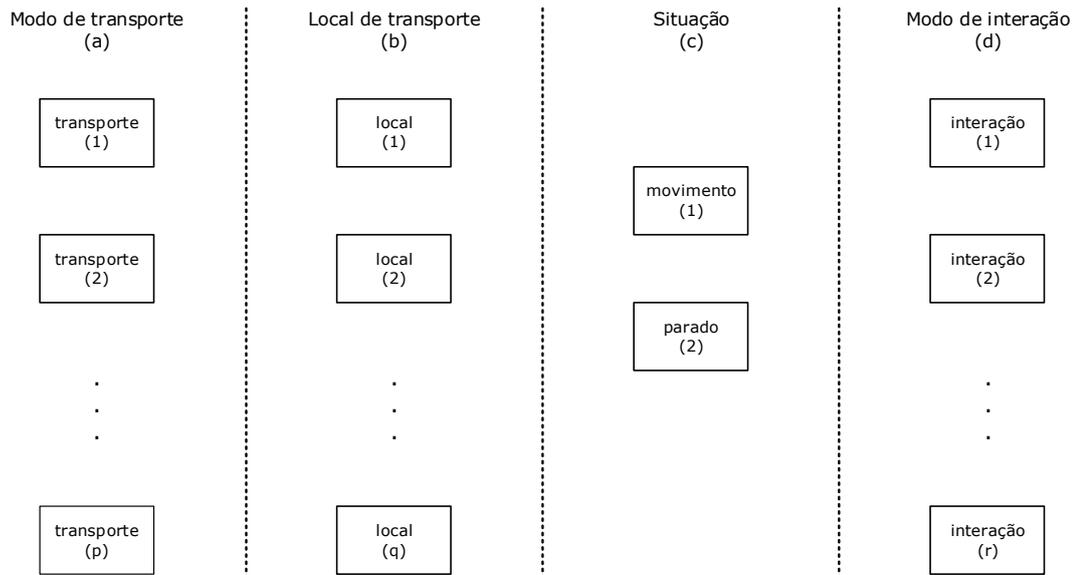


Figura 3.4: Esquema de nomenclatura de classes de transporte. Cada classe é dividida em quatro divisões: modo de transporte, local de transporte, situação e modo de interação. Cada divisão é identificada por uma letra e seu valor é identificado por um número. Por exemplo, a classe  $a_p b_q c_1 d_r$  indica que o *smartphone* está sendo transportado no modo de transporte  $p$ , no local  $q$ , o usuário está em movimento e o modo de interação é o  $r$ .

O sistema de classificação desenvolvido neste trabalho é capaz de identificar 3 modos de transporte, sendo eles: carro, bicicleta e a pé. De acordo com o modo de transporte identificado, há 3 possíveis locais nos quais o *smartphone* pode estar sendo transportado: em mãos, no painel ou no bolso. Também é possível identificar, em cada classe, se o usuário está se locomovendo (em movimento) ou se está parado. Por fim, há 3 possíveis modos de interação com o celular. O usuário pode estar fazendo uma ligação, olhando/digitando no aparelho ou, ainda, pode não estar interagindo com o dispositivo. A Figura 3.5 ilustra a nomenclatura e divisão de cada classe.

Para cada modo de transporte, há diferentes classes que podem ser identificadas, conforme veremos a seguir.

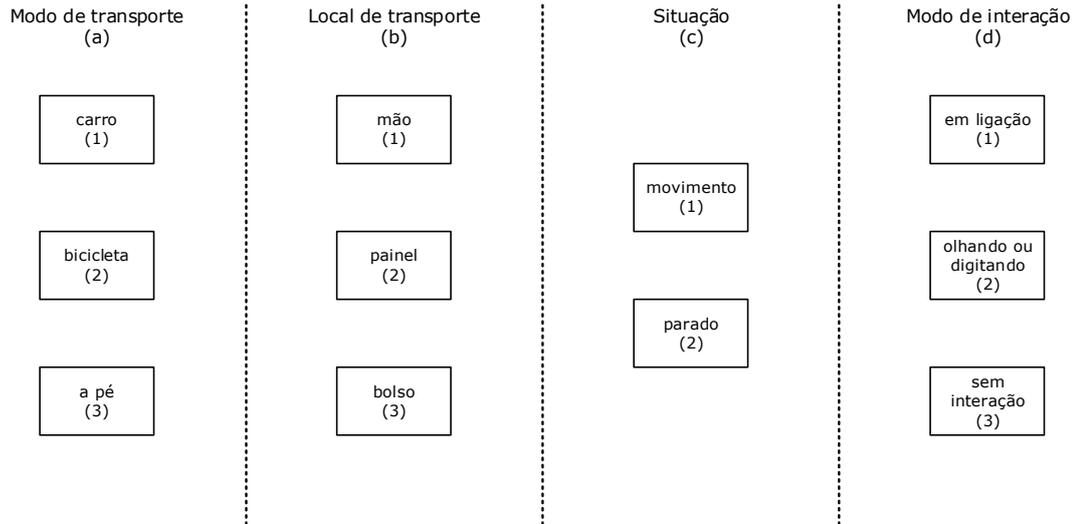


Figura 3.5: Classificações realizadas pelo sistema. Cada classe é representada através da união dos índices de cada divisão. A classe  $a_1b_1c_1d_1$ , por exemplo, indica que o *smartphone* está sendo transportado em um carro e levado em mãos enquanto o carro está em movimento e o usuário está realizando uma ligação. A classe  $a_3b_1c_1d_2$ , por sua vez, indica que o usuário está a pé com o *smartphone* em mãos enquanto caminha e olha ou digita algo, e assim por diante.

### 3.2.1 Modo de Transporte: carro

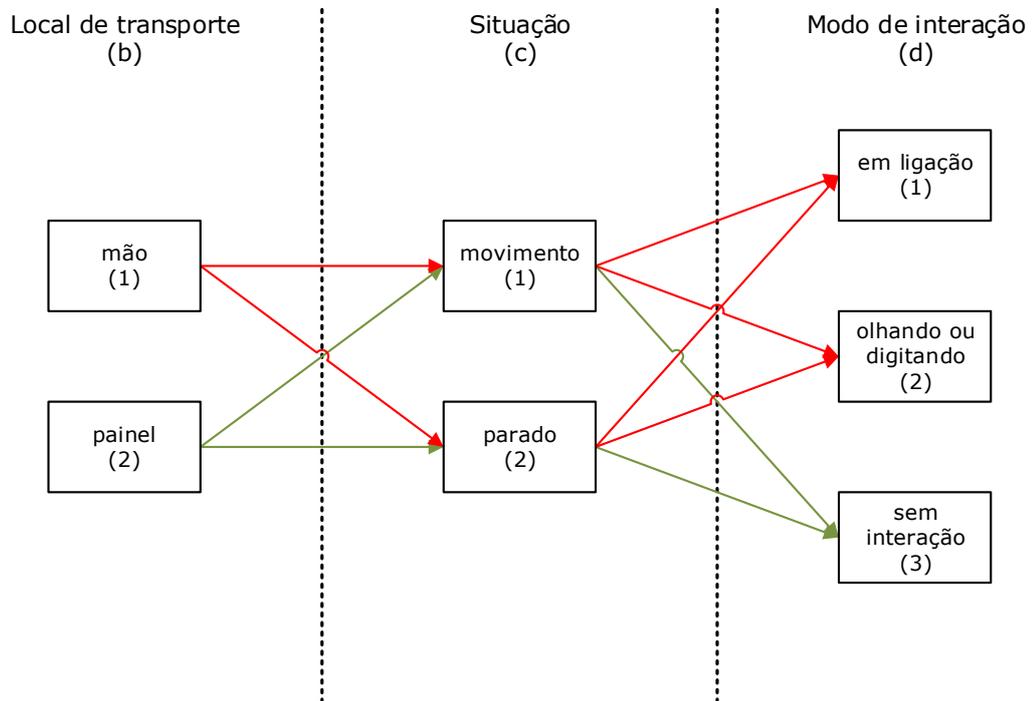


Figura 3.6: Classificações realizadas quando o usuário está utilizando um carro ( $a_1$ ). As setas em vermelho formam as possíveis classes para a situação em que o dispositivo é transportado em mãos (classes  $a_1b_1c_1d_1$ ,  $a_1b_1c_1d_2$ ,  $a_1b_1c_2d_1$  e  $a_1b_1c_2d_2$ ); a seta verde, as situações em que o *smartphone* é transportado no painel do carro (classes  $a_1b_2c_1d_3$  e  $a_1b_2c_2d_3$ ).

Dado que o modo de transporte utilizado pelo usuário é um carro ( $a_1$ ), o *smartphone* só pode estar sendo transportado em mãos ( $a_1b_1$ ) ou no painel ( $a_1b_2$ ). Em ambos os casos, o carro pode estar em movimento ( $a_1b_1c_1$  e  $a_1b_2c_1$ ) ou parado ( $a_1b_1c_2$  e  $a_1b_2c_2$ ).

Com o celular em mãos com o carro em movimento, o usuário pode estar realizando uma ligação ( $a_1b_1c_1d_1$ ) ou apenas olhando ou digitando ( $a_1b_1c_1d_2$ ). Com o aparelho em mãos com o carro parado, as mesmas interações são possíveis: realizando uma ligação ( $a_1b_1c_2d_1$ ) e olhando ou digitando ( $a_1b_1c_2d_2$ ).

As classes existentes para quando o celular está sendo transportado no painel do carro ( $a_1b_2c_1$  e  $a_1b_2c_2$ ) se resumem às situações em que não há interação com o *smartphone* ( $a_1b_2c_1d_3$  e  $a_1b_2c_2d_3$ ).

A Figura 3.6 ilustra as classes existentes. As setas vermelhas indicam as classes disponíveis na situação em que o dispositivo é transportado em mãos; as verdes, no painel do carro.

### 3.2.2 Modo de Transporte: bicicleta

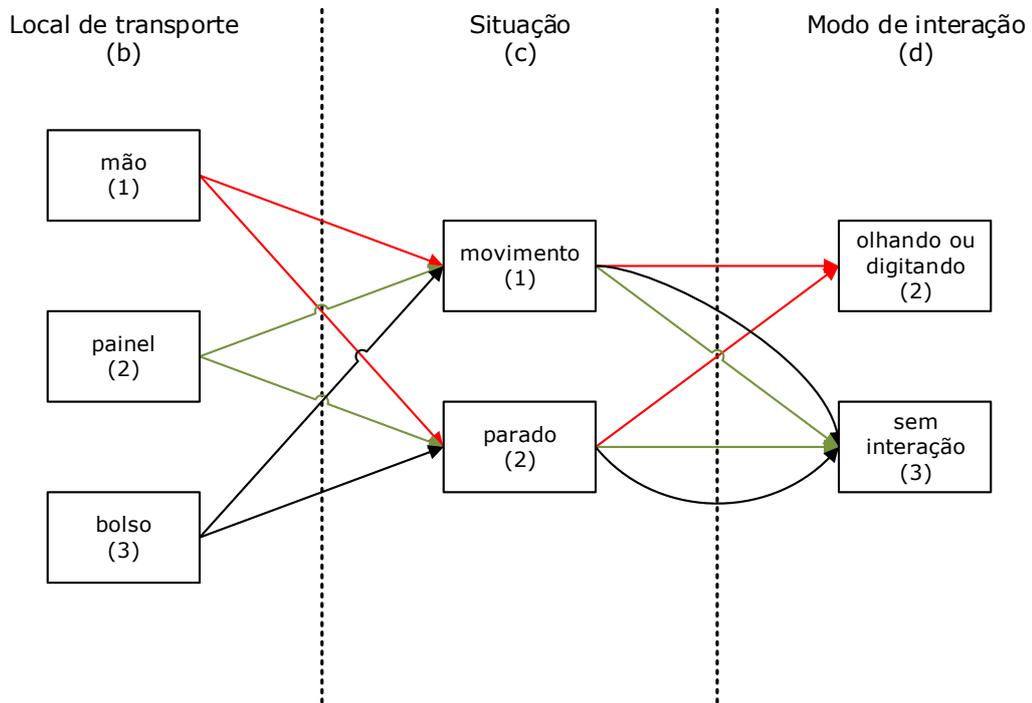


Figura 3.7: Classificações realizadas quando o usuário está utilizando uma bicicleta ( $a_2$ ). As setas em vermelho formam as possíveis classes para a situação em que o dispositivo é transportado em mãos (classes  $a_2b_1c_1d_2$  e  $a_2b_1c_2d_2$ ); a seta verde, as situações em que o *smartphone* é transportado no painel da bicicleta (classes  $a_2b_2c_1d_3$  e  $a_2b_2c_2d_3$ ); as pretas, quando no bolso (classes  $a_2b_3c_1d_3$  e  $a_2b_3c_2d_3$ ).

Dado que o modo de transporte sendo utilizado é uma bicicleta ( $a_2$ ), o *smartphone* pode estar sendo transportado em mãos ( $a_2b_1$ ), no painel ( $a_2b_2$ ) ou no bolso da calça/bermuda utilizada pelo ciclista ( $a_2b_3$ ). Em todas estas situações, a bicicleta pode estar em movimento ( $a_2b_1c_1$ ,  $a_2b_2c_1$  e  $a_2b_3c_1$ ) ou parada ( $a_2b_1c_2$ ,  $a_2b_2c_2$  e  $a_2b_3c_2$ ).

Se o celular estiver em mãos, as duas únicas classes identificáveis se referem às situações em que a bicicleta está em movimento com o ciclista olhando ou digitando no *smartphone* ( $a_2b_2c_1d_2$ ) e parada com o usuário olhando ou digitando ( $a_2b_2c_2d_2$ ).

Com o *smartphone* no painel ou no bolso, com a bicicleta em movimento ou parada, há um único modo identificável de interação com o celular: não interagindo ( $a_2b_2c_1d_3$ ,  $a_2b_3c_1d_3$ ,  $a_2b_2c_2d_3$  e  $a_2b_3c_2d_3$ )

A Figura 3.7 ilustra as classes existentes. As setas vermelhas indicam as classes disponíveis na situação em que o dispositivo é transportado em mãos; as verdes, no painel da bicicleta; as pretas, no bolso da calça/bermuda do usuário.

### 3.2.3 Modo de Transporte: a pé

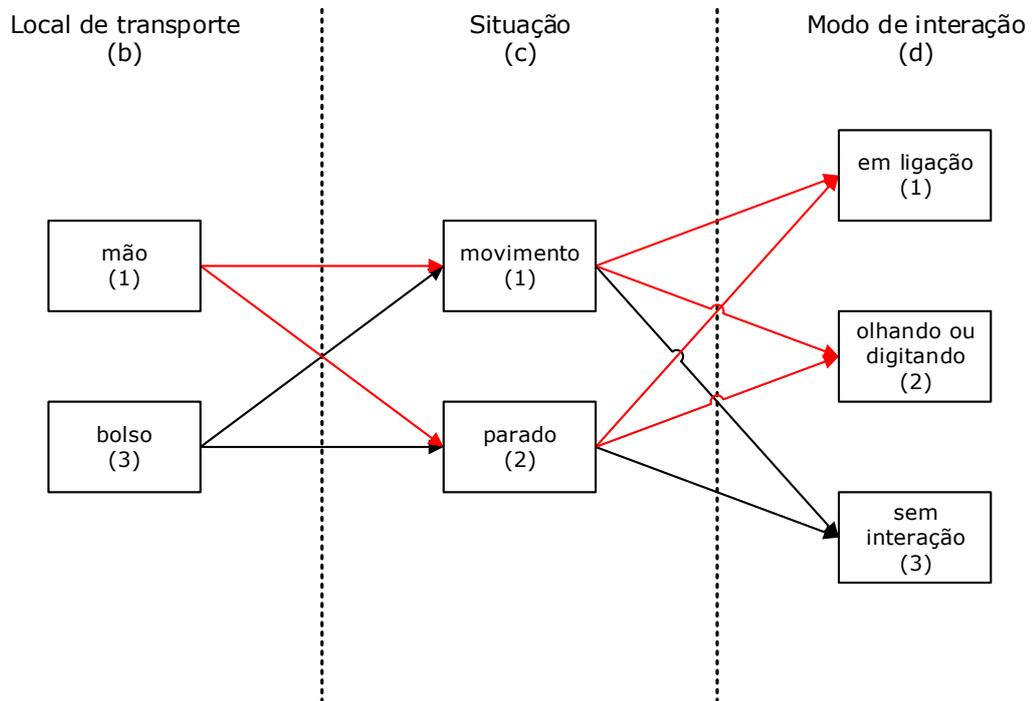


Figura 3.8: Classificações realizadas quando o usuário está a pé ( $a_3$ ). As setas em vermelho formam as possíveis classes para a situação em que o dispositivo é transportado em mãos (classes  $a_3b_1c_1d_1$ ,  $a_3b_1c_1d_2$ ,  $a_3b_1c_2d_1$  e  $a_3b_1c_2d_2$ ); a seta preta, as situações em que o *smartphone* é transportado no bolso do usuário (classes  $a_3b_3c_1d_3$  e  $a_3b_3c_2d_3$ ).

Dado que o usuário está se locomovendo a pé ( $a_3$ ), o *smartphone* pode estar sendo transportado em mãos ( $a_3b_1$ ) ou no bolso de uma calça/bermuda ( $a_3b_3$ ). Em todas estas situações, o usuário pode estar em movimento ( $a_3b_1c_1$  e  $a_3b_3c_1$ ) ou parado ( $a_3b_1c_2$  e  $a_3b_3c_2$ ).

Com o celular em mãos, tanto em movimento como parado, o usuário pode estar realizando uma ligação telefônica ( $a_3b_1c_1d_1$  e  $a_3b_1c_2d_1$ ) ou, então, olhando ou digitando no celular ( $a_3b_3c_1d_2$  e  $a_3b_3c_2d_2$ ).

Se o *smartphone* estiver sendo transportado no bolso de uma calça/bermuda, há apenas um modo de interação identificável pelo método proposto: sem interação ( $a_3b_3c_1d_3$  e  $a_3b_3c_2d_3$ ).

A Figura 3.8 ilustra as classes existentes. As setas vermelhas indicam as classes disponíveis na situação em que o dispositivo é transportado em mãos; as pretas, no bolso da calça/bermuda do usuário.

### 3.3 CLASSIFICADOR HIERÁRQUICO

Após a extração das características do sinal, é necessário apresentá-las ao classificador hierárquico da Figura 3.9.

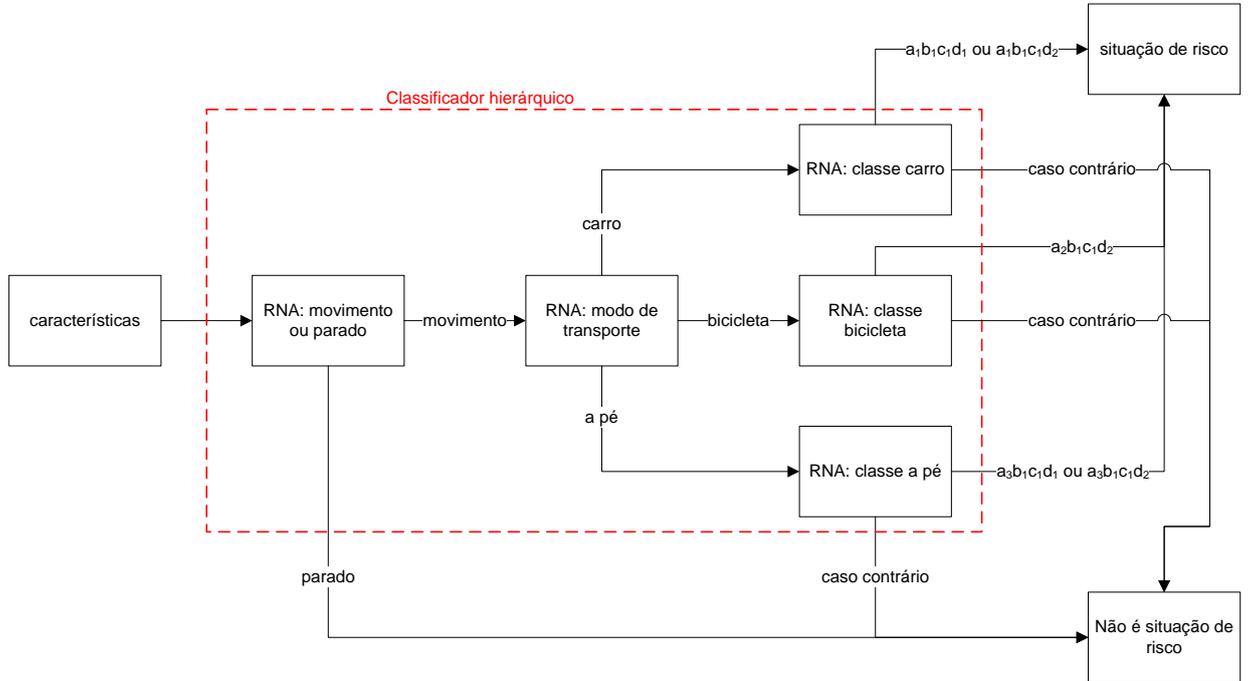


Figura 3.9: Representação do classificador hierárquico. Dado um conjunto de características, o sistema o classifica como "situação de risco" ou "não é situação de risco".

O primeiro classificador consiste em definir se a entrada representa uma classe na qual o usuário está em movimento (classes que possuem  $c_1$ ) ou parado (classes que possuem  $c_2$ ). A identificação de uma classe como pertencente ao grupo  $c_1$  é um indicativo de que pode haver risco; caso contrário, não há risco. A saída da RNA é um valor  $\phi$  que identifica a classe de interesse, conforme Equação 3.7.

$$\phi = \begin{cases} +1, & \text{se usuário em movimento} \\ -1, & \text{se usuário parado} \end{cases} \quad (3.7)$$

Ainda, o segundo classificador consiste em definir o modo de transporte sendo utilizado pelo usuário, ou seja, identificar se a locomoção está sendo realizada em um carro (classes que possuem  $a_1$ ), em uma bicicleta (classes que possuem  $a_2$ ) ou a pé (classes que possuem  $a_3$ ). A saída da RNA é um vetor  $\phi$  que identifica a classe de interesse, conforme Equação

3.8.

$$\phi = \begin{cases} \begin{bmatrix} +1 & -1 & -1 \end{bmatrix}, & \text{se carro} \\ \begin{bmatrix} -1 & +1 & -1 \end{bmatrix}, & \text{se bicicleta} \\ \begin{bmatrix} -1 & -1 & +1 \end{bmatrix}, & \text{se a pé} \end{cases} \quad (3.8)$$

As terceira, quarta e quinta redes neurais decidem se o *smartphone* está sendo utilizado de forma perigosa ou não, de acordo com o modo de transporte detectado pela segunda RNA (carro, bicicleta ou a pé). Para cada modo de transporte, há uma rede neural específica. As classes consideradas como sendo perigosas são:  $a_1b_1c_1d_1$ ,  $a_1b_1c_1d_2$ ,  $a_2b_1c_1d_2$ ,  $a_3b_1c_1d_1$  e  $a_3b_1c_1d_2$ . A saída de cada rede é um valor  $\phi$  que identifica se a situação é de risco ou não, conforme Equação 3.9.

$$\phi = \begin{cases} +1, & \text{se situação de risco} \\ -1, & \text{se situação de não-risco} \end{cases} \quad (3.9)$$

### 3.4 CONSTRUÇÃO DA BASE DE DADOS

O método aqui apresentado faz uso apenas do acelerômetro de um *smartphone*. Dessa forma, não é usado nenhum sensor que dependa de sinais externos, tais como GPS, GIS e rede do celular. Isso permite que a aquisição dos dados seja realizada independente do local. Além do acelerômetro, há outros sensores que poderiam ser utilizados, tais como o giroscópio e o magnetômetro. Entretanto, o acelerômetro é o sensor que fornece mais informações a respeito do modo de transporte do usuário e também apresenta o menor gasto de energia quando comparado com os outros sensores [27].

Todas as aquisições foram realizadas por meio de um *smartphone* com um aplicativo que realiza aquisição e escrita em arquivo dos valores das acelerações dos 3 eixos do acelerômetro do celular, conforme Figura 2.1. A taxa de amostragem é de 500 aquisições por segundo, ou seja, a cada segundo uma tripla  $(x_1, x_2, x_3)$  é adquirida e escrita em arquivo.

Conforme apresentado na Seção 3.2, o sistema é capaz de identificar 18 classes e, para cada uma delas, foram gravados pelo menos 19 minutos de dados. Sendo assim, a base de dados é composta por 342 minutos de dados.

A Figura 3.10 mostra o posicionamento do *smartphone* de acordo com o modo de transporte.

Nas classes pertencentes a  $a_1b_2$ , o dispositivo foi colocado em pé em um suporte posi-

cionado em frente à saída do ar-condicionado do carro, como mostrado na Figura 3.10a. O ar-condicionado e o som do carro permaneceram desligados durante o período de aquisição.

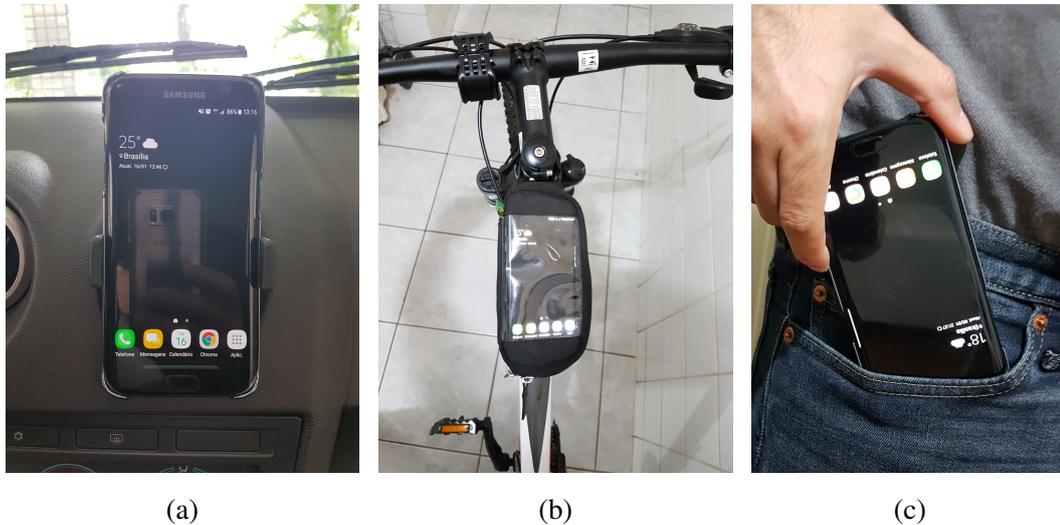


Figura 3.10: Posicionamento do *smartphone* em cada modo de transporte: (a) *smartphone* posicionado em suporte no painel do carro, em frente à saída do ar-condicionado. O dispositivo é colocado em pé com a tela voltada para o motorista. O ar-condicionado e o som do carro permaneceram desligados o tempo todo; (b) *smartphone* colocado dentro de um bolso posicionado no painel da bicicleta com a tela virada para cima; e (c) *smartphone* sendo colocado no bolso. A parte de trás do dispositivo encontra-se virada para a perna do usuário e o aparelho é inserido de cabeça para baixo.

Nas classes pertencentes a  $a_2b_2$ , o dispositivo foi colocado deitado com a tela virada para cima dentro de um bolso protetor posicionado no painel da bicicleta, como mostrado na Figura 3.10b.

Nas classes que possuem  $b_3$ , o dispositivo foi colocado no bolso direito de cabeça para baixo e com a parte de trás virada para a perna do usuário, como mostrado na Figura 3.10c.

A base de dados foi adquirida por meio de um único indivíduo e de forma direta, ou seja, o usuário realizou as medições se colocando em situações reais. As classes que representam perigo foram adquiridas em ambientes vazios e seguros, sem a possibilidade de causar acidentes.

Cada linha de cada arquivo contendo as gravações de cada classe possui quatro valores, separados por ponto e vírgula na seguinte ordem: as medições das acelerações nos eixos  $x_1$ ,  $x_2$ ,  $x_3$  e o tempo, em milissegundos, em relação à última aquisição. Após esse último elemento, é inserida uma quebra de linha.

## 4 RESULTADOS EXPERIMENTAIS

Neste capítulo, serão apresentados o método de treinamento do sistema, a escolha das redes neurais e os resultados obtidos.

### 4.1 TREINAMENTO DO SISTEMA

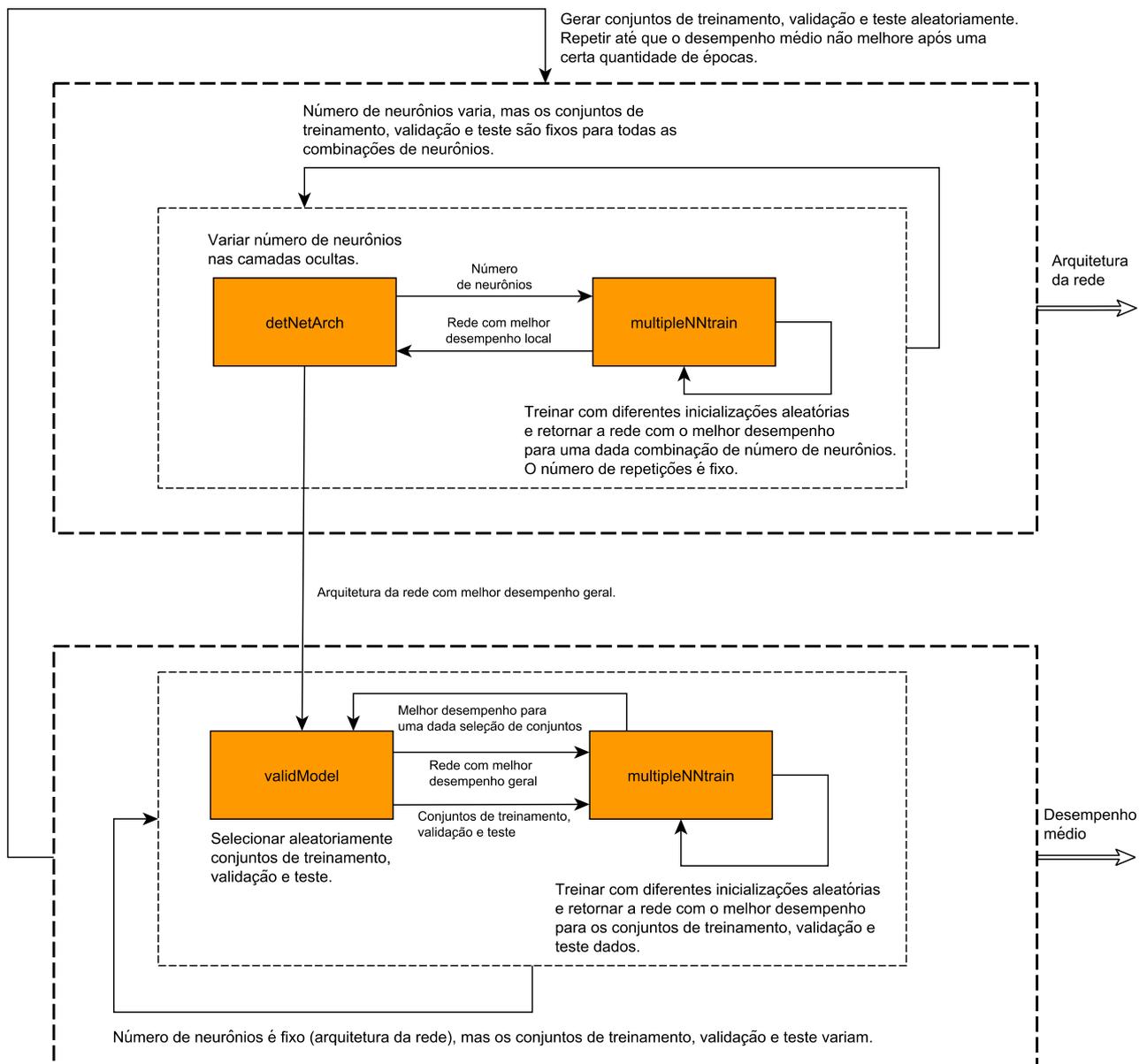


Figura 4.1: Fluxograma do processo de treinamento.

O processo de treinamento de cada rede neural do sistema é apresentado em forma de

fluxograma na Figura 4.1.

Os dados utilizados para a obtenção das métricas aqui apresentadas estão divididos em quatro conjuntos: teste geral, treinamento parcial, validação parcial e teste parcial.

O conjunto de teste geral consiste em 15% da base de dados. Antes de se iniciar o treinamento, 15% dos dados de cada classe são removidos e esses não participam do processo de treinamento de nenhuma rede neural. Esse conjunto é utilizado após a escolha definitiva da melhor rede, sendo um conjunto nunca antes visto pela RNA e, por isso, um bom conjunto de teste para ser aplicado e gerar as métricas finais.

Os conjuntos parciais são aqueles descritos na Seção 2.1.7. São os conjuntos de treinamento, de validação e de teste. Sobre os 85% restantes da base de dados, esses conjuntos são compostos pelas seguintes proporções: o conjunto de treinamento é composto por 70% da base; o de validação, por 15%; e o de teste, 15%.

Todas as redes neurais são do tipo *feed-forward*, são treinadas pelo método de Levenberg-Marquardt e o treinamento é supervisionado. Com exceção da camada de entrada, todos os neurônios utilizam a tangente hiperbólica como função de transferência.

São treinadas redes neurais para tomarem decisões em 5 situações diferentes, compondo o classificador hierárquico apresentado no diagrama da Figura 3.9. As entradas para cada rede seguem o formato especificado nas Equações 3.4 e 3.5.

Por fim, é necessário definir o processo de treinamento de cada rede neural. O início desse processo se dá com a definição do tempo  $s$  da janela. Todas as redes são treinadas utilizando as janelas de tempo definidas pelo conjunto  $T$ , conforme Equação 4.1.

$$T = \{ 2, 4, 6, 8, 10, 12, 14, 16 \} \quad (4.1)$$

Após a definição do tamanho da janela, é necessário treinar as redes para cada combinação de faixas possível. Isso é feito, primeiramente, através do uso de cada faixa em separado e, depois, realizando uma combinação com diferentes faixas. O conjunto que compreende todas as combinações é apresentado na Equação 4.2.

$$A = \{ Lo, Me, Hi, LoMe, LoHi, MeHi, LoMeHi \} \quad (4.2)$$

De forma geral, o processo de treinamento de cada uma das 5 redes neurais do classificador hierárquico passa pelas etapas abaixo.

1. Para cada janela de  $s$  segundos em  $T$ , os passos de 2 a 6 são executados.
2. A extração de características é realizada, formando o conjunto  $F$ .
3. Para cada elemento do conjunto  $A$  de faixas, os passos de 4 a 6 são executados.

4. 15% das amostras de cada classe de interesse são separadas e deixadas de fora do treinamento. Este conjunto é chamado de conjunto de teste geral.
5. A partir de  $F$ , são formados os subconjuntos de entradas ( $F_{movimento}$ ,  $F_{classeMovimento}$ ,  $F_{carro}$ ,  $F_{bicicleta}$  ou  $F_{aPe}$ ) e saídas para cada RNA.
6. A rede neural é treinada de acordo com o método de Monte Carlo descrito na Seção 4.1.1.

O processo de treinamento de cada uma das redes neurais que compõem o classificador hierárquico passa por duas etapas. Primeiramente, a arquitetura da rede neural é definida. Tendo definido a arquitetura, é realizada a validação do sistema pelo método de Monte Carlo.

A definição da arquitetura da RNA consiste em variar a quantidade de neurônios nas duas camadas escondidas de acordo com a quantidade  $Q$  de elementos fornecidos para o treinamento da rede e da quantidade  $\tau$  de classes nas quais a rede pode classificar cada entrada. A quantidade de neurônios na segunda camada oculta varia de 2 em 2 dentro de um certo intervalo, enquanto a quantidade de neurônios na primeira camada oculta permanece fixa. Após o treinamento da rede para cada quantidade de neurônios na segunda camada ter sido realizado, são adicionados mais dois neurônios na primeira camada e o processo se repete até que o número de neurônios nas camadas atinja um valor máximo.

A quantidade inicial de neurônios na primeira camada escondida,  $MinH_1$ , é incrementada de 2 em 2 até o valor máximo,  $MaxH_1$ , definidos pela Equação 4.3.

$$\begin{aligned} MinH_1 &= \left\lfloor \frac{MinH_1}{2} \right\rfloor \\ MaxH_1 &= \sqrt{\frac{Q}{\tau}} \end{aligned} \tag{4.3}$$

Com a quantidade  $\theta$  de neurônios fixada na primeira camada, a quantidade de neurônios na segunda camada oculta varia de 2 em 2 a partir do valor mínimo  $MinH_2$  até o máximo  $MaxH_2$ , calculados pela Equação 4.4.

$$\begin{aligned} MinH_2 &= \left\lfloor \frac{\theta}{1.5} \right\rfloor \\ MaxH_2 &= \left\lfloor \frac{\theta}{3} \right\rfloor \end{aligned} \tag{4.4}$$

Para cada arquitetura, a rede é treinada 10 vezes ou menos, caso algum treinamento resulte em 100% de acerto. A rede com a maior taxa de acerto é guardada. Ao final do treinamento de todas as arquiteturas possíveis, aquela que apresenta a maior taxa de acerto sobre o conjunto de teste parcial é escolhida para a próxima etapa: validação pelo método de Monte Carlo.

### 4.1.1 Treinamento com Monte Carlo

Conforme apresentado na Seção 2.1.7.4, o método de Monte Carlo não possui uma única implementação possível, sendo a definição de uma ideia que deve ser seguida. Para o treinamento de cada uma das redes neurais que compõem o classificador hierárquico, foi utilizada uma abordagem que implementa o método de Monte Carlo, conforme Figura 4.1.

Dada a arquitetura de uma rede neural, os passos abaixo são realizados.

1. A base de dados é embaralhada de forma aleatória, formando novos conjuntos de treinamento, validação e teste parciais.
2. A rede neural é treinada até 10 vezes (caso atinja uma taxa de acerto de 100%, o treinamento é dado como terminado), sendo selecionada aquela que apresenta a maior taxa de acerto.
3. A taxa de acerto é guardada.
4. Caso os passos anteriores ainda não tenham sido executados 50 vezes, deve-se retornar ao Passo 1.
5. É calculada a média das taxas de acerto de todas as 50 redes neurais selecionadas.
6. O valor médio calculado no Passo 5 é tomado como a taxa de acerto média daquela arquitetura.

Então, o processo de definir a arquitetura e de validá-la por Monte Carlo é realizado novamente. O processo de validação resulta em uma taxa de acerto média, que é comparada com a taxa obtida na execução anterior. Caso a taxa de acerto atual seja menor ou igual à taxa obtida na execução anterior, uma variável chamada  $perfTest$ , inicializada em 0, é incrementada. Caso a taxa atual seja maior, a variável é zerada. Os dois processos são repetidos até que pelo menos uma das condições abaixo sejam verdadeiras:

1.  $perfTest \geq 2$  ou
2. A quantidade de execuções do processo de definir arquitetura e validar a rede por Monte Carlo é maior ou igual a 20.

Ao término do processo de treinamento, a rede que apresenta a maior taxa de acerto média é selecionada e o conjunto de teste geral é submetido à ela. A taxa de acerto resultante representa o resultado final para aquela classe, dadas as faixas de interesse e a janela de dados.

## 4.2 RESULTADOS

Após o processo de treinamento das redes neurais, o conjunto de teste geral foi submetido a cada RNA.

A Tabela 4.1 mostra as taxas de acerto e as arquiteturas das redes obtidas, enquanto que a Figura 4.2 representa a superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto obtidas. Neste caso, foi utilizado todo o conjunto de teste geral. A rede neural classifica as entradas indicando se o usuário está em movimento (classes  $a_1b_1c_1d_1$ ,  $a_1b_1c_1d_2$ ,  $a_1b_2c_1d_3$ ,  $a_2b_1c_1d_1$ ,  $a_2b_2c_1d_3$ ,  $a_2b_3c_1d_3$ ,  $a_3b_1c_1d_1$ ,  $a_3b_1c_1d_2$  e  $a_3b_3c_1d_3$ ) ou se está parado (classes  $a_1b_1c_2d_1$ ,  $a_1b_1c_2d_2$ ,  $a_1b_2c_2d_3$ ,  $a_2b_1c_2d_2$ ,  $a_2b_2c_2d_3$ ,  $a_2b_3c_2d_3$ ,  $a_3b_1c_2d_1$ ,  $a_3b_1c_2d_2$ ,  $a_3b_3c_2d_3$ ). A melhor RNA obtida foi com o uso dos três faixas e com uma janela de 14 segundos.

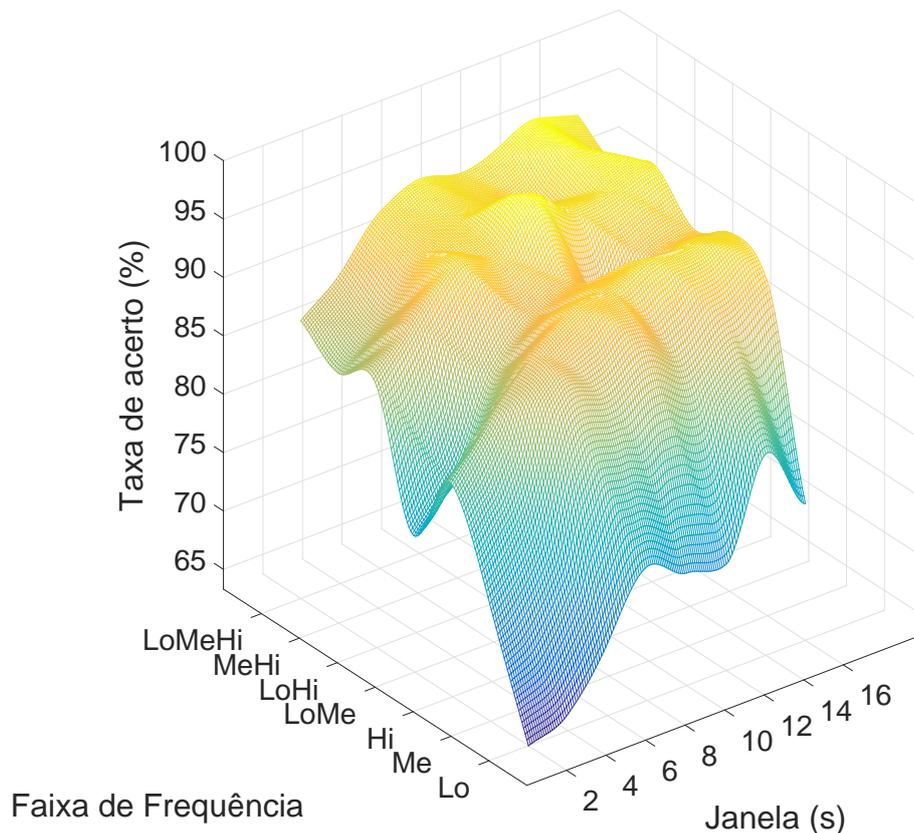


Figura 4.2: Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica se o celular está em movimento ou não. Resultados obtidos a partir da Tabela 4.1.

A Tabela 4.2 mostra as taxas de acerto e as arquiteturas das redes obtidas assumindo

Janela	2	4	6	8	10	12	14	16	Média
Faixas									
Lo	63,31% - 36 - 12	64,75% - 23 - 12	69,55% - 27 - 17	74,79% - 20 - 7	72,95% - 22 - 7	72,43% - 13 - 6	80,57% - 20 - 9	75,00% - 17 - 6	<b>71,67%</b>
Me	73,11% - 38 - 19	77,50% - 25 - 14	86,42% - 33 - 11	86,57% - 18 - 6	88,70% - 22 - 11	86,83% - 19 - 10	89,57% - 24 - 16	91,85% - 15 - 7	<b>85,07%</b>
Hi	81,41% - 34 - 17	85,87% - 31 - 12	89,71% - 23 - 8	92,33% - 18 - 8	92,81% - 20 - 7	94,24% - 23 - 14	94,31% - 14 - 9	94,02% - 13 - 6	<b>90,59%</b>
LoMe	74,97% - 64 - 29	79,15% - 23 - 10	86,83% - 21 - 11	90,14% - 16 - 11	91,44% - 22 - 15	87,24% - 19 - 10	91,47% - 24 - 10	92,39% - 15 - 7	<b>86,70%</b>
LoHi	85,67% - 46 - 29	89,57% - 25 - 8	93,42% - 25 - 14	92,88% - 22 - 9	96,57% - 14 - 9	94,24% - 15 - 9	94,79% - 12 - 4	95,65% - 21 - 7	<b>92,85%</b>
MeHi	84,71% - 38 - 13	87,93% - 37 - 16	92,59% - 31 - 20	93,97% - 16 - 7	93,84% - 16 - 5	93,00% - 13 - 6	96,21% - 12 - 4	93,48% - 13 - 4	<b>91,97%</b>
LoMeHi	87,11% - 54 - 32	89,85% - 23 - 14	93,83% - 31 - 20	95,34% - 24 - 14	94,52% - 16 - 5	95,47% - 17 - 10	96,68% - 20 - 13	95,65% - 13 - 4	<b>93,56%</b>

Tabela 4.1: Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem se a janela em questão indica movimento ou não. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente.

que o usuário encontra-se em movimento, independentemente do modo de transporte. A Figura 4.3 representa a superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto obtidas. Nesse caso, foram utilizados os dados do conjunto de teste geral e que fazem parte da divisão  $c_1$ . A rede classifica cada entrada como pertencente ao modo de transporte carro (classes  $a_1b_1c_1d_1$ ,  $a_1b_1c_1d_2$  e  $a_1b_2c_1d_3$ ), bicicleta (classes  $a_2b_1c_1d_1$ ,  $a_2b_2c_1d_3$  e  $a_2b_3c_1d_3$ ) ou a pé (classes  $a_3b_1c_1d_1$ ,  $a_3b_1c_1d_2$  e  $a_3b_3c_1d_3$ ).

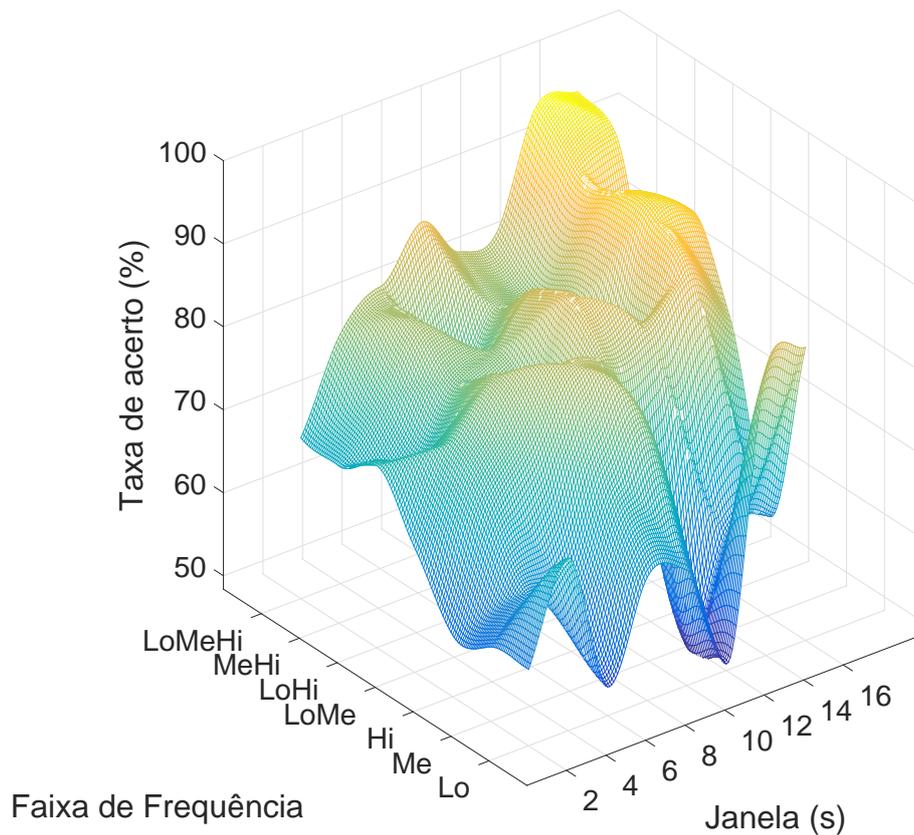


Figura 4.3: Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica qual o modo de transporte sendo utilizado. Resultados obtidos a partir da Tabela 4.2.

A Tabela 4.3 mostra as taxas de acerto e as arquiteturas das redes obtidas para as situações em que o usuário está se locomovendo em um carro, sendo que a Figura 4.4 representa a superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto obtidas. Nesse caso, foram utilizados os dados do conjunto de teste geral e que pertencem às classes  $a_1b_1c_1d_1$ ,  $a_1b_1c_1d_2$  e  $a_1b_2c_1d_3$ . A rede classifica os dados de entrada como sendo situação de risco (as duas primeiras classes) ou não (última classe).

Janela / Faixas	2	4	6	8	10	12	14	16	Média
Lo	57,61% - 8 - 5	69,21% - 8 - 3	51,85% - 7 - 4	63,24% - 4 - 3	62,33% - 5 - 2	49,19% - 5 - 2	82,86% - 5 - 2	83,70% - 5 - 2	<b>65,00%</b>
Me	57,34% - 8 - 3	61,85% - 6 - 4	61,32% - 7 - 2	72,43% - 6 - 4	76,71% - 5 - 2	48,39% - 5 - 2	60,95% - 5 - 2	60,87% - 3 - 1	<b>62,48%</b>
Hi	55,83% - 12 - 8	69,21% - 8 - 5	75,31% - 9 - 5	82,70% - 6 - 4	80,82% - 5 - 2	82,26% - 5 - 2	87,62% - 5 - 2	81,52% - 5 - 2	<b>76,91%</b>
LoMe	63,37% - 8 - 5	74,11% - 6 - 4	80,25% - 7 - 4	76,76% - 4 - 3	86,30% - 5 - 2	76,61% - 5 - 2	92,38% - 5 - 2	91,30% - 5 - 2	<b>80,14%</b>
LoHi	70,32% - 14 - 7	69,48% - 8 - 3	76,95% - 7 - 4	78,92% - 4 - 3	84,25% - 5 - 2	81,45% - 5 - 2	92,38% - 5 - 2	78,26% - 5 - 2	<b>79,00%</b>
MeHi	67,35% - 10 - 7	73,57% - 8 - 3	81,07% - 5 - 2	72,43% - 6 - 2	79,45% - 5 - 2	79,84% - 5 - 2	92,38% - 5 - 2	94,57% - 5 - 2	<b>80,08%</b>
LoMeHi	67,76% - 8 - 3	78,47% - 8 - 5	82,30% - 9 - 5	88,11% - 6 - 4	82,88% - 5 - 2	83,87% - 5 - 2	97,14% - 5 - 2	96,74% - 5 - 2	<b>84,66%</b>

Tabela 4.2: Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem qual o modo de transporte sendo utilizado, dado que o usuário está em movimento. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente.

Faixas	Janela																Média
	2	4	6	8	10	12	14	16									
Lo	93,21% - 13 - 8	98,78% - 8 - 3	96,30% - 6 - 2	100% - 5 - 2	96,97% - 5 - 2	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	98,16%
Me	93,21% - 13 - 4	93,90% - 10 - 3	94,44% - 6 - 2	97,62% - 5 - 2	100% - 5 - 2	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	97,40%
Hi	95,68% - 19 - 8	90,24% - 8 - 5	94,44% - 12 - 4	88,10% - 7 - 4	90,91% - 5 - 2	96,43% - 4 - 2	95,83% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	93,95%
LoMe	97,53% - 21 - 13	100% - 14 - 7	100% - 6 - 2	100% - 5 - 2	100% - 5 - 2	100% - 4 - 2	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	99,69%
LoHi	98,15% - 11 - 4	98,78% - 8 - 3	100% - 6 - 2	97,62% - 5 - 2	96,97% - 5 - 2	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	98,94%
MeHi	98,15% - 11 - 4	100% - 8 - 3	98,15% - 6 - 4	100% - 5 - 2	96,97% - 5 - 2	100% - 4 - 1	87,50% - 6 - 4	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	97,60%
LoMeHi	96,91% - 11 - 4	98,78% - 8 - 3	100% - 6 - 2	100% - 5 - 2	100% - 5 - 2	96,43% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	100% - 4 - 1	99,02%

Tabela 4.3: Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem se o *smartphone* está sendo utilizado em uma situação de risco, dado que o usuário está em um carro em movimento. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente.

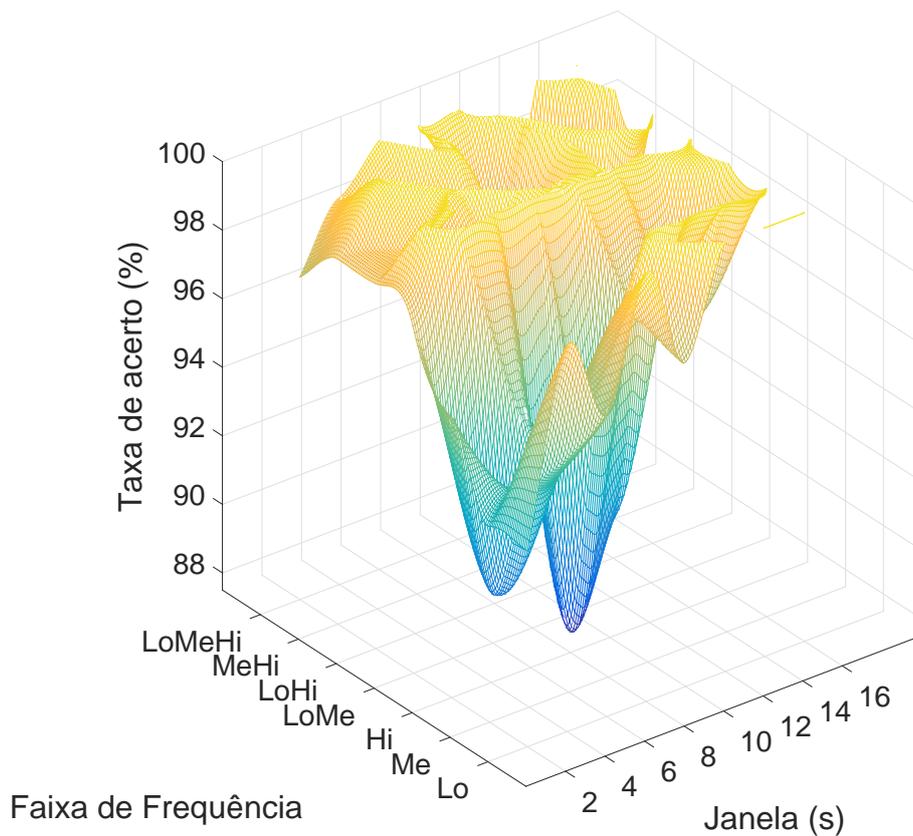


Figura 4.4: Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica se o celular está sendo utilizado em situação de risco ou não, dado que o modo de transporte é um carro. Resultados obtidos a partir da Tabela 4.3.

A Tabela 4.4 mostra as taxas de acerto e as arquiteturas das redes obtidas para as situações em que o usuário está se locomovendo em uma bicicleta, sendo que a Figura 4.5 representa a superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto obtidas. Neste caso, foram utilizados os dados do conjunto de teste geral e que pertencem às classes  $a_2b_1c_1d_1$ ,  $a_2b_2c_1d_3$  e  $a_2b_3c_1d_3$ . A rede classifica os dados de entrada como sendo situação de risco (a primeira classe) ou não (as duas últimas classes).

Janela		2	4	6	8	10	12	14	16	Média
Faixas										
Lo		73,46% - 17 - 10	67,07% - 12 - 4	77,78% - 6 - 4	80,49% - 7 - 2	69,70% - 7 - 2	85,71% - 4 - 1	69,56% - 4 - 1	80,95% - 6 - 2	<b>75,59%</b>
Me		59,26% - 11 - 6	62,20% - 10 - 3	57,41% - 10 - 5	70,73% - 7 - 4	69,70% - 5 - 2	64,29% - 8 - 5	78,26% - 4 - 1	76,19% - 6 - 2	<b>67,26%</b>
Hi		72,84% - 13 - 4	78,05% - 14 - 7	70,37% - 8 - 3	53,66% - 7 - 4	48,48% - 7 - 4	82,14% - 6 - 2	73,91% - 4 - 1	47,62% - 6 - 2	<b>65,88%</b>
LoMe		72,84% - 19 - 10	75,61% - 8 - 5	74,07% - 8 - 3	73,17% - 5 - 2	72,73% - 5 - 2	67,86% - 4 - 3	82,61% - 6 - 2	66,67% - 6 - 2	<b>73,20%</b>
LoHi		82,72% - 13 - 8	75,61% - 14 - 7	75,93% - 8 - 3	75,61% - 5 - 2	75,76% - 5 - 2	92,86% - 6 - 2	78,26% - 6 - 2	52,38% - 6 - 2	<b>76,14%</b>
MeHi		79,01% - 11 - 4	75,61% - 8 - 3	64,81% - 8 - 5	73,17% - 5 - 2	63,64% - 7 - 4	78,57% - 4 - 1	82,61% - 6 - 2	47,62% - 4 - 1	<b>70,63%</b>
LoMeHi		80,25% - 17 - 6	76,83% - 8 - 3	75,93% - 6 - 2	63,41% - 5 - 2	63,64% - 5 - 2	85,71% - 6 - 2	86,96% - 6 - 2	57,14% - 6 - 4	<b>73,73%</b>

Tabela 4.4: Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem se o *smartphone* está sendo utilizado em uma situação de risco, dado que o usuário está em uma bicicleta e em movimento. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios nas primeira e segunda camadas ocultas, respectivamente.

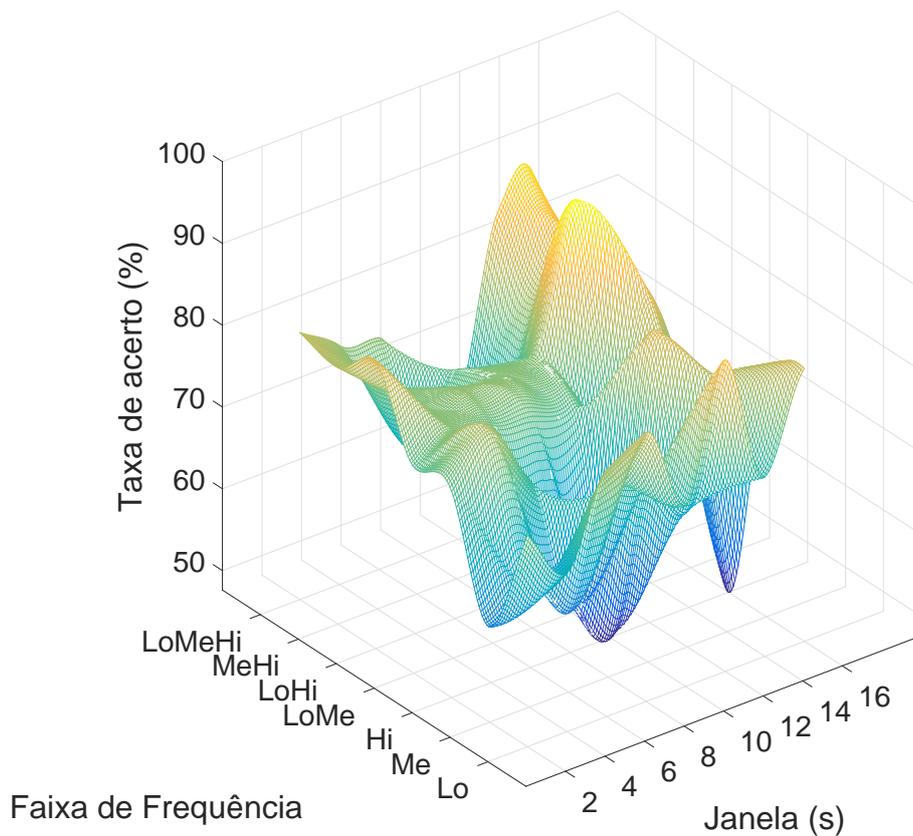


Figura 4.5: Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica se o celular está sendo utilizado em situação de risco ou não, dado que o modo de transporte é uma bicicleta. Resultados obtidos a partir da Tabela 4.4.

A Tabela 4.5 mostra as taxas de acerto e as arquiteturas das redes obtidas para as situações em que o usuário está se locomovendo a pé, sendo que a Figura 4.6 representa a superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto obtidas. Neste caso, foram utilizados os dados do conjunto de teste geral e que pertencem às classes  $a_3b_1c_1d_1$ ,  $a_3b_1c_1d_2$  e  $a_3b_3c_1d_3$ . A rede classifica os dados de entrada como sendo situação de risco (as duas primeiras classes) ou não (última classe).

Faixas	Janela																Média
	2	4	6	8	10	12	14	16									
Lo	74,07% - 13 - 4	73,17% - 8 - 5	83,33% - 12 - 6	66,67% - 5 - 2	81,25% - 5 - 2	82,14% - 6 - 4	60,87% - 8 - 5	80,00% - 4 - 3									<b>75,19%</b>
Me	79,63% - 13 - 8	78,05% - 10 - 7	94,44% - 8 - 5	88,10% - 7 - 2	93,75% - 5 - 2	89,29% - 4 - 1	100% - 4 - 1	100% - 4 - 1									<b>90,41%</b>
Hi	70,37% - 19 - 12	71,95% - 10 - 3	88,89% - 6 - 2	95,24% - 5 - 2	81,25% - 9 - 3	100% - 4 - 1	86,96% - 4 - 1	100% - 4 - 1									<b>86,83%</b>
LoMe	86,42% - 11 - 4	74,39% - 14 - 5	88,89% - 6 - 2	88,10% - 5 - 2	78,13% - 9 - 3	75,00% - 4 - 1	95,65% - 4 - 1	100% - 4 - 1									<b>85,82%</b>
LoHi	80,86% - 15 - 5	78,05% - 10 - 5	94,44% - 6 - 4	92,86% - 9 - 3	93,75% - 7 - 2	85,71% - 4 - 1	86,96% - 4 - 1	85,00% - 4 - 1									<b>87,20%</b>
MeHi	82,10% - 11 - 4	74,39% - 8 - 3	92,59% - 6 - 2	95,24% - 5 - 2	87,50% - 7 - 4	85,71% - 4 - 1	86,96% - 4 - 1	100% - 6 - 2									<b>88,06%</b>
LoMeHi	88,89% - 11 - 6	81,70% - 8 - 5	92,59% - 6 - 2	100% - 5 - 2	90,63% - 9 - 5	92,86% - 4 - 1	95,65% - 4 - 1	95,00% - 4 - 1									<b>92,16%</b>

Tabela 4.5: Tabela referente aos resultados obtidos pelo treinamento de redes neurais que decidem se o *smartphone* está sendo utilizado em uma situação de risco, dado que o usuário está a pé e em movimento. O tamanho da janela está em segundos, e os resultados apresentam a taxa de acerto sobre o conjunto de dados separado do treinamento e a quantidade de neurônios na primeira e segunda camadas ocultas, respectivamente.

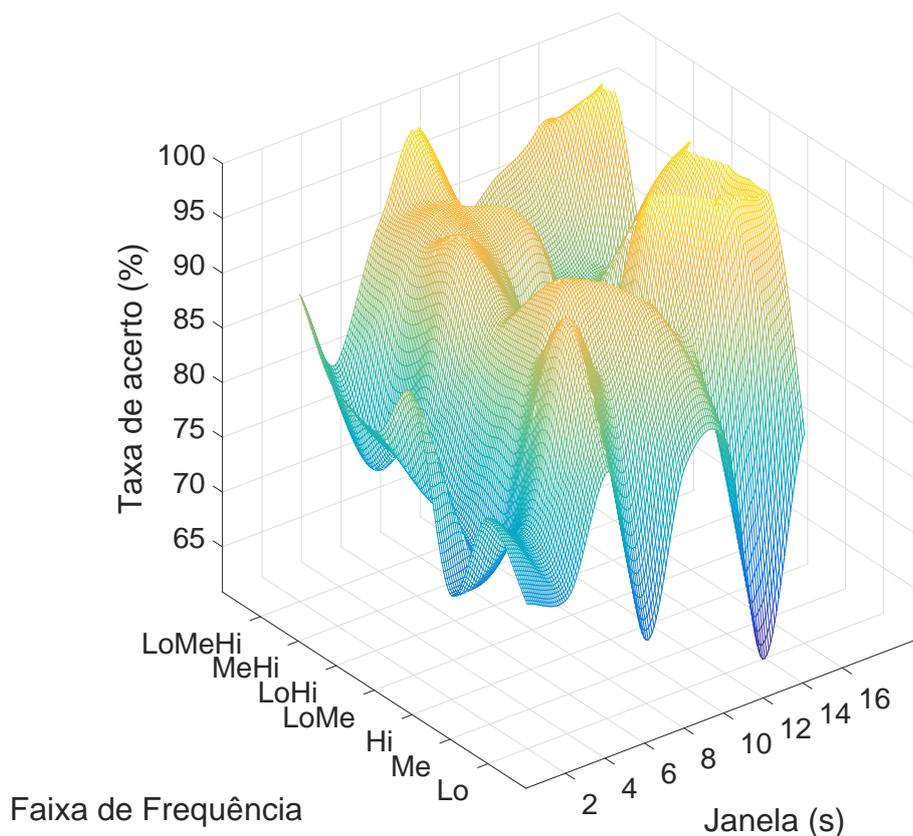


Figura 4.6: Superfície que relaciona as faixas de frequência e as janelas com as taxas de acerto para a RNA que indica se o celular está sendo utilizado em situação de risco ou não, dado que o modo de transporte é a pé. Resultados obtidos a partir da Tabela 4.5.

A tendência geral observada nas janelas é que quanto maior a janela e quanto mais faixas forem associadas, maior será a taxa de acerto, ou seja, quanto mais para a direita e para baixo nas tabelas, as taxas de acerto tendem a ser melhores. Dessa forma, analisando as médias de cada linha, é possível notar que o uso das três faixas resulta em uma maior taxa de acerto média na maioria dos casos.

Para testar o classificador hierárquico, foi adotado o uso das três faixas e foram realizados os testes sobre todo o conjunto de teste geral para cada tamanho de janela. Os resultados são apresentados na Tabela 4.6 e na Figura 4.7, com a taxa de acerto média do sistema sendo de 87,70%. A avaliação realizada classifica cada entrada apenas como sendo uma situação de risco ou não.

Janela	2	4	6	8	10	12	14	16	Média
Faixas	81,21%	85,32%	86,01%	90,41%	86,99%	87,65%	94,31%	91,30%	
LoMeHi									<b>87,70%</b>

Tabela 4.6: Tabela referente aos resultados obtidos pelo classificador hierárquico tendo como entrada os dados do conjunto de teste geral. O classificador utiliza todas as faixas e as redes neurais adotadas são aquelas apresentadas em suas respectivas tabelas. O tamanho da janela está em segundos.

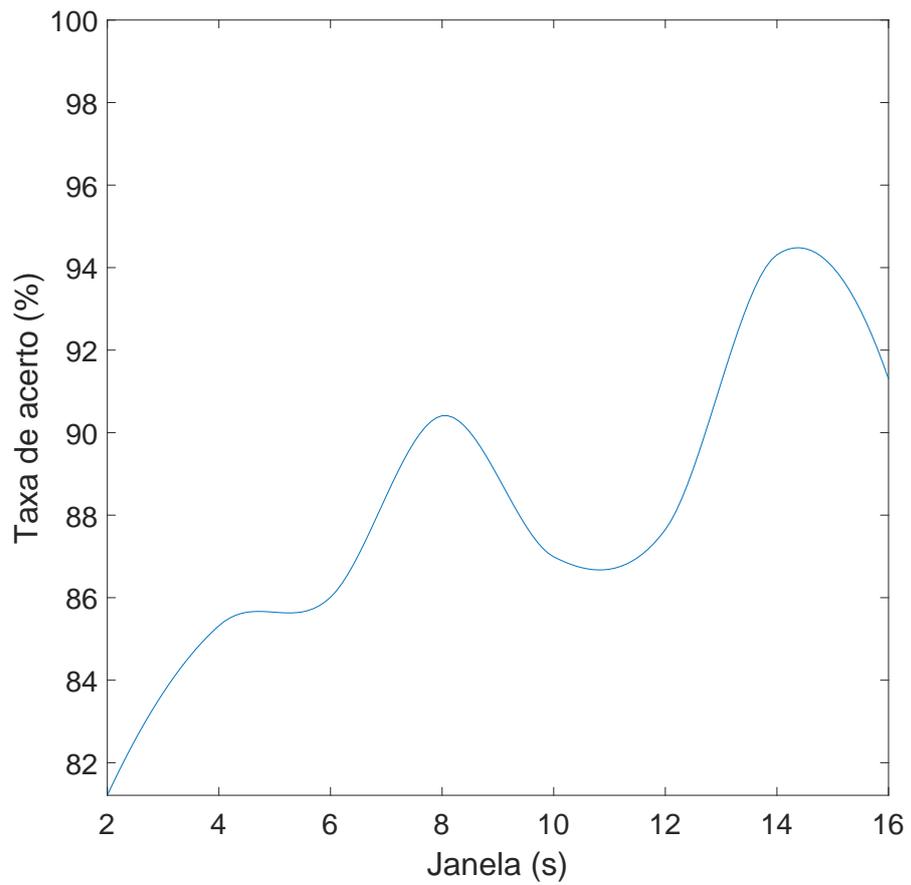


Figura 4.7: Curva que relaciona o tamanho das janelas com as taxas de acerto para o classificador hierárquico, dado que foram utilizadas as faixas de frequência do conjunto *LoMeHi*. Resultados obtidos a partir da Tabela 4.6.

## 5 CONCLUSÃO

O uso de *smartphones* é uma faca de dois gumes. Por um lado, é possível realizar uma grande gama de atividades e se comunicar com pessoas em praticamente qualquer lugar do mundo. Por outro lado, seu uso em momentos inadequados se torna um risco de vida tanto para o usuário quanto para aqueles ao seu redor. Dessa forma, é natural o surgimento de técnicas capazes de identificar situações de uso do celular que implicam em riscos de acidentes.

Enquanto outros trabalhos se limitam à detecção do modo de transporte, o método aqui apresentado realiza a detecção do modo de transporte (carro, bicicleta e a pé) e o detalha em 3 subníveis: local de transporte (mão, painel e bolso), situação (movimento e parado) e modo de interação (em ligação, olhando/digitando e sem interação).

Tornar o acelerômetro o único sensor no qual o método se baseia fornece uma grande vantagem sobre a maioria dos outros métodos de detecção de transporte: a não dependência de sinais externos, tais como a cobertura da rede móvel do celular ou a visibilidade do GPS. Dessa forma a técnica pode ser aplicada em quaisquer situação e lugar.

A técnica aqui apresentada alcança uma taxa de acerto de 94,31% com uma janela de 14 segundos de aquisições do acelerômetro, fazendo uso de apenas 36 características. Quando esses valores são comparados com os resultados obtidos na Tabela 2.1 é de suma importância notar que:

- a taxa de acerto obtida é competitiva com o estado da arte;
- alguns desses métodos [27, 29] utilizam mais de 300 características para que possam alcançar seus resultados, enquanto que o método aqui proposto o faz com apenas 36;
- não é necessário nenhum sensor além do acelerômetro do próprio celular e
- as classes identificadas permitem maior grau de detalhamento e, portanto, aumenta-se a capacidade de detectar nuances na identificação de situações de risco.

O algoritmo apresenta vantagens em relação ao estado da arte. É importante notar que, no melhor caso, é necessária uma janela de 14 segundos até que o padrão possa ser detectado. Com 8 segundos, entretanto, já foi possível atingir uma taxa de acerto de 90,41%. No caso médio, foi possível atingir a taxa de 87,70% de acerto.

Trabalhos futuros envolvem 3 pontos principais, descritos abaixo.

1. Adicionar mais modos de transporte detectáveis, tais como moto, patins, patinete e skate.

2. Realizar a coleta de mais dados para aumentar a base de dados disponível.
3. Fazer coletas com mais indivíduos, visto que a base de dados é composta por apenas um único usuário.

Por fim, o método proposto atinge o seu objetivo, pois é eficiente e capaz de detectar situações de risco de acidentes com alto grau de precisão, alcançando valores próximos aos valores descritos na literatura.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J. F. Júnior, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, “Driver behavior profiling: An investigation with different smartphone sensors and machine learning.” *PLoS ONE*, 2017. 1
- [2] W. Sigloch, “Pedestrians crossing streets: Distraction by smartphone poses risks,” DEKRA, Handwerkstraße 15, D-70565 Stuttgart, Tech. Rep., Abril 2016. [Online]. Available: [www.dekra.com/en/press](http://www.dekra.com/en/press) 1
- [3] J. W. Ayers, E. C. Leas, M. Dredze, J.-P. Allem, J. G. Grabowski, and L. Hill, “Pokémon go - a new distraction for drivers and pedestrians,” *JAMA Internal Medicine*, vol. 176, no. 12, pp. 1865–1866, 2016. 1
- [4] V. S. Thakare, N. N. Patil, and J. S. Sonawane, “Survey on image texture classification techniques,” *International Journal of Advancements in Technology*, 2013. 2.1.5
- [5] M. Hall-Beyer, “GlcM texture: A tutorial,” The University of Calgary, Tech. Rep., Março 2017. 2.1.5, 2.1.5
- [6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2000. 2.1.6
- [7] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936. 2.1.6
- [8] G. P. Zhang, “Neural networks for classification: A survey,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 30, pp. 451 – 462, 2000. 2.1.7
- [9] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, Agosto 2015. 2.1.7.1, 2.1.7.2
- [10] S. O. Haykin, *Neural Networks and Learning Machines*. Prentice Hall, 2008. 2.1.7.1
- [11] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox: User’s Guide*. MathWorks, 2015. 2.1.7.1.1, 2.1.7.3
- [12] K. Madsen, N. Nielsen, and O. Tingleff, “Methods for nonlinear least squares problems,” Technical University of Denmark, Tech. Rep., 2004. 2.1.7.3
- [13] M. T. Hagan and M. B. Menhaj, “Training feedforward networks with the marquardt algorithm,” *IEEE Transactions on Neural Networks*, vol. 5, no. 6, Novembro 1994. 2.1.7.3

- [14] H. Yu and B. M. Wilamowski, “Levenberg-marquardt training,” in *Intelligent Systems*, Fevereiro 2011. 2.1.7.3.2, 2.1.7.3.2
- [15] K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” *The Quarterly of Applied Mathematics*, pp. 164–168, Julho 1944. 2.1.7.3.2
- [16] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the Society for Industrial and Applied Mathematics*, pp. 431–441, Junho 1963. 2.1.7.3.2
- [17] M. I. A. Lourakis, “A brief description of the levenberg-marquardt algorithm implemented by *levmar*,” 2005. 2.1.7.3.2, 2.1.7.3.2, 2.1.7.3.2
- [18] B. D. Ripley, *Stochastic Simulation*. Wiley-Interscience, Março 2006. 2.1.7.4
- [19] S. S. Sawilowsky, “You think you’ve got trivials?” *Journal of Modern Applied Statistical Methods*, pp. 218–225, 2003. 2.1.7.4
- [20] H. L. Anderson, “Metropolis, monte carlo and the maniac,” *Los Alamos Science*, pp. 96–108, 1986. 2.1.7.4
- [21] N. Metropolis, “The beginning of the monte carlo method,” *Los Alamos Science Special Issue*, pp. 125–130, 1987. 2.1.7.4
- [22] S. Gordon, *The Normal Distribution*. The University of Sydney, 2006. 2.1.7.4
- [23] S. Shapsough, A. Hesham, Y. Elkhorary, I. A. Zualkernan, and F. Aloul, “Emotion recognition using mobile phones,” in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Setembro 2016, pp. 1–6. 2.2
- [24] A. Sahyoun, K. Chehab, O. Al-Madani, F. Aloul, and A. Sagahyoon, “Parknosis: Diagnosing parkinson’s disease using mobile phones,” in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Setembro 2016, pp. 1–6. 2.2
- [25] A. Anjum and M. U. Ilyas, “Activity recognition using smartphone sensors,” in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Janeiro 2013, pp. 914–919. 2.2
- [26] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, Março 2011. [Online]. Available: <http://doi.acm.org/10.1145/1964897.1964918> 2.2
- [27] B. Aguiar, J. Silva, T. Rocha, S. Carneiro, and I. Sousa, “Monitoring physical activity and energy expenditure with smartphones,” in *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Junho 2014, pp. 664–667. 2.2, 3.4, 5

- [28] P. Widhalm, P. Nitsche, and N. Brändie, “Transport mode detection with realistic smartphone sensor data,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Novembro 2012, pp. 573–576. 2.2
- [29] N. Cardoso, J. Madureira, and N. Pereira, “Smartphone-based transport mode detection for elderly care,” in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Setembro 2016, pp. 1–6. 2.2, 5
- [30] M. E. L. Mejia, J. R. G. Montalvo, F. M. Mena, and F. A. M. Ramirez, “Intelligent platform for non intrusive assistance of elderly people,” *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2433–2439, Maio 2016. 2.2
- [31] C. Figueira, R. Matias, and H. Gamboa, “Body location independent activity monitoring,” in *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4: BIOSIGNALS, (BIOSTEC 2016)*, 2016, pp. 190–197. 2.2