

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
BIBLIOTECA UNIVERSITÁRIA**

Alana de Melo e Souza

**EXTRAÇÃO E CLASSIFICAÇÃO DOS PARÂMETROS  
DO CORPO HUMANO PARA ANÁLISE E  
RECONHECIMENTO DA MARCHA**

Florianópolis

2017



Alana de Melo e Souza

**EXTRAÇÃO E CLASSIFICAÇÃO DOS PARÂMETROS  
DO CORPO HUMANO PARA ANÁLISE E  
RECONHECIMENTO DA MARCHA**

Dissertação submetido ao Programa  
de Pós-Graduação em Engenharia de  
Automação e Sistemas para a obtenção  
do Grau de Mestre em Engenharia de  
Automação e Sistemas.  
Universidade Federal de Santa Cata-  
rina  
Orientador: Prof. Dr. Marcelo Ri-  
cardo Stemmer

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Souza , Alana de Melo e

Extração e classificação dos parâmetros do corpo humano  
para análise e reconhecimento da marcha / Alana de Melo e  
Souza ; orientador, Marcelo Ricardo Stemmer -  
Florianópolis, SC, 2017.

103 p.

Dissertação (mestrado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico. Programa de Pós-Graduação em  
Engenharia de Automação e Sistemas.

Inclui referências

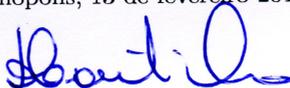
1. Engenharia de Automação e Sistemas. 2. Reconhecimento  
de padrões. 3. Kinect. 4. Análise Marcha. 5. Biometria. I.  
Ricardo Stemmer, Marcelo . II. Universidade Federal de  
Santa Catarina. Programa de Pós-Graduação em Engenharia de  
Automação e Sistemas. III. Título.

Alana de Melo e Souza

**EXTRAÇÃO E CLASSIFICAÇÃO DOS PARÂMETROS  
DO CORPO HUMANO PARA ANÁLISE E  
RECONHECIMENTO DA MARCHA**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Engenharia de Automação e Sistemas”, e aprovado em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

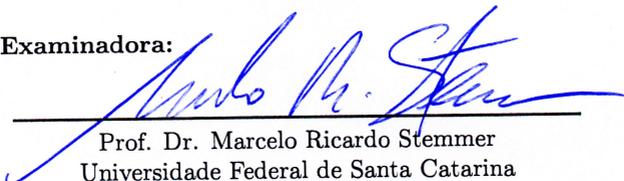
Florianópolis, 13 de fevereiro 2017.



---

Prof. Dr. Daniel Ferreira Coutinho  
Universidade Federal de Santa Catarina  
Coordenador

**Banca Examinadora:**



---

Prof. Dr. Marcelo Ricardo Stemmer  
Universidade Federal de Santa Catarina  
Orientador



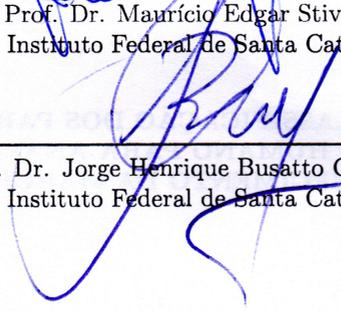
---

Prof. Dr. Mário Lucio Roloff  
Instituto Federal de Santa Catarina



---

Prof. Dr. Maurício Edgar Stivanello  
Instituto Federal de Santa Catarina



---

Prof. Dr. Jorge Henrique Busatto Casagrande  
Instituto Federal de Santa Catarina

Dedico esse trabalho aos meus pais que me deram muito apoio nos momentos mais difíceis da minha vida, aos meus professores que me ensinaram que por mais que achamos que o nosso conhecimento já está bem profundo, estamos enganados, pois o conhecimento é algo que está sempre se renovando.



## **AGRADECIMENTOS**

Aos meus pais, Heloisa e Helder, e minha irmã Alessandra, por toda estrutura, apoio, incentivo e carinho.

Meu querido amigo Edwin Calla, que sempre ficou de meu lado em todo momento.

Ao meu orientador Marcelo Ricardo Stemmer, pelo empenho dedicado à elaboração deste trabalho.

À Universidade Federal de Santa Catarina (UFSC) e aos professores do Programa de Pós-Graduação em Engenharia de Automação e Sistemas (PPGEAS) por contribuírem para o meu aprendizado.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) pelo apoio financeiro.



Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.

(Charles Chaplin, 1977)



## RESUMO

A análise da marcha humana é considerada como uma nova ferramenta biométrica pela capacidade de obter as métricas do corpo à distância. Os identificadores biométricos possuem propriedades que tecnologicamente podem medir e analisar as características do corpo humano, utilizados como forma de identificação e controle de acesso para segurança. O reconhecimento através da apropriada interpretação dos parâmetros da marcha é um problema importante para classificação de padrões. Este trabalho possui como finalidade desenvolver um sistema de processamento de imagens que seja capaz de extrair padrões do movimento para a análise da marcha e apresentar um diagnóstico comparativo entre diferentes tipos de redes neurais, para a aplicação de técnicas que possam determinar a qualidade e eficácia das estatísticas para a identificação humana. Para este objetivo, utilizou-se dados de voluntários a partir do aplicativo desenvolvido em C# com base na análise tridimensional feita pela câmera *Kinect* da *Microsoft*, onde é possível identificar o esqueleto humano e extrair automaticamente os parâmetros cinéticos e cinemáticos. Os resultados obtidos revelaram a viabilidade para o processo de extração dos parâmetros da marcha e do reconhecimento do corpo humano.

**Palavras-chave:** Reconhecimento de padrões. Análise de marcha. *Kinect*. Biometria.



## ABSTRACT

The analysis of human gait is considered as a new biometric tool for the ability to obtain the metrics of the body at a distance. Biometric identifiers have properties that technology can measure and analyze the characteristics of the human body, used as a form of identification and access control for security. The recognition through suitable interpretation of parameters of the gait is a major problem for pattern classification. This work has as purpose to develop an image processing system that is able to extract patterns of movement for gait analysis and to present a comparative diagnosis between different types of neural networks, for applying techniques that can to determine the quality and efficacy of the statistics for human identification. For this objective, we used data from volunteers from the application developed in C# based on three-dimensional analysis made by *Microsoft's Kinect* camera, where it is possible to identify the human skeleton and automatically extract the kinetic and kinematic parameters. The results obtained proved the feasibility to extraction process of gait parameters and the recognition of the human body.

**Keywords:** Pattern Recognition. Gait analysis. Kinect. Biometrics.



## LISTA DE FIGURAS

Figura 1	Número de publicações sobre o tema de pesquisa ao decorrer dos anos. ....	27
Figura 2	Pioneiros no estudo da biomecânica. ....	32
Figura 3	Diferentes tipos de sistemas biométricos. ....	34
Figura 4	Coordenadas espaciais do <i>Kinect</i> . ....	39
Figura 5	Características do ciclo da marcha. ....	40
Figura 6	Movimentos básicos do corpo. ....	42
Figura 7	Planos e eixos do corpo humano. ....	43
Figura 8	Ângulo relativo entre dois segmentos. ....	44
Figura 9	Demonstração dos segmentos. ....	45
Figura 10	Ângulo absoluto: braço(a), quadril(b), coxa(c), perna(d)	46
Figura 11	Diagrama de corpo livre, define o sistema pela $F_a =$ Força de atrito, $F_p =$ Força peso e $F_s =$ Força de reação do solo. .	47
Figura 12	Sensor <i>Kinect</i> . ....	52
Figura 13	Aplicativo WPF. ....	53
Figura 14	Ângulos de flexão plantar do tornozelo direito. ....	55
Figura 15	Ângulos de flexão do joelho esquerdo. ....	56
Figura 16	Força de Reação do Solo. ....	56
Figura 17	Resultados - SVM. ....	62
Figura 18	Estrutura da MLP. ....	64
Figura 19	Resultados - MLP. ....	65
Figura 20	Resultados - RF - Erro <i>Out-of-bag</i> . ....	66
Figura 21	Resultados - RF. ....	67
Figura 22	Arquitetura PNN. ....	68
Figura 23	Resultados - PNN. ....	69
Figura 24	Resultados - NB. ....	71
Figura 25	Resultados - LVQ. ....	74
Figura 26	Resultados - KNN. ....	76
Figura 27	Resultados - <i>k-means</i> - 1ª Matriz. ....	77
Figura 28	Resultados - <i>k-means</i> - 2ª Matriz. ....	78
Figura 29	Modelo de árvore de decisão. ....	79
Figura 30	Resultados - Árvore de decisão. ....	80

Figura 31	Arquitetura - DNN.....	81
Figura 32	Arquitetura - 1 <sup>o</sup> Autoencoder .....	82
Figura 33	Arquitetura - 2 <sup>o</sup> Autoencoder .....	82
Figura 34	Arquitetura - <i>Softmax</i> .....	83
Figura 35	Resultados - DNN.....	84

## LISTA DE TABELAS

Tabela 1	Resultados dos classificadores.....	85
----------	-------------------------------------	----



## LISTA DE ABREVIATURAS E SIGLAS

<i>API</i>	<i>Application programming interface</i>
<i>ANN</i>	<i>Artificial neural network</i>
<i>CMOS</i>	<i>Complementary metal oxide semiconductor</i>
<i>3D</i>	<i>Três dimensões</i>
<i>DNN</i>	<i>Deep neural network</i>
<i>FRS</i>	<i>Força de reação do solo</i>
<i>kNN</i>	<i>k-Nearest neighbors</i>
<i>LQV</i>	<i>Learning vector quantization neural network</i>
<i>MLP</i>	<i>Multilayer neural perceptron</i>
<i>NB</i>	<i>Naive bayes</i>
<i>1R</i>	<i>One rule</i>
<i>PNN</i>	<i>Probabilistic neural network</i>
<i>PIN</i>	<i>Personal identification number</i>
<i>RGB</i>	<i>Red, green and blue</i>
<i>RF</i>	<i>Random forest</i>
<i>SDK</i>	<i>Software developer kit</i>
<i>SVM</i>	<i>Support vector machine</i>
<i>SOM</i>	<i>Self-organizing maps</i>
<i>WPF</i>	<i>Windows presentation foundation</i>



## LISTA DE SÍMBOLOS

$\overrightarrow{AB}$	Ângulo relativo entre os segmentos A e B.
$A_x$	Aceleração instantânea no eixo $x$ .
$A_y$	Aceleração instantânea no eixo $y$ .
$A_z$	Aceleração instantânea no eixo $z$ .
$AB_x$	Ângulo entre os segmentos A e B no eixo $x$ .
$AB_y$	Ângulo entre os segmentos A e B no eixo $y$ .
$AB_z$	Ângulo entre os segmentos A e B no eixo $z$ .
$a$	Aceleração do corpo.
$\overrightarrow{CB}$	Ângulo relativo entre os segmentos C e B.
$CB_x$	Ângulo entre os segmentos C e B no eixo $x$ .
$CB_y$	Ângulo entre os segmentos C e B no eixo $y$ .
$CB_z$	Ângulo entre os segmentos C e B no eixo $z$ .
$CG_x$	Centro de gravidade em $x$ .
$F_a$	Força de atrito.
$F_p$	Força peso.
$F_s$	Força de reação do solo.
$F_x$	Força de atrito na direção ântero-posterior.
$F_y$	Força de atrito na direção mediolateral.
$F_z$	Força normal no eixo $z$ .
$g$	Aceleração da gravidade.
$k$	Amostras dos vetores de entrada.
$\lambda$	Amostras classificadas corretamente.
$m$	Massa do corpo.
$M_x$	Momento de força dos segmentos em $x$ .
$m_S$	Massa dos seguimentos.
$\mu$	Coefficiente de atrito dinâmico.
$N$	Força normal.
<i>oob</i>	<i>Erro out-of-bag.</i>
$P$	Precisão da classificação.
$\Phi$	Amostras desejadas.
$t$	Tempo em segundos.
$\theta$	Theta, o ângulo relativo resultante dos segmentos.

$V_x$	Velocidade instantânea no eixo $x$ .
$V_y$	Velocidade instantânea no eixo $y$ .
$V_z$	Velocidade instantânea no eixo $z$ .
$V$	Velocidade total do movimento.
$x$	Distância em metros no eixo $x$ .
$x_{proximal}$	Distância do segmento superior próximo em $x$ .
$x_{distal}$	Distância do segmento inferior próximo em $x$ .
$x_{joelho}$	Distância do segmento do joelho superior próximo em $x$ .
$x_{tornozelo}$	Distância do segmento do tornozelo inferior próximo em $x$ .
$y$	Distância em metros no eixo $y$ .
$y_{proximal}$	Distância do segmento superior próximo em $y$ .
$y_{distal}$	Distância do segmento inferior próximo em $y$ .
$y_{joelho}$	Distância do segmento do joelho superior próximo em $y$ .
$y_{tornozelo}$	Distância do segmento do tornozelo inferior próximo em $y$ .
$z$	Distância em metros no eixo $z$ .

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	25
1.1	MOTIVAÇÃO E FORMULAÇÃO DO PROBLEMA	25
1.2	OBJETIVOS	28
1.2.1	Objetivo geral	28
1.2.2	Objetivos específicos	28
1.3	ESTRUTURA DA DISSERTAÇÃO	28
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	31
2.1	INTRODUÇÃO	31
2.1.1	O movimento humano na história	31
2.1.2	Cinesiologia	32
2.1.3	Biomecânica	33
2.1.4	Sistemas biométricos	33
2.1.5	Reconhecimento da marcha	35
2.2	TRABALHOS RELACIONADOS	36
2.3	CONCLUSÃO	37
<b>3</b>	<b>CARACTERÍSTICAS DA MARCHA HUMANA</b>	39
3.1	INTRODUÇÃO	39
3.1.1	O ciclo da marcha	40
3.2	PARÂMETROS DE INFLUÊNCIA	41
3.2.1	Descrição dos movimentos básicos	41
3.2.2	Sistema de planos e eixos	41
3.2.3	Cinemática linear	42
3.2.4	Cinemática angular	44
3.2.5	Cinética linear	47
3.3	CONCLUSÃO	49
<b>4</b>	<b>DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA DE EXTRAÇÃO AUTOMÁTICA DOS PARÂMETROS DA MARCHA</b>	51
4.1	INTRODUÇÃO	51
4.2	DESENVOLVIMENTO	51
4.2.1	Kinect e pacotes de desenvolvimento	51
4.2.2	Aplicativo	52
4.2.3	Resultados gráficos	53
4.3	CONCLUSÃO	57
<b>5</b>	<b>IDENTIFICAÇÃO BIOMÉTRICA DE INDIVÍDUOS ATRAVÉS DO RECONHECIMENTO DE PADRÕES</b>	59
5.1	INTRODUÇÃO	59

5.1.1	Base de dados . . . . .	59
5.2	CLASSIFICADORES . . . . .	60
5.2.1	<i>Support Vector Machine</i> - SVM . . . . .	60
5.2.1.1	Aplicação e resultados - SVM . . . . .	61
5.2.2	<i>Multilayer Perceptron Neural Network</i> - MLP . . . . .	62
5.2.2.1	Aplicação e resultados - MLP . . . . .	63
5.2.3	<i>Random Forest</i> - RF . . . . .	64
5.2.3.1	Aplicação e resultados - RF . . . . .	65
5.2.4	<i>Probabilistic Neural Network</i> - PNN . . . . .	67
5.2.4.1	Aplicação e resultados - PNN . . . . .	68
5.2.5	<i>Naive Bayes</i> - NB . . . . .	70
5.2.5.1	Aplicação e resultados - NB . . . . .	70
5.2.6	<i>Learning Vector Quantization Neural Network</i> - LVQ . . . . .	71
5.2.6.1	Aplicação e resultados - LVQ . . . . .	72
5.2.7	<i>k-Nearest Neighbors</i> - kNN . . . . .	73
5.2.7.1	Aplicação e resultados - kNN . . . . .	75
5.2.8	<i>k-means</i> . . . . .	75
5.2.8.1	Aplicação e resultados - <i>k-means</i> . . . . .	76
5.2.9	Árvore de decisão . . . . .	78
5.2.9.1	Aplicação e Resultados - Árvore de decisão . . . . .	79
5.2.10	Deep Neural Network - DNN . . . . .	80
5.2.10.1	Aplicação e resultados - DNN . . . . .	81
5.3	CONCLUSÃO . . . . .	84
6	CONCLUSÃO . . . . .	87
6.1	TRABALHOS FUTUROS . . . . .	89
	REFERÊNCIAS . . . . .	91

# 1 INTRODUÇÃO

A marcha é um modo das pessoas andarem e é considerada como novo método biométrico de identificar os seres humanos. Identificadores biométricos possuem propriedades que tecnologicamente podem modelar e rastrear as características do corpo humano. O fato de sua métrica ser obtida à distância, promove sua diferença com relação aos outros métodos biométricos.

É possível efetuar a análise para o reconhecimento do movimento devido as suas características serem definidas por segmentos corporais, que são descritos em termos de localização e orientação, para criar o sistema espacial humano (ROBERTSON et al., 2013).

Um sistema promissor para a extração dos parâmetros da marcha é o sensor *Kinect* da *Microsoft*, lançado em 2011 juntamente com um pacote de desenvolvimento, SDK (*Software Development Kit*), que permite aos desenvolvedores criarem aplicações em linguagem de programação C# utilizando o *software Microsoft Visual Studio* (GEERSE; COOLEN; ROERDINK, 2015).

Seu estudo baseado em sequências de vídeos tridimensionais aumentou relativamente devido às necessidades de interpretar os movimentos cinéticos e cinemáticos, tendo em vista facilitar a identificação de problemas físicos voltados à saúde e a biomecânica, ou também aplicados à área de segurança e tecnologia (ZERPA et al., 2015).

O presente trabalho consiste em extrair os parâmetros da marcha humana para estudos direcionados à sua análise, a partir do desenvolvimento de um protótipo de *software* que consiga gerar valores para efetuar o diagnóstico correto com base em técnicas de reconhecimento de padrões em aprendizagem de máquina, a fim de distinguir indivíduos no campo biométrico.

## 1.1 MOTIVAÇÃO E FORMULAÇÃO DO PROBLEMA

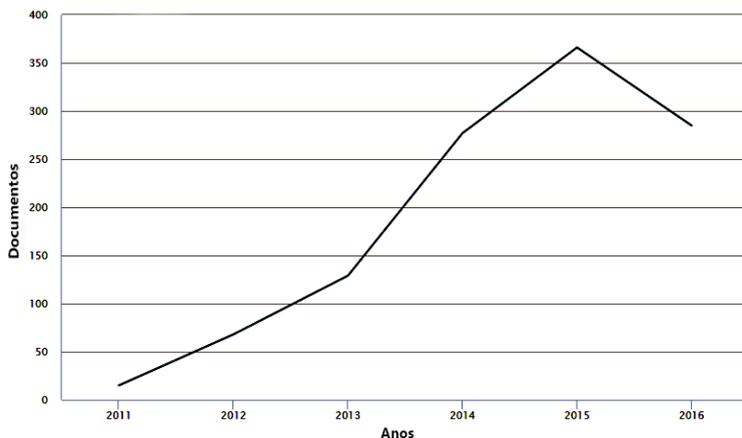
A presente pesquisa é motivada pelo grande interesse em sistemas biométricos para identificar pessoas, e a análise da marcha é uma nova ferramenta para isto. Deste modo, o trabalho se propõe a estudar esta nova técnica biométrica e criar uma ferramenta para a extração de características de imagens de vídeo, onde a marcha é o objeto de estudo. De acordo com o problema apresentado o autor conduziu-se por uma metodologia, ao pesquisar em projetos e artigos já existentes no meio

acadêmico sobre o assunto, para solucionar e enriquecer o campo de estudo. Quanto aos procedimentos metodológicos para execução deste trabalho destaca-se:

- Especificar quais os requisitos para a elaboração do protótipo de *software*, de acordo com as características da marcha humana;
- Definir e testar as ferramentas físicas, o sensor *Kinect*, para a captura dos dados;
- Coletar dados de movimentos com o *Kinect* da *Microsoft*;
- Implementar um código em *C#* utilizando o pacote de desenvolvimento SDK, para captura de vídeo;
- Fazer os cálculos, para extração dos parâmetros da marcha, em função da cinética e cinemática do movimento na biomecânica;
- Implementar um código em *C#* que extraia características sequenciais da marcha e mostre o esqueleto como modelo de locomoção humana;
- Criar um aplicativo WPF (*Windows Presentation Foundation*) para extrair os dados da marcha, em um formato que seja flexível para manipular;
- Efetuar a conversão de dados do aplicativo WPF para o *Matlab*, a fim de aplicar a classificação dos dados *offline*;
- Plotar os gráficos dos movimentos cinéticos e cinemáticos no *Matlab* para análise dos resultados e validação do aplicativo;
- Implementar um algoritmo no *Matlab* que possa efetuar a análise dos dados, a partir do reconhecimento de padrões para múltiplas classes em diferentes classificadores;
- Analisar e avaliar o avanço dos algoritmos de aprendizagem de máquina para o reconhecimento da marcha humana.

Como comentado anteriormente, este tema tem despertado interesse na área acadêmica, em busca de resultados relevantes para o desenvolvimento tecnológico. Na Figura 1, mostra-se o resultado de uma busca de dados na plataforma SCOPUS (ELSEVIER, 2016), que permite analisar o número de publicações em conferências e revistas ao decorrer dos anos, utilizando as seguintes palavras-chave para a pesquisa:

Figura 1: Número de publicações sobre o tema de pesquisa ao decorrer dos anos.



Fonte: Elsevier (2016).

*Gait Analysis with Kinect, Skeleton with Kinect e Human Motion with Kinect.*

É visto na Figura 1 que desde as primeiras pesquisas houve um crescimento relativo de trabalhos produzidos em função deste tema até o atual ano. Abaixo serão apresentados alguns investigadores desta linha de pesquisa da extração, análise e classificação da marcha humana utilizando o sensor *Kinect* como a ferramenta de estudo.

De uma forma geral, os sistemas biométricos para análise da marcha humana baseados em vídeo, utilizam um sistema de visão capaz de obter imagens em 3D (Três dimensões) (como por exemplo, o sensor *Kinect* ou um estereoscópio) de um indivíduo em movimento. No qual efetua a captura dos ângulos de algumas juntas do corpo ao longo de um determinado tempo de aquisição de imagem e extração de algumas características adicionais derivadas destas, como velocidade, aceleração e forças que serão descritas no Capítulo 3.

Esses diversos parâmetros são então utilizadas primeiramente para treinar um sistema de reconhecimento de indivíduos aplicando técnicas de aprendizagem de máquina, e que após o treinamento serviu para a identificação biométrica propriamente dita. Como se trata de um problema de reconhecimento de padrões, tais sistemas utilizam técnicas como redes neurais, *Support Vector Machine* (SVM), redes bayesianas,

*k-Nearest Neighbors* (kNN) e outras para o treinamento e classificação (identificação).

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

Este trabalho tem por finalidade, a pesquisa e elaboração de um protótipo de sistema de processamento de imagens de vídeo, que seja capaz de extrair características do movimento humano para aplicar técnicas de reconhecimentos de padrões, a fim de analisar e realizar o reconhecimento biométrico do indivíduo inspecionado.

### 1.2.2 Objetivos específicos

- Realizar as capturas de movimento da marcha humana pelo sensor *Kinect*, utilizando métodos baseados em visão computacional.
- Elaborar uma plataforma para capturas frontais e laterais do corpo em um ambiente controlado.
- Implementar código utilizando o pacote de desenvolvimento para que seja possível visualmente a identificação do esqueleto do indivíduo.
- Desenvolver um protótipo de ferramenta que possa auxiliar nos processos e estudos da análise de marcha.
- Aplicar técnicas de diferentes classificadores, para o processo e análise de reconhecimento de padrões, a fim de reconhecer o indivíduo a partir da marcha.

## 1.3 ESTRUTURA DA DISSERTAÇÃO

- Capítulo 2: aborda a fundamentação teórica do tema proposto nesta pesquisa para compreender o movimento humano, o que as pessoas, a arte, ciência e a tecnologia contribuíram para a análise da marcha humana.
- Capítulo 3: relata as definições básicas do movimento humano,

teoria e cálculos para implementar a extração dos parâmetros cinéticos e cinemáticos do corpo.

- Capítulo 4: descreve o desenvolvimento do protótipo de *software*, a partir dos estudos da biomecânica para a extração dos parâmetros da marcha.
- Capítulo 5: apresenta os mecanismos estatísticos utilizados na classificação dos dados de entrada, extraídos pelo protótipo de *software* desenvolvido para aquisição dos parâmetros da marcha, e os resultados obtidos de cada classificador.
- Capítulo 6: conclui e lista os trabalhos futuros que poderão ser realizados a partir desta investigação.



## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 INTRODUÇÃO

Visando aprofundar o conhecimento sobre o assunto, na fundamentação teórica serão abordadas questões de como compreender o movimento humano, o que as pessoas, a arte, ciência e a tecnologia contribuíram para a análise da marcha humana e quão avançada está a tecnologia para realizar estudos dos movimentos com uma base de dados e quais tipos de sistemas utilizados.

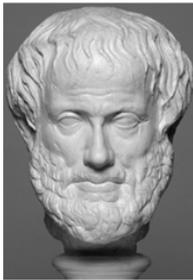
#### 2.1.1 O movimento humano na história

Devido as curiosidades e necessidades, o movimento humano tem sido estudado desde muito tempo na história da humanidade. Na Figura 2 podemos ver os pioneiros no estudo da biomecânica. O filósofo grego Aristóteles (384a.C - 322a.C) publicou um texto sobre a marcha dos animais, além de outros estudos que complementaram o assunto. Esse texto contém temas que abordam qual seria o motivo de homens e pássaros, embora sendo bípedes, possuem uma curvatura oposta nas pernas e o conhecimento básico geométrico para cálculos e análises (ROSENHAHN; KLETTE; METAXAS, 2008).

Leonardo da Vinci (1452 - 1519), na renascença, desenvolveu vários estudos detalhados sobre a anatomia humana, uma parte desses foram direcionados para a cinemática do movimento humano. Em 1897 o fotógrafo inglês Edward Muybridge introduziu os primeiros estudos sobre movimentos a partir do uso de várias câmeras para captar os saltos efetuados por equinos, e lançou sua obra *Animal Locomotion*. A partir destes ensinamentos e de outros estudiosos que também contribuíram para estas descobertas, obtivemos um ponto de partida para o desenvolvimento de estudos na área (ROSENHAHN; KLETTE; METAXAS, 2008).

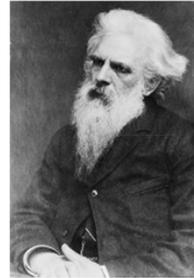
A ciência que estuda o movimento humano surgiu devido ao fascínio dos estudiosos analisarem o comportamento motor animal. A maneira como nada um peixe, como o pássaro voa, como o homem anda e quais são os limites da força muscular, levaram o homem a desenvolver a ciência do movimento humano, denominada cinesiologia (LEHMKUHL; SMITH; WEISS, 1989).

Figura 2: Pioneiros no estudo da biomecânica.



Filósofo grego  
Aristóteles  
(384a.C - 322a.C)  
Marcha dos animais.

Leonardo da Vinci  
(1452 - 1519)  
Cinemática do Movimento  
Humano.



Edward Muybridge  
(1830 - 1904)  
Múltiplas câmeras para  
captar movimentos equinos.  
Obra: "Animal Locomotion".

Fonte: Rosenhahn, Klette e Metaxas (2008).

### 2.1.2 Cinesiologia

A cinesiologia é uma área da ciência que estuda o movimento humano e animal, seu termo em inglês *kinesiology* tem origem da palavra grega *kinesis* que significa movimento e *logos* que significa estudo (SILVA, 2015).

Segundo (OLIVEIRA et al., 2011) a cinesiologia é uma disciplina multidisciplinar, envolvendo conhecimentos de anatomia, fisiologia articular e aplica os mesmos aos movimentos humanos simples e complexos. Estudar cinesiologia é relacionar a mesma de forma permanente com o estudo das multiplicidades dos movimentos executados pelo ser humano, compreendendo as forças internas e externas atuantes e seus limites, possibilitando em muitos casos até a prevenção de lesões.

Esta área se consolidou como ciência a partir do século XX, devido aos avanços tecnológicos e metodológicos, como as técnicas de avaliação do movimento. Assim surge a biomecânica, uma subdisciplina derivada da cinesiologia, que efetua as análises físicas dos diferentes sistemas biológicos, inclusive o movimento humano (SILVA, 2015).

### 2.1.3 Biomecânica

A biomecânica descende dos estudos referentes ao movimento e análises físicas do corpo humano. Fisicamente o corpo humano pode ser definido como um sistema de segmentos articulados. Este movimento provem de forças internas que atuam fora do eixo circular do corpo ou fora deste corpo (AMADIO et al., 1999).

É preciso seis coordenadas para representar a posição e orientação de um corpo físico em um espaço tridimensional, pois este corpo possui seis graus de liberdade para movimentar-se. Para que essas coordenadas sejam designadas é preciso medir a posição 3D de pelo menos três marcadores não colineares em cada segmento (ARAÚJO; ANDRADE; BARROS, 2005).

É necessário evidenciar a complexidade estrutural do movimento e a aplicação dos seus métodos de avaliação das diversas áreas da ciência, sua investigação consiste em combinações simultâneas e sincronizadas de procedimentos cinemáticos e dinâmicos que são fundamentais para examiná-lo (AMADIO et al., 1999).

O movimento é descrito quando é ligado ao sistema de coordenadas relativas a variação da posição e orientação durante uma caminhada. Os seis segmentos corporais devem trabalhar juntos com um sistema de coordenadas, pois a orientação deve estar ligada com os eixos anatômicos do corpo, para chegar nos resultados desejados (ARAÚJO; ANDRADE; BARROS, 2005).

Em ações estáticas a posição e a orientação não variam com o tempo, é possível caracterizá-la apenas com a orientação dos segmentos corporais ligados a força da gravidade. Já as ações dinâmicas variam com o tempo em relação a posição e orientação (SOUSA, 2008).

### 2.1.4 Sistemas biométricos

Um sistema biométrico pode operar em dois modos descritos por Ross, Jain e Prabhakar (2004):

- Modo verificação: neste o sistema tenta validar a identidade do usuário por ferramentas como cartões inteligentes, números de PIN (*Personal Identification Number*) ou nome de usuários, fundamentado na comparação com o banco de dados, para validar se o reconhecimento da identidade é positivo.
- Modo identificação: o sistema identifica usuários buscando a par-

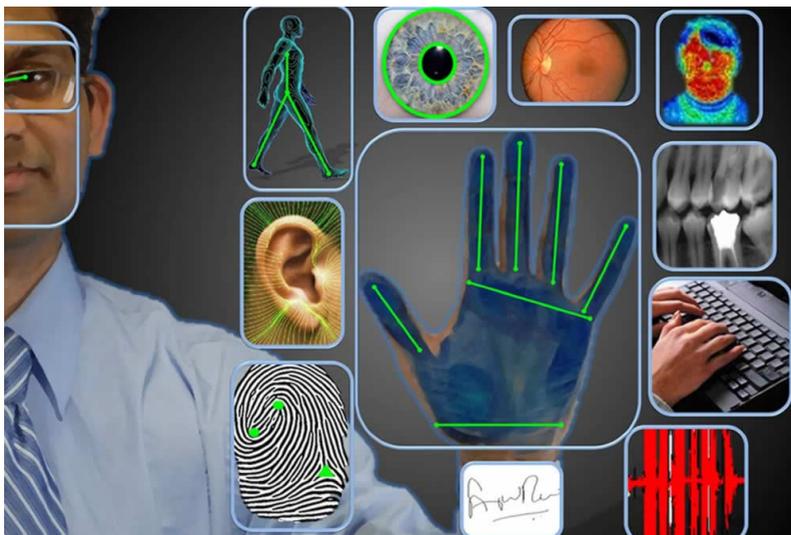
tir de modelos físicos e comportamentais, para impedir que outros indivíduos usem várias identidades, como por exemplo a leitura de digitais, escaneamento da retina, análise de marcha, etc.

Um indivíduo biométrico possui características fisiológicas ou comportamentais humanas que podem ser usadas como uma característica biométrica (ROSS; JAIN; PRABHAKAR, 2004).

A extração e o processamento de informações que caracterizam cada indivíduo de forma natural estão sendo bastante utilizados em mecanismos de identificação. Na qual manifestam-se novas tecnologias que simplificam a execução de vários atos, movimentos e atividades comuns no cotidiano de monitoramento, identificação e controle de acesso (TAKEDA et al., 2014).

Com o intuito de autenticar usuários para assegurar que a pessoa que está acessando o sistema é quem diz ser, sem a utilização de senhas alfanuméricas, pois são formas de autenticação vulneráveis ao usuário caso caia em mãos erradas, surgem as características biométricas físicas. Elas baseiam-se em utilizar impressões digitais, exames de retina, leitura da palma da mão, análise da voz, facial e entre outros como sistemas e produtos biométricos mais seguros existentes atualmente mostrados na Figura 3 (HEINEN; OSÓRIO, 2004).

Figura 3: Diferentes tipos de sistemas biométricos.



Fonte: TECH (2016).

O uso do caminhar como um sistema biométrico ainda é uma novidade, quando contraposto com os outros métodos de reconhecimento. As variações do jeito de caminhar são utilizadas como um identificador biométrico para detectar as pessoas individualmente (JAIN; FLYNN; ROSS, 2008). São discernidos alguns parâmetros como a velocidade na cinemática linear, a movimentação do quadril, tornozelo e joelho na cinemática angular e as forças atuantes no corpo pela cinética linear.

### 2.1.5 Reconhecimento da marcha

Caminhar é tão comum e muito importante para nos movimentarmos. A análise deste movimento decorre da medição e avaliação de características biomecânicas que estão associadas com várias tarefas. Há progressos significativos sobre a análise de marcha, que auxiliam no desenvolvimento dos sistemas de gravação de movimento, exigindo menos tempo de processamento de dados (??).

A partir da análise de marcha podemos assumir quais são as funcionalidades do corpo quando estão em movimento, é possível obter a estimação do movimento da marcha efetuando cálculos referentes a posição espacial do centro de massa e os centros de giro das articulações do corpo (MARTÍNEZ; GÓMEZ; ROMERO, 2009).

A marcha pode ser definida como uma combinação de movimentos, eles devem acontecer em um padrão temporal específico para ela ocorrer e possuir um método repetitivo entre etapas cíclicas, com a alternância dos pés, isto faz com que seja um fenômeno único (BOYD; LITTLE, 2005).

O movimento possui ciclos que incluem ações desde o contato inicial de uma extremidade até a outra extremidade entrar em contato com o solo, possuindo a fase de apoio e a oscilante. A fase de apoio corresponde a 60% do ciclo, sendo o momento em que o pé encontra-se no chão e a fase oscilante possui os 40% restantes do ciclo (SOUSA, 2008). Mesmo que as fases da marcha humana possuam um padrão de comportamento, cada ser humano produz imagens diferentes durante seu movimento.

Pequenas variações no estilo de marcha podem ser usados como um identificador biométrico para identificar as pessoas individualmente, a extração dos parâmetros da marcha, geram diversas amostras que podem ser empregadas para recursos de aprendizagem para a classificação.

## 2.2 TRABALHOS RELACIONADOS

Sistemas de classificação são muito utilizados para a análise da marcha humana, há diversos estudos com foco no tema. Estes estudos aplicaram diferentes técnicas, utilizando o *Kinect* e câmeras convencionais, a fim de encontrar a solução para o problema.

Haro (2014) efetua a análise da marcha humana utilizando uma câmera convencional, com o objetivo de descrever e especificar as utilidades dos parâmetros para aplicações clínicas em termos de doenças. E quais são os processos de aquisição e análise da imagem com propósito de definir o perfil de comportamento da marcha em função do tempo.

Kewatkar e Kathle (2012) utiliza uma câmera convencional, o autor propõe um projeto baseado no reconhecimento humano, sendo a marcha o objeto de estudo no desenvolvimento de um programa utilizando a biblioteca OpenCV. E como procedimento de avaliação e classificação dos dados, foi aplicado o algoritmo *k-Nearest Neighbors* (kNN), um método não paramétrico em que as entradas consistem em  $k$  exemplos de treinamento, em que uma amostra é classificada pelo voto da maioria dos vizinhos, sendo que ela está atribuída à classe mais comum entre os seus  $k$  vizinhos.

Nieto-Hidalgo et al. (2016) apresentou uma especificação geral da marcha, como anormal ou normal, para identificar as características com a execução dos seguintes classificadores utilizando o *Kinect*. O *Support Vector Machine* (SVM) encontra uma linha de separação chamada de hiperplano entre dados de duas ou mais classes, essa linha busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes. O Naive Bayes (NB), um classificador probabilístico com base na aplicação do teorema de Bayes e o *k-Nearest Neighbors* (kNN).

Gianaria et al. (2014) utilizou o sensor *Kinect* e aplicou o SVM para mostrar que o conjunto de características comportamentais relacionadas com os movimentos da cabeça, joelhos e cotovelos, é uma ferramenta eficaz para a caracterização da marcha e reconhecimento de pessoas.

Araujo, Graña e Andersson (2013) utiliza o *Kinect* e propõe a avaliação do esqueleto humano pelo classificador *Multi-Layer Perceptron* (MLP), uma rede neural artificial *feedforward* composta por várias camadas de neurônios ligadas entre si por sinapses de pesos, que utiliza a aprendizagem supervisionada com a técnica de retro propagação para treinar a rede. O classificador C4.5 é utilizado para gerar uma árvore de decisão e mineração de dados sendo considerado um classificador proba-

bilístico. O *Random Forest* (RF) ou florestas de decisão aleatória, é um classificador que opera através da construção de um grande número de árvores de decisão, é a média de múltiplas árvores de decisão profundas, formadas em diferentes partes de um mesmo conjunto de treinamento para a redução da variância, e por fim, o classificador kNN citado anteriormente.

A proposta de Ball et al. (2012) investigou a possibilidade de reconhecer pessoas individualmente a partir da marcha utilizando o *Kinect* e aplicou o algoritmo *K-means*, que é um método de *clustering*, utilizado em mineração de dados. Este classificador particiona várias amostras em grupos, onde cada uma pertence ao conjunto mais próximo da média.

O sensor *Kinect* foi utilizada por Preis et al. (2012), que determina a aplicação de três diferentes classificadores para avaliar as características do corpo. Primeiramente o algoritmo *One Rule* (1R), gera uma regra de classificação para cada classe com base em um único recurso nos dados de treinamento, e seleciona a amostra com menor erro como regra. O autor utiliza também os classificadores C4.5 e Naive Bayes citados anteriormente.

## 2.3 CONCLUSÃO

Neste capítulo apresentou-se um pouco da história da marcha humana, quais foram as primeiras ideologias sobre a área de pesquisa, com o surgimento da cinesiologia e da biomecânica e presentes estudos relacionados com o tema. Tal qual estas disciplinas são de grande importância para a aplicação tecnológica ao desenvolver qualquer sistema biométrico para a identificação de usuários.



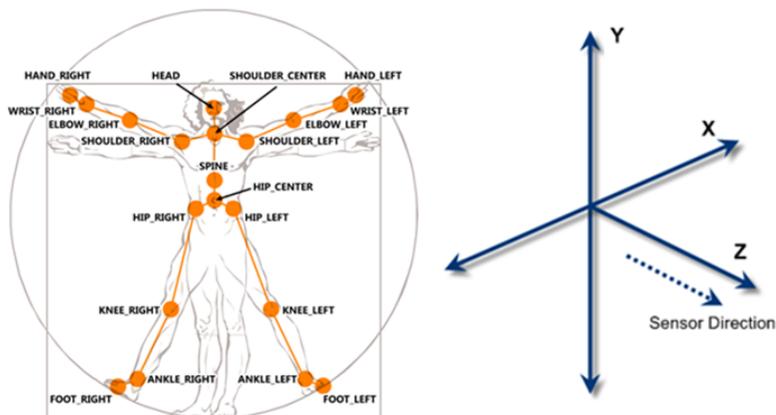
### 3 CARACTERÍSTICAS DA MARCHA HUMANA

#### 3.1 INTRODUÇÃO

O presente capítulo mostrará um esboço de alguns aspectos do movimento humano para a aplicação deste estudo. A análise deste movimento decorre da medição de características biomecânicas que estão associadas a várias tarefas (??). É o que nos faz interagir com o ambiente, desde um exercício na academia, um simples passo na caminhada pela rua ou até a participação em um campeonato de futebol. Ele decorre da mudança de lugar, da posição e da postura em algum ponto no ambiente.

O sistema de coordenadas associadas à mudança da posição e orientação no decorrer de uma caminhada, expõe as características do movimento. O sensor *Kinect* fornece as coordenadas espaciais distribuídas em 20 pontos específicos no corpo humano, formando um modelo tridimensional do esqueleto como na Figura 4 (CUJI et al., 2013).

Figura 4: Coordenadas espaciais do *Kinect*.



Fonte: Research (2016).

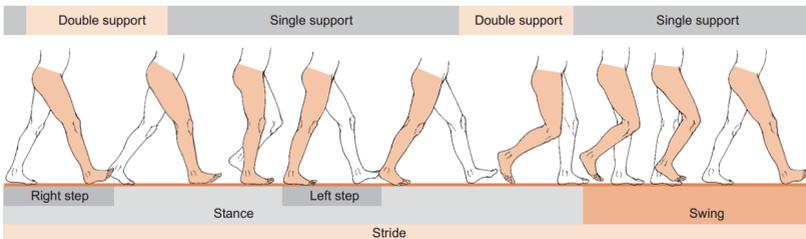
### 3.1.1 O ciclo da marcha

A marcha é um movimento cíclico, composta por um ciclo básico de repetições, o ciclo da marcha. Este consiste no tempo em que o pé de referência (direito ou esquerdo) está no chão ou não está em contato com o solo (OATIS, 2003).

A Figura 5 exemplifica pontos significantes para analisar o ciclo, que incluem:

- *Stride*: é o movimento ocorrido durante o ciclo da marcha em ambos os membros, definido como a passada.
- *Single e Double Support*: São dois breves períodos do ciclo, o apoio único(*single*) tem a duração de contato com o solo de um pé até o contato novamente com o mesmo pé. O apoio duplo(*double*) é o período em que o primeiro pé está em contato com o solo até o outro pé entrar em contato com o solo.
- *Right e Left Step*: representa o passo direito e o passo esquerdo, respectivamente.
- *Stance Phase*: é a fase de apoio que representa 60% do ciclo, na qual o membro de referência está em contato com o solo.
- *Swing Phase* : é a fase de balanço, representada por 40% do ciclo, onde o membro de referência não está em contato com o solo.

Figura 5: Características do ciclo da marcha.



Fonte: Oatis (2003).

## 3.2 PARÂMETROS DE INFLUÊNCIA

### 3.2.1 Descrição dos movimentos básicos

De acordo com Hamill e Knutzen (2014) na combinação das articulações do corpo ocorrem seis diferentes movimentos exibidos na Figura 6.

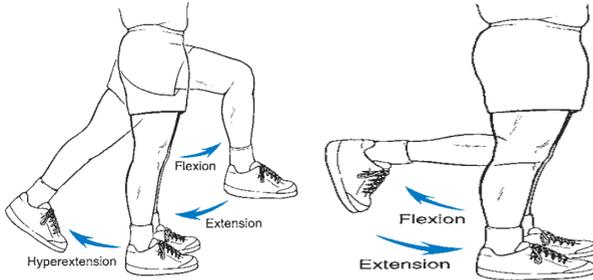
- Flexão (a) e (b): movimento no qual o ângulo relativo da articulação entre dois segmentos adjacentes diminui.
- Extensão (a) e (b): movimento de alisamento em que o ângulo relativo da articulação entre dois segmentos adjacentes aumenta, conforme a articulação retorna para a posição inicial. A hiperextensão (a) ocorre quando o movimento vai além da faixa normal da flexão.
- Abdução (c): movimento de abertura do segmento na direção oposta a linha média do corpo.
- Adução (c): movimento de retorno do segmento em direção a linha média do corpo. A hiperadução (c) ocorre quando o movimento vai além da faixa normal da adução.
- Rotação medial (d): movimento de rotação interno.
- Rotação lateral (d): movimento de rotação externo.

### 3.2.2 Sistema de planos e eixos

O sistema de coordenadas referente à variação e posição durante uma caminhada é composto por diferentes planos e eixos anatômicos representados na Figura 7 (WHITTLE, 2014).

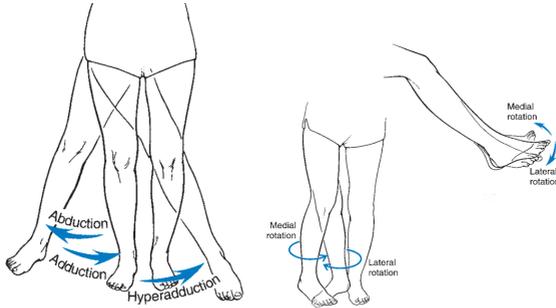
- Plano sagital: ocorre o movimento para frente ou para trás sobre o eixo mediolateral.
- Plano frontal: também chamado de plano coronal, ocorre o movimento para direita ou para esquerda sobre o eixo ântero-posterior.
- Plano transversal: também chamado de plano horizontal, ocorre o movimento para dentro ou para fora sobre o eixo longitudinal.

Figura 6: Movimentos básicos do corpo.



(a) Flexão, extensão e hiperextensão.

(b) Flexão e extensão.



(c) Abdução, adução e hiperadução.

(d) Rotação medial e lateral.

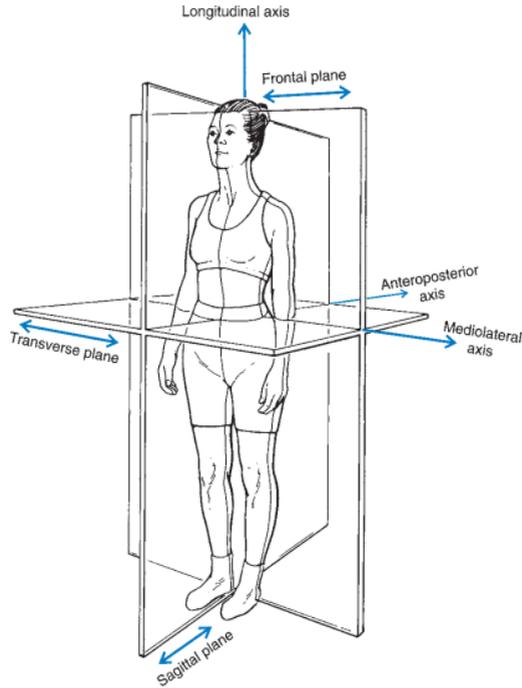
Fonte: Hamill e Knutzen (2014).

### 3.2.3 Cinemática linear

A análise cinemática linear envolve o estudo do movimento em linha reta para determinar se o objeto está se movendo, qual a altura que atinge ou a que distância se desloca em metros, representados pelos eixos  $x$ ,  $y$  e  $z$ . As equações (3.1), (3.2) e (3.3) são utilizadas para calcular as velocidades instantâneas  $V_x$ ,  $V_y$  e  $V_z$  pois há necessidade de calculá-las em um instante específico, para melhor análise biomecânica (HAMILL; KNUTZEN, 2014).

Por diferenciação numérica é possível descrever os meios para calcular as derivadas  $dx$ ,  $dy$  e  $dz$  em função de  $dt$ , na qual  $t$  é determinado pelo tempo em segundos (DEMAILLY, 2012):

Figura 7: Planos e eixos do corpo humano.



Fonte: Hamill e Knutzen (2014).

$$V_x = \frac{dx}{dt} = \frac{x(t+1) - x(t-1)}{\Delta t} \quad (3.1)$$

$$V_y = \frac{dy}{dt} = \frac{y(t+1) - y(t-1)}{\Delta t} \quad (3.2)$$

$$V_z = \frac{dz}{dt} = \frac{z(t+1) - z(t-1)}{\Delta t} \quad (3.3)$$

Pelo cálculo da magnitude vetorial é determinada pela equação (3.4) a velocidade total ( $V$ ) do movimento:

$$V = \sqrt{V_x^2 + V_y^2 + V_z^2} \quad (3.4)$$

Segundo Hamill e Knutzen (2014), a aceleração consiste na va-

riação das velocidades  $dV_x$ ,  $dV_y$  e  $dV_z$  em função do tempo, assim foi aplicado o mesmo com as acelerações instantâneas  $A_x$ ,  $A_y$  e  $A_z$ , calculadas conforme as equações (3.5), (3.6) e (3.7).

Para o cálculo das derivadas de 2ª ordem é considerado as expansões por série de Taylor (DEMAILLY, 2012):

$$A_x = \frac{dV_x}{dt} = \frac{x_{(t+1)} - 2x_{(t)} + x_{(t-1)}}{(\Delta t)^2} \quad (3.5)$$

$$A_y = \frac{dV_y}{dt} = \frac{y_{(t+1)} - 2y_{(t)} + y_{(t-1)}}{(\Delta t)^2} \quad (3.6)$$

$$A_z = \frac{dV_z}{dt} = \frac{z_{(t+1)} - 2z_{(t)} + z_{(t-1)}}{(\Delta t)^2} \quad (3.7)$$

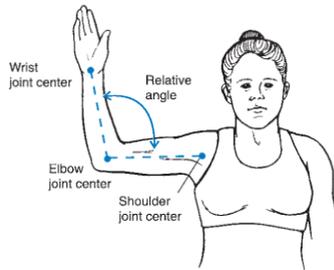
Pela equação (3.8) calcula-se a aceleração total ( $A$ ) utilizando a magnitude vetorial:

$$A = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (3.8)$$

### 3.2.4 Cinemática angular

Para determinar a posição da articulação, é preciso definir o ângulo de cada articulação. O sensor *Kinect* possui um algoritmo capaz de efetuar a conversão das medidas por pixels em unidades vetoriais, as coordenadas x, y e z, no que possibilita o cálculo dos ângulos das articulações (CUJI et al., 2013).

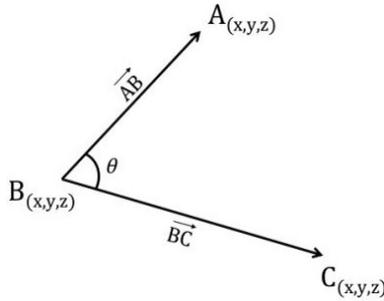
Figura 8: Ângulo relativo entre dois segmentos.



Fonte: Hamill e Knutzen (2014).

Na biomecânica calcula-se os ângulos de duas maneiras, o primeiro é chamado ângulo relativo, na qual é medido em relação a uma referência móvel e define o ângulo incluído entre o eixo longitudinal de dois segmentos demonstrado na Figura 8 (HAMILL; KNUTZEN, 2014).

Figura 9: Demonstração dos segmentos.



Fonte: Sulino (2014).

O ângulo relativo dos segmentos produzidos será obtido pelas coordenadas tridimensionais de cada ponto anatômico (SULINO, 2014). As equações (3.9) e (3.10) determinam os vetores  $\vec{AB}$  e  $\vec{CB}$  referentes aos dois segmentos, retratados na Figura 9:

$$\vec{AB} = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} - \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} \quad (3.9)$$

$$\vec{CB} = \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} - \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} \quad (3.10)$$

A magnitude dos vetores de acordo com as equações (3.11) e (3.12) define o seu comprimento em três dimensões. O mesmo é calculado utilizando uma variação do teorema de Pitágoras, devido a formação de triângulos quando o vetor é dividido em componentes. Estas representam as diferentes forças atuando sobre um objeto e o valor total destas forças, a magnitude (LENGYEL, 2005).

$$\|\vec{AB}\| = \sqrt{AB_x^2 + AB_y^2 + AB_z^2} \quad (3.11)$$

$$\|\vec{CB}\| = \sqrt{CB_x^2 + CB_y^2 + CB_z^2} \quad (3.12)$$

É obtido o produto escalar pela equação (3.13), a fim de calcular o ângulo  $\theta$ :

$$\vec{AB} \odot \vec{CB} = \begin{pmatrix} AB_x \\ AB_y \\ AB_z \end{pmatrix} \odot \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} \quad (3.13)$$

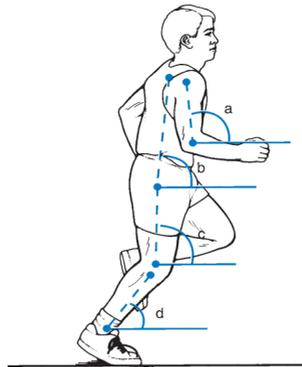
E finalmente foi calculado o ângulo  $\theta$  pela função (3.14) e (3.15) do cosseno e arco-cosseno, onde  $\theta$  é o ângulo relativo resultante dos segmentos:

$$\cos \theta = \frac{\vec{AB} \odot \vec{CB}}{|\vec{AB}| \times |\vec{CB}|} \quad (3.14)$$

$$\theta = \left( \cos \theta \times \frac{180}{\pi} \right) \quad (3.15)$$

De acordo com Hamill e Knutzen (2014) o segundo é o ângulo absoluto, um ângulo de inclinação do segmento do corpo medido em relação a uma referência fixa e descreve a orientação do segmento no espaço demonstrado na Figura 10.

Figura 10: Ângulo absoluto: braço(a), quadril(b), coxa(c), perna(d)



Fonte: Hamill e Knutzen (2014).

Este pode ser calculado utilizando a relação trigonométrica da tangente, onde ela é definida com base nos lados de um triângulo

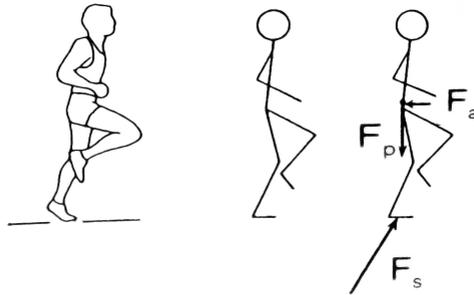
retângulo. A equação (3.16) demonstra como efetuar o cálculo:

$$\begin{aligned}\tan \theta &= \frac{y_{proximal} - y_{distal}}{x_{proximal} - x_{distal}} \\ \tan \theta_{perna} &= \frac{y_{joelho} - y_{tornozelo}}{x_{joelho} - x_{tornozelo}}\end{aligned}\quad (3.16)$$

### 3.2.5 Cinética linear

A cinética linear é a área de estudo que examina as forças que atuam em um sistema como o corpo humano, definindo as forças causadoras do movimento. O diagrama de corpo livre é um instrumento extremamente útil nos estudos biomecânicos (HAMILL; KNUTZEN, 2014). A Figura 11 exhibe o desenho do sistema, a identificação das forças atuantes definem o problema e determinam como a análise deve processar.

Figura 11: Diagrama de corpo livre, define o sistema pela  $F_a$  = Força de atrito,  $F_p$  = Força peso e  $F_s$  = Força de reação do solo.



Fonte: Enoka (2000).

A equação (3.17) determina o vetor peso que é direcionado para o centro da Terra, e se origina de um ponto chamado centro de gravidade. Este ponto representa o equilíbrio entre todos os segmentos corporais, isto indica que a massa do sistema está em equilíbrio em torno dele (HAMILL; KNUTZEN, 2014).

Assim temos que definir o cálculo estimativo da massa e a localização do centro de gravidade de cada um para encontrar a localização vetorial do centro de gravidade total. Os segmentos definidos para o cálculo são determinados pelo número de articulações em que ocorre o deslocamento angular durante o movimento.

A partir de estudos feitos por Chandler et al. (1975) em cadáveres, temos dados mais concretos disponíveis para calcular as equações regressivas. Ao somar o momento de força de cada segmento em função das coordenadas x, y e z pela equação (3.18), é possível determinar a localização do centro de gravidade do corpo humano. O mesmo cálculo é feito para os outros vetores (ENOKA, 2000).

$$F_p = m.g \quad (3.17)$$

$$\begin{aligned} \sum M_x &= F_{p,S} \times x \\ CG_x &= \frac{\sum M_x}{F_p} \end{aligned} \quad (3.18)$$

A força de reação do solo (FRS) foi calculada por um método utilizado por Bobbert, Schamhardt e Nigg (1991), no qual os valores cinemáticos são derivados das acelerações dos centros de massa de cada um dos segmentos, onde a FRS retrata as acelerações deles individualmente, devido ao seu movimento. A FRS é o somatório das forças verticais de todos os segmentos do corpo, em função da gravidade ( $g = 9,8m/s^2$ ). A equação (3.19) determina esta força:

$$F_s = \sum m_S(a - g) \quad (3.19)$$

Segundo Hamill e Knutzen (2014) é difícil medir o coeficiente de atrito e a força de atrito com precisão sem equipamentos sofisticados. Ambos, no entanto, podem ser medidos com uma plataforma de força. Assim é possível determinar a resultante da força de atrito pela equação (3.20) e os componentes de cisalhamento  $F_x$  e  $F_y$ , respectivamente são as forças de atrito nas direções ântero-posterior e mediolateral. O componente  $F_z$  é utilizado como a Força Normal (N), e estima-se o coeficiente de atrito dinâmico pela equação (3.21):

$$Fa = \mu.N \quad (3.20)$$

$$\mu = \frac{F_y}{F_z} \quad (3.21)$$

### 3.3 CONCLUSÃO

Neste capítulo são apresentados os métodos de calcular os parâmetros cinéticos e cinemáticos do corpo que influenciam na análise da marcha. Para este trabalho foram utilizados apenas os parâmetros inferiores: quadris, joelhos, tornozelos e pés para a extração dos dados cinemáticos angulares. Foram calculados os dados cinemáticos lineares, tais como, velocidade e aceleração pois estes influenciam na computação dos dados da cinética linear ao calcular a força de reação do solo. A força atrito não foi aplicada neste estudo visto que segundo (HAMILL; KNUTZEN, 2014) necessita de equipamentos sofisticados para sua estimação.



## 4 DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA DE EXTRAÇÃO AUTOMÁTICA DOS PARÂMETROS DA MARCHA

### 4.1 INTRODUÇÃO

Neste Capítulo aborda-se os procedimentos efetuados para as etapas de desenvolvimento do protótipo de software utilizando a câmera *Kinect* da *Microsoft* e os métodos de extração dos parâmetros, baseados nos cálculos apresentados pelos estudos da biomecânica citados no capítulo anterior.

O presente trabalho consiste em extrair os parâmetros da marcha humana para estudos direcionados à sua análise de reconhecimento de indivíduos no campo biométrico. Propõe-se um protótipo de software que consiga gerar valores por comportamento gráfico das variáveis, sejam elas definidas pelos ângulos das articulações ou coordenadas da posição determinaremos a força do movimento em função do ciclo da marcha.

Nas próximas seções será apresentada a metodologia aplicada, uma análise da validação dos valores das variáveis e seus respectivos resultados.

### 4.2 DESENVOLVIMENTO

#### 4.2.1 Kinect e pacotes de desenvolvimento

Pode-se visualizar na Figura 12 o sensor *Kinect*. Sua câmera *RGB* consegue capturar os três componentes básicos da cor. A câmera possui também um sensor de profundidade, com um projetor de infravermelhos e combinado com o *Complementary Metal Oxide Semiconductor* (CMOS) monocromático. Ele gera imagens em um espaço *3D* (três dimensões) e captura imagens em condições bastante variáveis e até desfavoráveis de iluminação, permitindo a identificação dos pontos de marcação no corpo, assim elimina a utilização de marcadores e procedimentos de calibração (ZERPA et al., 2015).

Para obter o mapa de profundidade o *Kinect* utiliza um modelo geométrico derivado da triangulação estéreo (em imagem computacional é quando duas câmeras tiram diferentes fotos da mesma cena a par-

Figura 12: Sensor *Kinect*.

Fonte: Research (2016).

tir de distintos pontos de vista). A triangulação na geometria estéreo conta com a calibração de duas câmeras de visualização do mesmo objeto para a obtenção de imagens em 3D (SMISEK; JANCOSK; PAJDLA, 2013).

Para trabalhar com esta câmera é necessário ter conhecimento de três pacotes de *software* para o seu funcionamento no sistema operacional *Microsoft Windows*. São eles:

- OpenNI: é o pacote principal que contém a documentação completa e exemplos de visualização, transmissão e gravação do Kinect.
- NITE: desenvolvido em C# possui documentações e exemplos sobre o rastreamento do esqueleto feito pelo sensor.
- SensorKinet: é um *driver* desenvolvido pela PrimeSense, usado com a *interface* OpenNI.

A Microsoft criou o pacote *Software Developer Kit* (SDK), exclusivamente para o desenvolvimento de aplicativos utilizando o *Kinect*. Contendo *Application programming interface* (API), interface de dispositivos, a documentação técnica e amostras do código fonte.

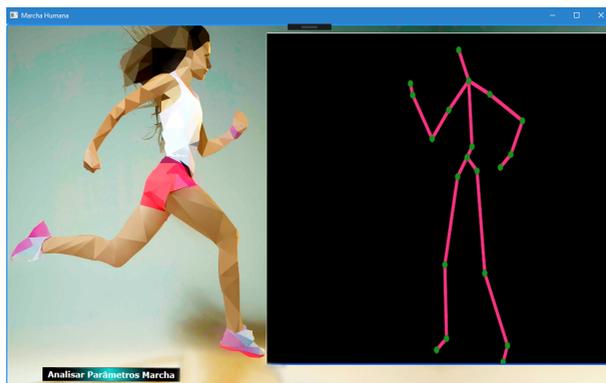
#### 4.2.2 Aplicativo

O método teórico da análise cinética e cinemática do movimento, pode ser implementado computacionalmente por um aplicativo *Windows Presentation Foundation* (WPF) mostrado na Figura 13, um mo-

delo de programação unificado pela plataforma *Microsoft Visual Studio* 2015, na linguagem C#, que permite simular, visualizar o esqueleto humano e o movimento, e analisar os dados obtidos graficamente pelos cálculos citados, na qual estes valores foram armazenados em arquivos .txt.

Para aquisição de imagens e extração dos parâmetros foi utilizado o *Kinect*. Ele foi posicionado paralelamente ao indivíduo, a fim de analisá-lo nos planos sagital, frontal e transversal. Para que o sensor possa identificar o esqueleto na tela do aplicativo, o voluntário inicia o movimento de caminhar no espaço predeterminado, seguindo as especificações técnicas que variam de 1.8m à 2.4m de distância (CUJI et al., 2013) e automaticamente extrai os parâmetros do corpo de acordo com o tempo específico de um ciclo da marcha.

Figura 13: Aplicativo WPF.



Fonte: Própria autoria.

O sistema de medição *Kinect* permite ao usuário identificar visualmente os pontos de interesse da alimentação de vídeo, foi utilizado o algoritmo de rastreamento de código aberto dos marcadores do corpo, no qual permitiu a implementação dos cálculos da biomecânica simultaneamente para a medição de distâncias e parâmetros a serem tomados.

#### 4.2.3 Resultados gráficos

Os valores alcançados foram gerados pelo protótipo *software* desenvolvido, comparados e julgados a partir de resultados presentes nos

estudos da biomecânica apontados por Hamill e Knutzen (2014). Foram executados os testes em cada voluntário, ao total de 10 pessoas, para avaliar a correlação entre os valores obtidos e validar os resultados.

Foi extraído a FRS como parâmetro cinético e os parâmetro cinemáticos da parte inferior do corpo, a seguir:

- Ângulo oblíquo da pelve.
- Ângulo de rotação da pelve.
- Ângulo de flexão do quadril direito.
- Ângulo de flexão do quadril esquerdo.
- Ângulo de adução do quadril direito.
- Ângulo de adução do quadril esquerdo.
- Ângulo de flexão do joelho direito.
- Ângulo de flexão do joelho esquerdo.
- Ângulo de flexão do tornozelo direito.
- Ângulo de flexão do tornozelo esquerdo.

No sistema 3D, para obter uma identificação precisa é necessário possuir como referência os eixos x, y e z. O conceito de profundidade pelo eixo z, medial e lateral, deve ser acrescentado aos outros vetores bidimensionais da altura pelo eixo y, para cima e para baixo, e largura pelo eixo x, para frente e para trás (HAMILL; KNUTZEN, 2014).

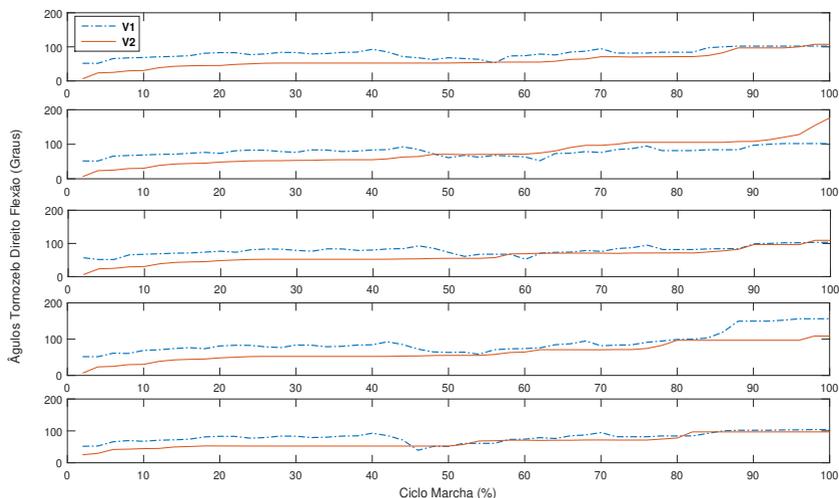
A intersecção dos eixos é definida pelo ponto (0,0,0), no eixo x os movimentos são horizontais e para a direita, no eixo y são verticais e para cima e no eixo z são horizontais e para frente, assim todas as coordenadas no primeiro quadrante são positivas. Os movimentos negativos são para a esquerda em x, para baixo em y e para trás em z (HAMILL; KNUTZEN, 2014).

Após o contato com o solo, a reação à carga agrega ao peso do corpo, assim a cinemática é descrita pelos movimentos angulares inferiores do corpo neste caso. Por método de comparação, é possível analisar os resultados dos voluntários 1 e 2 e voluntários 5 e 9 pelas Figuras 14 e 15 respectivamente. Elas exibem os valores angulares do movimento de flexão plantar do tornozelo direito e movimento de flexão do joelho esquerdo, resultantes de cinco testes distintos.

O gráficos apresentaram os resultados em função de um padrão temporal, o ciclo da marcha (ANDERSON, 1999). De acordo com Oatis

(2003), Knutsson (1972), Murray (1967) a partir de estudos da biomecânica, o ciclo da marcha tem duração de aproximadamente um segundo em uma caminhada. Para este projeto o aplicativo contabiliza os valores em função deste período, definindo o percentual da marcha.

Figura 14: Ângulos de flexão plantar do tornozelo direito.



Fonte: Elaboração própria.

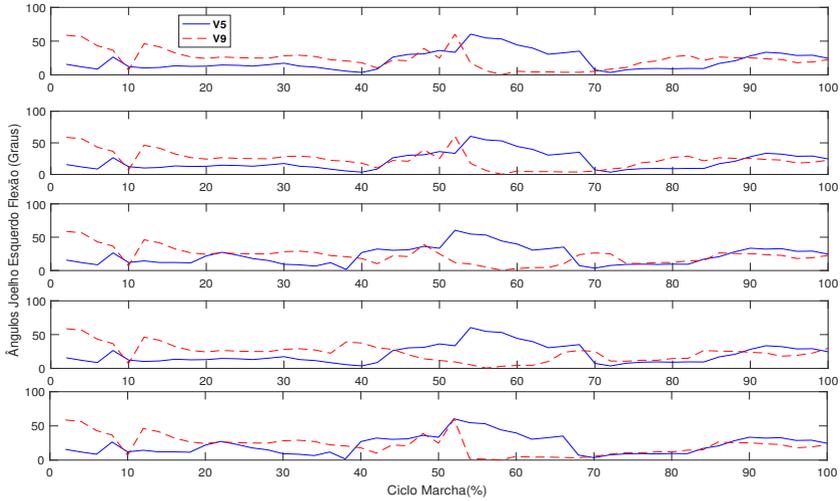
É visto que entre voluntários diferentes há variação dos valores para o mesmo movimento e os resultados para os testes do mesmo voluntário permanecem com valores aproximados, não idênticos, pois é evidente que existem diferenças na marcha em relação a magnitude vetorial do deslocamento em função da variação da velocidade e aos intervalos temporais.

O mesmo acontece com os demais parâmetros, eles apresentaram as mesmas semelhanças aos intervalos angulares, ao analisar os testes de cada voluntário separadamente.

Os resultados cinéticos são representados na Figura 16 pela força de reação do solo, seu conceito é descrito pela lei de ação e reação de Newton, admite as reações do solo em função da aceleração de todos os segmentos do corpo (ENOKA, 2000).

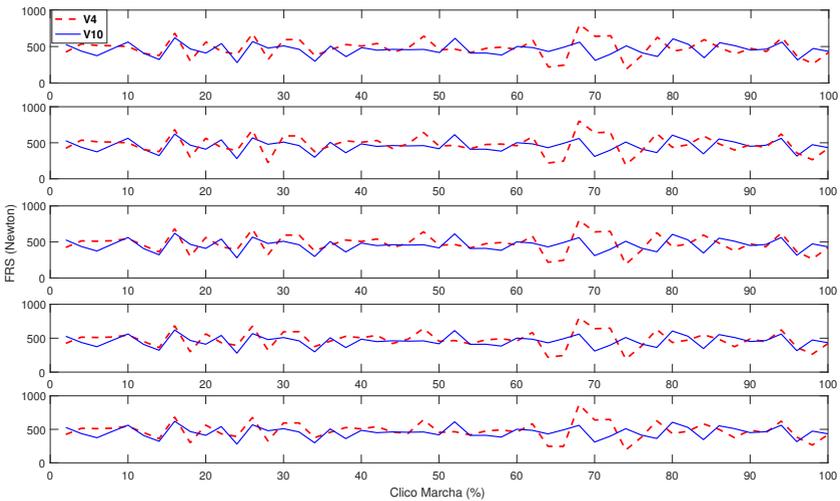
A FRS se caracteriza por possuir uma forma bimodal, onde obtemos dois valores máximos, o primeiro pico representa a fase em que todo corpo é abaixado após o contato do pé, ocorrendo a sustentação completa do peso e a massa corporal é acelerada para cima. A força

Figura 15: Ângulos de flexão do joelho esquerdo.



Fonte: Elaboração própria.

Figura 16: Força de Reação do Solo.



Fonte: Elaboração própria.

diminui quando o joelho flexiona, acarretando uma redução parcial da carga. O segundo pico representa o empuxo em função do solo, para o

movimento do passo seguinte (HAMILL; KNUTZEN, 2014).

### 4.3 CONCLUSÃO

A partir das análises biomecânicas identificados na revisão da literatura, foi possível desenvolver o aplicativo de extração automática dos parâmetros. Com objetivo de desenvolver um sistema de medição de baixo custo utilizando o sensor *Kinect*, obtém-se um mecanismo que consegue determinar as características da análise de marcha humana para a aplicação em reconhecimento de padrões que será abordado no próximo capítulo.



## 5 IDENTIFICAÇÃO BIOMÉTRICA DE INDIVÍDUOS ATRAVÉS DO RECONHECIMENTO DE PADRÕES

### 5.1 INTRODUÇÃO

Nesse Capítulo serão apresentados os mecanismos estatísticos utilizados na classificação dos dados de entrada, extraídos pelo aplicativo de aquisição dos parâmetros da marcha, apresentado no capítulo anterior. Utilizando a ferramenta *Toolbox* do *software Matlab*, aponta-se a estrutura empregada e os resultados de cada classificador durante a implementação e validação.

O reconhecimento de padrões é um ramo da aprendizagem de máquina que incide sobre o reconhecimento e regularidades em dados, por técnicas de aprendizado supervisionado ou sem supervisão (NAM, 2016). Estes domínios têm evoluído substancialmente das raízes da inteligência artificial, estatística e engenharia (BISHOP, 2006).

Esta técnica atribui parâmetros para um determinado valor de entrada, dado que a classificação efetua tentativas para atribuir a cada valor de entrada um determinado conjunto de classes. Estes algoritmos têm como objetivo proporcionar uma resposta razoável para todas as entradas possíveis e realizar a correlação entre as entradas mais prováveis, tendo em conta a sua variação estatística (BURGES, 1998; NAM, 2016).

O aprendizado supervisionado determina um conjunto de dados de treinamento de entrada, que consiste em um conjunto de amostras que foram devidamente direcionadas com a saída desejada. Aplica-se um procedimento de aprendizagem, e em seguida, gera um modelo que tenta melhorar o desempenho sobre os dados de treinamento. A aprendizagem não supervisionada, assume dados de treino sem haver uma saída desejada para coincidir com a resposta, por tentativas a rede busca encontrar padrões inerentes, que possam ser utilizados para determinar o valor correto da saída para novas instâncias de dados (FU, 1968),(BISHOP, 2006).

#### 5.1.1 Base de dados

Para os dados de entrada foi criada uma matriz de amostras composta por valores numéricos de 11 parâmetros da marcha, apresentados no Capítulo 4, constituída por 10 voluntários (8 homens e 2

mulheres), com idade entre 20 e 30 anos e massa corporal entre 50 e 110 kg. Foram efetuados 5 testes em cada voluntário, cada teste contém os 11 parâmetros, e cada parâmetro 50 amostras. Ao total a base de dados gera uma matriz de entrada de 11 x 2500.

Para os classificadores que utilizam o aprendizado supervisionado foi determinada uma matriz de saída desejada com 10 classes, dado que cada classe representa um voluntário e estão direcionadas aos parâmetros dos dados de entrada. O *dataset* armazena os parâmetros da marcha humana contendo ao total 10 classes e 27500 amostras.

## 5.2 CLASSIFICADORES

De acordo com Lorena (2006) há vários problemas que envolvem a discriminação de dados em classes ou categorias, em função de uma tarefa chamada de classificação. Através do conhecimento das classes, que formam o conjunto de dados, os algoritmos de aprendizagem de máquina podem ser utilizados por classificadores que conseguem fazer a discriminação desejada para novos dados do mesmo domínio.

### 5.2.1 *Support Vector Machine* - SVM

Gianaria et al. (2014) aborda que este classificador é um algoritmo robusto de clusterização e amplamente utilizado para classificar pessoas em função de suas características biométricas e comportamentais. Este método faz um mapeamento de dados em um espaço de entrada dimensional e constrói um hiperplano ou conjuntos de hiperplanos de separação entre as classes. O SVM utiliza métodos de aprendizado supervisionado que analisam os dados para reconhecimento de padrões para classificá-los (SUYKENS; VANDEWALLE, 1999).

O problema para a classificação pode ser declarado em um espaço de dimensão finita, dado que as classes para discriminar não são linearmente separáveis neste espaço. Portanto, define-se que o espaço de dimensão finita inicial tem que ser mapeado para um espaço muito maior, tornando a separação mais fácil. Utilizando a função de Kernel o SVM assegura uma carga computacional razoável, ao calcular mais facilmente os produtos de cada amostra referente a sua resposta desejada (SUTHAHARAN, 2016; HEARST et al., 1998).

Os hiperplanos no espaço de dimensão superior são definidos como o conjunto de amostras cujo produto escalar é uma constante.

Os vetores que definem os hiperplanos são definidos por uma combinação linear entre os parâmetros e as amostras. Assim os pontos são mapeados no espaço de treinamento a partir da relação destes vetores (MEYER; WIEN, 2015).

É efetuado o somatório de cada amostra de teste e saída desejada, para determinar o grau de proximidade entre os dois termos. Deste modo, a soma é utilizada para medir a proximidade relativa de cada ponto de teste em função dos pontos provenientes de outras classes a serem discriminadas (PRESS et al., 2007).

### 5.2.1.1 Aplicação e resultados - SVM

SVM foi desenvolvido para solução de problemas com apenas duas classes, também conhecidos como binários. Mas para este trabalho requer a discriminação de dados em mais de duas classes. É visto que este método é um classificador linear binário não probabilístico, assim podemos obter a performance da classificação não linear utilizando métodos de Kernel o que implica em uma mapeamento de características de altas dimensões (KUMAR; SINGH, 2015).

Para a classificação não linear, o espaço dimensional da função é mapeado em um espaço maior, para que seja possível encontrar um hiperplano de separação linear como margem máxima neste espaço (GANARIA et al., 2014).

Implementou-se no *software Matlab* o algoritmo SVM, utilizando a estrutura *svmtrain*, *svmclassify* e a função *kernelfunction* polinomial foi configurada para mapear os dados de treinamento. As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks (2016d, 2016e). Ao utilizar a função Kernel não necessita cálculo explícito dos dados no espaço, apenas precisa-se computar os produtos internos entre todos os pares de dados (SHMILOVICI, 2005).

Os dados de entrada foram divididos em grupos de treinamento e teste. Os parâmetros são computados usando o procedimento de validação cruzada para que o conjunto de treinamento seja dividido em subconjuntos de igual tamanho e um único é testado usando o classificador treinado nos restantes. O processo é repetido várias vezes para cada um e os resultados são calculados por estimativa.

Para avaliação do desempenho usamos a precisão da classificação pela Equação (5.1), onde  $\lambda$  é o número de amostras classificadas corre-

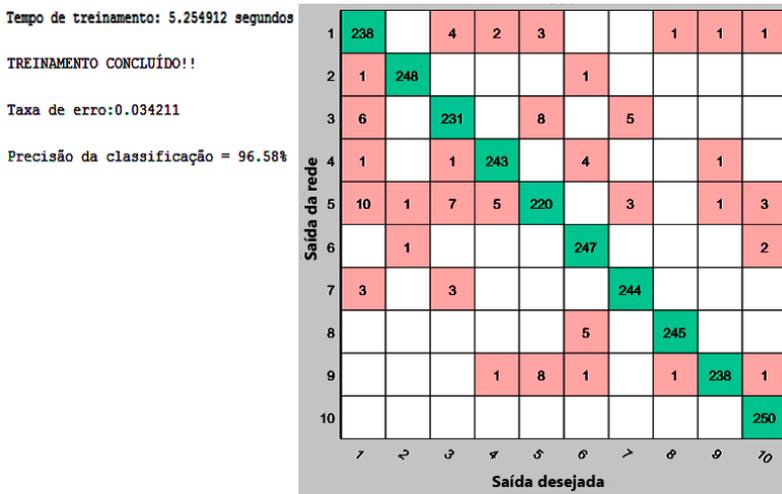
tamente e  $\Phi$  é número de amostras desejadas (GIANARIA et al., 2014):

$$P = \frac{\lambda}{\Phi} * 100\% \quad (5.1)$$

A precisão da classificação obtida pelo SVM foi de 96,58%. De acordo com a Figura 17, pode-se analisar a matriz confusão onde a diagonal em destaque, na cor verde, representa as amostras classificadas corretamente. Os demais valores, na cor em vermelho, são os erros de classificação.

O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pelo SVM, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. Apenas o voluntário 10 obteve 100% de acerto na classificação. O tempo decorrido para o treinamento foi de 5,255 segundos e a taxa de erro foi 0,034.

Figura 17: Resultados - SVM.



Fonte: Matlab.

### 5.2.2 Multilayer Perceptron Neural Network - MLP

A rede neural de múltiplas camadas, é um tipo de rede *feed-forward* que forma grupos sensoriais com uma camada de entrada, uma

ou mais camadas escondidas e uma camada de saída, onde sinal de entrada é propagado através desta arquitetura (HAYKIN et al., 2009). Este classificador possui um potencial maior por tratar de dados não linearmente separáveis de múltiplas saídas (ORTIZ, 2010).

As camadas de neurônios ocultas permitem que a rede aprenda tarefas complexas extraíndo informações dos padrões de entrada em cada etapa do processo de treinamento. A aplicação de várias camadas ocultas pode ocasionar o aumento do espaço de padrões que a rede representa, tornando-a um processo dinâmico (ORTIZ, 2010; ORHAN; HEKIM; OZER, 2011).

O treinamento desta rede é feito utilizando o algoritmo *Backpropagation*, o qual tem como princípio utilizar o gradiente descendente e estimar o erro das camadas intermediárias pelas estimativa do resultado encontrado no erro da camada de saída. Este algoritmo é supervisionado (BISHOP, 2006) e usa pares de entrada e saída da rede para efetuar o ajuste dos pesos (WANG; SHUN, 2016; KAUR et al., 2016).

Na fase de propagação, após determinado os dados de entrada, a resposta é propagada como entrada da próxima camada, até identificar a reposta da rede e calcular o erro, na camada de saída. Durante a retropropagação do erro, a partir da camada de saída até a camada de entrada, são feitos os ajustes dos pesos (HAYKIN et al., 2009).

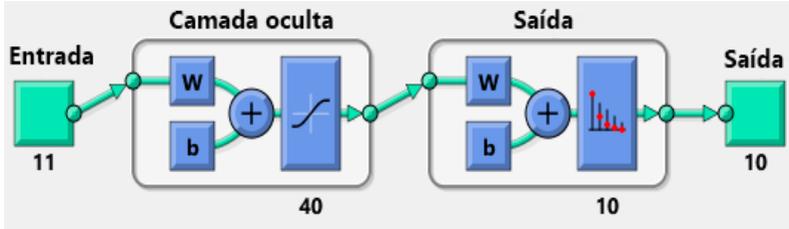
Durante o treinamento a saída é comparada com o valor desejado para computar o erro global da rede, que influencia na correção dos pesos, na retro propagação, então calcula-se o erro e este é retroalimentado para as camadas intermediárias, para o ajuste dos pesos. Os pesos são atualizados até atingir o número de épocas (iterações) necessárias, e os ajustes são feitos conjunto aos erros para cada padrão apresentado pela rede (ORTIZ, 2010).

### 5.2.2.1 Aplicação e resultados - MLP

Nesta etapa foi implementado a rede neural de múltiplas camadas para analisar os parâmetros da marcha humana utilizando o *software Matlab*. A estrutura da rede é composta pela camada de entrada com 2500 amostras dos 11 parâmetros, uma camada oculta sendo a primeira com 40 neurônios e a camada de saída com 10 classes referente aos voluntários, exibida na Figura 18. Os dados de entrada foram divididos em grupos de treinamento, validação e teste.

Para o treinamento utilizamos a função *Backpropagation* que utiliza derivadas Jacobianas, estes algoritmos podem ser mais rápidos

Figura 18: Estrutura da MLP.



Fonte: Matlab.

mas requerem mais memória para seu processamento, conhecido como *Levenberg-Marquardt backpropagation* (*trainlm*). Na camada oculta utiliza-se a função de transferência sigmoide simétrica (*tansig*), e na camada de saída a função de transferência *Softmax*. As funções utilizadas são pré definidas pelo *software Matlab*, de acordo com a documentação MathWorks (2016a), estas realizam a conversão dos dados de entrada da camada em dados de saída.

Para avaliação do desempenho usamos a precisão da classificação pela Equação (5.1), assim adquire-se o valor da MLP em 97,6%. De acordo com a Figura 19, pode-se analisar a matriz confusão onde a diagonal em destaque, na cor verde, representa as amostras classificadas corretamente. Os demais valores, na cor em vermelho, são os erros de classificação.

O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pelo MLP, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. As amostras dos voluntários 8 e 10 foram 100% classificadas. O tempo decorrido para o treinamento foi de 0,961 segundos e a taxa de erro foi 0,024.

### 5.2.3 *Random Forest* - RF

*Random forest* ou floresta aleatória é um classificador que consiste em uma coleção de classificadores estruturados por árvores de decisão, onde cada amostra é distribuída de forma idêntica em cada árvore. A saída final é dada quando obter o resultado mais comum para cada amostra, assim a nova amostra é alimentada em todas as árvores com o propósito de elegerem um voto de maioria para cada modelo de classificação (GARG et al., 2016; BREIMAN, 2001).



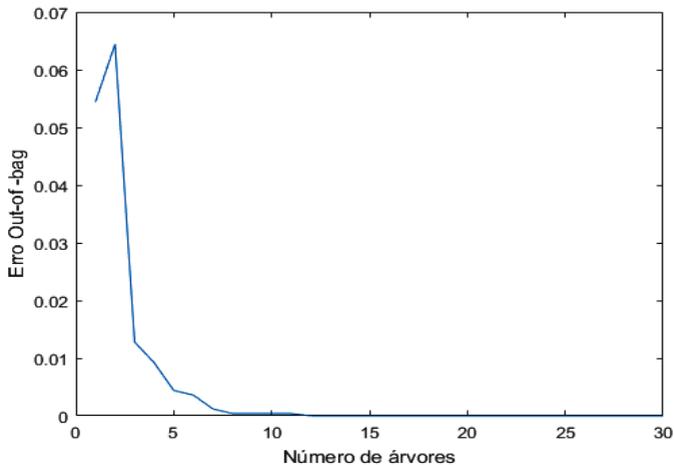
média dos modelos e geralmente produz melhores resultados (BREIMAN, 1996, 2001). As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks (2016d, 2016e).

Esta técnica incorpora a regularização dos dados e somente é necessário escolher bons parâmetros para os algoritmos de base. A média dos modelos leva a eliminar os modelos instáveis que podem ser produzidos a partir de dados tendenciosos (SALLES et al., 2015).

Para avaliação do desempenho utilizamos o método de predição *out-of-bag* (SALLES et al., 2015). O valor obtido da precisão pela RF foi de 97,4% de acerto. De acordo com a Figura 20 pode-se observar o comportamento do erro em função do número de árvores. Conforme a Figura 21, pode-se analisar a matriz confusão onde a diagonal em destaque, representa as amostras classificadas corretamente. Os demais valores, são os erros de classificação.

O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pelo RF, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. As amostras dos voluntários 8 e 10 foram 100% classificadas. O tempo decorrido para o treinamento foi de 1,923 segundos e a taxa de erro foi 0,026.

Figura 20: Resultados - RF - Erro *Out-of-bag*.



Fonte: Matlab.

Figura 21: Resultados - RF.

```

Processando banco de dados...
Treinando a rede...
Duração:      1.922956 segundos

TREINAMENTO CONCLUÍDO!!
Taxa de erro = 0.025600
Precisão da classificação = 97.4%
Matriz Confusão:

```

1	236	0	1	1	5	3	2	1	1	0
2	1	248	0	0	0	0	0	0	0	1
3	3	0	242	0	3	0	2	0	0	0
4	1	0	0	247	1	1	0	0	0	0
5	7	1	4	3	229	0	4	0	1	1
6	2	3	0	1	0	243	0	0	1	0
7	0	1	1	0	0	2	246	0	0	0
8	0	0	0	0	0	0	0	250	0	0
9	0	0	0	2	1	0	1	0	245	1
10	0	0	0	0	0	0	0	0	0	250
	1	2	3	4	5	6	7	8	9	10

Classes desejadas

Fonte: Matlab.

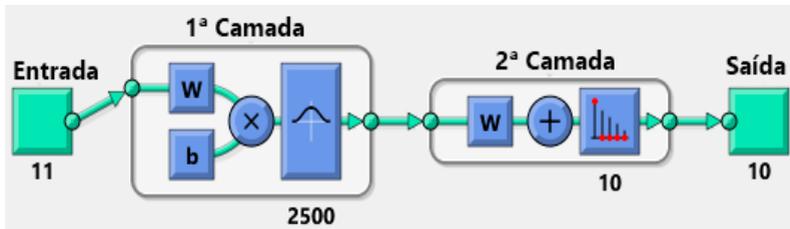
### 5.2.4 Probabilistic Neural Network - PNN

As redes neurais probabilísticas foram desenvolvidas como um diferente paradigma, e com novas regras de aprendizagem, para determinar estatisticamente um conjunto de amostras de treinamento e classificar os novos padrões com bases nestas estatísticas definidas por Specht (1990). Operacionalmente é uma rede de fácil treinamento, ao contrário de muitas outras redes ela opera completamente em paralelo sem a necessidade de realimentação de neurônios individuais para as entradas (SPECHT, 1990; ABDULLAH; ABDEL-QADER, 2016).

Segundo (TEMURTAS, 2009) a rede PNN oferece uma solução para problemas de classificação de padrões, a partir de uma abordagem estatística dos classificadores Bayesianos (LANGLEY; IBA; THOMPSON, 1992; MIR et al., 2016) e também utiliza estimadores de Parzen (LAPKO; LAPKO, 2016; PW, 1976) desenvolvidos para construir funções probabilísticas de densidade pela teoria de Bayes (GONG et al., 2016; SHAHADAT; PAL, 2015; SAHA et al., 2016).

Na arquitetura mostrada na Figura 22, quando uma entrada é apresentada, a primeira camada calcula as distâncias do vetor de entrada para os dados treinamento, e produz um vetor cujos elementos indicam quão próximo estão os dados de entrada para os de treinamento. A segunda camada resume essas contribuições para cada classe de entradas, produzindo como sua saída um vetor de probabilidades. Finalmente, uma função de transferência concorrente na saída da segunda camada escolhe o máximo dessas probabilidades, produzindo 1 ou 0 para a determinar a precisão da rede (MATHWORKS, 2016b; SPECHT, 1990).

Figura 22: Arquitetura PNN.



Fonte: Matlab.

É visto que a quantidade de neurônios na camada da função de transferência de base radial (PRATHIBA et al., 2016; MATHWORKS, 2016c), é diferente de outras redes neurais. O número de neurônios é igual a distância vetorial entre seu vetor peso e seu vetor de entrada, multiplicado pelo bias. À medida que essa distância diminui, a saída aumenta. Assim, o neurônio de base radial atua como um detector que produz o valor 1 sempre que a entrada é idêntica ao seu vetor peso (MATHWORKS, 2016b; TEMURTAS, 2009).

#### 5.2.4.1 Aplicação e resultados - PNN

Para este estudo foi elaborada um rede neural probabilística de multicamadas implementada no *Matlab*, na qual utiliza-se a função *newpnn*. As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks (2016b). Sua estrutura mostrada na Figura 22 é composta de uma camada oculta ligada a uma camada de saída. Sendo o vetor de entrada definido com os valores dos parâmetros da marcha e a saída com 10 classes, referente a cada voluntário.

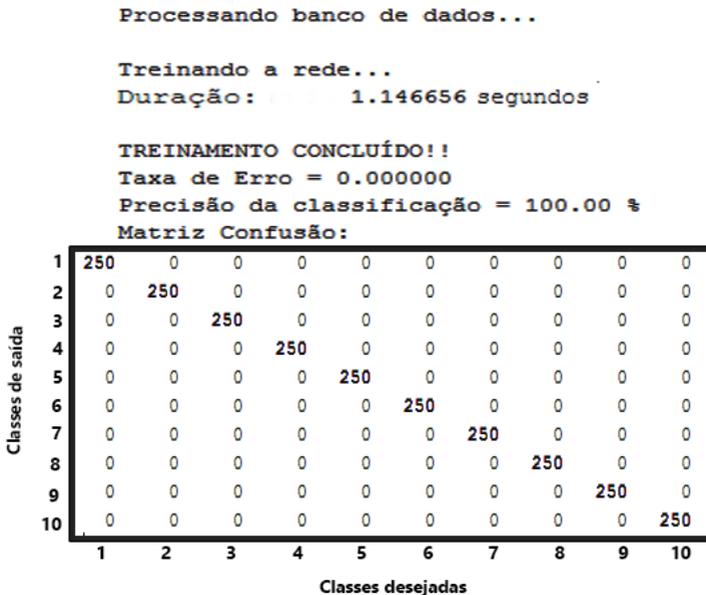
O vetor de entrada é destinado para os nós da camada oculta

pelos pesos, este nó calcula a distância euclidiana (OZYILDIRIM; AVCI, 2016; ABDULLAH; ABDEL-QADER, 2016) entre o centro e o vetor de entrada da rede, entregando o resultado para a função de base radial (*Radial basis function* - RBF) do tipo gaussiana (PRATHIBA et al., 2016; MATHWORKS, 2016c; OZYILDIRIM; AVCI, 2016).

Para avaliação do desempenho usamos a precisão da classificação pela Equação (5.1). O valor obtido pela PNN foi de 100% . De acordo com a Figura 23 pode-se analisar a matriz confusão onde a diagonal em destaque, representa as amostras classificadas corretamente. Os demais valores, são os erros de classificação.

O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pelo PNN, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. As amostras de todos voluntários foram 100% classificadas. O tempo decorrido para o treinamento foi de 1,147 segundos e a taxa de erro foi 0,000.

Figura 23: Resultados - PNN



Fonte: Matlab.

### 5.2.5 *Naive Bayes* - NB

*Naive Bayes* é um classificador probabilístico que utiliza o teorema de Bayes (JØSANG, 2016; PFEIFER, 2016) para aplicação. São altamente escaláveis, exigindo uma série de parâmetros lineares no número de variáveis no problema de aprendizagem. O treinamento pode ser feito através da avaliação de uma expressão de forma fechada, em vez da aproximação iterativa utilizado por outros tipos de classificadores (RUSSELL et al., 2003).

Este modelo atribui classes para instâncias de problemas, representados como vetores de características, onde essas classes são extraídas de um conjunto finito. A classificação tem como objetivo um algoritmo de aprendizagem que treine o conjunto de dados de entrada em função das classes determinadas (XU, 2016; LEWIS, 1998).

Apesar de ser simplista, classificadores *Naive Bayes* têm trabalhado muito bem em diferentes e complexas situações do mundo real, precisando apenas de um número de dados de treinamento para estimar os parâmetros necessários para a classificação (ZHANG, 2004).

#### 5.2.5.1 Aplicação e resultados - NB

Para este estudo foi elaborado um classificador *Naive Bayes* implementado no *Matlab*, utilizando a função *fitcnn*. Dado que o vetor de entrada é definido com os valores dos parâmetros da marcha e a saída desejada com 10 classes, referente a cada voluntário. As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks (2016e).

Para avaliação do desempenho usamos a função *resubloss*, que define a precisão da classificação. Esta função mede a imprecisão de modelos de classificação (MATHWORKS, 2016e). Ao comparar o mesmo tipo de perda entre muitos modelos, a baixa perda indica um melhor modelo. A precisão obtida pela NB foi de 97.6%.

De acordo com a Figura 24 pode-se analisar a matriz confusão onde a diagonal em destaque, representa as amostras classificadas corretamente. Os demais valores, são os erros de classificação.

O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pelo NB, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. As amostras voluntário 1 foram 100% classificadas. O tempo decorrido para o treinamento foi de 2,069 segundos e a taxa de

erro foi 0,024.

Figura 24: Resultados - NB

```

Processando banco de dados...
Treinando a rede...
Duração:      2.068834 segundos

TREINAMENTO CONCLUÍDO!!

Taxa de erro:0.024000

Precisão da classificação = 97.6%

Matriz Confusão:

```

1	250	0	0	0	0	0	0	0	0	
2	0	242	1	1	0	5	0	0	1	
3	2	0	242	0	1	0	0	0	5	
4	0	1	0	240	0	2	0	7	0	
5	0	0	1	0	246	3	0	0	0	
6	0	4	0	1	2	243	0	0	0	
7	2	0	0	0	1	0	244	0	3	
8	0	0	0	2	0	0	0	247	1	
9	0	0	5	0	0	0	0	0	245	
10	0	0	0	0	3	6	0	0	0	241
	1	2	3	4	5	6	7	8	9	10

Classes desejadas

Fonte: Matlab.

### 5.2.6 Learning Vector Quantization Neural Network - LVQ

O LVQ foi introduzido por Kohonen (1986) e pôde ser expressado como um caso especial de rede neural artificial (Artificial neural network - ANN), dado que aplica-se a técnica de treinamento *winner-take-all* (BIEHL; GHOSH; HAMMER, 2006; AMEZCUA; MELIN; CASTILLO, 2016).

Este algoritmo é um precursor de um tipo de rede neural artificial que é treinada utilizando o aprendizado não supervisionado para produzir a representação de baixa dimensão do espaço de entrada das amostras de treino, chamado de *self-organizing maps* (SOM) (KOHONEN, 1995; KOHONEN; SOMERVUO, 1998; NEBEL; VILLMANN, 2016).

*Winner-take-all* é um caso de aprendizagem competitiva em redes neurais. Os nós da saída da rede inibem mutualmente um ao outro, enquanto simultaneamente são ativados pelas conexões reflexivas. Depois apenas um nó na camada de saída será ativado, correspondente

a entrada mais forte. (GROSSBERG, 1982; BIEHL; GHOSH; HAMMER, 2006; AMEZCUA; MELIN; CASTILLO, 2016).

O algoritmo de treinamento *winner-take-all* determina, para cada amostra, a classe que está mais próxima da entrada de acordo com a distância. Assim a classe vencedora é adaptada, ou seja, o vencedor que aproximou-se corretamente classifica a amostra ou o mais distante classifica incorretamente os dados (GROSSBERG, 1982; AMEZCUA; MELIN; CASTILLO, 2016).

De acordo com (TEMURTAS, 2009) uma rede LVQ particiona os dados em conjuntos de treinamento e teste, baseando-se na similaridade deles. Esta rede tem uma camada competitiva que classifica a entrada dos vetores, e uma camada de saída linear para transformar as classes da camada competitiva, chamadas de subclasses, em classificações definidas pelo usuário.

Aprendizagem competitiva é utilizada em redes neurais que possuem uma camada oculta, chamada de camada competitiva (SALATA, 2011). As classes definidas na camada de saída linear são denominadas classes de destino (VILLMANN et al., 2016).

Segundo Salata (2011), para cada vetor de entrada, os neurônios competitivos competem uns aos outros para encontrar o neurônio vencedor, na qual seu vetor peso tem que possuir a maior semelhança com esse outro vetor de entrada em particular.

O algoritmo LVQ é implementado através de uma rede neural utilizando uma camada competitiva. Assim é necessário a inicialização dos pesos, e em cada época um vetor é apresentado como entrada para a rede. A distância de cada centroide é calculada e o neurônio vencedor possui o menor valor da distância Euclidiana (OZYILDIRIM; AVCI, 2016; ABDULLAH; ABDEL-QADER, 2016). Por fim atualiza os vetores de pesos, devido ao movimento do neurônio vencedor em relação ao vetor de entrada. Onde a quantidade de movimentos é definida por uma taxa de aprendizado (SALATA, 2011; RAO; MURTHY, 2016).

Segundo Kohonen (1995) o sistema LVQ é utilizado em uma variedade de aplicações práticas e pode ser aplicado a problemas de classificação multi-classe de forma natural.

### 5.2.6.1 Aplicação e resultados - LVQ

Para este estudo foi elaborado um classificador LVQ implementado no *Matlab*, utilizando a função *lvqnet*. As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks

(2016a).

Utilizou-se neste trabalho a constituição da rede por uma camada oculta, a camada competitiva, e a camada de saída linear. Os dados de entrada foram definidos com os valores dos parâmetros da marcha e a saída com 10 classes, referente a cada voluntário. O vetor de entrada é encaminhado para os nós da camada oculta através dos pesos e o nó da camada oculta calcula a distância euclidiana entre o centro e o vetor de entrada da rede, e retorna o resultado para as funções competitivas, assim definindo uma estrutura de multicamadas.

Ao efetuar o treinamento com a matriz de entrada composta pelos 5 testes de cada voluntário, o algoritmo se torna incapaz de processar os dados, necessitando de uma grande demanda de processamento da máquina e em certas etapas não consegue efetuar todo processo. Neste caso determinamos um valor menor de amostras na entrada da rede, de 27500 para 1650 amostras, composta por apenas 1 teste de extração dos parâmetros referentes aos voluntários 1, 2 e 3.

Para avaliação do desempenho usamos a Equação (5.1) que define a precisão da classificação. A precisão obtida pelo LVQ foi de 36,0%. De acordo com a Figura 25, pode-se analisar a matriz de confusão onde a diagonal em destaque, na cor verde, representa as amostras classificadas corretamente. Os demais valores, na cor em vermelho, são os erros de classificação.

O eixo vertical na matriz de confusão corresponde ao resultado da classificação das classes determinado pelo LVQ, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. Mesmo ao reduzir a quantidade de amostras a ser classificadas, o algoritmo não apresentou uma precisão satisfatória para o treinamento dos parâmetros da marcha humana.

O tempo decorrido para o treinamento de menor número de amostras foi de 17,129 segundos e a taxa de erro foi 0,640. Quando efetuado o treinamento para 27500 amostras, o tempo de treinamento foi superior a 1 hora, e apresentava erro na operação do programa ou fechava o software.

### 5.2.7 *k-Nearest Neighbors* - kNN

O classificador kNN é um método não paramétrico aplicado para classificação em reconhecimento de padrões (ANAVA; LEVY, 2016). Nesta classificação a entrada é formada por amostras de treinamento e a saída é uma associação de classes. Uma amostra é classificada pelo voto da

Figura 25: Resultados - LVQ

```
Treinando a rede...
Elapsed time is 17.128566 seconds.

TREINAMENTO CONCLUÍDO!!
Taxa de erro:0.640000

Precisão da classificação = 36.0%
```

Classes de saída	1	9 6.0%	41 27.3%	0 0.0%	18.0% 82.0%
	2	5 3.3%	45 30.0%	0 0.0%	90.0% 10.0%
	3	8 5.3%	42 28.0%	0 0.0%	0.0% 100%
		40.9% 59.1%	35.2% 64.8%	NaN% NaN%	36.0% 64.0%
		1	2	3	Classes desejadas

Fonte: Matlab.

maioria de seus vizinhos, que é atribuída à classe mais comum entre os seus vizinhos mais próximos (ALTMAN, 1992; ZHANG et al., 2016).

Os exemplos de treinamento são vetores em um espaço recurso multidimensional, para cada classe. A fase de treinamento do algoritmo consiste apenas armazenar os vetores de características e as classes das amostras de treinamento. Na fase de classificação, o vetor de teste é classificado pela atribuição da classe mais frequente entre as definidas pelas amostras de treinamento mais próximo a esse ponto de consulta (ANAVA; LEVY, 2016; JASKOWIAK; CAMPELLO, 2011).

Os dados podem ser escalares ou vetores multidimensionais, contidos em um espaço métrico. A distância utilizada para a métrica das variáveis contínuas é a distância euclidiana. Muitas vezes, a precisão da classificação de kNN pode ser significativamente melhorada se a métrica de distância é treinada com algoritmos especializados (WU; YANG; WANG, 2016; JASKOWIAK; CAMPELLO, 2011), como *large margin nearest neighbor* (WEINBERGER; BLITZER; SAUL, 2005; WEINBERGER; SAUL, 2009) ou *neighbourhood components analysis* (GOLDBERGER et al., 2004).

### 5.2.7.1 Aplicação e resultados - kNN

Para este estudo foi elaborado um classificador kNN implementado no *Matlab*, utilizando a função *fitcknn*. Dado que o vetor de entrada é definido com os valores dos parâmetros da marcha e a saída desejada com 10 classes, referente a cada voluntário. As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks (2016e).

Para avaliação do desempenho usamos a função *resubloss*, que define a precisão da classificação. Esta função mede a imprecisão de modelos de classificação (MATHWORKS, 2016e). Ao comparar o mesmo tipo de perda entre muitos modelos, a baixa perda indica um melhor modelo. A precisão obtida pela kNN foi de 99,2%.

De acordo com a Figura 26 pode-se analisar a matriz confusão onde a diagonal em destaque, representa as amostras classificadas corretamente. Os demais valores, são os erros de classificação.

O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pelo kNN, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. As amostras dos voluntários 1, 3, 4, 8, 9 e 10 foram 100% classificadas. O tempo decorrido para o treinamento foi de 0,626 segundos e a taxa de erro foi 0,008.

### 5.2.8 *k-means*

*k - means* é um método de quantização vetorial, muito utilizado para análise de *cluster* em mineração de dados. Este algoritmo particiona as amostras em grupos, onde cada amostra pertence ao conjunto das mais próximas da média. Isso resulta na fragmentação do espaço de dados e tende encontrar aglomerados de amostras neste espaço para comparação e classificar novos dados (VATTANI, 2011; PÉREZ et al., 2016).

Dado um conjunto de amostras, o algoritmo particiona as  $n$  amostras, para minimizar o *within cluster sums of squares* (WCSS) (KAMINKA et al., 2016), caracterizado como a soma das distâncias de cada ponto para o centro dos grupos, na qual seu objetivo é encontrar onde a média dos pontos (SINGH; NAGAR; SANT, 2016).

Segundo Mackay (2003) o algoritmo alterna entre duas etapas:

- Etapa de atribuição: ele atribui cada amostra ao cluster cuja média produza o WCSS. Uma vez que a soma dos quadrados é o

Figura 26: Resultados - KNN

```

Processando banco de dados...
Treinando a rede...
Duração:          0.626032 segundos

TREINAMENTO CONCLUÍDO!!
Taxa de erro:0.008000

Precisão da classificação = 99.2%

Matriz Confusão:

```

1	250	0	0	0	0	0	0	0	0	0
2	0	246	0	0	0	2	1	0	1	0
3	0	0	250	0	0	0	0	0	0	0
4	0	0	0	250	0	0	0	0	0	0
5	0	0	1	0	249	0	0	0	0	0
6	3	1	2	1	2	238	0	1	0	2
7	2	0	0	0	0	0	247	0	0	1
8	0	0	0	0	0	0	0	250	0	0
9	0	0	0	0	0	0	0	0	250	0
10	0	0	0	0	0	0	0	0	0	250
	1	2	3	4	5	6	7	8	9	10

Classes desejadas

Fonte: Matlab.

quadrado da distância Euclidiana, isto é, o mais próximo. Onde cada amostra é atribuída exatamente em um conjunto, mesmo que pudesse ser atribuída a dois ou mais deles.

- Etapa de atualização: esta etapa calcula as novas médias para ser os centroides, ponto associado ao centro geométrico, das observações nos novos *clusters*. A média aritmética é um estimador de mínimos quadrados, isso também minimiza WCSS.

O algoritmo converge quando as atribuições não mudam mais, como ele é um algoritmo heurístico, não há nenhuma garantia que irá convergir para o ótimo global, e o resultado pode depender dos grupos iniciais (MACKAY, 2003; KAMINKA et al., 2016).

### 5.2.8.1 Aplicação e resultados - *k-means*

Para este estudo foi elaborado um classificador *k-means* implementado no *Matlab*, utilizando a função *kmeans*. Os dados de entrada

foram definidos com os valores dos parâmetros da marcha e a saída com 10 classes, referente a cada voluntário. As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks (2016e).

Para avaliação do desempenho usamos a função (5.1) que define a precisão da classificação. Ao efetuar o treinamento com a matriz de entrada composta pelos 5 testes de cada voluntário, o algoritmo não conseguiu classificar as amostras, totalizando a primeira performance do k-means em 13,2%.

De acordo com a Figura 27 pode-se analisar a 1ª matriz confusão onde a diagonal em destaque, representa as amostras classificadas corretamente. Os demais valores, são os erros de classificação. O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pelo *k-means*, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizadas para o treinamento.

Figura 27: Resultados - *k-means* - 1ª Matriz

```

Processando banco de dados...
Treinando a rede...
Duração:          0.434070 segundos

TREINAMENTO CONCLUÍDO!!
Taxa de erro:0.867600

Precisão da classificação = 13.2%

Matriz Confusão:

```

1	<b>13</b>	7	72	12	4	51	24	61	3	3
2	6	<b>20</b>	7	14	20	36	90	57	0	0
3	12	8	<b>76</b>	17	5	48	1	83	0	0
4	30	0	12	<b>152</b>	0	0	51	5	0	0
5	24	2	49	65	<b>0</b>	49	20	41	0	0
6	14	12	0	48	5	<b>13</b>	90	54	10	4
7	11	6	32	7	0	43	<b>41</b>	110	0	0
8	60	10	0	10	50	50	50	<b>10</b>	10	0
9	1	0	29	37	0	17	68	92	<b>6</b>	0
10	25	0	0	185	0	0	40	0	0	<b>0</b>
	1	2	3	4	5	6	7	8	9	10

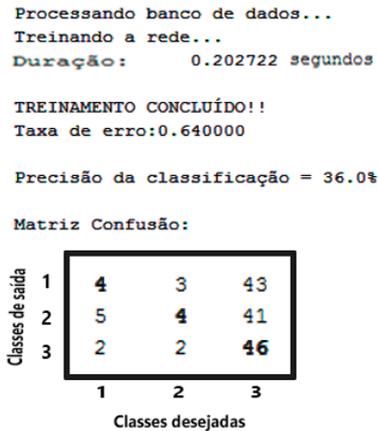
Classes desejadas

Fonte: Matlab.

Logo determinamos um valor menor de amostras na entrada da rede, de 27500 para 1650 amostras, composta por apenas um teste de

extração dos parâmetros referentes aos voluntários 1, 2 e 3. A Figura 28 mostra a 2ª matriz confusão, mesmo reduzindo a quantidade de amostras, o algoritmo não apresentou uma precisão satisfatória para o treinamento dos parâmetros da marcha humana, totalizando a segunda precisão em 36%. O tempo decorrido para o treinamento da 1ª e 2ª performance foi de 0,434 e 0,203 segundos respectivamente e a taxa de erro foi 0,868 e 0,640.

Figura 28: Resultados - *k-means* - 2ª Matriz



Fonte: Matlab.

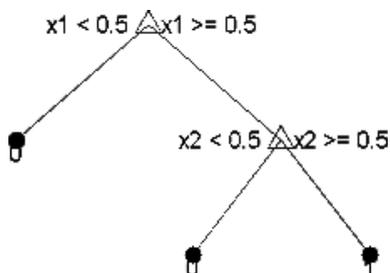
### 5.2.9 Árvore de decisão

A Árvore de decisão mapeia observações sobre uma amostra, representada pelos ramos, e as conclusões sobre o valor desejado desta amostra é representado pelas folhas. Assim sua classificação apresenta respostas que são nominais. Este método é uma das abordagens de modelagem preditiva utilizadas em estatísticas, mineração de dados e aprendizagem de máquina (ROKACH; MAIMON, 2014; OLIVER; HAND, 2016).

Na Figura 29 temos um exemplo de árvore de decisão para classificação. Esta árvore prevê a classificação em duas classes  $x_1$  e  $x_2$ . Iniciam no nó superior representado por um triângulo. A primeira decisão é saber se  $x_1$  é menor que 0.5, caso seja, segue-se o ramo esquerdo e a árvore classifica os dados como do tipo 0. Se, no entanto,  $x_1$  exceder

0.5 seguirá o ramo direito para o nó inferior direito. Aqui a árvore questiona se  $x_2$  é menor do 0.5, caso seja, direciona para o ramo esquerdo e a árvore classifica do tipo 0. Caso seja maior que 0.5, direciona para o ramo direito e a árvore classifica os dados do tipo 1.

Figura 29: Modelo de árvore de decisão.



Fonte: MathWorks (2016e).

### 5.2.9.1 Aplicação e Resultados - Árvore de decisão

Para este estudo foi elaborado um classificador por árvore de decisão implementado no *Matlab*, utilizando a função *fitctree*. Dado que o vetor de entrada definido com os valores dos parâmetros da marcha e a saída desejada com 10 classes, referente a cada voluntário. As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks (2016e).

Para avaliação do desempenho usamos a função *resubloss*, que define a precisão da classificação. Esta função mede a imprecisão de modelos de classificação (MATHWORKS, 2016e). Ao comparar o mesmo tipo de perda entre muitos modelos, a baixa perda indica um melhor modelo. A precisão obtida foi de 96%.

De acordo com a Figura 30 pode-se analisar a matriz confusão onde a diagonal em destaque, representa as amostras classificadas corretamente. Os demais valores, são os erros de classificação. O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pela árvore de decisão, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. As amostras do voluntário 10 foram 100% classificadas. O tempo decorrido para o treinamento foi de 0,997 segundos e a taxa de erro foi 0,039.

Figura 30: Resultados - Árvore de decisão

```

Processando banco de dados...
Treinando a rede...
Duração:      0.997116 segundos

TREINAMENTO CONCLUÍDO!!
Taxa de erro:0.039600

Precisão da classificação = 96.0%

Matriz Confusão:|

```

1	242	0	1	0	3	0	0	3	1	
2	3	238	0	1	0	2	0	0	4	
3	5	4	231	0	3	0	3	0	4	
4	3	0	1	242	1	0	0	0	3	
5	12	0	4	6	223	1	1	0	3	
6	2	0	2	4	0	241	0	0	1	
7	0	0	0	0	1	1	246	0	2	
8	0	0	0	0	0	0	0	250	0	
9	0	0	1	3	3	0	4	0	238	
10	0	0	0	0	0	0	0	0	250	
	1	2	3	4	5	6	7	8	9	10

Classes desejadas

Fonte: Matlab.

### 5.2.10 Deep Neural Network - DNN

*Deep Neural Networks* são desenvolvidas como uma rede neural artificial *feedforward*, com várias camadas ocultas entre as camadas de entrada e saída. DNNs podem modelar dados não lineares complexos, e suas camadas extras permitem a composição dos dados das camadas mais baixas, modelando dados complexos com menos unidades (BEN-GIO, 2009; MIAO et al., 2016).

Aprendizado profundo (*Deep learning*) é uma nova área de aprendizagem de máquina com base em conjuntos de algoritmos que tentam modelar abstrações de alto nível em dados usando várias camadas de processamento, normalmente utiliza-se redes neurais artificiais e a maioria dos modelos da arquitetura profunda são baseados em aprendizado não supervisionado (GRAUPE, 2016).

De acordo com Deng e Yu (2014) as características de uma *Deep Neural Networks* são:

- Utilizam uma cascata de múltiplas camadas de processamento

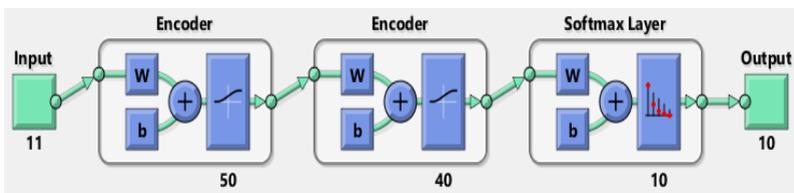
para a extração e transformação das características, onde cada camada sucessiva utiliza uma saída a partir da camada anterior como entrada, e os algoritmos podem ser por treinamento supervisionado ou não supervisionado;

- Baseiam-se na aprendizagem de múltiplos níveis de recursos, sendo que as características de nível superior são derivadas de recursos de nível inferior para formar uma representação hierárquica;
- Fazem parte do campo de aprendizagem de máquina mais amplo com representações de aprendizagem de dados;
- Aprendem vários níveis de representações que correspondem a diferentes níveis de abstração, que por fim formam uma hierarquia de conceitos.

#### 5.2.10.1 Aplicação e resultados - DNN

Nesta etapa foi implementado a estrutura *Deep Neural Network* para analisar os parâmetros da marcha humana utilizando o *Matlab*. As funções utilizadas são pré definidas pelo *software*, de acordo com a documentação MathWorks (2016e). A estrutura da rede exibida na Figura 31 é formada por três camadas, a primeira e a segunda utiliza *autoencoders* e a última uma camada *softmax*.

Figura 31: Arquitetura - DNN



Fonte: Matlab.

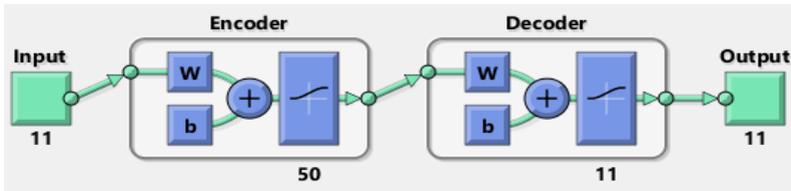
A matriz de entrada é formada por 2500 amostras dos 11 parâmetros, e a saída com 10 classes referente aos voluntários. Os dados de entrada foram divididos em grupos de treinamento, validação e teste.

Para treinar o primeiro *autoencoder*, utiliza-se a função *trainAutoencoder*, inicia-se os dados de entrada de treinamento sem utilizar os dados da saída desejada. Ele é uma rede neural artificial utilizada para a aprendizagem não supervisionada de códigos eficientes. Seu objetivo

é aprender uma representação (codificação) para um conjunto de dados, tipicamente para o efeito de redução de dimensionalidade (LIU et al., 2014).

Na Figura 32 é visto a arquitetura do 1º *autoencoder*, é determinada pelos dados de entrada com 11 parâmetros da marcha. Em seguida a primeira camada oculta com 50 neurônios, sendo esta o codificador, que mapeia os dados. E logo após, a segunda camada, o decodificador que reverte o mapeamento para reconstituir a entrada original. Sendo que a camada de saída possui o mesmo número de neurônios como a camada de entrada, com a finalidade de reconstruir as suas próprias entradas, sem prever suas saídas, utilizando o treinamento sem supervisão.

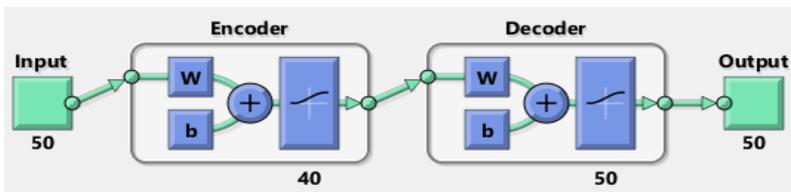
Figura 32: Arquitetura - 1º Autoencoder



Fonte: Matlab.

Para treinar o segundo *autoencoder*, utiliza-se a mesma estrutura anterior, como mostrado na Figura 33. Mas com a diferença de que os dados de entrada serão os dados gerados a partir do treinamento do *autoencoder* anterior e o número de neurônios na camada oculta são reduzidos para 40, de modo que o codificador do segundo *autoencoder* treine uma representação menor de dados de entrada.

Figura 33: Arquitetura - 2º Autoencoder



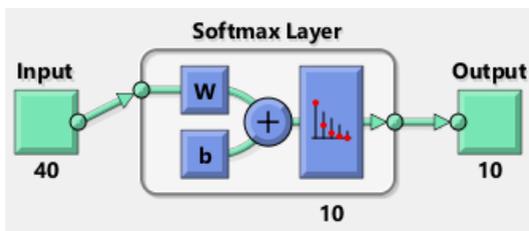
Fonte: Matlab.

Em seguida treina-se a última camada *softmax* utilizando a função *trainSoftmaxLayer*, ao contrário dos *autoencoders*, o treinamento desta

camada será de modo supervisionado, onde determinamos os dados da camada de saída com a matriz de saída desejada. Observa-se que os dados de entrada serão os dados gerados a partir do treinamento do segundo *autoencoder*.

Mostra-se na Figura 34, que em redes neurais a função *softmax* é implementada na última camada para classificação, na qual essa rede treina sob regime de perda, resultando em uma variante não linear de regressão logística multinomial. Segundo Bishop (2006) função *softmax* é uma generalização da função logística, na qual é usada para representar uma distribuição categórica, ou seja, uma distribuição de probabilidade sobre diferentes resultados possíveis. Após efetuado os

Figura 34: Arquitetura - *Softmax*



Fonte: Matlab.

treinamentos de cada camada, utiliza-se a função *stack* para formar a rede de profundidade DNN vista da Figura 31. Como ajuste fino, podemos melhorar o desempenho da rede através da realização da retropropagação, utilizando a função *train* em toda rede de multicamada. É feito a reciclagem dos dados de treinamento de forma supervisionada, utilizando o grupo de testes.

Para avaliação do desempenho usamos a precisão da classificação pela Equação (5.1), assim adquire-se o valor da DNN em 99,8%. De acordo com a Figura 35, pode-se analisar a matriz confusão onde a diagonal em destaque, na cor verde, representa as amostras classificadas corretamente. Os demais valores, na cor em vermelho, são os erros de classificação.

O eixo vertical na matriz confusão corresponde ao resultado da classificação das classes determinado pelo DNN, após o treinamento. O eixo horizontal representa as classes da saída desejada, utilizada para o treinamento. As amostras dos voluntários 1, 3, 4 e 9 apresentaram erros na classificação. O tempo decorrido para o treinamento foi de 99,505 segundos e a taxa de erro foi 0,0016.

Figura 35: Resultados - DNN

Finaliza treinamento da Deep Network!

Tempo de treinamento total:

Duração: 99.504639 segundos

Taxa de erro:0.001600 |

Precisão da classificação: 99.8% amostras

1	249	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99.6%
	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%
2	0	250	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99.6%
	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%
3	1	0	249	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99.6%
	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%
4	0	0	0	249	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
5	0	0	0	0	250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
6	0	0	0	0	0	250	0	0	0	0	0	0	0	0	0	0	0	0	0	99.6%
	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%
7	0	0	0	0	0	0	250	0	0	0	0	0	0	0	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
8	0	0	0	0	0	0	0	250	0	0	0	0	0	0	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
9	0	0	0	0	0	0	0	0	0	249	0	0	0	0	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
10	0	0	0	0	0	0	0	0	0	0	250	0	0	0	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	99.6%	100%	99.6%	99.6%	100%	100%	100%	100%	100%	99.6%	100%	99.6%	100%	99.6%	100%	99.6%	100%	99.6%	100%	99.3%
	0.4%	0.0%	0.4%	0.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%	0.0%	0.0%	0.0%	0.4%	0.0%	0.4%	0.0%	0.4%	0.0%	0.2%
	1	2	3	4	5	6	7	8	9	10										
											Saída desejada									

Fonte: Matlab.

### 5.3 CONCLUSÃO

Neste Capítulo exibimos os resultados obtidos de 10 diferentes tipos de mecanismos para classificar os parâmetros da marcha humana. A classificação é efetuada em 10 pessoas sendo definidas por 10 classes, na qual este sistema é determinado por múltiplas entradas e múltiplas saídas, e se for necessário realizar a identificação de uma pessoa, isto será um caso particular das abordagens utilizadas neste capítulo.

São utilizados o *Neural Network Toolbox* e *Statistics and Machine Learning Toolbox* para aplicação dos classificadores implementados no *software Matlab*. Em cada um foi apresentado seus métodos para criar a estrutura, processar o treinamento do algoritmo e expor a precisão, seguindo as documentações MathWorks (2016d, 2016e, 2016a, 2016b, 2016c).

Na Tabela 1 podemos analisar a precisão da classificação e o tempo de treinamento de cada um deles. O tempo de computação é uma medida importante para avaliar diferentes modelos de classificação. O cálculo foi implementado em um *notebook* com processador Intel(R) Core(TM) i7-4510U 2,60 GHz, 8 GB de RAM e o sistema operacional Windows 10 Pro de 64 bits. São adicionados dois comandos no código, *tic* e *toc*, para efetuar a leitura do tempo decorrido durante

Tabela 1: Resultados dos classificadores.

<b>Classificadores</b>	<b>Precisão (%)</b>	<b>Tempo(s)</b>
SVM	96,58	5,255
MLP	97,60	0,961
RF	97,40	1,923
PNN	100,00	1,147
NB	97,60	2,069
LVQ	36,00	17,129
kNN	99,20	0,626
k-means	13,20	0,434
Árvore de decisão	96,00	0,997
DNN	99,80	99,505

Fonte: Elaboração própria.

o treinamento.

Por comparação o classificador com melhor precisão foi o *Probabilistic Neural Network* (PNN), seguido do *Deep Neural Networks* (DNN) e o *k-Nearest Neighbors* (kNN), que apresentou o melhor tempo de treinamento. Os classificadores que tiveram o pior desempenho foram *k-means* e *Learning Vector Quantization Neural Network* (LVQ).



## 6 CONCLUSÃO

A análise da marcha humana tem aplicações em diferentes campos de estudo, como por exemplo: na medicina, com propósito de detecção de doenças relacionadas ao movimento; Em construções de jogos que utilizam as imagens para reprodução do inanimado em personagens virtuais; Nos esportes para o diagnóstico do rendimento do corpo baseado nos movimentos que realizam. Este trabalho traz como contribuição a iniciação dos estudos da marcha humana voltados à novos desenvolvimentos no campo biométrico. Esta tecnologia surge como uma ferramenta de apoio indispensável para a obtenção de resultados instantâneos e precisos, dado que a segurança se torna uma das prioridades nos tempos atuais.

No Capítulo 2, retratou-se a história do movimento humano, e quais foram as primeiras ideias sobre a área de pesquisa. A partir da colaboração de importantes estudiosos de diferentes épocas, com a formulação de como o corpo humano executa tais atividades e dinâmicas.

Assim surge um meio de compreender as multiplicidades dos movimentos executados pelo ser humano, e as forças internas e externas atuantes e seus limites, envolvendo o conhecimento da anatomia e fisiologia articular. Ao aplicar os mesmos em uma ciência multidisciplinar, denominadas por cinesiologia e biomecânica. Tornando o estudo destas duas disciplinas a fonte de desenvolvimento e aplicação da teoria em sistemas biométricos, para a análise do movimento humano.

O Capítulo 3 aborda os conceitos chave para a introdução da investigação do estudo da biomecânica humana, na qual expõe o comportamento dinâmico do sistema através de equações do movimento segundo a literatura. Assim, foram apresentados os termos mais comuns, as metodologias mais frequentes, e as variáveis requeridas pela teoria para avaliação biomecânica da marcha. Ao mostrar os métodos para calcular os parâmetros cinéticos e cinemáticos do corpo que influenciam nesta análise.

Para este projeto foram utilizados apenas os cálculos dos parâmetros da parte inferior do corpo, sendo eles: os quadris, joelhos, tornozelos e pés, para a aplicação da extração dos valores cinemáticos angulares. Os dados cinemáticos lineares, tais como, a velocidade e aceleração geram influência nos cálculos dos dados da cinética linear. Portanto são empregados diretamente na aquisição da força de reação do solo. A força atrito não foi utilizada neste estudo visto que segundo Hamill e Knutzen (2014) se necessita de equipamentos sofisticados e

mais precisos para sua estimação.

A primeira aplicação prática das técnicas da biomecânica é abordada no Capítulo 4. Nesta etapa o presente trabalho retratou um método automatizado de calcular os parâmetros da marcha humana, ao desenvolver um aplicativo, baseado na utilização da câmera *Kinect* que forneceu uma ótima precisão na segmentação temporal e no desempenho para computar os dados.

Para validar os resultados efetuou-se múltiplos testes em pessoas, extraindo variáveis relacionadas ao movimento do caminhar humano com grande aproximação aos valores apresentados na teoria da biomecânica. Visto que foi realizado apenas a análise dos membros inferiores do corpo, o *Kinect* possui uma alta variabilidade para executar o rastreamento em todo o esqueleto, podendo em futuras aplicações aprimorar os resultados com mais parâmetros. Em função da implementação, dispomos de uma ferramenta confiável, um sistema de medição de baixo custo, que consegue determinar as características da análise da marcha humana para a aplicação em reconhecimento de padrões.

Após realizada a extração dos parâmetros da marcha, por critério de classificação utilizamos mecanismos de reconhecimento de padrões para avaliar se estes dados são coerentes em função da identificação automática de pessoas a fim de utilizá-los em aplicações de sistemas biométricos.

O Capítulo 5 mostrou a topologia de cada classificador para criar sua estrutura, processar o treinamento do algoritmo e expor a precisão. Foram obtidos resultados de 10 diferentes tipos de mecanismos para classificar os parâmetros da marcha humana em sistemas de múltiplas classes, implementados no *software Matlab* e utilizando as ferramentas *Neural Network Toolbox Statistics and Machine Learning Toolbox*.

Os resultados mostraram que os classificadores PNN, DNN e kNN, foram os algoritmos que alcançaram melhor desempenho em relação a taxa de precisão, na casa dos 99%. Os algoritmos SVM, MLP, NB, RF e árvore de decisão, também tiveram um bom desempenho na faixa de 97%. O classificador kNN foi o que obteve menor tempo de processamento do treinamento em 0,626 segundos. Entre os dez apenas dois algoritmos, LVQ e k-means, apresentaram a pior precisão, em 36% e 13,2% respectivamente.

A partir dos resultados obtidos pela análise dos classificadores, é visto que as metodologias e estruturas de reconhecimento de padrões, garantem o uso dos dados extraídos dos parâmetros da marcha para aplicações do reconhecimento de pessoas. Desta maneira, com os obje-

tivos alcançados pelos resultados dos classificadores, pode-se perceber que é viável o projeto para extração e classificação dos parâmetros da marcha humana. Pois este trabalho abre diversas lacunas para trabalhos futuros voltados ao desenvolvimento de um software de segurança que consiga identificar as pessoas pelo seu modo de andar.

## 6.1 TRABALHOS FUTUROS

Visando aprofundar o estudo dos dados da marcha humana em função de aprimorar a pesquisa apresentada, vamos mostrar de forma pontual, os trabalhos futuros que poderão dar continuidade à investigação:

- Realizar testes da atual pesquisa em computadores com outros processadores, para observar e analisar o desempenho da classificação, em função do tempo decorrido.
- Otimizar o aplicativo desenvolvido utilizando o sensor *Kinect*, com a aplicação de todos os parâmetros do corpo humano.
- Implementar os classificadores juntamente ao aplicativo, a fim de poder identificar um indivíduo em tempo real.
- Definir os requisitos de sistema mínimos para a utilização do software, em função do comportamento, rendimento e transferências de dados.
- Desenvolver um *software* que reconheça pessoas a partir da utilização de câmeras convencionais, para que este produto possa ser utilizado em variados locais que disponham de qualquer câmera.
- Criar uma aplicação de celular (*mobile app*), para trabalhar em conjunto com o software de segurança principal, em função de dar praticidade e acessibilidade aos usuários.
- Devido a necessidade de grandes processamentos, é visado desenvolver um aplicativo que utilize uma infraestrutura capaz de armazenar e recuperar quantidades gigantescas de dados em nuvem.
- Efetuar a análise de outras topologias de classificadores que possam lidar com dados lineares e não lineares, com uma capacidade preditiva, como o *Gradient Tree Boosting*.



## REFERÊNCIAS

- ABDULLAH, O.; ABDEL-QADER, I. A pnn-jensen-bregman divergence symmetrization for a wlan indoor positioning system. In: **IEEE. Electro Information Technology (EIT), 2016 IEEE International Conference on.** [S.l.], 2016. p. 0362–0367.
- ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. **The American Statistician**, Taylor & Francis, v. 46, n. 3, p. 175–185, 1992.
- AMADIO, A. et al. Introdução à análise do movimento humano - descrição e aplicação dos métodos biomecânicos de medição. **Revista Brasileira Fisioterapia**, v. 3, n. 2, p. 41–54, 1999.
- AMEZCUA, J.; MELIN, P.; CASTILLO, O. A neural network with a learning vector quantization algorithm for multiclass classification using a modular approach. In: **Recent Developments and New Direction in Soft-Computing Foundations and Applications.** [S.l.]: Springer, 2016. p. 171–184.
- ANAVA, O.; LEVY, K. k-nearest neighbors: From global to local. In: **Advances In Neural Information Processing Systems.** [S.l.: s.n.], 2016. p. 4916–4924.
- ANDERSON, F. C. **A dynamic optimization solution for a complete cycle of normal gait.** [S.l.: s.n.], 1999.
- ARAÚJO, A. G. N.; ANDRADE, L. M.; BARROS, R. M. L. de. Sistema para análise cinemática da marcha humana baseado em videogrametria. **Fisioterapia e Pesquisa**, v. 11, n. 1, p. 3–10, 2005.
- ARAUJO, R. M.; GRAÑA, G.; ANDERSSON, V. Towards skeleton biometric identification using the microsoft kinect sensor. In: **ACM. Proceedings of the 28th Annual ACM Symposium on Applied Computing.** [S.l.], 2013. p. 21–26.
- BALL, A. et al. Unsupervised clustering of people from skeleton data. In: **ACM. Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction.** [S.l.], 2012. p. 225–226.

BENGIO, Y. Learning deep architectures for ai. **Foundations and trends® in Machine Learning**, Now Publishers Inc., v. 2, n. 1, p. 1–127, 2009.

BIEHL, M.; GHOSH, A.; HAMMER, B. Learning vector quantization: The dynamics of winner-takes-all algorithms. **Neurocomputing**, Elsevier, v. 69, n. 7, p. 660–670, 2006.

BISHOP, C. M. Pattern recognition. **Machine Learning**, v. 128, 2006.

BOBBERT, M. F.; SCHAMHARDT, H. C.; NIGG, B. M. Calculation of vertical ground reaction force estimates during running from positional data. **Journal of biomechanics**, Elsevier, v. 24, n. 12, p. 1095–1105, 1991.

BOYD, J. E.; LITTLE, J. J. Biometric gait recognition. Springer, p. 19–42, 2005.

BREIMAN, L. Bagging predictors. **Machine learning**, Springer, v. 24, n. 2, p. 123–140, 1996.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BURGES, C. J. A tutorial on support vector machines for pattern recognition. **Data mining and knowledge discovery**, Springer, v. 2, n. 2, p. 121–167, 1998.

CHANDLER, R. et al. **Investigation of inertial properties of the human body**. [S.l.], 1975.

CUJI, D. et al. Gait cycle analysis through Kinect and stationary wavelet transform. **IEEE**, 2013.

DEMAILLY, J.-P. **Analyse numérique et équations différentielles**. [S.l.]: EDP Sciences, 2012.

DENG, L.; YU, D. Deep learning. **Signal Processing**, Citeseer, v. 7, p. 3–4, 2014.

ELSEVIER. **SciVerse Scopus**. 2016. Disponível em: <<https://www.scopus.com/>>.

ENOKA, R. M. **Bases neuromecânica da cinesiologia**. [S.l.]: São Paulo: Manole, 2000.

FU, K. **Sequential methods in pattern recognition and machine learning**. [S.l.]: Academic press, 1968.

GARG, R. et al. A bootstrap machine learning approach to identify rare disease patients from electronic health records. **arXiv preprint arXiv:1609.01586**, 2016.

GEERSE, D. J.; COOLEN, B. H.; ROERDINK, M. Kinematic validation of a multi - Kinect v2 instrumented 10 - meter walkway for quantitative gait assessments. **Plos one**, Public library of science, v. 10, n. 10, p. e0139913, 2015.

GIANARIA, E. et al. Human classification using gait features. In: SPRINGER. **International Workshop on Biometric Authentication**. [S.l.], 2014. p. 16–27.

GOLDBERGER, J. et al. Neighbourhood components analysis. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2004. p. 513–520.

GONG, T. et al. A bayes theory-based modeling algorithm to end-to-end network traffic. In: EDP SCIENCES. **ITM Web of Conferences**. [S.l.], 2016. v. 7, p. 09024.

GRAUPE, D. **Deep Learning Neural Networks: Design and Case Studies**. [S.l.]: World Scientific Publishing Co Inc, 2016.

GROSSBERG, S. Contour enhancement, short term memory, and constancies in reverberating neural networks. In: **Studies of mind and brain**. [S.l.]: Springer, 1982. p. 332–378.

HAMILL, J.; KNUTZEN, K. M. **Biomechanical basis of human movement**. [S.l.]: Lippincott Williams & Wilkins, 2014.

HARO, D. M. Laboratorio de análisis de marcha y movimiento. **Revista Médica Clínica Las Condes**, Elsevier, v. 25, n. 2, p. 237–247, 2014.

HAYKIN, S. S. et al. **Neural networks and learning machines**. [S.l.]: Pearson Upper Saddle River, NJ, USA:, 2009.

HEARST, M. A. et al. Support vector machines. **IEEE Intelligent Systems and their Applications**, IEEE, v. 13, n. 4, p. 18–28, 1998.

HEINEN, M. R.; OSÓRIO, F. S. Biometria comportamental: Pesquisa e desenvolvimento de um sistema de autenticação de usuários utilizando assinaturas manuscritas. **INFOCOMP Journal of Computer Science**, v. 3, n. 2, p. 32–37, 2004.

JAIN, A. K.; FLYNN, P.; ROSS, A. A. **Handbook of Biometrics**. [S.l.]: Springer, 2008.

JAMES, G. et al. **An introduction to statistical learning**. [S.l.]: Springer, 2013.

JASKOWIAK, P. A.; CAMPELLO, R. Comparing correlation coefficients as dissimilarity measures for cancer classification in gene expression data. In: **Proceedings of the Brazilian Symposium on Bioinformatics**. [S.l.: s.n.], 2011. p. 1–8.

JØSANG, A. Generalising bayes' theorem in subjective logic. In: **International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)**. [S.l.: s.n.], 2016.

KAMINKA, G. et al. Repetitive branch-and-bound using constraint programming for constrained minimum sum-of-squares clustering. In: IOS PRESS. **ECAI 2016: 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands-Including Prestigious Applications of Artificial Intelligence (PAIS 2016)**. [S.l.], 2016. v. 285, p. 462.

KAUR, S. et al. Gene selection for tumor classification using resilient backpropagation neural network. In: IEEE. **Advances in Computing, Communication, & Automation (ICACCA)(Fall), International Conference on**. [S.l.], 2016. p. 1–5.

KEWATKAR, S.; KATHLE, S. Human gait recognition by opencv. **Journal Contents for Volume**, v. 3, n. 1, 2012.

KNUTSSON, E. An analysis of parkinsonian gait. **Brain**, Oxford Univ Press, v. 95, n. 3, p. 475–486, 1972.

KOHONEN, T. **Learning vector quantization for pattern recognition**. [S.l.], 1986.

KOHONEN, T. Learning vector quantization. In: **Self-Organizing Maps**. [S.l.]: Springer, 1995. p. 175–189.

KOHONEN, T.; SOMERVUO, P. Self-organizing maps of symbol strings. **Neurocomputing**, Elsevier, v. 21, n. 1, p. 19–30, 1998.

KUMAR, R.; SINGH, S. Character recognition from google street view images. 2015.

LANGLEY, P.; IBA, W.; THOMPSON, K. An analysis of bayesian classifiers. In: **Aaai**. [S.l.: s.n.], 1992. v. 90, p. 223–228.

LAPKO, A.; LAPKO, V. Nonparametric estimate of a parzen-type probability density with an implicitly specified form of the kernel. **Measurement Techniques**, Springer, v. 59, n. 6, p. 571–576, 2016.

LEHMKUHL, L. D.; SMITH, L. K.; WEISS, E. Cinesiologia clínica de brunstrom. **Clinical Kinesiology from Brunstrom**, Manole São Paulo, 1989.

LENGYEL, E. **Mathematics for 3D game programming and computer graphics**. [S.l.]: Cengage learning, 2005.

LEWIS, D. D. Naive (bayes) at forty: The independence assumption in information retrieval. In: SPRINGER. **European conference on machine learning**. [S.l.], 1998. p. 4–15.

LIOU, C.-Y. et al. Autoencoder for words. **Neurocomputing**, Elsevier, v. 139, p. 84–96, 2014.

LORENA, A. C. Investigação de estratégias para a geração de máquinas de vetores de suporte multiclases. **Instituto de Ciências Matemáticas e de Computação (ICMC), Ciências de Computação e Matemática Computacional-Universidade de São Paulo**, 2006.

MACKAY, D. An example inference task: clustering. **Information theory, inference and learning algorithms**, Cambridge University Press Cambridge, p. 284–292, 2003.

MARTÍNEZ, F.; GÓMEZ, F.; ROMERO, E. Análise do vídeo para o estimation do movimento humano: Uma revisão. **Revista Med**, Universidad Militar Nueva Granada, v. 17, n. 1, p. 95–106, 2009.

MATHWORKS. **Neural Network Toolbox User's Guide**. 2016. Disponível em:

<[https://www.mathworks.com/help/pdf\\_doc/nnet/nnet\\_ug.pdf](https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf)>.

MATHWORKS. **Probabilistic Neural Networks**. 2016. Disponível em: <<https://www.mathworks.com/help/nnet/ug/probabilistic-neural-networks.html>>.

MATHWORKS. **Radial Basis Functions for Model Building**. 2016. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/nnet/nnet\\_ug.pdf](https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf)>.

MATHWORKS. **Statistics and Machine Learning Toolbox Release Notes**. 2016. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/stats/rn.pdf](https://www.mathworks.com/help/pdf_doc/stats/rn.pdf)>.

MATHWORKS. **Statistics and Machine Learning Toolbox User's Guide**. 2016. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/stats/stats.pdf](https://www.mathworks.com/help/pdf_doc/stats/stats.pdf)>.

MEYER, D.; WIEN, F. T. Support vector machines. **The Interface to libsvm in package e1071**, 2015.

MIAO, H. et al. Modelhub: Towards unified data and lifecycle management for deep learning. **arXiv preprint arXiv:1611.06224**, 2016.

MIR, N. M. et al. An experimental evaluation of bayesian classifiers applied to intrusion detection. **Indian Journal of Science and Technology**, v. 9, n. 12, 2016.

MURRAY, M. P. Gait as a total pattern of movement: Including a bibliography on gait. **American Journal of Physical Medicine & Rehabilitation**, LWW, v. 46, n. 1, p. 290–333, 1967.

NAM, D. H. Comparative studies in methods of feature recognition with machine learning for affective computing: A survey. In: THE STEERING COMMITTEE OF THE WORLD CONGRESS IN COMPUTER SCIENCE, COMPUTER ENGINEERING AND APPLIED COMPUTING (WORLDCOMP). **Proceedings on the International Conference on Artificial Intelligence (ICAI)**. [S.l.], 2016. p. 122.

NEBEL, D.; VILLMANN, T. Optimization of statistical evaluation measures for classification by median learning vector quantization. In: **Advances in Self-Organizing Maps and Learning Vector Quantization**. [S.l.]: Springer, 2016. p. 281–291.

- NIETO-HIDALGO, M. et al. A vision based proposal for classification of normal and abnormal gait using rgb camera. **Journal of biomedical informatics**, Elsevier, v. 63, p. 82–89, 2016.
- OATIS, C. A. The mechanics and pathomechanics of human movement. Williams and Wilkins, Lippincott, 2003.
- OLIVEIRA, A. L. de et al. **Licenciatura em Educação Física: Cinesiologia**. [S.l.]: UFGP, 2011.
- OLIVER, J. J.; HAND, D. J. On pruning and averaging decision trees. In: **Machine Learning: Proceedings of the Twelfth International Conference**. [S.l.: s.n.], 2016. p. 430–437.
- ORHAN, U.; HEKIM, M.; OZER, M. Eeg signals classification using the k-means clustering and a multilayer perceptron neural network model. **Expert Systems with Applications**, Elsevier, v. 38, n. 10, p. 13475–13481, 2011.
- ORTIZ, C. Classificação da força de reação do solo durante a marcha de sujeitos com fratura de membro inferior (dissertação). **Rio de janeiro: Universidade Federal do Rio de Janeiro**, 2010.
- OZYILDIRIM, B. M.; AVCI, M. One pass learning for generalized classifier neural network. **Neural Networks**, Elsevier, v. 73, p. 70–76, 2016.
- PÉREZ, J. et al. Optimization of the k-means algorithm for the solution of high dimensional instances. In: AIP PUBLISHING. **International Conference of Numerical Analysis and Applied Mathematics 2015(ICNAAM 2015)**. [S.l.], 2016. v. 1738, n. 1, p. 310002.
- PFEIFER, N. Experimental probabilistic pragmatics beyond bayes' theorem. **Sprachwissenschaft**, v. 35, n. 1, p. 89–96, 2016.
- PRATHIBA, R. et al. Multiple output radial basis function neural network with reduced input features for on-line estimation of available transfer capability. **Journal of Control Engineering and Applied Informatics**, v. 18, n. 1, p. 95–106, 2016.
- PREIS, J. et al. Gait recognition with kinect. In: NEW CASTLE, UK. **1st international workshop on kinect in pervasive computing**. [S.l.], 2012. p. P1–P4.

- PRESS, W. H. et al. Section 16.5. support vector machines. **Numerical Recipes: The Art of Scientific Computing**, 2007.
- PW, R. On the choice of smoothing parameters for parzen estimators of probability density functions. **IEEE Transactions on Computers**, 1976.
- RAO, D. R.; MURTHY, J. Comparative analysis: Heart diagnosis classification using bp-lvq neural network models for analog and digital data. **Global Journal of Computer Science and Technology**, v. 16, n. 5, 2016.
- RESEARCH, M. **Programming with the Kinect for Windows SDK**. 2016. Disponível em: <<https://goo.gl/AhY3HF>>.
- ROBERTSON, G. et al. **Research methods in biomechanics**. [S.l.]: Human Kinetics, 2013.
- ROKACH, L.; MAIMON, O. **Data mining with decision trees: theory and applications**. [S.l.]: World scientific, 2014.
- ROSENHAHN, B.; KLETTE, R.; METAXAS, D. **Human Motion: Computational imaging and vision**. [S.l.]: Springer, 2008.
- ROSS, A.; JAIN, A. K.; PRABHAKAR, S. An introduction to biometric recognition. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 14, n. 1, p. 4-20, 2004.
- RUSSELL, S. J. et al. **Artificial intelligence: a modern approach**. [S.l.]: Prentice hall Upper Saddle River, 2003.
- SAHA, S. et al. Human skeleton matching for e-learning of dance using a probabilistic neural network. In: IEEE. **Neural Networks (IJCNN), 2016 International Joint Conference on**. [S.l.], 2016. p. 1754-1761.
- SALATA, J. Implementation of competitive learning networks for weka. **ICT Research Blog**, 2011.
- SALLES, T. et al. Broof: Exploiting out-of-bag errors, boosting and random forests for effective automated classification. In: ACM. **Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval**. [S.l.], 2015. p. 353-362.

SHAHADAT, N.; PAL, B. An empirical analysis of attribute skewness over class imbalance on probabilistic neural network and naive bayes classifier. In: **IEEE. 2015 International Conference on Computer and Information Engineering (ICCIIE)**. [S.l.], 2015. p. 150–153.

SHMILOVICI, A. Support vector machines. In: **Data Mining and Knowledge Discovery Handbook**. [S.l.]: Springer, 2005. p. 257–276.

SILVA, V. R. **Cinesiologia e biomecânica**. [S.l.]: Rio de Janeiro: SESES, 2015.

SINGH, M.; NAGAR, H.; SANT, A. K-mean and em clustering algorithm using attendance performance improvement primary school student. 2016.

SMISEK, J.; JANCOSSEK, M.; PAJDLA, T. 3d with kinect. In: **Consumer Depth Cameras for Computer Vision**. [S.l.]: Springer, 2013. p. 3–25.

SOUSA, A. S. P. de. **Análise da marcha baseada numa correlação multifactorial**. Dissertação (Mestrado em Engenharia Biomédica) — Faculdade de Engenharia da Universidade do Porto, 2008.

SPECHT, D. F. Probabilistic neural networks. **Neural networks**, Elsevier, v. 3, n. 1, p. 109–118, 1990.

SULINO, R. M. Avaliação automatizada de uma habilidade motora fundamental com o Kinect. Universidade Estadual Paulista (UNESP), 2014.

SUTHAHARAN, S. Support vector machine. In: **Machine Learning Models and Algorithms for Big Data Classification**. [S.l.]: Springer, 2016. p. 207–235.

SUYKENS, J. A.; VANDEWALLE, J. Least squares support vector machine classifiers. **Neural processing letters**, Springer, v. 9, n. 3, p. 293–300, 1999.

TAKEDA, R. Y. et al. Identificação biométrica de indivíduos baseada em parâmetros dinâmicos extraídos da marcha humana. 2014.

TECH, V. **Biometric Services in Dubai from VRS Tech for businesses and organizations enable you to keep your facility safe and monitored all the time.** 2016. Disponível em: <<http://www.vrstech.com/biometric-systems-security-services.html>>.

TEMURTAS, F. A comparative study on thyroid disease diagnosis using neural networks. **Expert Systems with Applications**, Elsevier, v. 36, n. 1, p. 944–949, 2009.

VATTANI, A. K-means requires exponentially many iterations even in the plane. **Discrete & Computational Geometry**, Springer, v. 45, n. 4, p. 596–616, 2011.

VILLMANN, T. et al. Learning vector quantization classifiers for roc-optimization. **Computational Statistics**, Springer, p. 1–22, 2016.

WANG, K.-Y.; SHUN, H.-Y. Applying back propagation neural networks in the prediction of management associate work retention for small and medium enterprises. 2016.

WEINBERGER, K. Q.; BLITZER, J.; SAUL, L. K. Distance metric learning for large margin nearest neighbor classification. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2005. p. 1473–1480.

WEINBERGER, K. Q.; SAUL, L. K. Distance metric learning for large margin nearest neighbor classification. **Journal of Machine Learning Research**, v. 10, n. Feb, p. 207–244, 2009.

WHITTLE, M. W. **Gait analysis: an introduction**. [S.l.]: Butterworth-Heinemann, 2014.

WU, X.; YANG, J.; WANG, S. Tea category identification based on optimal wavelet entropy and weighted k-nearest neighbors algorithm. **Multimedia Tools and Applications**, Springer, p. 1–15, 2016.

XU, S. Bayesian naïve bayes classifiers to text classification. **Journal of Information Science**, SAGE Publications, p. 0165551516677946, 2016.

ZERPA, C. et al. The use of microsoft Kinect for human movement analysis. **International journal of sports science**, Scientific & Academic Publishing, v. 5, n. 4, p. 120–127, 2015.

ZHANG, H. The optimality of naive bayes. **AA**, v. 1, n. 2, p. 3, 2004.

ZHANG, Y. et al. Comparison of machine learning methods for stationary wavelet entropy-based multiple sclerosis detection: decision tree, k-nearest neighbors, and support vector machine. **Simulation**, SAGE Publications, v. 92, n. 9, p. 861–871, 2016.