

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS

VICTOR HUGO MORAIS DIAS

**ANÁLISE COMPARATIVA ENTRE O PADRÃO DE MARCHA EM ADULTOS E
IDOSOS UTILIZANDO TÉCNICAS DE *MACHINE LEARNING***

João Monlevade,
2022

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS

VICTOR HUGO MORAIS DIAS

**ANÁLISE COMPARATIVA ENTRE O PADRÃO DE MARCHA EM ADULTOS E
IDOSOS UTILIZANDO TÉCNICAS DE *MACHINE LEARNING***

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Orientador: Talles Henrique de Medeiros

João Monlevade,
2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

D541a Dias, Victor Hugo Moraes.

Análise comparativa entre o padrão de marcha em adultos e idosos utilizando técnicas de machine learning. [manuscrito] / Victor Hugo Moraes Dias. - 2022.

56 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Talles Henrique de Medeiros.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de Computação .

1. Aprendizado do computador. 2. Biomecânica. 3. Processamento de sinais. 4. Redes neurais (Computação). I. Medeiros, Talles Henrique de. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.5/.8

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS



FOLHA DE APROVAÇÃO

Victor Hugo Moraes Dias

ANÁLISE COMPARATIVA ENTRE O PADRÃO DE MARCHA EM ADULTOS E IDOSOS UTILIZANDO TÉCNICAS DE MACHINE LEARNING

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação

Aprovada em 07 de março de 2022

Membros da banca

Dr. - Talles Henrique de Medeiros - Orientador (Universidade Federal de Ouro Preto)
Dr. - Luiz Carlos Bambirra Torres - (Universidade Federal de Ouro Preto)
Dra. - Janniele Aparecida Soares Araújo - (Universidade Federal de Ouro Preto)

Prof. Talles Henrique de Medeiros, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 10/03/2022



MAGISTERIO SUPERIOR, em 10/03/2022, às 09:59, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0290493** e o código CRC **C7CE0FFE**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.002687/2022-29

SEI nº 0290493

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: (31)3808-0819 - www.ufop.br

AGRADECIMENTOS

Primeiramente gostaria de agradecer aos meus pais, Alessandro e Viviane, pelo suporte e apoio incondicional que recebi, por me incentivarem nos momentos difíceis e compreenderem minha ausência enquanto eu me dedicava à minha graduação.

Aos amigos que fiz ao longo deste período, com quem convivi intensamente durante os últimos anos, pelo companheirismo e pela troca de experiências que me permitiram crescer não só como pessoa, mas também como profissional. Em especial, aos amigos que fiz no *Cheerleading*, pela motivação e por estarem ao meu lado em todos os momentos.

A todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa, enriquecendo o meu processo de aprendizado. Em especial à minha namorada, Sara, por todo o apoio e ajuda recebidos nessa reta final de graduação.

"Everybody is a genius. But if you judge a fish by its ability to climb a tree, it will spend its whole life believing that it is stupid"

(Albert Einstein)

RESUMO

O objetivo deste trabalho é gerar um modelo capaz de discriminar um indivíduo e seu padrão de marcha com base em dados cinemáticos. A base de dados utilizada para estudo consiste em dados coletados por marcadores em 42 indivíduos, sendo eles 24 jovens adultos (idades 27.6 ± 4.4 anos) e 18 adultos mais velhos (idades 62.7 ± 8.0 anos). Para a realização do estudo foram utilizadas técnicas de processamento de sinais, de análise de dados e de *Machine Learning*. Ao final dos testes, o algoritmo que obteve melhores resultados em sua classificação foi o *Support Vector Machine*, com uma acurácia média de 85% para os marcadores localizados no quadril.

Palavras-chave: Biomecânica. Aprendizagem de Máquina. Máquina de Vetor de Suporte. Rede Neural Artificial.

ABSTRACT

The objective of this work is to obtain a model capable of discriminating an individual and his gait pattern based on kinetic data. The database used for the study consists of data from markers in 42 individuals, being 24 young adults (ages 7.6 ± 4.4 years) and 18 older adults (ages 62.7 ± 8.0 years). To carry out the study, signal processing, data analysis and Machine Learning methods were used. At the end of the tests, the one that obtained the best results in its Support Vector Machine classification, with an average accuracy of 85% with an average accuracy of 85% for the markers located in the hip.

Key-words: Biomechanics. *Machine Learning. Support Vector Machine. Artificial Neural Network.*

LISTA DE ILUSTRAÇÕES

FIGURAS

Figura 1 -	Tipos de aprendizado de máquina	17
Figura 2 -	Modelo computacional de um neurônio	20
Figura 3 -	Direção do vetor x	21
Figura 4 -	Vetores x , y e o ângulo entre eles	22
Figura 5 -	Representação gráfica da relação entre os ângulos θ , β e α	22
Figura 6 -	Dados bidimensionais separados por uma linha	23
Figura 7 -	Dados bidimensionais linearmente inseparáveis	24
Figura 8 -	Solução ótima utilizando o algoritmo SVM	25
Figura 9 -	Exemplo de erro de classificação	26
Figura 10 -	SVM robusto para outliers	26
Figura 11 -	Base de dados não linear	27
Figura 12 -	Base de dados não linear normalizada	27
Figura 13 -	Representação conceitual dos elementos da biomecânica	29
Figura 14 -	Sistema de câmeras	30
Figura 15 -	Protocolo de conjunto de marcadores	30
Figura 16 -	Código utilizado para preparar a base de dados para os algoritmos de ML	39
Figura 17 -	Código utilizado para treinar o classificador ANN	40
Figura 18 -	Código utilizado para gerar a matriz de confusão do classificador ANN	42
Figura 19 -	Código utilizado para treinar o classificador SVM	43
Figura 20 -	Código utilizado para gerar a matriz de confusão do classificador SVM	46

LISTA DE ILUSTRAÇÕES

GRÁFICOS

Gráfico 1 -	Aplicação da Transformada de Fourier no marcador R.ASIS na dimensão X do arquivo WBDS20walkT05mkr.txt	35
Gráfico 2 -	Aplicação da Transformada de Fourier no marcador R.ASIS na dimensão Y do arquivo WBDS20walkT05mkr.txt	36
Gráfico 3 -	Marcador R.ASIS na dimensão Z do arquivo WBDS20walkT05mkr.txt	36
Gráfico 4 -	Picos da Transformada de Fourier do marcador R.Ankle na dimensão X do arquivo WBDS20walkT05mkr.txt destacados	37
Gráfico 5 -	Picos da Transformada de Fourier do marcador R.Ankle na dimensão Y do arquivo WBDS20walkT05mkr.txt destacados	37
Gráfico 6 -	Picos da Transformada de Fourier do marcador R.Ankle na dimensão Z do arquivo WBDS20walkT05mkr.txt destacados	38

LISTA DE TABELAS

Tabela 1 -	Teste para o melhor valor de K para o algoritmo ANN considerando todos os marcadores	40
Tabela 2 -	Teste para o melhor valor de K para o algoritmo ANN considerando os marcadores localizados no quadril	41
Tabela 3 -	Teste para o melhor valor de K para o algoritmo ANN considerando os marcadores localizados no joelho	41
Tabela 4 -	Teste para o melhor valor de K para o algoritmo ANN considerando os marcadores localizados nos tornozelos	42
Tabela 5 -	Matriz de confusão para k=10 dos marcadores localizados no quadril	42
Tabela 6 -	Teste para o melhor valor de K para o algoritmo SVM considerando todos os marcadores	44
Tabela 7 -	Teste para o melhor valor de K para o algoritmo SVM considerando os marcadores localizados no quadril	44
Tabela 8 -	Teste para o melhor valor de K para o algoritmo SVM considerando os marcadores localizados no joelho	45
Tabela 9 -	Teste para o melhor valor de K para o algoritmo SVM considerando os marcadores localizados no tornozelo	45
Tabela 10 -	Matriz de confusão para k=10 dos marcadores localizados no quadril	46

LISTA DE ABREVIATURAS

ANN	<i>Artificial Neural Networks</i>
DFT	Transformada Discreta de Fourier
FFT	Transformada Rápida de Fourier
Hz	<i>Hertz</i>
mkr	<i>marker-coordinate</i>
ML	<i>Machine Learning</i>
R.Ankle	Proeminência Lateral do Maléolo Lateral Direito
R.ASIS	Espinha Ilíaca Anterior Superior Direita
s	segundo
SVM	<i>Support Vector Machine</i>
t	tempo
TF	Transformada de Fourier
WBDS	<i>Walking Biomechanics Dataset</i>

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVOS	15
1.2 JUSTIFICATIVA	16
1.3 METODOLOGIA	16
2 REFERENCIAL TEÓRICO	17
2.1 MACHINE LEARNING	17
2.1.1 Artificial Neural Networks (ANN)	18
2.1.2 Support Vector Machine (SVM)	20
2.2 BIOMECÂNICA	28
2.3 BASE DE DADOS	29
2.4 PROCESSAMENTO DE SINAIS	31
2.4.1 Sinais De Tempo Discretos	31
2.4.2 Periodicidade De Sinais	32
2.5 ANÁLISE DE FOURIER	32
2.5.1 Transformada De Fourier	33
3 ANÁLISE DE DADOS	35
4 BASE DE DADOS GERADA	38
5 MODELOS DE MACHINE LEARNING	39
5.1 Testes realizados com o ANN	39
5.2 Testes realizados com o SVM	43
6 RESULTADOS E CONCLUSÃO	47
REFERÊNCIAS	48
APÊNDICE A - Códigos utilizados para a Análise de Dados	52
APÊNDICE B - Códigos utilizados para geração da nova base de dados	53
APÊNDICE C - Códigos utilizados para testes dos algoritmos de Machine Learning	55

1 INTRODUÇÃO

Em 1959, Arthur Samuel, pioneiro nos campos de jogos de computador, inteligência artificial e aprendizado de máquina, publicou um artigo onde definiu o aprendizado de máquina como o “campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados” (SAMUEL, 1959 apud BHAVSAR *et al.*, 2017).

Atualmente, com a abundância de bases de dados disponíveis, a demanda pela tecnologia vem crescendo cada vez mais. Muitas indústrias estão investindo nessa área para melhorar seus processos e produtos, desde a indústria de base até a medicina moderna.

O conceito se baseia na ideia de que máquinas e sistemas podem aprender de forma parcial ou totalmente autônoma. Para isso, os sistemas utilizam algoritmos complexos e um grande volume de dados, criando conexões e tornando-os capazes de executar tarefas automaticamente, fazendo a leitura de informações, identificando padrões, coletando dados de usuários e tomando decisões com base na sua própria experiência.

A proposta escolhida envolve a biomecânica do movimento humano, a marcha. Segundo Mezêncio *et al.* (2021, p. 88) “a biomecânica é uma ciência do grupo das ciências biológicas que trata de análises físico-matemáticas de sistemas biológicos e, como consequência, de movimentos humanos”. Estes movimentos podem ser analisados de acordo com leis e normas mecânicas diretamente relacionados aos parâmetros específicos do sistema biológico.

Nos últimos anos, devido à progressão das técnicas de medição, armazenamento e processamento de dados, a análise do movimento teve uma melhora significativa, contribuindo para o desenvolvimento da Biomecânica e para um maior interesse de estudos nessa área (AMADIO *et al.*, 1999, p. 42).

1.1 OBJETIVOS

O propósito deste trabalho é gerar um modelo capaz de discriminar, com eficiência, o indivíduo e seu padrão de marcha. Para isso, foram utilizadas técnicas de processamento de sinais, análise de dados e *Machine Learning* (ML).

Esta pesquisa pretende analisar padrões de marcha considerando informações de cinética e cinemática dos indivíduos, além de desenvolver modelos preditivos empregando as duas técnicas mais utilizadas em aprendizagem de

máquina na biomecânica do movimento humano: *Support Vector Machine* (SVM) e *Artificial Neural Networks* (ANN) (HALILAJ *et al.*, 2018). A base de dados utilizada neste trabalho foi coletada e disponibilizada por Fukuchi *et al.* (2017).

1.2 JUSTIFICATIVA

Acredita-se que a partir da análise dos padrões de marcha, disponibilizados na base de dados, será possível entender melhor a biomecânica humana em atividades físicas com o intuito de identificar e prevenir possíveis doenças ou lesões de natureza musculoesquelética. O desenvolvimento dos modelos preditivos viabiliza a identificação de padrões de marcha tanto em adultos jovens quanto em idosos, que possuem limitações naturais do processo de envelhecimento.

1.3 METODOLOGIA

No estudo realizado por Fukuchi *et al.* (2018) os dados foram coletados a partir das medidas das cinéticas e cinemáticas de 42 participantes, incluindo 24 jovens adultos (idade 27.6 ± 4.4 anos, altura 171.1 ± 10.5 cm e massa 68.4 ± 12.2 kg) e 18 adultos mais velhos (idade 62.7 ± 8.0 anos, altura 161.8 ± 9.5 cm e massa 66.9 ± 10.1 kg) caminhando em diferentes velocidades.

Com base no artigo de Halilaj *et al.* (2018), que fez uma extensa revisão bibliográfica em mais de 3 mil artigos que envolvem Biomecânica e *Machine Learning* (ML), escolheu-se as duas técnicas mais utilizadas de ML na biomecânica do movimento humano.

Seguindo o artigo de Taspinar (2018), foi utilizada a Análise de Fourier que diz que, se um sinal contém componentes que são periódicos por natureza, pode-se decompô-lo em seus componentes periódicos, extraindo as características chave do sinal. Neste trabalho extraiu-se os dez primeiros picos, a partir da Transformada de Fourier (TF) de cada sensor, em todos os eixos, para cada indivíduo, na velocidade auto-selecionada confortável.

2 REFERENCIAL TEÓRICO

Neste capítulo é realizado o embasamento teórico para a pesquisa, explicando e definindo os principais conceitos utilizados como biomecânica, processamento de sinais, ML e as técnicas utilizadas. Além disso, detalha-se melhor a respeito da base de dados utilizada como referência.

2.1 MACHINE LEARNING

Machine Learning é uma subárea da inteligência artificial que vem crescendo muito nos últimos anos. O conceito é baseado em algoritmos que aprendem por experiência e melhoram suas performances com o decorrer do tempo. Esta abordagem costuma ser utilizada para a detecção de padrões em conjuntos de dados.

Frequentemente, após analisar os dados manualmente, não é possível obter as melhores conclusões ou extrair o padrão corretamente, nesses casos, o uso das técnicas de ML é ideal para resolver esses problemas (RICHERT; COELHO, 2013). Sabendo que existem diversos tipos (Figura 1), o problema em questão deve ser analisado para que seja possível escolher a técnica mais adequada.

Figura 1 - Tipos de aprendizado de máquina



Fonte: Raphaell, [202-].

Os algoritmos de aprendizagem supervisionada necessitam de uma assistência externa. A base de dados utilizada deve ser dividida em duas partes: treino e teste. A parte de treino possui as variáveis de saída que precisam ser previstas ou classificadas. Todos os algoritmos aprendem alguns tipos de padrões

do conjunto de dados de treinamento e os aplicam ao conjunto de dados de teste para predição ou classificação (KOTSIANTIS, 2007).

A aprendizagem não-supervisionada é uma técnica onde não existe a informação dos rótulos, ou seja, não se tem as saídas desejadas a serem estimadas e, por este motivo, diz-se que os dados são não-rotulados (ESCOVEDO, 2020). Quando novos dados são introduzidos, o algoritmo usa o que foi aprendido previamente para categorizar os dados. De modo geral, são utilizados para *clustering* e redução de recursos.

Já a aprendizagem semi-supervisionada, é uma técnica que combina o poder dos algoritmos supervisionados e não supervisionados. É bastante útil em processos de mineração de dados onde dados não rotulados estão presentes e a rotulação destes dados é um processo tedioso (ZHU; GOLDBERG, 2009.).

Por fim, a aprendizagem por reforço, que é um tipo de aprendizagem que toma decisões com base em quais ações tornam o resultado mais positivo. A máquina não tem conhecimento de qual ação tomar até que ocorra alguma situação. As ações tomadas podem afetar as situações e ações futuras. Essa técnica depende somente de dois critérios: busca por tentativa e erro e resultado atrasado (SUTTON, 1992).

2.1.1 Artificial Neural Networks (ANN)

O trabalho com redes neurais artificiais tem sido motivado desde o começo pelo reconhecimento de que o cérebro humano processa informações de forma inteiramente diferente do computador digital convencional (HAYKIN, 2007).

O cérebro humano é capaz de organizar seus neurônios de forma a realizar certas tarefas muito mais rapidamente que o mais rápido computador existente. O reconhecimento perceptivo é um exemplo, onde um indivíduo pode reconhecer um rosto familiar em um cenário desconhecido, levando frações de segundos para realizar essa tarefa. Computadores convencionais podem levar até dias para executar tarefas com complexidades muito inferiores (HAYKIN, 2007).

Haykin (2007) explica que um neurônio em “desenvolvimento” é sinônimo de um cérebro plástico, o que permite que o sistema nervoso humano se adapte ao seu meio ambiente e desenvolva suas próprias regras através da experiência. Da mesma forma são as redes neurais construídas com neurônios artificiais.

De modo geral, a rede neural é uma máquina projetada através de componentes eletrônicos, ou por programação (no caso de um computador digital), com a finalidade de modelar a maneira como o “cérebro” realiza uma tarefa. Para alcançar um bom desempenho, as redes neurais interligam as células computacionais, que são chamadas neurônios ou unidades de processamento (HAYKIN, 2007).

A partir dos conceitos vistos, pode-se definir uma rede neural como um processador paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso, sua estrutura é semelhante ao cérebro em dois principais aspectos, estabelecidos por Haykin (2007):

- a) O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem;
- b) Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

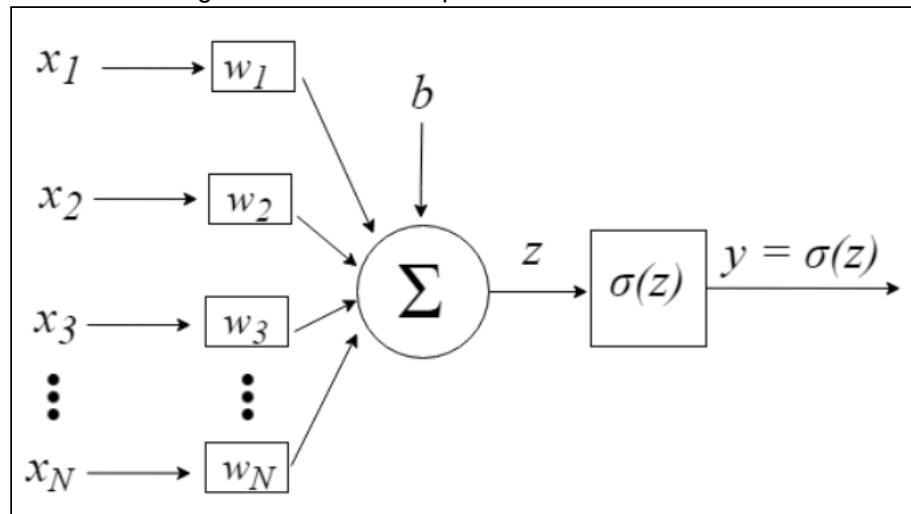
Algoritmo de aprendizagem é o nome dado ao procedimento utilizado para realizar o processo de aprendizagem, sua função é modificar os pesos sinápticos da rede de forma ordenada com o intuito de alcançar um objetivo desejado. A modificação dos pesos sinápticos é o método mais tradicional para o projeto de redes neurais.

O poder computacional de uma rede neural se dá primeiro por sua estrutura e segundo pela sua habilidade de aprender, logo, de generalizar. Segundo Haykin (2007) “a generalização se refere ao fato de a rede ser capaz de produzir saídas adequadas para entradas que não estavam presentes durante o treinamento (aprendizagem)”. Juntas, estas capacidades tornam possível para as redes neurais resolver problemas complexos que atualmente são intratáveis.

Entretanto, é importante ressaltar que a capacidade de processamento das redes neurais artificiais está longe de ser próxima à capacidade humana. Um problema complexo deve ser decomposto em tarefas relativamente simples e assim atribuído a uma rede neural como um subconjunto de tarefas que coincidem com suas capacidades inerentes.

A seguir apresenta-se a formulação matemática para o ANN. O modelo computacional que melhor descreve um neurônio é ilustrado na Figura 2:

Figura 2 - Modelo computacional de um neurônio



Fonte: Leite, (2018).

Os sinais de entrada são representados pelo vetor $x = [x_1, x_2, \dots, x_n]$. Ao chegarem no neurônio, os dados são multiplicados pelos respectivos pesos sinápticos, que são elementos do vetor $w = [w_1, w_2, \dots, w_n]$, gerando o valor z . O valor gerado é denominado de potencial de ativação e é expresso como:

$$z = \sum_{i=1}^N x_i w_i + b \quad (1)$$

O termo b (bias), que não é afetado pela entrada nessa expressão, provê um grau de liberdade a mais. O valor z passa então por uma função matemática de ativação σ , com a característica de ser não linear, responsável por limitar tal valor a um certo intervalo. Sendo assim, produz um valor final y de saída do neurônio. Alguns exemplos de funções de ativação usadas são a degrau, sigmóide, tangente hiperbólica, etc.

2.1.2 Support Vector Machine (SVM)

O SVM é um algoritmo de aprendizagem de máquina supervisionado. É aplicado na solução de problemas lineares e não lineares e funciona muito bem para uma grande variedade de problemas práticos. Ele pode ser usado para desafios de classificação ou de regressão e sua ideia é bastante simples: o algoritmo cria uma linha (ou um hiperplano) que separa os dados em classes (PAULA FILHO, 2019).

A seguir, são apresentados alguns conceitos matemáticos utilizados na definição do SVM, como o comprimento e a direção de um vetor, o produto escalar,

a separabilidade linear, o conceito de hiperplano e como o algoritmo utiliza o classificador.

O comprimento de um vetor x é chamado de norma e é escrita como $||x||$. A Fórmula para o cálculo da norma euclidiana de um vetor $x = (x_1, x_2, ..., x_n)$ é:

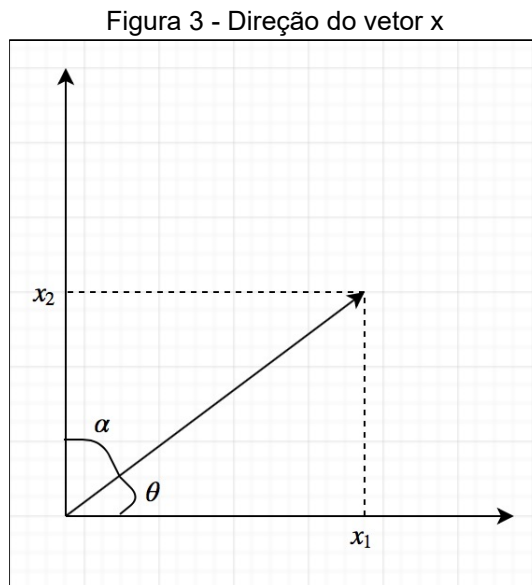
$$||x|| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2} \quad (2)$$

A direção de um vetor $x = (x_1, x_2, ..., x_n)$ é escrita como w e é definida como:

$$w = \left(\frac{x_1}{||x||}, \frac{x_2}{||x||} \right) \quad (3)$$

A Figura 3 mostra que o $\cos(\theta) = \frac{x_1}{||x||}$ e o $\cos(\alpha) = \frac{x_2}{||x||}$. Portanto, a direção do vetor w também pode ser escrita como:

$$w = (\cos(\theta), \cos(\alpha)) \quad (4)$$

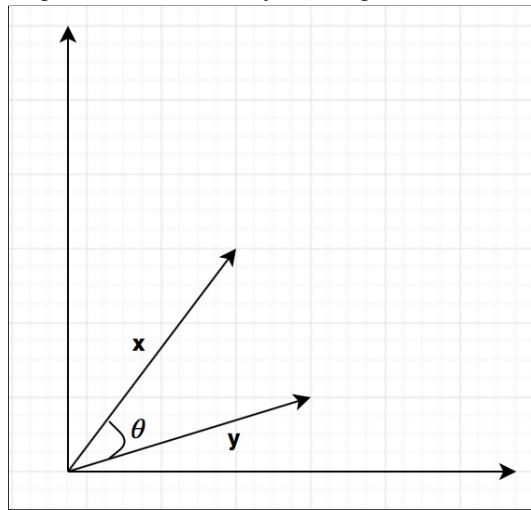


Fonte: Fan, (2018).

O produto escalar entre dois vetores retorna um escalar. Isso demonstra como os vetores estão relacionados. A Figura 4 mostra vetores (x e y) e o ângulo entre eles (θ). A Fórmula do produto escalar é definida como:

$$x \cdot y = \|x\| \|y\| \cos(\theta) \quad (5)$$

Figura 4 - Vetores x , y e o ângulo entre eles

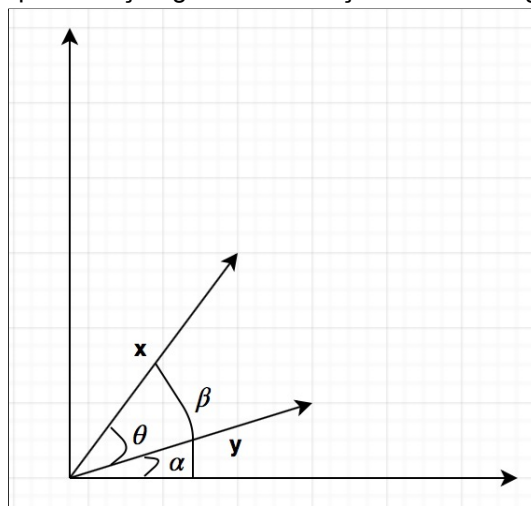


Fonte: Fan, (2018).

Analisando a Figura 5, é possível perceber que $\theta = \beta - \alpha$. Logo, pode-se deduzir que:

$$\begin{aligned} \cos(\theta) &= \cos(\beta - \alpha) \\ \cos(\theta) &= \cos(\beta)\cos(\alpha) - \sin(\beta)\sin(\alpha) \\ \cos(\theta) &= \frac{x_1}{\|x\|} \frac{y_1}{\|y\|} + \frac{x_2}{\|x\|} \frac{y_2}{\|y\|} \\ \cos(\theta) &= \frac{x_1 y_1 + x_2 y_2}{\|x\| \|y\|} \end{aligned} \quad (6)$$

Figura 5 - Representação gráfica da relação entre os ângulos θ , β e α



Fonte: Fan, (2018).

Substituindo o valor encontrado de $\cos(\theta)$ na Fórmula geométrica do produto escalar, obtemos:

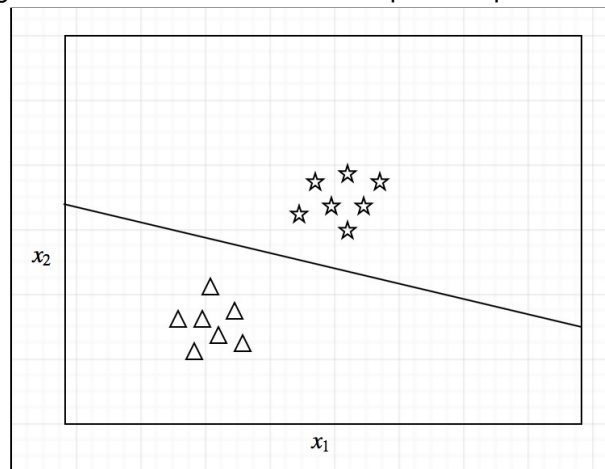
$$\begin{aligned}
 x \cdot y &= \|x\| \|y\| \frac{x_1 y_1 + x_2 y_2}{\|x\| \|y\|} \\
 x \cdot y &= x_1 y_1 + x_2 y_2
 \end{aligned}
 \tag{7}$$

Esta é a fórmula algébrica do produto escalar. De modo geral, o produto escalar pode ser computado, para dois vetores com N dimensões, com o seguinte somatório:

$$x \cdot y = \sum_{i=1}^n x_i y_i
 \tag{8}$$

A separabilidade linear é um conceito importante para o SVM. A Figura 6 mostra dados bidimensionais separados por uma linha. Neste caso, dizemos que os dados são linearmente separáveis.

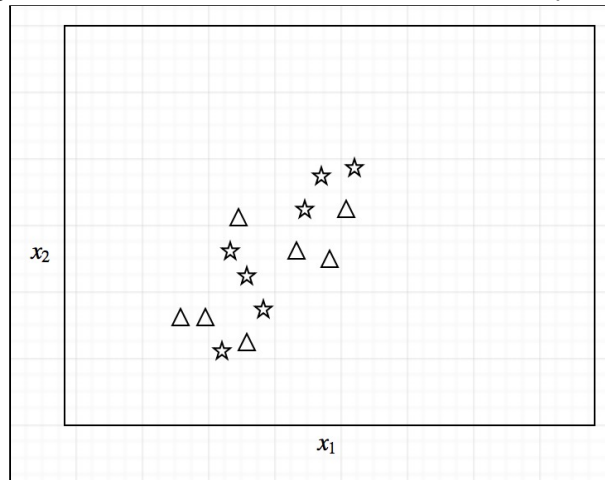
Figura 6 - Dados bidimensionais separados por uma linha



Fonte: Fan, (2018).

Em alguns casos, não será possível realizar a separação linear dos dados. A Figura 7 apresenta um exemplo de dados que não são linearmente separáveis, ou seja, não é possível traçar uma reta que os separe.

Figura 7 - Dados bidimensionais linearmente inseparáveis



Fonte: Fan, (2018).

Quando os dados presentes são multidimensionais, ao invés de se utilizar uma linha para separá-los, utiliza-se um hiperplano. Considerando o caso de dados bidimensionais e linearmente separáveis, a função da linha é $y = ax + b$. Renomeando x por x_1 e y por x_2 , obtém-se:

$$ax_1 + x_2 + b = 0 \quad (9)$$

Definindo $x = (x_1, x_2)$ e $w = (a, -1)$, obtemos:

$$w \cdot x + b = 0 \quad (10)$$

Esta equação é derivada de vetores bidimensionais, porém funciona para qualquer número de dimensões e representa a equação do hiperplano.

Uma vez que obtivemos o hiperplano, podemos então utilizá-lo para fazer previsões. Definimos uma função de hipótese h como:

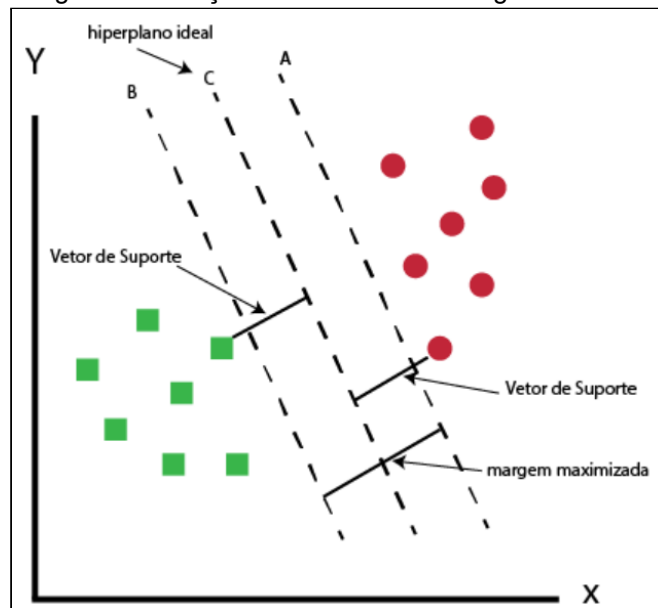
$$\begin{cases} h(x_i) = +1, & \text{se } x \geq 0 \\ h(x_i) = -1, & \text{se } x < 0 \end{cases} \quad (11)$$

O ponto acima ou sob hiperplano será classificado como classe +1, e o ponto abaixo do hiperplano será classificado como classe -1.

De modo geral, o objetivo do SVM é encontrar um hiperplano que possa separar os dados com precisão. Porém, existem infinitas possibilidades de linhas que poderiam separar essas duas classes. Para contornar isso, o SVM encontra os pontos mais próximos das duas classes. Esses pontos são os chamados vetores de

suporte, como mostrado na Figura 8. A distância entre as linhas e os vetores de suporte é chamada de margem. O objetivo do algoritmo é maximizar a margem.

Figura 8 - Solução ótima utilizando o algoritmo SVM

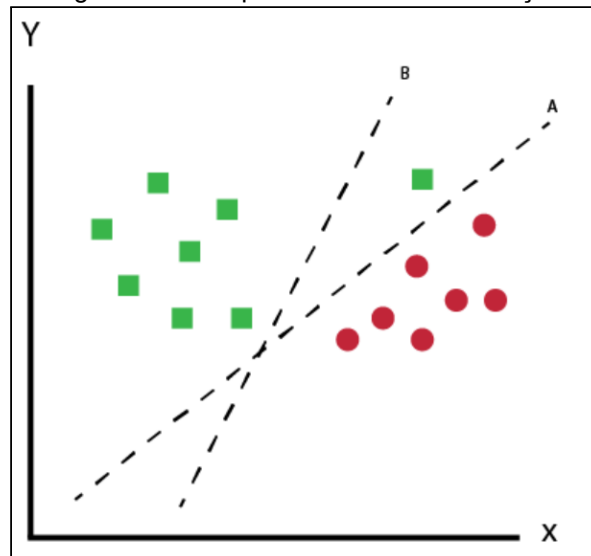


Fonte: Paula Filho, (2019).

Assim, o SVM tenta estabelecer um limite de decisão de forma que a separação entre as classes seja a maior possível. Em alguns casos, pode ser que o algoritmo não escolha uma linha ou hiperplano que seja a maior possível, pois o SVM seleciona o primeiro hiperplano que classifica as classes com precisão, tudo isso antes de maximizar a margem.

No exemplo abaixo (Figura 9), apesar de o hiperplano B possuir uma margem maior em comparação com o hiperplano A, ele possui um erro de classificação, enquanto A classificou tudo corretamente. Logo, o melhor hiperplano é o A.

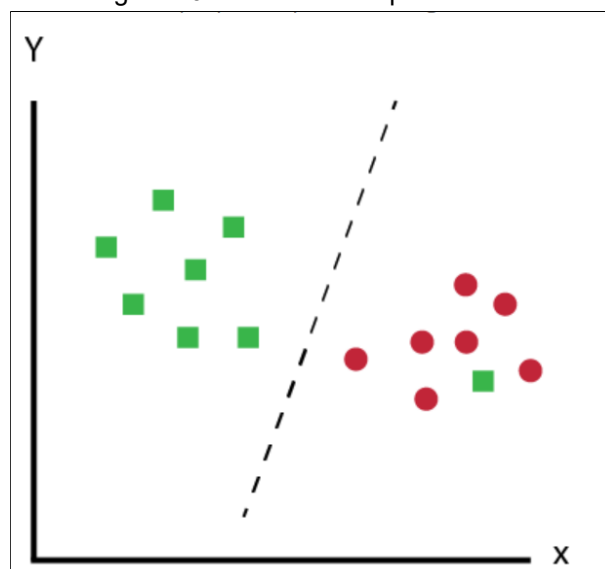
Figura 9 - Exemplo de erro de classificação



Fonte: Paula Filho, (2019).

Pode acontecer de não ser possível separar as duas classes usando uma linha reta, pois algum *outlier* (ponto fora da curva) está presente nos dados. Nesse caso, o SVM tem um recurso para ignorar esses valores discrepantes e encontrar o hiperplano que tem margem máxima. Pode-se dizer que o SVM é robusto para *outliers* (Figura 10).

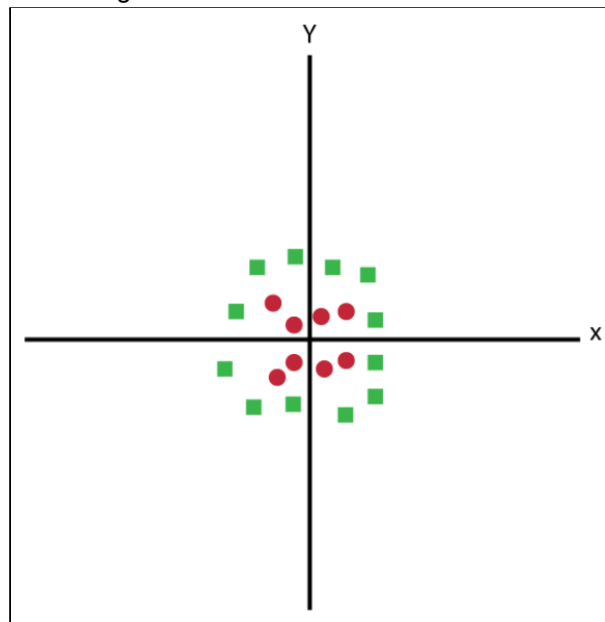
Figura 10 - SVM robusto para outliers



Fonte: Paula Filho, (2019).

Por fim, podem haver casos onde não se pode obter um hiperplano linear entre as duas classes, como mostrado na Figura 11:

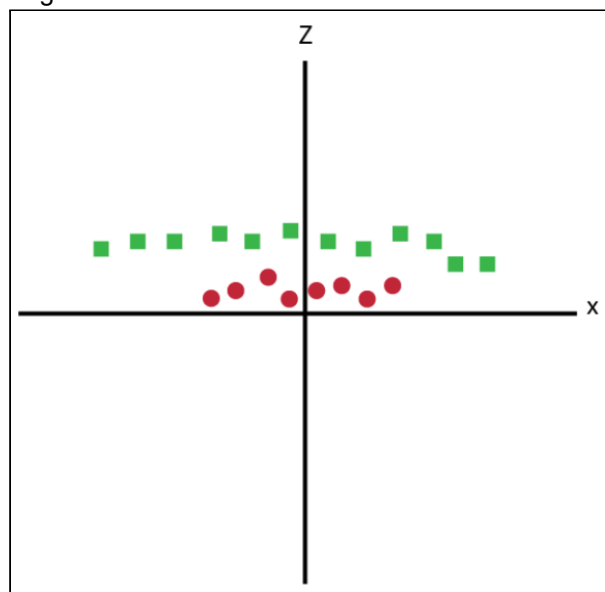
Figura 11 - Base de dados não linear



Fonte: Paula Filho, (2019).

Porém, o SVM resolve esse problema facilmente inserindo um recurso adicional. Basta adicionar a condição de que $z = x^2 + y^2$. Com isso, todos os valores para z são positivos, pois z é a soma quadrática de x e y . Logo, plotando novamente o Gráfico da Figura 11 com esse recurso adicionado obtém-se o Gráfico da Figura 12:

Figura 12 - Base de dados não linear normalizada



Fonte: Paula Filho, (2019).

Esse processo é feito automaticamente pelo algoritmo através de uma técnica chamada de truque do kernel. As funções, chamadas de núcleos (kernels)

convertem um problema não separável em um problema separável transformando o espaço de entrada dimensional baixo em um espaço de entrada dimensional mais alto, sendo bastante úteis no problema de separação não linear (PAULA FILHO, 2019).

2.2 BIOMECÂNICA

A biomecânica pode ser definida como uma ciência entre as ciências derivadas das ciências naturais. Seu principal objetivo é realizar análises físicas de sistemas biológicos, e, por conseguinte, análises físicas de movimentos do corpo humano (AMADIO, 1989).

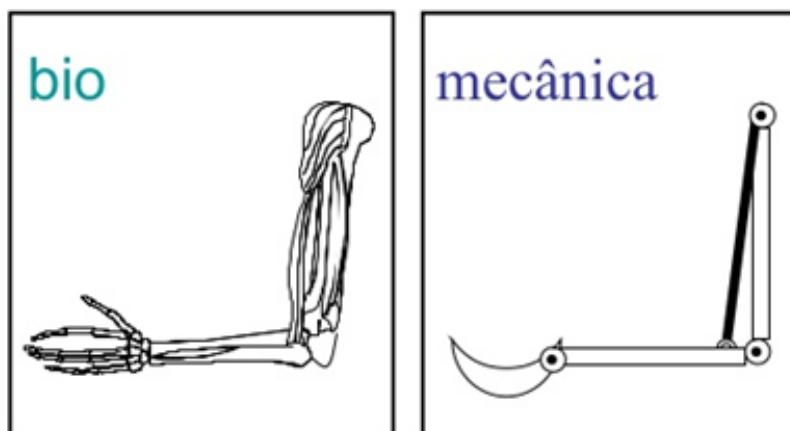
Em tese, devemos considerar que a estrutura funcional de um sistema biológico passou por um processo organizacional evolutivo de otimização, e que esse processo se diferencia sensivelmente do caminho de aperfeiçoamento técnico do movimento. Em contraposição a um corpo rígido, a estrutura biológica do corpo humano permite a produção de força através da contração muscular, que transforma o corpo num sistema independente e assim, o movimento acontece (AMADIO et al., 1999).

Logo, o corpo humano pode ser definido fisicamente como um complexo sistema de segmentos articulados em equilíbrio estático ou em equilíbrio dinâmico, onde o movimento é causado por forças internas atuando fora do eixo articular, provocando deslocamentos angulares dos segmentos, e por forças externas ao corpo. Desta forma, podemos definir que a ciência que descreve, analisa e modela os sistemas biológicos é chamada de Biomecânica, uma ciência altamente interdisciplinar dada a natureza do fenômeno investigado (AMADIO et al., 1999).

Através da biomecânica e de suas áreas correlatas podemos analisar as causas e fenômenos do movimento. Para que seja possível entender melhor a complexidade do movimento humano e explicarmos suas causas, é necessário que outros aspectos da análise multidisciplinar sejam também considerados. Além da biomecânica fazem parte desse campo de estudo e de pesquisa outras importantes disciplinas como a fisiologia geral, a bioquímica, o ensino do movimento, a psicologia, a física (mecânica), a matemática, a eletrônica - instrumentação e processamento de sinais, etc. (AMADIO et al., 1999).

Assim, a Biomecânica do movimento busca explicar como as formas de movimento dos corpos de seres vivos acontecem na natureza a partir de parâmetros cinemáticos e dinâmicos (ZERNICKE, 1981 apud AMADIO et al., 1999).

Figura 13 - Representação conceitual dos elementos da biomecânica



Fonte: Resende, (2021).

De modo geral, a biomecânica aplica as leis da física no que diz respeito por exemplo à alavancas e polias, como ilustrado na Figura 13, para definir e compreender as forças envolvidas nos sistemas biológicos

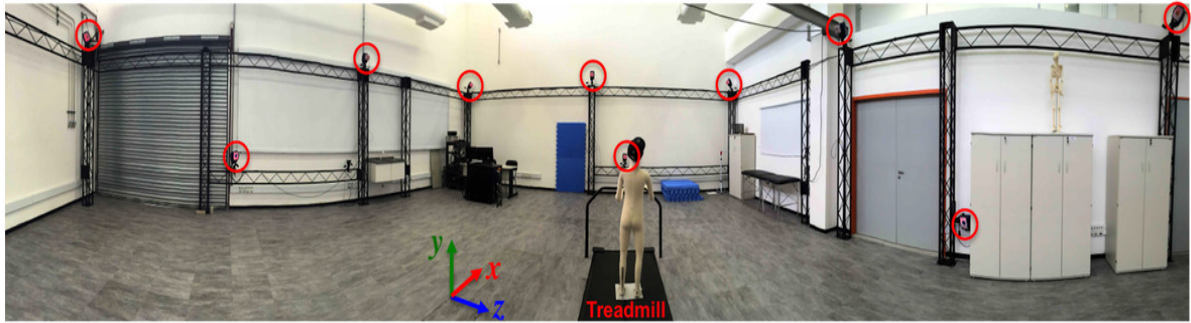
2.3 BASE DE DADOS

A base de dados utilizada neste trabalho foi coletada e disponibilizada por Fukuchi *et al.* (2017), onde todas as caminhadas foram realizadas com os participantes descalços, utilizando calções confortáveis (no caso das mulheres, um sutiã esportivo também era utilizado). Foi pedido a cada participante que selecionasse uma velocidade de caminhada que eles consideravam confortável.

Após isto, foi-lhes pedido que andassem sobre a esteira, realizando-se o seguinte protocolo: para se familiarizar com a velocidade da esteira, os participantes caminharam à sua velocidade confortável durante cinco minutos. Depois, cada participante caminhou durante 90 segundos em cada uma das oito condições de velocidade de marcha (40%, 55%, 70%, 85%, 85%, 100%, 115%, 130%, e 145% desta velocidade).

Para a coleta de dados, foi utilizado um sistema de captura de movimento (Figura 14) que continha 12 câmeras, cinco plataformas de força embutidos no chão, e uma esteira com dupla correia, instrumentada numa sala de 10 × 12 m no Laboratório de Biomecânica e Controle de Motores da Universidade Federal de ABC, Brasil.

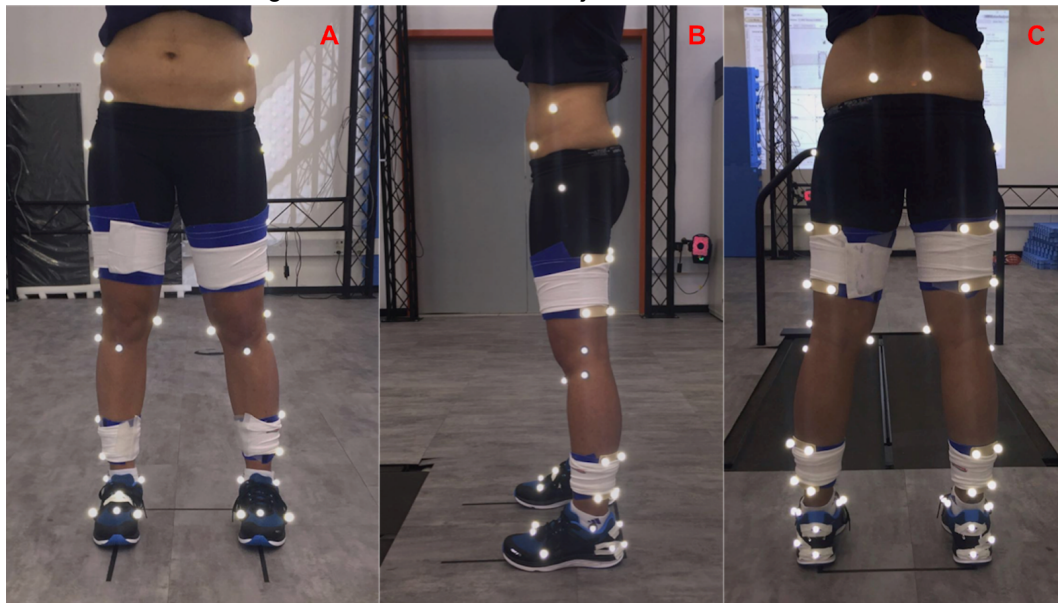
Figura 14 - Sistema de câmeras



Fonte: Fukuchi et al., (2017).

A cada tentativa, os dados cinemáticos e cinéticos foram registrados por cerca de 20 dispositivos *wearables* (Figura 15), durante os últimos 30 segundos, onde cada registro possui seus componentes nos eixos x, y e z. Durante as caminhadas, os participantes foram convidados a caminhar naturalmente e foram autorizados a segurar os corrimãos da esteira, se necessário.

Figura 15 - Protocolo de conjunto de marcadores



Fonte: Fukuchi et al., (2017).

Os dados coletados foram armazenados em vários arquivos de texto seguindo a seguinte nomenclatura: WBDS (*Walking Biomechanics Dataset*), xx números que representam cada participante (do 1 ao 42), *walk* para representar a ação, T que representa *Treadmill* (esteira, em uma tradução livre), xx que representa a porcentagem de velocidade em que a esteira estava com base na velocidade

confortável (1 ao 8), mkr (*marker-coordinate*) que representa a coordenada dos marcadores e .txt que representa a extensão do arquivo.

Sendo assim, “WBDS01walkT01mkr.txt” indica o arquivo que contém as coordenadas dos marcadores, do indivíduo 01 caminhando na velocidade da esteira correspondente a 40% da velocidade confortável auto-selecionada.

A parte da base de dados utilizada neste estudo contém cerca de 42 arquivos em formato de texto. Cada arquivo contém cerca de 4.500 linhas e 67 colunas. A primeira coluna representa o instante de tempo, enquanto as outras 66 representam a exata posição espacial dos marcadores naquele instante. O nome, a sigla e a descrição dos marcadores utilizados podem ser encontrados no artigo de Fukuchi *et al.* (2018).

O objetivo do estudo foi criar uma base de dados sobre cinemática da marcha em 3D e sobre dados cinéticos, ambos influenciados pela idade, ambiente e velocidades de marcha. Com a inclusão de mais participantes, pode-se usar a base como dados normativos de marcha.

2.4 PROCESSAMENTO DE SINAIS

Segundo Haykin e Veen (2001 apud FONSECA *et al.* 2018) “um sinal é formalmente definido como uma função de uma ou mais variáveis, a qual veicula informações sobre a natureza de um fenômeno físico”. Assim, os sinais podem ter muitas formas diferentes: pode-se pensar em sinais de áudio, imagens, sinais de vídeo, dados de sonar e radar. Na sua essência, quase tudo pode ser interpretado como um sinal, desde que traga informação dentro de si.

2.4.1 Sinais De Tempo Discretos

De modo geral, os sinais são divididos em duas categorias: sinais de tempo contínuo e sinais de tempo discreto (TASPINAR, 2018). Os sinais de tempo contínuo (analógicos) são a maioria dos sinais encontrados na natureza, e têm uma variável independente de natureza contínua, que está presente em cada passo de tempo dentro do seu domínio, ou seja, em $t = 0,1$ s, em $t = 0,1435$ s, em $t = 0,1435297$ s, etc.

Já os sinais de tempo discreto apenas são definidos em passos de tempo específicos. Por exemplo, se o período de um sinal discreto for 0,1 s, será definido a $t = 0,1$ s, $t = 0,2$ s, $t = 0,3$ s, etc ... (mas não a $t = 0,143$ s).

Sempre que quisermos digitalizar um sinal analógico com o intuito de o analisar e visualizar num computador, este torna-se discreto. Para isso, é importante escolher uma boa taxa de amostragem, pois se a taxa de amostragem escolhida for muito baixa, o sinal discreto deixará de conter as características do sinal analógico original e as características importantes que o definem serão perdidas.

Para ser mais específico, Taspinar (2018) explica que um sinal é subamostragem se a taxa de amostragem for menor do que a taxa de Nyquist. A taxa de Nyquist é o dobro da maior frequência presente no sinal.

2.4.2 Periodicidade De Sinais

Um sinal periódico contém um padrão que se repete após um período de tempo específico (TASPINAR, 2018). O tempo necessário para ele se repetir é chamado período P e a distância que o sinal percorre neste período é chamada comprimento de onda λ . A frequência f é o inverso do Período; se um sinal tem um Período de 1 segundo, a sua frequência é 1 Hz, e se o período é 10 segundos, a frequência é 0,1 Hz. O período, o comprimento de onda e a frequência estão relacionados entre si através da Fórmula 12, onde s é a velocidade do som:

$$f = \frac{1}{P} = \frac{s}{\lambda} \quad (12)$$

2.5 ANÁLISE DE FOURIER

A análise de Fourier é um campo de estudo utilizado para analisar a periodicidade em sinais periódicos. Taspinar (2018) explica que se um sinal contém componentes periódicos, a análise de Fourier pode ser utilizada para decompor este sinal, dizendo em que frequência se encontram estes componentes periódicos.

Por exemplo, se o ritmo cardíaco de um indivíduo for medido e apresentar um ritmo de 60 batimentos por minuto, o sinal terá uma frequência de 1 hertz (Hz) (Período de 1 segundo = frequência de 1Hz). Caso o indivíduo esteja fazendo uma tarefa repetitiva em que mexe os dedos a cada 2 segundos, o sinal da mão terá uma frequência de 0,5Hz (Período de 2 segundos = frequência de 0,5 Hz). Um eletrodo colocado no braço deste indivíduo irá medir a combinação dos dois sinais, dessa forma, a análise de Fourier realizada nos sinais combinados deve mostrar um pico no espectro de frequência 0,5 Hz e um a 1 Hz.

Assim, dois (ou mais) sinais diferentes (com frequências diferentes, amplitudes, etc.) podem ser misturados entre si para formar um novo sinal composto. O inverso também é verdade, cada sinal - por mais complexo que pareça - pode ser decomposto numa soma dos seus sinais mais simples. Estes sinais mais simples são funções trigonométricas (ondas senoidais e cossenoidais). A função matemática que transforma um sinal do domínio do tempo para o domínio da frequência é chamada Transformada de Fourier, e a função que faz o contrário é chamada de Transformada de Fourier Inversa.

2.5.1 Transformada De Fourier

Em 1822, o matemático Barão Jean-Baptiste-Joseph Fourier desenvolveu a transformada de Fourier, publicada pela primeira vez no livro *Théorie Analytique de la Chaleur* (SANTOS, 2019). A partir do interesse de Fourier em saber como o calor flui nos materiais é que descobriu-se uma importante ferramenta matemática para várias áreas da ciência. A transformada de Fourier (Fórmula 13) é uma função originada da série de Fourier que explica exatamente quais frequências estão no sinal original de uma forma de onda qualquer e é definida a seguir (SANTOS, 2019):

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nx) + b_n \sin(nx)] \quad (13)$$

Partindo da definição da série de Fourier, a transformada de Fourier é encontrada a partir de duas mudanças: os limites de integração se tornam infinitos (tornando o período da função ilimitado) e o parâmetro de frequência ω deve aceitar todos os valores reais. Por definição:

$$F(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-i\omega x} dx \quad (14)$$

Tomando como exemplo o projeto apresentado por Santos (2019), no caso em que existem três ondas senoidais de diferentes frequências representando três notas musicais distintas, as ondas podem ser combinadas em uma única, gerando uma nota musical de forma não periódica, pouco parecida com as senóides originais.

O resultado da descoberta de Fourier mostra que independente da complexidade da forma da onda final, é possível demonstrá-la como uma

combinação de senos e cossenos, desde que se saiba as frequências exatas que devem ser combinadas, além da frequência final apresentada.

Com base na definição (Fórmula 14), $f(x)$ representa a onda complexa, o termo $e^{-i\omega x}$ representa as senóides que originaram o sinal e a integral permite a escolha das frequências de cada senóide necessária para a representação do sinal. O resultado $F(\omega)$ retornado nos fornece a magnitude e o tempo de atraso de cada sinal gerador que será combinado.

Em relação às aplicações digitais não é possível falar de continuidade, por esse motivo, é necessária uma alternativa em relação à TF. Para isso, existe uma variação da mesma, a Transformada Discreta de Fourier (DFT). A DFT é utilizada para sinais conhecidos apenas em instantes separados por tempos de amostragem, ou seja, dados discretos.

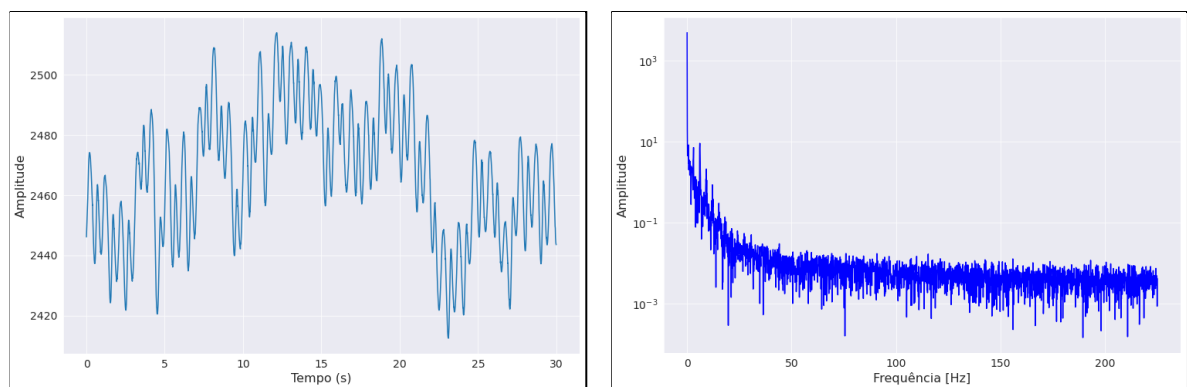
Considerando que qualquer sinal, até mesmo os não periódicos, pode ser reconstruído com bastante precisão adicionando diferentes sinais senoidais, foi criado um algoritmo, a Transformada Rápida de Fourier (FFT). A FFT é um algoritmo que calcula a DFT e permite observar quais frequências estão presentes em qualquer sinal analógico e quais dessas são as mais dominantes. Caso o leitor queira se aprofundar mais no assunto sugere-se a leitura de Oppenheim e Willsky (2010), onde os autores abordam o assunto de forma mais aprofundada.

3 ANÁLISE DE DADOS

A base de dados tomada como referência para este trabalho contém muitas amostras, tornando inviável a análise dos dados de forma visual. Levando em consideração que os dados registrados durante a caminhada dos participantes podem ser expressos como um sinal de tempo discreto e por se tratar de uma atividade cíclica, decidiu-se fazer uso da Análise de Fourier para decompor os sinais existentes em seus componentes periódicos. Além disso, escolheu-se a velocidade auto selecionada confortável para cada indivíduo por se tratar de um dado mais próximo da realidade.

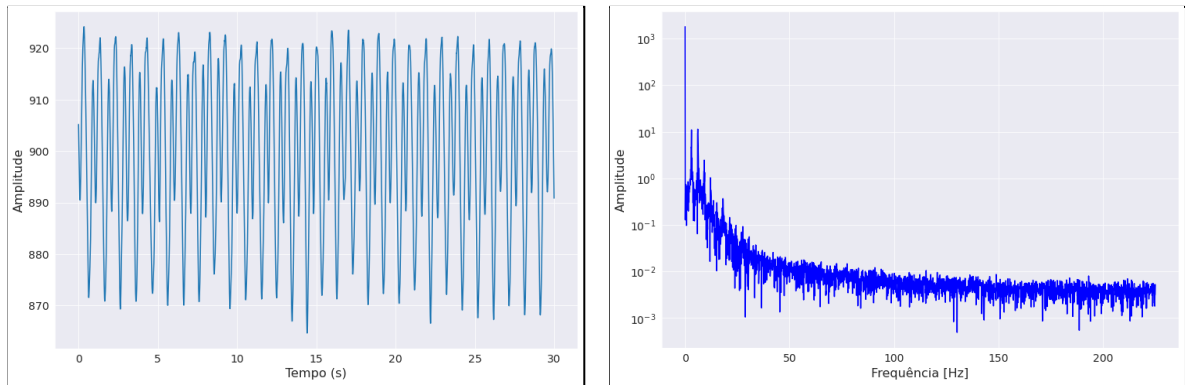
A seguir, são apresentados os Gráficos 1, 2 e 3 com o resultado da aplicação da transformada de Fourier sobre dados coletados do marcador R.ASIS (Espinha ilíaca anterior superior direita) nas dimensões x, y e z de um indivíduo qualquer. É possível notar que mesmo após a Análise de Fourier, não se pode tirar conclusões com base apenas em observações. Além disso, foi necessária a manipulação da escala do Gráfico da transformada para a base logarítmica, tornando possível uma visualização mais detalhada.

Gráfico 1 - Aplicação da Transformada de Fourier no marcador R.ASIS na dimensão X do arquivo WBDS20walkT05mkr.txt



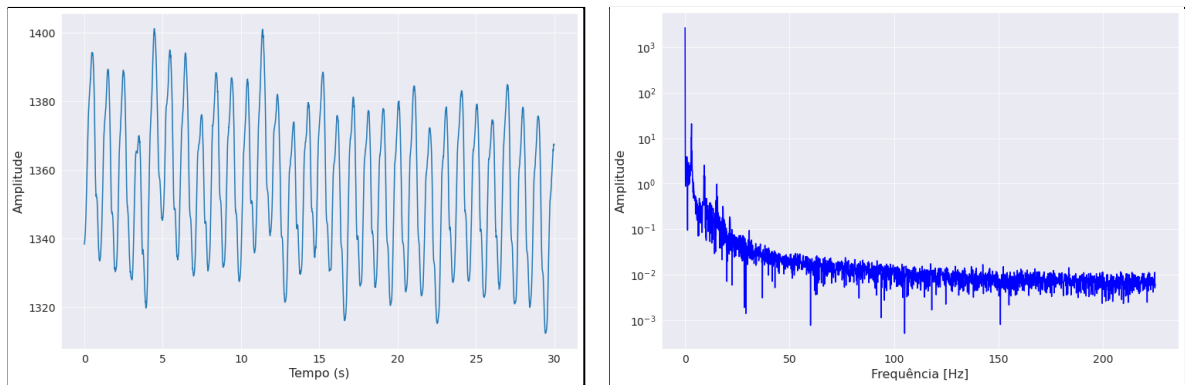
Fonte: Elaborado pelo autor, (2022).

Gráfico 2 - Aplicação da Transformada de Fourier no marcador R.ASIS na dimensão Y do arquivo WBDS20walkT05mkr.txt



Fonte: Elaborado pelo autor, (2022).

Gráfico 3 - Marcador R.ASIS na dimensão Z do arquivo WBDS20walkT05mkr.txt

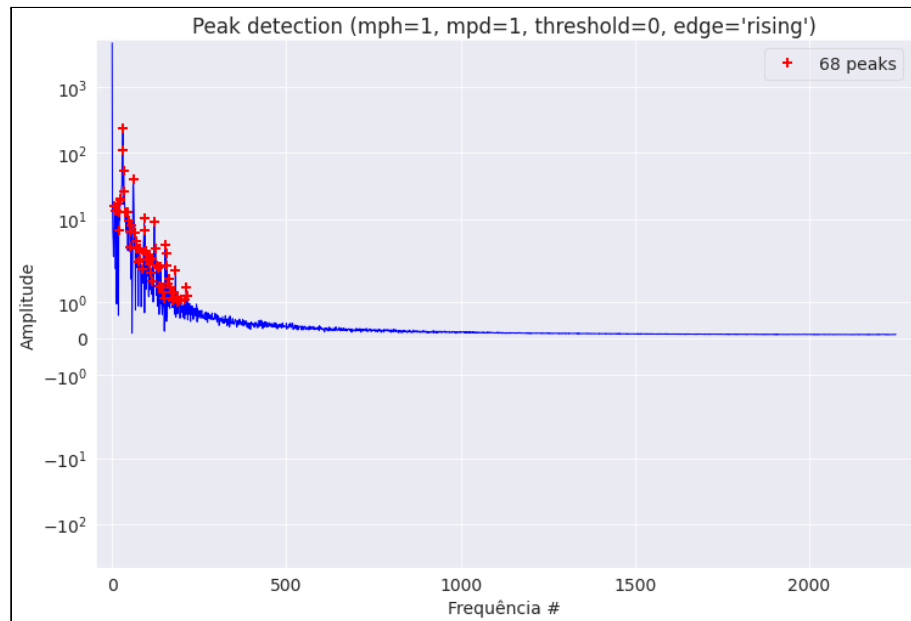


Fonte: Elaborado pelo autor, (2022).

A execução da transformada de Fourier retorna um vetor de tamanho N de valores de magnitude e de frequência, também chamados de picos, sendo N um valor variável.

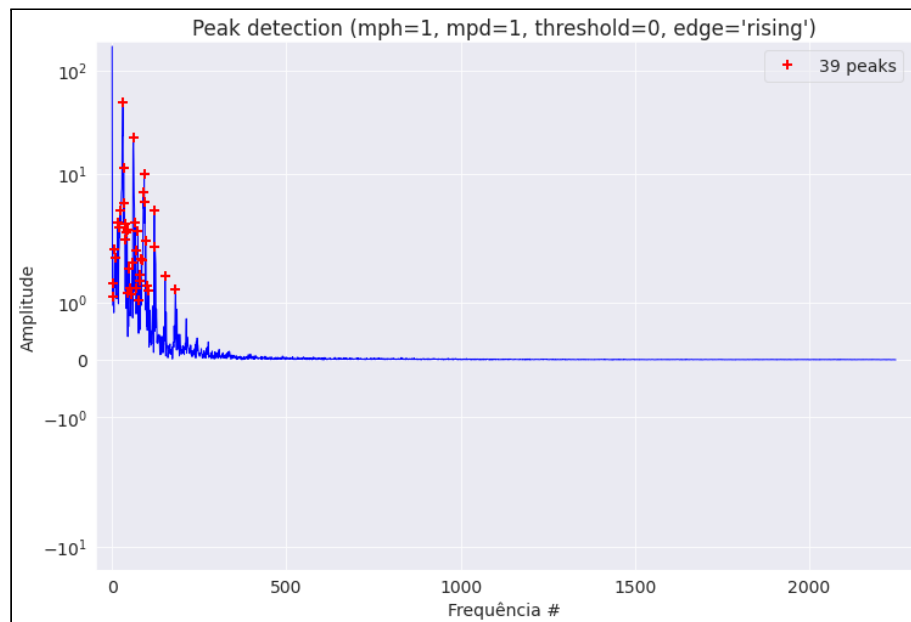
Os Gráficos 4, 5 e 6 mostram os picos da Transformada de Fourier do marcador R.Ankle nas dimensões x, y e z. Como demonstrado nos Gráficos, ainda é difícil fazer uma análise visual.

Gráfico 4 - Picos da Transformada de Fourier do marcador R.Ankle na dimensão X do arquivo WBDS20walkT05mkr.txt destacados



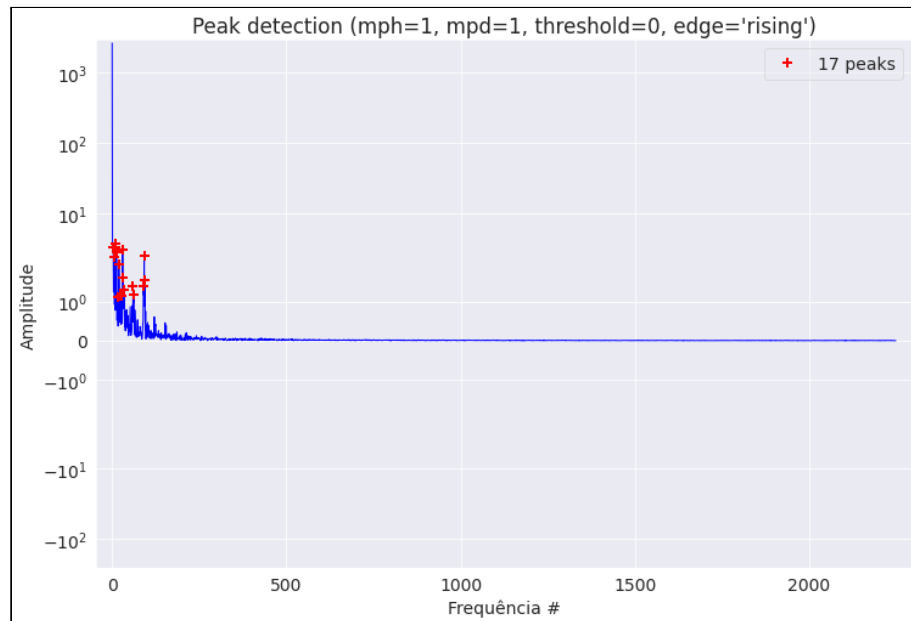
Fonte: Elaborado pelo autor, (2022)

Gráfico 5 - Picos da Transformada de Fourier do marcador R.Ankle na dimensão Y do arquivo WBDS20walkT05mkr.txt destacados



Fonte: Elaborado pelo autor, (2022).

Gráfico 6 - Picos da Transformada de Fourier do marcador R.Ankle na dimensão Z do arquivo WBDS20walkT05mkr.txt destacados



Fonte: Elaborado pelo autor, (2022).

A escolha dos componentes do sinal transformado se fez necessária para maior controle do volume de dados enviados aos algoritmos, visto que a quantidade de componentes extraídos do sinal é imprevisível.

A partir dessas características extraídas, é possível montar uma nova base de dados para alimentar os algoritmos de *Machine Learning* visando a classificação dos dados da forma desejada. Os códigos utilizados nesta seção estão disponíveis no Apêndice A.

4 BASE DE DADOS GERADA

Com a análise dos dados feita, uma nova base de dados foi gerada com o intuito de facilitar os testes com os algoritmos. Essa nova base de dados contém os dados de 39 dos 42 indivíduos, pois em 3 desses indivíduos a quantidade de sensores que falharam na coleta inviabilizou o processamento e a normalização dos dados.

A Transformada de Fourier foi realizada para todos os sensores, em todas as coordenadas, dos 39 indivíduos selecionados. Foram extraídos 10 componentes para cada sensor, ou seja, foram escolhidos os 10 primeiros picos de cada FFT, a fim de testar os algoritmos com diferentes combinações até que os melhores resultados fossem encontrados. Não foram escolhidos mais picos pois quanto mais distantes da origem, mais os picos se aproximam de ruídos ao invés de

caracterizarem o sinal. Os códigos utilizados para a geração da nova base de dados estão disponíveis no Apêndice B, bem como o *link* do arquivo gerado.

5 MODELOS DE *MACHINE LEARNING*

Com base no artigo de Halilaj et al. (2018), as duas técnicas escolhidas para a realização do trabalho foram *Artificial Neural Networks* e *Support Vector Machine*. A seguir, são apresentados os testes realizados com cada algoritmo. Os códigos utilizados para os testes estão disponíveis no Apêndice C.

5.1 Testes realizados com o ANN

Com a criação da nova base de dados, foi utilizado o trecho de código da Figura 16 para ler os dados gerados e formatá-los de modo que o algoritmo ANN consiga interpretá-los. Nos testes realizados utilizou-se a linguagem de programação Python em sua versão 3.10.2, as bibliotecas Pandas, NumPy e pacotes do scikit-learn.

Figura 16 - Código utilizado para preparar a base de dados para os algoritmos de ML

```
def readFiles(arquivo):  
    dados = pd.read_excel(arquivo)  
    dados = dados.drop('Indivíduo / Velocidade', 1)  
    y_ = dados['Classificação']  
    dados = dados.drop('Classificação', 1)  
  
    X = np.array(dados)  
    y = np.array(y_)  
    return X, y
```

Fonte: Elaborado pelo autor, (2022).

Na Figura 17 está apresentada a função responsável por treinar os modelos e retornar os resultados da classificação. Optou-se por utilizar a validação cruzada para melhor avaliar a capacidade de generalização do modelo. Os dados eram salvos em um arquivo separado para posterior análise.

Figura 17 - Código utilizado para treinar o classificador ANN

```
def mlpScore(arquivo):
    X, y = readFiles(arquivo)

    clf = MLPClassifier()
    scores = cross_val_score(clf, X, y, cv=10, scoring='accuracy')

    print(scores.mean())
    with open('MLP', 'a') as f:
        f.write('\nFile: ' + str(arquivo.strip('./')))
        f.write('\nMean: ' + str(scores.mean()))
```

Fonte: Elaborado pelo autor, (2022).

Foram realizados testes com os valores de $k = 1, \dots, 10$, sendo k o número de picos. Além de variar o número de picos, os testes também foram realizados separando os marcadores por regiões, sendo elas: quadril, joelho e tornozelo. As médias obtidas após cada execução do algoritmo de Figura 17 são exibidas nas Tabelas a seguir:

Tabela 1 - Teste para o melhor valor de K para o algoritmo ANN considerando todos os marcadores.

Todos os Sensores	
Valor de K	Acurácia Média
1	0,59
2	0,62
3	0,62
4	0,52
5	0,53
6	0,69
7	0,62
8	0,458
9	0,59
10	0,53

Fonte: Elaborado pelo autor, (2022).

Tabela 2 - Teste para o melhor valor de K para o algoritmo ANN considerando os marcadores localizados no quadril.

Quadril	
Valor de K	Acurácia Média
1	0.72
2	0.62
3	0.425
4	0.575
5	0.57
6	0.6
7	0.57
8	0.65
9	0.59
10	0.75

Fonte: Elaborado pelo autor, (2022).

Tabela 3 - Teste para o melhor valor de K para o algoritmo ANN considerando os marcadores localizados no joelho.

Joelho	
Valor de K	Acurácia Média
1	0.64
2	0.62
3	0.48
4	0.64
5	0.61
6	0.72
7	0.59
8	0.625
9	0.57
10	0.425

Fonte: Elaborado pelo autor, (2022).

Tabela 4 - Teste para o melhor valor de K para o algoritmo ANN considerando os marcadores localizados nos tornozelos.

Tornozelo	
Valor de K	Acurácia Média
1	0.492
2	0.575
3	0.6
4	0.47
5	0.48
6	0.542
7	0.542
8	0.558
9	0.52
10	0.425

Fonte: Elaborado pelo autor, (2022).

Após a análise dos dados, calculou-se a matriz de confusão, com o código exibido na Figura 18, do melhor resultado obtido (marcadores localizados no quadril para k=10), com o intuito de verificar a acurácia e a sensibilidade do modelo.

Figura 18 - Código utilizado para gerar a matriz de confusão do classificador ANN

```
def mlpConfusionMatrix(arquivo):
    X, y = readFiles(arquivo)

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=3)

    clf = MLPClassifier()
    clf.fit(X_train, y_train)
    print(confusion_matrix(y_test, clf.predict(X_test)))
```

Fonte: Elaborado pelo autor, (2022).

Tabela 5 - Matriz de confusão para k=10 dos marcadores localizados no quadril

MATRIZ DE CONFUSÃO		
	VERDADEIRO	FALSO
VERDADEIRO	5	0
FALSO	3	4

Fonte: Elaborado pelo autor, (2022).

O resultado obtido com a matriz de confusão nos mostra que, existem 8 itens classificados na classe VERDADEIRO; 3 itens estão incorretamente classificados e 5 itens estão corretamente classificados. Além disso, existem 4 itens classificados na classe FALSO; 4 desses itens estão corretamente classificados e nenhum item está incorretamente classificado. Vale ressaltar que o vetor de entrada para a geração da matriz de confusão possui apenas 30% dos dados, cerca de 12 indivíduos, pois os outros 70%, cerca de 27 indivíduos, foram utilizados para treinar o classificador.

Com base nos testes realizados, entendeu-se que o melhor valor de K seria 10 para os sensores isolados do quadril, obtendo assim uma acurácia de 75%. A matriz de confusão nos diz que a acurácia desse modelo, apesar de não muito alta, atende aos requisitos.

5.2 Testes realizados com o SVM

Nesta etapa, foi utilizado o mesmo trecho de código da Figura 16 para ler os dados gerados e formatá-los de modo que o algoritmo SVM consiga interpretá-los. Manteve-se as mesmas condições de testes utilizados no algoritmo ANN variando o parâmetro k, e também os marcadores separados por região. Na Figura 19 está apresentada a função responsável por treinar os modelos e retornar os resultados da classificação. Nos testes realizados utilizou-se a linguagem de programação Python em sua versão 3.10.2, as bibliotecas Pandas, NumPy e pacotes do scikit-learn.

Figura 19 - Código utilizado para treinar o classificador SVM

```
def svmScore(arquivo):
    X, y = readFiles(arquivo)

    clf = svm.SVC(kernel='linear', C=1.0)
    scores = cross_val_score(clf, X, y, cv=10, scoring='accuracy')

    with open('SVM', 'a') as f:
        f.write('\nFile: ' + str(arquivo.strip('./')))
        f.write('\nMean: ' + str(scores.mean()))
```

Fonte: Elaborado pelo autor, (2022).

Os mesmos parâmetros utilizados para validação do modelo ANN foram utilizados para o SVM. As médias obtidas após cada execução do algoritmo da Figura 19 são exibidas nas Tabelas a seguir:

Tabela 6 - Teste para o melhor valor de K para o algoritmo SVM considerando todos os marcadores.

Todos os Sensores	
Valor de K	Acurácia Média
1	0.62
2	0.61
3	0.63
4	0.67
5	0.61
6	0.64
7	0.64
8	0.57
9	0.59
10	0.54

Fonte: Elaborado pelo autor, (2022).

Tabela 7 - Teste para o melhor valor de K para o algoritmo SVM considerando os marcadores localizados no quadril.

Quadril	
Valor de K	Acurácia Média
1	0.62
2	0.62
3	0.57
4	0.48
5	0.58
6	0.62
7	0.64
8	0.62
9	0.625
10	0.85

Fonte: Elaborado pelo autor, (2022).

Tabela 8 - Teste para o melhor valor de K para o algoritmo SVM considerando os marcadores localizados no joelho.

Joelho	
Valor de K	Acurácia Média
1	0.575
2	0.575
3	0.62
4	0.72
5	0.62
6	0.59
7	0.675
8	0.69
9	0.6
10	0.575

Fonte: Elaborado pelo autor, (2022).

Tabela 9 - Teste para o melhor valor de K para o algoritmo SVM considerando os marcadores localizados no tornozelo.

Tornozelo	
Valor de K	Acurácia Média
1	0.47
2	0.558
3	0.48
4	0.58
5	0.53
6	0.558
7	0.53
8	0.58
9	0.58
10	0.5

Fonte: Elaborado pelo autor, (2022).

Após a análise dos dados, calculou-se a matriz de confusão, com o código exibido na Figura 20, do melhor resultado obtido, que acabou por ser o mesmo do

ANN (marcadores localizados no quadril para $k=10$), também com o intuito de verificar a acurácia e a sensibilidade do modelo.

Figura 20 - Código utilizado para gerar a matriz de confusão do classificador SVM

```
def svmConfusionMatrix(arquivo):
    X, y = readFiles(arquivo)

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=3)

    clf = svm.SVC(kernel='linear', C=1.0)
    clf.fit(X_train, y_train)
    print(confusion_matrix(y_test, clf.predict(X_test)))
```

Fonte: Elaborado pelo autor, (2022).

Tabela 10 - Matriz de confusão para $k=10$ dos marcadores localizados no quadril

MATRIZ DE CONFUSÃO		
	VERDADEIRO	FALSO
VERDADEIRO	5	0
FALSO	1	6

Fonte: Elaborado pelo autor, (2022).

O resultado obtido com a matriz de confusão nos mostra que, existem 6 itens classificados na classe VERDADEIRO; 1 itens está incorretamente classificado e 5 itens estão corretamente classificados. Além disso, existem 6 itens classificados na classe FALSO; 6 desses itens estão corretamente classificados e nenhum item está incorretamente classificado. Assim como no ANN, o vetor de entrada para a geração da matriz de confusão possui apenas 30% dos dados, cerca de 12 indivíduos, pois os outros 70%, cerca de 27 indivíduos, foram utilizados para treinar o classificador.

Obteve-se o melhor valor para $K=10$ para os sensores isolados do quadril, obtendo assim uma acurácia média de 85%. A matriz de confusão nos diz que a acurácia desse modelo atende totalmente aos requisitos.

Com mais algum estudo e refinamento dos algoritmos a precisão do algoritmo pode ser melhorada, podendo ultrapassar os 90%, como ocorreu em alguns casos na validação cruzada.

6 RESULTADOS E CONCLUSÃO

Ao final dos testes foi possível gerar um modelo capaz de discriminar de maneira eficiente o indivíduo e seu padrão de marcha. O modelo em questão foi o algoritmo *Support Vector Machine*, que obteve uma acurácia média de 85% em sua classificação considerando os marcadores localizados na região do quadril.

Isso significa que foi possível, a partir dos dados cinemáticos, diferenciar os padrões de marcha de adultos jovens e idosos. Com este resultado alcançou-se os objetivos propostos neste trabalho.

Sendo assim, o modelo desenvolvido é capaz de identificar alterações no padrão de marcha, levando em consideração dados cinemáticos, o que pode ajudar no diagnóstico de doenças e/ou lesões de natureza músculo-esquelética. Vale ressaltar que, apesar de alguns bons resultados terem sido encontrados durante os testes, resultados expressivos foram obtidos apenas nos marcadores localizados na região do quadril.

Por fim, a base de dados utilizada na pesquisa continha amostras de poucos indivíduos. Para um estudo mais aprofundado com um resultado mais fidedigno, faz-se necessário um universo amostral maior.

REFERÊNCIAS

AMADIO, A.C. **Fundamentos da biomecânica do esporte**: considerações sobre a análise cinética e aspectos neuromusculares do movimento. 1989. 119p. Tese (Livre Docência) - Escola de Educação Física, Universidade de São Paulo, São Paulo, 1989.

AMADIO, A. C.; COSTA, P. H. L.; SACCO, I. C. N.; SERRÃO, J. C.; ARAUJO, R. C.; MOCHIZUKI, L.; DUARTE, M. Introdução à análise do movimento humano: descrição e aplicação dos métodos biomecânicos de medição. **Revista Brasileira de Fisioterapia**, São Paulo, v. 3, n. 2, p. 41-54, 1999.

Disponível em:

https://edisiplinas.usp.br/pluginfile.php/5462886/mod_resource/content/1/artigo1_aula1.pdf. Acesso em: 01 mar. 2022.

BARELA, A. M. F. **Análise biomecânica do andar de adultos e idosos nos ambientes aquático e terrestre**. 2005. Tese (Doutorado em Educação Física) - Escola de Educação Física e Esporte, Universidade de São Paulo, São Paulo, 2005. Disponível em:

https://teses.usp.br/teses/disponiveis/39/39132/tde-29062006-150959/publico/tese_amfb.pdf. Acesso em: 01 mar. 2022.

BHAVSAR, P.; SAFRO, I.; BOUAYNAYA, N.; POLIKAR, R.; DERA, D. Chapter 12: Machine Learning in transportation data analytics. *In*: CHOWDHURY, M.; APON, A.; DEY, K. (ed.). **Data Analytics for intelligent transportation systems**. Netherlands: Elsevier, 2017. p. 283-307, 2017. ISBN: 9780128097151. Disponível em:

<https://www.sciencedirect.com/science/article/pii/B9780128097151000122>. Acesso em: 28 fev. 2022.

ESCOVEDO, T. Machine Learning: Conceitos e Modelos. **Medium**, [s.l.], 29 jun. 2020. Parte II: Aprendizado Não-Supervisionado. Disponível em: <https://tatianaesc.medium.com/machine-learning-conceitos-e-modelos-parte-ii-aprendizado-n%C3%A3o-supervisionado-fb6d83e4a520>. Acesso em: 01 mar. 2022.

FAN, S. **Understanding the mathematics behind Support Vector Machines**. 07 mai. 2018. *Online*. Disponível em:

<https://shuzhanfan.github.io/2018/05/understanding-mathematics-behind-support-vector-machines/>. Acesso em: 01 mar. 2022.

FONSECA, A. S.; RODRIGUES, J. S. S. R.; SOUSA, A. F. G. Relevância e resultados da monitoria 2017.4 para as disciplinas de análise de sistemas lineares e sinais e sistemas. *In*: SEMINÁRIO DE PROJETOS DE ENSINO, 3., 2018, Marabá. **Anais [...]**. Marabá: DPROJ, 2018. Disponível em: <https://periodicos.unifesspa.edu.br/index.php/spe/article/view/616>. Acesso em: 01 mar. 2022.

FUKUCHI, C. A.; FUKUCHI, R. K.; DUARTE, M. A public dataset of overground and treadmill walking kinematics and kinetics in healthy individuals. **PeerJ**, [s.l.], v. 6, p. 1-17, abr. 2018. DOI:

<https://doi.org/10.7717/peerj.4640>. Disponível em:
<https://peerj.com/articles/4640/>. Acesso em: 01 fev. 2022.

FUKUCHI, R. K.; FUKUCHI, C. A.; DUARTE, M. A public dataset of running biomechanics and the effects of running speed on lower extremity kinematics and kinetics. **PeerJ**, [s.l.], v. 5, p. 1-24, 09 mai. 2017. DOI: <https://doi.org/10.7717/peerj.3298>. Disponível em: <https://peerj.com/articles/3298/>. Acesso em: 28 fev. 2022.

HALILAJ, E.; RAJAGOPAL, A.; FITERAU, M.; HICKS, J. L.; HASTIE, T. J.; DELP, S. L. Machine Learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities. **Journal of Biomechanics**, [s.l.], v. 81, p. 1-11, nov. 2018. DOI: <https://doi.org/10.1016/j.jbiomech.2018.09.009>. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0021929018307309>. Acesso em: 01 fev. 2022.

HAYKIN, S. **Redes Neurais: Princípios e prática**. Porto Alegre: Bookman, 2007. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788577800865/pageid/1>. Acesso em: 05 ago. 2021.

HOF, L. Scaling gait data to body size. **Gait & Posture**, v. 4, n. 3, p. 222-223, mai. 1996. DOI: [https://doi.org/10.1016/0966-6362\(95\)01057-2](https://doi.org/10.1016/0966-6362(95)01057-2). Disponível em: <https://www.sciencedirect.com/science/article/pii/0966636295010572>. Acesso em: 01 mar. 2022.

KOTSIANTIS, S. B. Supervised Machine Learning: a review of classification techniques. **Informatica**, Ljubljana, v. 31, n. 3, p. 249-268, out. 2007. ISSN: 0350-5596. Disponível em: <https://www.informatica.si/index.php/informatica/article/view/148/140>. Acesso em: 01 mar. 2022.

LEITE, T. M. Redes neurais, perceptron multicamadas e o algoritmo backpropagation. **Medium**, [s.l.], 10 mai. 2018. Disponível em: <https://medium.com/ensina-ai/redes-neurais-perceptron-multicamadas-e-o-algoritmo-backpropagation-eaf89778f5b8>. Acesso em: 03 ago. 2021.

MEZÊNCIO, B.; FERREIRA, J. C.; AMADIO, A. C. Biomecânica do Movimento Humano. **Revista Corpoconsciência**, [s.l.], v. 25, n. 2, p. 87-109, mai./ago. 2021. DOI: 10.51283/rc.v25i2.12682. Disponível em: <https://periodicoscientificos.ufmt.br/ojs/index.php/corpoconsciencia/article/view/12682>. Acesso em:

OPPENHEIM, A. V.; WILLSKY, A. S. **Sinais e sistemas**. 2. ed. São Paulo: Pearson, 2010. 568 p., ISBN 9788576055044.

PAULA FILHO, A. C. Scikit-Learn: Support Vector Machine ou máquina de vetores de suporte. **Código Fluente**, [s.l.], 11 jun. 2019. Disponível em:

<https://www.codigofluente.com.br/aula-08-scikit-learn-maquina-de-vetores-de-suporte/>. Acesso em: 20 jul. 2021.

RAPHAELL, B. Desmistificando termos em Machine Learning: tipos de aprendizado. Alura, [s.l.], 16 dez. [202-], **Data Science**. Disponível em: <https://www.alura.com.br/artigos/desmistificando-termos-machine-learning-tipo-s-aprendizado>. Acesso em: 01 mar. 2022.

REDES Neurais: O que são e qual sua importância? **SAS**, [s.l.], [201-?]. Disponível em: https://www.sas.com/pt_br/insights/analytics/neural-networks.html. Acesso em: 01 mar. 2022.

RESENDE, P. H. S. Biomecânica. **Arqueologia e Pré-história**. [s.l.]. 2021. Disponível em: <https://arqueologiaeprehistoria.com/biomecanica/>. Acesso em: 29 ago. 2021.

RICHERT, W.; COELHO, L. P. **Building Machine Learning Systems with Python**. Birmingham: Packt Publishing, 2013. ISBN: 9781782161400. Disponível em: <http://totoharyanto.staff.ipb.ac.id/files/2012/10/Building-Machine-Learning-Systems-with-Python-Richert-Coelho.pdf>. Acesso em: 01 fev. 2022.

SAMUEL, A. L. Some studies in Machine Learning using the game of checkers. **IBM Journal of Research and Development**, [s. l.], v. 3, n. 3, p. 211-229, 1959. Disponível em: <https://www.cs.virginia.edu/~evans/greatworks/samuel1959.pdf>. Acesso em: 28 fev. 2022.

SANTOS, C. S. Usando a transformada de Fourier para construir um visualizador de música com Python e Arduino. **Medium**, [s.l.], 18 jun. 2019. Disponível em: <https://medium.com/@cafelouco/usando-a-transformada-de-fourier-para-construir-um-visualizador-de-m%C3%BA-sica-com-python-e-ardu%C3%ADno-57c98e723cdc>. Acesso em: 01 mar. 2022.

SUTTON, R. S. Introduction: The Challenge of Reinforcement Learning. **Machine Learning**, 8. Boston: Kluwer Academic Publishers, 1992. p. 225-227, 1992. Disponível em: <http://incompleteideas.net/papers/challengeofRL.pdf> . Acesso em: 01 mar. 2022.

TASPINAR, A. Machine Learning with Signal Processing Techniques. **ML Fundamentals**, [s.l.], 04 abr. 2018. Disponível em: <https://ataspinar.com/2018/04/04/machine-learning-with-signal-processing-techniques/> Acesso em: 01 fev. 2022.

ZHU, X.; GOLDBERG, A. B. Introduction to semi-supervised learning. *In*: BRACHMAN, R. J.; DIETTERICH, T. (ed.). **Synthesis Lectures on Artificial Intelligence and Machine Learning**. [s.l.]: Morgan & Claypool Publishers,

2009. p. 1-130, 2009. ISBN: 9781598295481. DOI:
<https://doi.org/10.2200/S00196ED1V01Y200906AIM006>. Disponível em:
<https://www.morganclaypool.com/doi/pdf/10.2200/S00196ED1V01Y200906AIM006>. Acesso em: 01 mar. 2022.

APÊNDICE A - Códigos utilizados para a Análise de Dados

```
#ler arquivos de um único indivíduo
tentativas = []
for i in range(1,8):
    try:
        arquivo = './WBDS20walkT0' + str(i) + 'mkr.txt'
        dados = pd.read_csv(arquivo, sep='\t')
        dados = dados.dropna()
        tentativas.append(dados)
    except:
        continue

#plotar gráfico dos valores de cada marcador individual
plt.plot(tentativas[4]['Time'], tentativas[4]['R.ASISZ'])
plt.xlabel('Tempo (s)', fontsize=16)
plt.ylabel('Amplitude', fontsize=16)

#obter os valores da FFT
def get_fft_values(y_values, T, N, f_s):
    f_values = np.linspace(0.0, 1.0/(2.0*T), N//2)
    fft_values_ = fft(y_values)
    fft_values = 2.0/N * np.abs(fft_values_[0:N//2])
    return f_values, fft_values

#plotar gráfico da FFT dos valores de cada marcador individual
t_n = 10
N = 4500
T = t_n / N
f_s = 1/T
marcador = np.array(tentativas[4]['R.AnkleY'])
f_values, fft_values = get_fft_values(marcador, T, N, f_s)
plt.plot(f_values, fft_values, linestyle='--', color='blue')
plt.xlabel('Frequência [Hz]', fontsize=16)
plt.ylabel('Amplitude', fontsize=16)
plt.yscale('log')
plt.title("Domínio de frequência do sinal", fontsize=16)
plt.show()

#marcar picos no gráfico da FFT
detect_peaks(fft_values, mph=1, show=True)
```

APÊNDICE B - Códigos utilizados para geração da nova base de dados

#link para a base de dados gerada
<https://drive.google.com/drive/folders/17AfBSrjIsswkuWhqQOWot1gSv6UXcVWu?usp=sharing>

```
#lê os arquivos de um único indivíduo
tentativas = []
for i in range(1,8):
    try:
        arquivo = './WBDS20walkT0' + str(i) + 'mkr.txt'
        dados = pd.read_csv(arquivo, sep='\t')
        dados = dados.dropna()
        tentativas.append(dados)
    except:
        continue
indivíduos = []
indivíduo = []
for i in range(7):
    for column in tentativas[i]:
        indivíduo.append(np.array(tentativas[i]))
        indivíduos.append(indivíduo)

#obtem os n primeiros picos da FFT
def get_first_n_peaks(x, y, n_peaks=5):
    x_, y_ = list(x), list(y)
    if len(x_) >= n_peaks:
        return x_[:n_peaks], y_[:n_peaks]
    else:
        missing_n_peaks = n_peaks-len(x_)
        return x_ + [0]*missing_n_peaks, y_ + [0]*missing_n_peaks
```

```
#calcula a fft e retorna os 10 primeiros picos
index = []
index_list = []
t_n = 10
N = 4500
T = t_n / N
f_s = 1/T

for column in tentativas[0]:
    f_values, fft_values =
get_fft_values(np.array(tentativas[4][column]), T, N, f_s)
    index.append(detect_peaks(fft_values, mph=0.1))
    index_list.append(get_first_n_peaks(fft_values, index[0], 10))
for i in range(67):
    print(index_list[i][0])
```

APÊNDICE C - Códigos utilizados para testes dos algoritmos de *Machine Learning*

```
#lê e formata um arquivo da nova base de dados
def readFiles(arquivo):
    dados = pd.read_excel(arquivo)
    dados = dados.drop('Indivíduo / Velocidade', 1)
    y_ = dados['Classificação']
    dados = dados.drop('Classificação', 1)
    X = np.array(dados)
    y = np.array(y_)
    return X, y

#treina o classificador ANN
def mlpScore(arquivo):
    X, y = readFiles(arquivo)
    clf = MLPClassifier()
    scores = cross_val_score(clf, X, y, cv=10, scoring='accuracy')
    print(scores.mean())
    with open('MLP', 'a') as f:
        f.write('\nFile: ' + str(arquivo.strip('./')))
        f.write('\nMean: ' + str(scores.mean()))

#treina o classificador SVM
def svmScore(arquivo):
    X, y = readFiles(arquivo)
    clf = svm.SVC(kernel='linear', C=1.0)
    scores = cross_val_score(clf, X, y, cv=10, scoring='accuracy')
    with open('SVM', 'a') as f:
        f.write('\nFile: ' + str(arquivo.strip('./')))
        f.write('\nMean: ' + str(scores.mean()))

#matriz de confusão do classificar ANN
def mlpConfusionMatrix(arquivo):
    X, y = readFiles(arquivo)
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.30, random_state=3)
    clf = MLPClassifier()
    clf.fit(X_train, y_train)
    print(confusion_matrix(y_test, clf.predict(X_test)))
```

```
#matriz de confusão do classificar SVM
def svmConfusionMatrix(arquivo):
    X, y = readFiles(arquivo)
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.30, random_state=3)
    clf = svm.SVC(kernel='linear', C=1.0)
    clf.fit(X_train, y_train)
    print(confusion_matrix(y_test, clf.predict(X_test)))
```