

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Lucas Raupp Hans

**Sistema de classificação de tipos de movimentos no esporte
tênis utilizando sensores inerciais e técnicas de
aprendizado de máquina supervisionado**

Porto Alegre
2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Lucas Raupp Hans

**Sistema de classificação de tipos de movimentos no esporte
tênis utilizando sensores inerciais e técnicas de
aprendizado de máquina supervisionado**

Projeto de Diplomação II, apresentado ao
Departamento de Engenharia Elétrica da Escola de
Engenharia da Universidade Federal do Rio Grande
do Sul, como requisito parcial para Graduação em
Engenharia Elétrica

UFRGS

Orientador: Prof. Dr. Tiago Weber

Porto Alegre
2022

LISTA DE FIGURAS

Figura 1 - Fases de execução do saque.....	14
Figura 2 - Fase de preparação (<i>forehand</i>).....	15
Figura 3 - Fase de aceleração (<i>forehand</i>).....	16
Figura 4 - Fase de contato (<i>forehand</i>).....	16
Figura 5 - Fase de finalização (<i>forehand</i>).....	17
Figura 6 - Fase de recuperação (<i>forehand</i>).....	17
Figura 7 - Técnica para execução do movimento do <i>backhand</i>	18
Figura 8 - Fase de preparação (<i>backhand</i>).....	19
Figura 9 - Fase de aceleração (<i>backhand</i>).....	19
Figura 10 - Fase de contato (<i>backhand</i>).....	20
Figura 11 - Fase de finalização (<i>backhand</i>).....	20
Figura 12 - Fase de recuperação (<i>backhand</i>).....	21
Figura 13 - Fluxo de um algoritmo supervisionado de aprendizado de máquina.....	24
Figura 14 - Exemplo de Árvore de Decisões simples.....	25
Figura 15 - Representação do modelo de SVM.....	27
Figura 16 - Abordagem One-to-One do SVM.....	28
Figura 17 - Abordagem One-to-Rest do SVM.....	29
Figura 18 - Exemplos de grades hiperparamétricas para uma, duas e três dimensões.....	32
Figura 19 - Procedimento geral da metodologia.....	39
Figura 20 - Diagrama de protótipo.....	40
Figura 21 - Esquemático do circuito.....	41
Figura 22 - Projeto da PCI.....	42
Figura 23 - Protótipo do sistema de aquisição e classificação de sinais inerciais e indicação da orientação dos eixos.....	42
Figura 24 - Sequência do movimento de saque.....	43
Figura 25 - Sequência do movimento de <i>backhand</i>	44
Figura 26 - Sequência do movimento de <i>forehand</i>	45
Figura 27 - Tela de execução da coleta de dados.....	46
Figura 28 - Tela de acesso aos resultados.....	47
Figura 29 - Representação do processo de pré processamento de geração das características.....	48
Figura 30 - Fluxograma do processo de detecção e classificação.....	53
Figura 31 - Representação de uma repetição de cada movimento.....	56
Figura 32 - Média e Desvio Padrão das características para aceleração.....	58
Figura 33 - Média e Desvio Padrão das características para velocidade angular.....	59
Figura 34 - Matriz de correlação das características escolhidas.....	61
Figura 35 - Relacionamento entre as amostras de cada característica.....	62
Figura 36 - Derivação Média de todos os movimentos coletados.....	63
Figura 37 - Matriz de Confusão dos modelos.....	65
Figura 38 - Curva ROC dos modelos.....	66
Figura 39 - Comparação da taxa de acerto entre outros trabalhos.....	69

LISTA DE TABELAS

Tabela 1 - Hiperparâmetros e parâmetros para alguns modelos de aprendizado de máquina..	31
Tabela 2 - Intervalos dos hiperparâmetros	51
Tabela 3 - Características obtidas para os métodos selectKBest, RFE e RFECV.....	60
Tabela 4 - Média da taxa de acerto das partições para o teste <i>cross_val_score</i>	64
Tabela 5 - Hiperparâmetros Escolhidos.....	65
Tabela 6 - Latência média de predição executada em <i>Python</i> com conjunto de dados teste ...	67
Tabela 7 - Latência média de predição no sistema embarcado com toda a base de dados.....	68
Tabela 8 - Tamanho dos modelos.....	68

SUMÁRIO

1. INTRODUÇÃO	9
1.1 JUSTIFICATIVA.....	9
1.2 PROBLEMA.....	10
1.3 OBJETIVO GERAL	10
1.4 OBJETIVOS ESPECÍFICOS	10
2. FUNDAMENTAÇÃO TEÓRICA.....	11
2.1 TRABALHOS RELACIONADOS.....	11
2.2 CARACTERIZAÇÃO DOS MOVIMENTOS.....	12
2.2.1 <i>Descrição da Técnica de Movimento Saque</i>	13
2.2.2 <i>Descrição da Técnica de Movimento Forehand</i>	15
2.2.3 <i>Descrição da Técnica de Movimento Backhand</i>	18
2.3 DETECÇÃO DE MOVIMENTOS COM SENSORES INERCIAIS	21
2.3.1 <i>Acelerômetro, Giroscópio e Sensores Magnéticos</i>	21
2.3.2 <i>Requisitos para medição de movimentos em esportes</i>	22
2.3.3 <i>Parâmetros obtidos dos sensores</i>	23
2.4 APRENDIZADO DE MÁQUINA	24
2.4.1 <i>Aprendizado Supervisionado</i>	24
2.4.1.1 <i>Árvore de Decisões</i>	25
2.4.1.2 <i>Florestas Aleatórias</i>	26
2.4.1.3 <i>Support Vector Machine</i>	26
2.4.1.3.1 <i>Classificação SVM com Múltiplas Classes</i>	27
2.5 DETECÇÃO DE MOVIMENTOS DE TÊNIS	29
2.6 CARACTERÍSTICAS UTILIZADAS PARA CLASSIFICAÇÃO DE MOVIMENTOS	29
2.7 ESCOLHA DE CARACTERÍSTICAS	30
2.7.1 <i>Relevância das Características</i>	31
2.8 OTIMIZAÇÃO DE HIPERPARÂMETROS	31
2.8.1 <i>Otimização com Busca de Grade</i>	32
2.9 APRENDIZADO DE MÁQUINA EM SISTEMAS EMBARCADOS.....	33
2.9.1 <i>Poda computacional (pruning)</i>	34
3. METODOLOGIA.....	35
3.1 MATERIAIS E FERRAMENTAS.....	35
3.1.1 <i>Hardware Utilizado</i>	35
3.1.1.1 ESP32	35
3.1.1.2 GY-521	35
3.1.1.3 TP4056	36
3.1.2 <i>Ferramentas Computacionais</i>	37
3.1.2.1 <i>Plataforma Arduino</i>	37
3.1.2.2 <i>Adafruit MPU6050</i>	37
3.1.2.3 <i>Google Firebase</i>	37
3.1.2.4 <i>React</i>	38

3.1.2.5 Scikit-learn	38
3.1.2.6 MicroML generator	38
3.2 PROCEDIMENTO	39
3.2.1 <i>Desenvolvimento do Protótipo</i>	40
3.2.1.2 Projeto do protótipo	40
3.2.2 <i>Coleta da Base de Dados</i>	42
3.2.2.1 Ferramenta para aquisição de dados	46
3.2.3 <i>Pré Processamento dos Dados e Geração das Características</i>	47
3.2.3.1 Características Geradas.....	48
3.2.3.2 Separação em Conjuntos Treinamento e Teste.....	49
3.2.4 <i>Escolha das Melhores Características</i>	49
3.2.5 <i>Otimização dos Hiperparâmetros</i>	50
3.2.6 <i>Método de Detecção dos Movimentos</i>	51
3.2.7 <i>Implementação do Modelo no Sistema Embarcado</i>	52
3.2.7.1 Detecção e Classificação dos movimentos.....	52
3.2.8 <i>Análise de Desempenho de cada Modelo</i>	54
4. ANÁLISE DE RESULTADOS.....	56
4.1 ESTUDO DA BASE DE DADOS.....	56
4.2 ESCOLHA DAS CARACTERÍSTICAS	60
4.3 DEFINIÇÃO DOS PARÂMETROS DE DETECÇÃO	63
4.4 OTIMIZAÇÃO DOS HIPERPARÂMETROS E GERAÇÃO DOS MODELOS	63
4.4.1 <i>Otimização e Avaliação dos modelos</i>	64
4.4.2 <i>Avaliação da Latência e Tamanho dos Modelos</i>	67
4.5 COMPARAÇÃO COM TRABALHOS DA LITERATURA	69
5. CONCLUSÕES.....	70
5.1 TRABALHOS FUTUROS.....	70
REFERÊNCIAS BIBLIOGRÁFICAS	72

RESUMO

A classificação automática de movimentos (golpes) de tênis para a geração de dados relacionados ao desempenho durante a prática esportiva pode auxiliar atletas no aprimoramento dos movimentos e direcionamento de estratégias de treinamento. A presente pesquisa detalha o processo de desenvolvimento de um dispositivo vestível, que é acoplado ao pulso do tenista, e se propõe a detectar e classificar os três principais movimentos do tênis: saque, *forehand* e *backhand*. O dispositivo é composto, principalmente, pelos módulos *ESP-WROOM-32D* e *MPU-6050*, possuindo o segundo dois sensores: um acelerômetro e um giroscópio. Ambos sensores possuem três eixos, utilizados para mensurar aceleração e velocidade angular durante a prática do esporte. Tanto os sensores não foram caracterizados isoladamente quanto o sistema, restringindo a extrapolação das análises feitas para outros sistemas e necessitando ainda que os resultados do sistema de medidas sejam validados. Uma base de dados, composta pelos três movimentos, foi coletada pelo autor durante a pesquisa e avaliada em relação a 90 características que representam cada um dos golpes estudados. Para classificação, as melhores características foram selecionadas com o uso de *RFECV (Recursive Feature Elimination with cross-validation)* e três modelos de aprendizado de máquina supervisionado (Árvore de Decisão, Floresta Aleatória e SVM) foram otimizados com o uso de busca de grade, comparados e avaliados. Todos os modelos apresentaram assertividade média de 100 % na classificação dos movimentos propostos para os testes realizados, com uma latência média no sistema embarcado de **2384 ± 181 us** para o modelo SVM (melhor resultado). Para os modelos de Árvore de Decisão e Floresta Aleatória, com a otimização dos hiperparâmetros, foi obtido um valor de 2 para a profundidade das árvores, indicando uma baixa complexidade dos modelos e uma boa capacidade das características escolhidas em diferenciar os movimentos. Todos os modelos se mostraram capazes de classificar, em tempo real, os movimentos propostos.

Palavras-chave: Tênis, Aprendizado de Máquina, Acelerômetro, Giroscópio.

ABSTRACT

The automatic classification of tennis strokes for the generation of data related to the performance during sports practice can help athletes in the improvement of movements and targeting training strategies. The present research details the development process of a wearable device, which is attached to the tennis player's wrist, and proposes to detect and classify the three main tennis movements: serve, forehand and backhand. The device is mainly composed by the modules *ESP-WROOM-32D* and *MPU-6050*, the second having two sensors: an accelerometer and a gyroscope. Both Sensors have three axes, used to measure acceleration and angular velocity during the practice of the sport. Both the sensors were not characterized in isolation as well as the device, restricting the extrapolation of the analyzes made to other systems and still requiring the results of the measurement system to be validated. A database, composed by the three movements, was collected by the author during the research and evaluated based on 90 characteristics that represent each of the studied strokes. For classification, the best features were selected using RFECV (Recursive Feature Elimination with cross-validation) and three supervised machine learning models (Decision Tree, Random Forest and SVM) were optimized using grid search, compared and evaluated. All models showed an accuracy of 100 % for the movements classification in the executed tests, with an average latency in the embedded system of $2.38 \pm 0.18 \text{ ms}$ for the SVM model (best result). For the Decision Tree and Random Forest models, with the hyperparameters optimization, a value of 2 was obtained for the trees depth, indicating a low complexity of the models and a good capacity of the chosen characteristics to differentiate the movements. All models were able to classify, in real time, the proposed movements.

Keywords: Tennis, Machine Learning, Accelerometer, Gyroscope.

1. INTRODUÇÃO

A utilização de sensores durante a prática de tênis auxilia os jogadores a analisar técnicas, aprimorar treinamentos e prevenir lesões. O uso da tecnologia na prática esportiva vem em uma grande crescente e, no tênis, já é permitida mesmo durante competições (CAMOMILLA et al., 2018).

Em partidas envolvendo atletas de alto rendimento já são utilizados sistemas de vídeo *Hawk-Eye*, que se baseiam em várias câmeras de alta velocidade posicionadas ao redor da quadra e processamento de vídeo. Este tipo de sistema é capaz de marcar bolas dentro e fora, prover replays instantâneos para os árbitros e possibilitar o desafio em marcações por parte dos jogadores. Além disso, o sistema armazena dados de jogadores, suporta acompanhamento da trajetória da bola e análise dos golpes executados (HAWKEYE, 2016). Porém, este tipo de tecnologia é extremamente cara e, por isso, é somente utilizada em uma exclusiva parcela dos eventos profissionais (ESPINOSA et al., 2015).

Movimentos de tênis, mais conhecidos na literatura como golpes de tênis, são, preferencialmente, detectados a partir de computação gráfica ou sensores de movimento (ZHAO et al., 2017). O primeiro é em geral muito caro em relação a outras abordagens com equipamentos mais simples e, pouco acessível. Já o segundo, é amplamente utilizado em vários esportes, tem grande potencial para classificação de movimentos e no tênis é comumente acoplado a raquete ou até mesmo ao pulso do jogador (KOS e KRAMBERGER, 2017).

Considerando as tecnologias atuais empregadas para aquisição de dados durante práticas esportivas, o presente trabalho se propõe a buscar uma solução menos complexa e acessível, capaz de classificar os golpes de tênis e prover dados para auxiliar no treinamento e evolução de atletas.¹

1.1 JUSTIFICATIVA

Para partidas profissionais de tênis, geralmente têm-se algumas estatísticas disponíveis, como: número de aces alcançados, pontos em primeiro e segundo saque, pontos ganhos próximo a rede, bolas vencedoras, erros não forçados, etc. A partir destes dados, que treinadores e especialistas podem conseguir informações valiosas sobre o jogo e identificar pontos fortes e fracos dos atletas (KOS et al., 2016).

¹ Não foi realizada a caracterização dos sensores utilizados tanto de forma isolada quanto no sistema. Desta forma se faz ainda necessária a validação dos dados obtidos dos sensores.

Segundo Ebner e Findling (2019), a utilização de sensores vestíveis podem ajudar na análise de sua técnica ao executar os golpes, aprimorar a efetividade e direcionar estratégias para aplicar em treinamentos e até mesmo auxiliar na prevenção de lesões. Este tipo de sensor vem sendo cada vez mais integrado em aplicações esportivas, possuem um grande potencial para detecção de movimentos e já se provaram efetivos em análises de biomecânica esportiva.

Quantificar o trabalho executado por um atleta é fundamental para o processo de treinamento em esportes de alto desempenho. No tênis, contabilizar os golpes realizados e também o tipo de cada golpe é uma tarefa importante para tomadas de decisão e evolução dos atletas. O padrão atual para contagem de golpes é realizado manualmente, o que requer tempo, está sujeito a erro humano e pode até mesmo não ser viável em determinados casos. (EBNER e FINDLING, 2019).

A detecção e classificação de golpes de tênis pode também ser definida como um problema de reconhecimento de gestos. Este é um campo importante na área de HCI (*Human-computer interaction*), a qual ajuda no aprimoramento e criação de novos métodos de comunicação que são próximos à natureza humana (KOS et al., 2016).

1.2 PROBLEMA

É possível desenvolver um dispositivo capaz de identificar e classificar os golpes de tênis saque, *forehand* e *backhand*, através de sensores inerciais acoplados ao pulso do jogador e técnicas de aprendizado de máquina supervisionado?

1.3 OBJETIVO GERAL

Desenvolver um dispositivo capaz de identificar e classificar os golpes de tênis saque, *forehand* e *backhand*, através de sensores inerciais acoplados ao pulso do jogador e técnicas de aprendizado de máquina supervisionado.

1.4 OBJETIVOS ESPECÍFICOS

- Desenvolver um sistema de aquisição de uma base de dados para treinamento;
- Realizar treinamento do sistema de classificação sem considerar recursos limitados de hardware;
- Implementar e testar o sistema para análise de assertividade do protótipo.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados e discutidos conteúdos relacionados e necessários para a compreensão da pesquisa. Serão apresentados fundamentos utilizados para classificação de movimentos durante a prática esportiva, técnicas de aprendizado de máquina que podem auxiliar neste processo e um estudo sobre as técnicas de execução dos movimentos propostos para classificação.

2.1 TRABALHOS RELACIONADOS

No contexto de dispositivos vestíveis, existem algumas diferentes soluções para classificação de movimentos de tênis: Sensor integrado ao cabo da raquete; Sensor acoplado às cordas da raquete; Sensor acoplado ao *grip* da raquete; Sensor acoplado ao pulso do jogador.

A primeira opção é um sistema desenvolvido pela empresa *Babolat*, chamado de *Babolat PLAY*. Esta é também opção mais cara, já que o jogador precisa comprar uma raquete que já possui o sensor integrado. A raquete registra e classifica os diferentes golpes (saque, *smash*, *forehand*, *backhand*, voleio) através de sensores como acelerômetro, giroscópio e piezoelétrico, e sincroniza dados com dispositivos móveis para visualização de estatísticas em partidas e treinamentos (BABOLAT, 2015). O dispositivo possui até 6 horas de autonomia, mas não há muita informação sobre o design do sistema. Outro dispositivo desenvolvido pela mesma empresa é o *Babolat POP*, que possui performance similar ao *Babolat PLAY*, porém é um dispositivo que é acoplado ao pulso do jogador (BABOLAT, 2015).

Cannaghan et al. (2020) utilizam um sistema de reconhecimento e classificação dos golpes de tênis baseado em uma câmera de baixo custo. A pesquisa se propõe a identificar os três principais golpes do tênis (saque, *forehand*, *backhand*) e foi testada com uma base de dados obtida com jogadores durante partidas competitivas.

Kos et al. (2016) propõem um dispositivo em miniatura, posicionado no antebraço do jogador. O sistema utiliza um acelerômetro e um giroscópio microeletromecânico (MEMS) com 6 graus de liberdade. Através do processamento dos dados fornecidos pelos dois sensores, são classificados os três movimentos mais comuns do tênis (saque, *forehand*, *backhand*). Devido a uma memória limitada e um fraco poder de processamento, é proposto um algoritmo de classificação bastante simples, mas que consegue entregar o que se é proposto.

Uma série de trabalhos vêm utilizando sensores inerciais para detecção de gestos e classificação de movimentos no tênis (KOS et al., 2016; EBNER e FINDLING, 2019; PARDO et al, 2019). Muitos destes trabalhos combinam diferentes sensores posicionados de diferentes

formas, como por exemplo raquete e pulso do jogador. Ebner e Findling (2019) trazem uma comparação entre o posicionamento de sensores no pulso, na raquete e em ambos. Na pesquisa são investigados a detecção e classificação de 8 tipos de golpes a partir de medições de aceleração e velocidade angular. Para detecção dos golpes são utilizados valores de pico com limites pré definidos e janelamento, já para o reconhecimento dos mesmos diferentes conjuntos de recursos e modelos de classificação são aplicados. Os resultados mostram uma performance bastante similar entre o posicionamento no pulso ou na raquete, sendo ambos capazes de atingir resultados satisfatórios.

2.2 CARACTERIZAÇÃO DOS MOVIMENTOS

Para diferenciar golpes de tênis, diferentes tipos de golpes são considerados na literatura. Os principais são o saque, *forehand* e o *backhand* (EBNER e FINDLING, 2019). Já Whiteside et al. (2017) estabelecem oito conhecidos tipos de golpes: saque, *forehand*, *forehand slice*, *forehand volley*, *backhand*, *backhand slice*, *backhand volley*, e *smash*, sendo que podem haver variações na execução de cada um destes movimentos.

Durante as partidas o movimento da raquete é controlado pelo tenista, porém a bola, que é golpeada pelo adversário, chega na quadra em diferentes velocidades, rotações, alturas, distâncias e profundidades, o que dificulta a execução e sincronização dos golpes. Salienta-se que entre os golpes de tênis, apenas durante a execução do saque o jogador possui controle total sobre a bola antes de golpeá-la (BRAGA NETO, 2003).

Segundo Braga Neto (2008) o *forehand* e o *backhand* são os golpes técnicos mais utilizados em uma partida de tênis. São chamados de golpes de base ou golpes de fundo, pois na maioria das vezes são executados próximos à linha do fundo da quadra, após o toque da bola com o solo. No Brasil, inicialmente o *forehand* foi chamado de golpe de direita e o *backhand* de golpe de esquerda, porém tais denominações podem causar confusões dependendo da dominância do tenista. O *forehand* é o golpe executado do mesmo lado do corpo no qual se segura a raquete, ficando com a palma da mão apontada para a rede. Para o caso de um tenista canhoto, pode-se gerar certa incoerência dizer que seu golpe de direita foi executado do lado esquerdo do corpo, então é convencional que se siga a nomenclatura em inglês (*forehand* e *backhand*).

2.2.1 Descrição da Técnica de Movimento Saque

Segundo Kruger (2014), o saque é o golpe mais complexo no jogo de tênis e é o único golpe onde o jogador possui total controle sobre o mesmo, sendo um bom saque aquele que tiver uma boa combinação de consistência, precisão, variação, potência e o disfarce do golpe. Cada vez mais os saques indefensáveis (*aces*) estão presentes nos jogos de tênis, o que motiva o interesse em estudar este fundamento que é determinante no rendimento do praticante (BRAGA NETO, 2003).

Ao longo dos anos o saque apresentou muitas mudanças técnicas, porém a forma como se segura a raquete (empunhadura), no geral, não foi alterada. A maioria dos tenistas utilizam a empunhadura chamada de *continental* (KRUGER, 2014).

Kruger (2014), na busca de definir os melhores métodos para o ensino de tênis, explica que este movimento pode ser dividido em cinco fases: preparação, *backswing* e *toss*, laçada, contato e finalização.

Preparação: Fase em que o tenista se prepara para iniciar o movimento de saque, mantendo uma base confortável das pernas para uma boa estabilização e balanço para a fase seguinte.

Backswing e Toss: Nesta fase deve ser executado um movimento sincronizado, enquanto o braço dominante inicia um movimento circular para trás, chamado de *backswing*, o braço não dominante realiza um movimento vertical de forma a conduzir a bola para o lançamento, chamado de *toss*. Para o lançamento da bola ser feito corretamente, o ideal é que não sejam feitos movimentos nas articulações do punho e cotovelo. Depois de feito o lançamento, é iniciado um movimento de flexão e extensão das pernas.

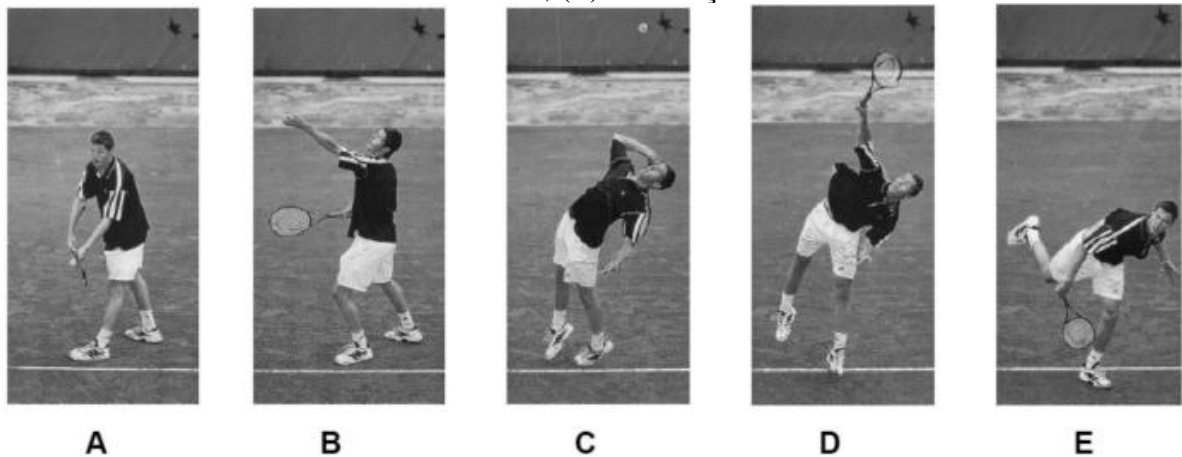
Laçada: Nesta fase a raquete se encontra atrás das costas. Enquanto a cabeça da raquete realiza um movimento descendente, os pernas realizam um movimento de extensão em direção à bola, seguido do momento de aceleração da raquete para frente e para cima em direção à bola.

Contato: Nesta fase ocorre o contato da raquete com a bola. O contato deve ocorrer no ponto mais alto possível.

Finalização: Depois de realizar o contato da raquete com a bola, a fase de finalização continua-se o movimento com a rotação interna do braço e a pronação do antebraço, seguindo a raquete na direção oposta do corpo.

A Figura 1 mostra as diferentes fases de execução do movimento de saque.

Figura 1 - Fases de execução do saque: (A) preparação; (B) *backswing e toss*; (C) laçada; (D) contato; (E) finalização.



Fonte: Braga Neto (2003).

O movimento do saque pode também sofrer algumas variações. Elliot e Wood (1983), dividiram duas diferentes técnicas de saque baseadas no posicionamento dos pés: *foot-back* e *foot-up*.

- *Foot-back*: no início do movimento a perna de trás fica afastada e esta distância é mantida até o início da fase aérea (entre contato e finalização), o que produz uma grande força horizontal;
- *Foot-up*: no início do movimento a perna de trás também fica mais afastada, porém durante a execução do saque é feita uma aproximação da perna de trás em relação a perna da frente, o que permite uma maior altura de contato com a bola.

Além disso, é possível diferenciar o saque pelo tipo de efeito e rotação empregado na bola (KRUGER, 2014).

- *Flat*: Também chamado de saque chapado, não possui rotação ou efeito e permite maiores velocidades;
- *Kick*: Este saque possui mais efeito, com a rotação da bola de baixo para cima, sendo um saque mais lento;
- *Slice*: Este saque possui um efeito lateral, com velocidades intermediárias em relação aos outros dois.

2.2.2 Descrição da Técnica de Movimento *Forehand*

Hofer (2004) descreve o conceito de cadeia cinética aplicada ao golpe *forehand*: na fase de preparação (*backswing*), o tenista (destro) concentra o peso do corpo sobre o pé direito, flexiona os joelhos e realiza a rotação do quadril para trás, posicionando-se de lado para a rede. A partir de então é iniciado a condução da raquete em direção à bola (*forward swing*) com a extensão dos joelhos. Em seguida é executada a rotação do quadril para frente e depois a rotação dos ombros. O braço e a raquete são os últimos componentes a serem acionados e para que o movimento seja eficiente, os tempos de cada segmento corporal devem ser respeitados, de forma a estarem sincronizados entre si e em relação ao movimento da bola.

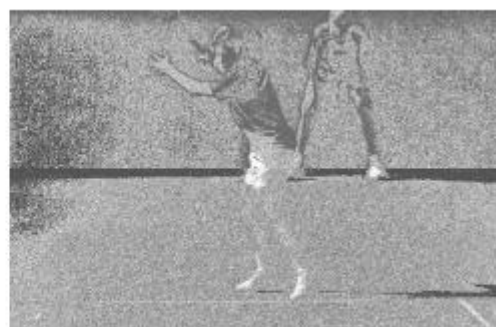
Grabb (2003) e De Mora (2002) definem variações nas técnicas do movimento de *forehand*, podendo este golpe ser executado como um *forehand open stance* ou *forehand square stance*, ambas divididas em cinco fases durante sua execução: preparação (*backswing*), aceleração (*forward swing*), contato, finalização (*follow-through*) e recuperação. As fases de execução são apresentadas a seguir, considerando-se um tenista destro.

Preparação: Também chamada de *backswing*, na qual a raquete é conduzida para trás. Nesta fase é executada a rotação do quadril para trás, posicionando o jogador lateralmente em relação à rede. Para o *forehand open stance* o executante posiciona as pernas paralelamente em relação à rede. Já para o *forehand square stance* o executante posiciona as pernas perpendicularmente em relação à rede. Para ambas as variações, o braço esquerdo é posicionado à frente do corpo para dar equilíbrio, a cabeça da raquete é posicionada acima da altura da cabeça e o apoio dos pés é concentrado sobre o pé direito.

Figura 2 - Fase de preparação: (A) *forehand open stance* (B) *forehand square stance*.



A

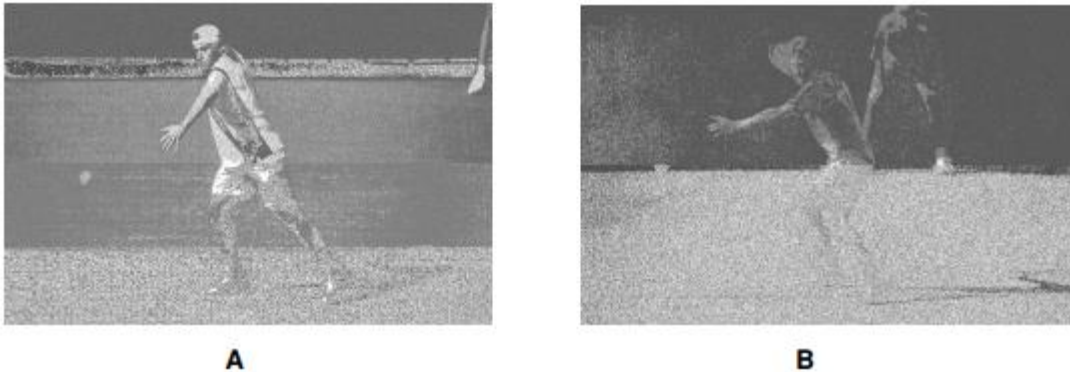


B

Fonte: (A) Grabb (2003); (B) De Mora (2002).

Aceleração: Conhecida também como *forward swing*, nesta fase o jogador leva a raquete para frente, em direção à bola, através também da rotação do quadril. O braço esquerdo é conduzido mais à frente do corpo para auxiliar na rotação do quadril, a cabeça da raquete é posicionada na altura dos ombros e o apoio se mantém sobre o pé direito.

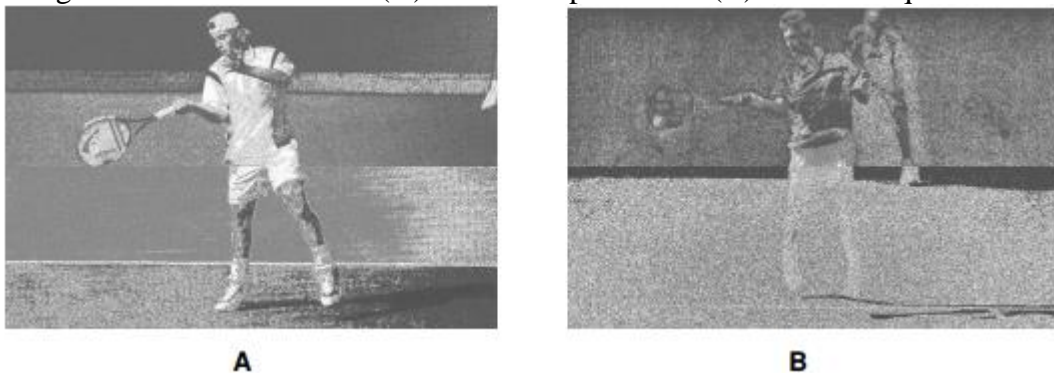
Figura 3 - Fase de aceleração: (A) *forehand open stance* (B) *forehand square stance*.



Fonte: (A) Grabb (2003); (B) De Mora (2002).

Contato: Fase em que é realizado o contato da raquete com a bola. O quadril é rotacionado de forma a posicionar o jogador novamente de frente para a rede. O contato, normalmente, ocorre próximo à altura da cintura, porém pode depender da altura que se encontra a bola. Caso o contato seja feito acima desta altura, o jogador pode perder o contato dos pés com o solo.

Figura 4 - Fase de contato: (A) *forehand open stance* (B) *forehand square stance*.

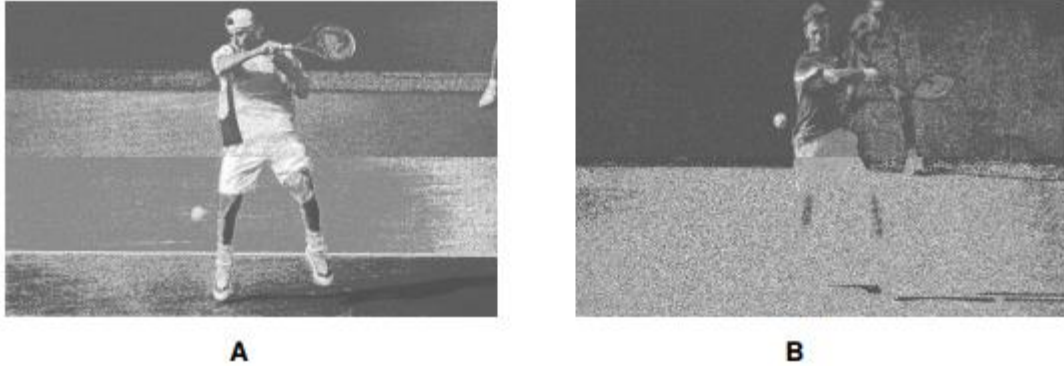


Fonte: (A) Grabb (2003); (B) De Mora (2002).

Finalização: Esta fase, também chamada de *follow-through*, ocorre após o contato com a bola. A raquete sofre uma desaceleração e ocorre a rotação interna do ombro, finalizando com a raquete do lado oposto em que ocorreu o contato com a bola. A mão esquerda passa a segurar

a raquete, auxiliando a conduzi-la novamente à posição inicial. Assim como na fase de contato, os pés podem perder o contato com o solo a depender da altura do contato e da potência utilizada durante a execução do golpe.

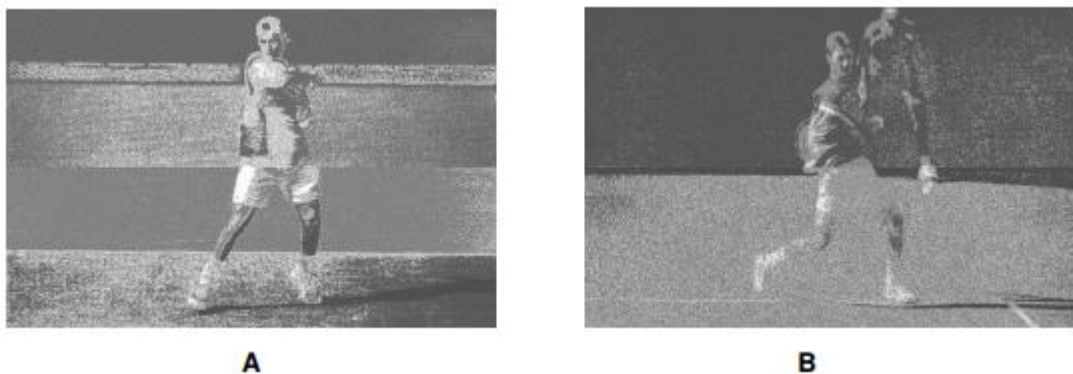
Figura 5 - Fase de finalização: (A) *forehand open stance* (B) *forehand square stance*.



Fonte: (A) Grabb (2003); (B) De Mora (2002).

Recuperação: Nesta fase o apoio passa a ser feito sobre o pé esquerdo. A partir de então o tenista passa a se movimentar de forma a buscar o centro da quadra para dar seguimento ao jogo.

Figura 6 - Fase de recuperação: (A) *forehand open stance* (B) *forehand square stance*.



Fonte: (A) Grabb, (2003); (B) De Mora (2002).

As duas primeiras fases (preparação e aceleração) são as que caracterizam a técnica do movimento: *forehand open stance* e *forehand square stance*. O movimento de preparação é o que define a posição dos pés, acomodando o quadril e caracterizando ambas as técnicas.

Além do tipo de preparação, o *forehand* pode ser classificado quanto à rotação transferida para a bola, sendo elas: *flat*, *underspin* e *topspin*.

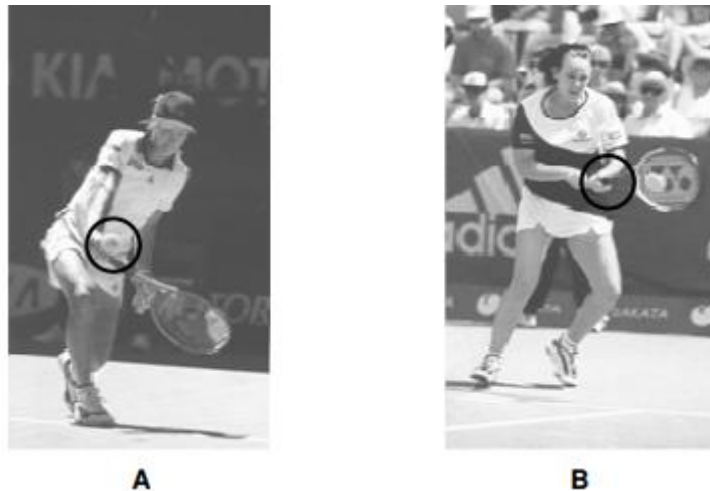
- *Flat*: Não é imprimida uma rotação significativa na bola.

- *Underspin*: Rotação gerada pelo contato das cordas da raquete com e a parte inferior da bola. Essa variação é conhecida também como *slice*.
- *Topspin*: Rotação gerada pelo contato das cordas da raquete com a parte superior da bola

2.2.3 Descrição da Técnica de Movimento *Backhand*

O golpe de *backhand*, o qual é realizado com a parte de trás da mão apontada para a rede, pode ser executado de duas diferentes formas: com uma ou duas mãos segurando o cabo da raquete no momento do contato com a bola (Figura 7).

Figura 7 - Técnica para execução do movimento do backhand: (A) com uma mão (B) com duas mãos.



Fonte: (A) Shiras (2003); (B) Rolley (1999).

Lansdorp (2004) e Trabert (1999) apresentam as técnicas *backhand* com uma mão (BH1) e *backhand* com duas mãos (BH2), como elas sendo também divididas em cinco fases: preparação (*backswing*), aceleração (*forward swing*), contato, finalização (*follow-through*) e recuperação. As fases de execução são apresentadas a seguir, considerando-se um tenista destro.

Preparação: Nesta fase a raquete é conduzida para trás. Em BH1 as mãos estão separadas e em BH2 as mãos estão juntas. Nesta fase ocorre a rotação do quadril para trás, posicionando o tenista lateralmente em relação à rede.

Figura 8 - Fase de preparação: (A) com uma mão (B) com duas mãos.



A



B

Fonte: (A) Lansdorp (2004); (B) Trabert (1999).

Aceleração: Nesta fase o jogador conduz a raquete para frente, em direção à bola. Durante o movimento é realizada a rotação do quadril para frente. O apoio é feito sobre o pé direito, que fica posicionado um pouco mais à frente. A cabeça da raquete é movimentada até a linha da cintura. Comparando as duas técnicas durante a aceleração, em BH2 é feita uma maior flexão do quadril e joelho.

Figura 9 - Fase de aceleração: (A) com uma mão (B) com duas mãos.



A



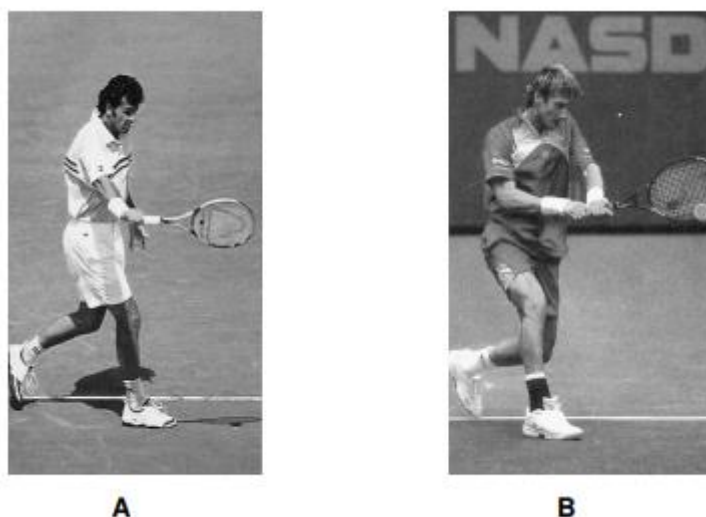
B

Fonte: (A) Lansdorp (2004); (B) Trabert (1999).

Contato: Esta é a fase em que mais se evidencia a diferença das duas técnicas. Nesta fase ocorre o contato da bola com a raquete. Para BH2 a rotação do quadril é mais acentuada,

ficando o tenista já posicionado de frente para a rede e provocando uma maior concentração de peso sobre o pé direito. Para BH1, durante o contato o tenista ainda se encontra posicionado lateralmente em relação à rede.

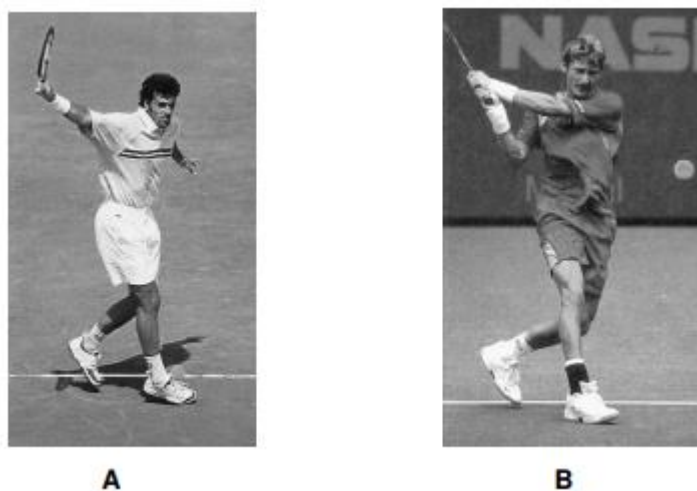
Figura 10 - Fase de contato: (A) com uma mão (B) com duas mãos.



Fonte: (A) Lansdorp (2004); (B) Trabert (1999).

Finalização: Nesta fase é feita a desaceleração da raquete. Para BH1 a rotação do quadril ainda não posiciona o jogador de frente para a rede. Para BH2 é evidenciada uma maior rotação do tronco, posicionando o tenista novamente de frente para a rede. Em ambas as técnicas, a cabeça da raquete já está posicionada do lado oposto ao qual foi realizado o contato.

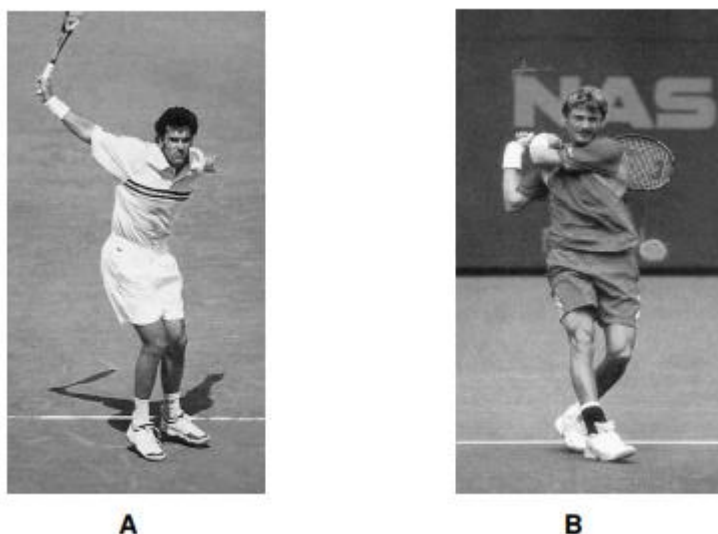
Figura 11 - Fase de finalização: (A) com uma mão (B) com duas mãos.



Fonte: (A) Lansdorp (2004); (B) Trabert (1999).

Recuperação: Tanto para BH1 quanto BH2, nesta fase o corpo continua apoiado sobre o pé direito, o que serve de apoio para impulsionar o jogador a voltar para o centro da quadra.

Figura 12 - Fase de recuperação: (A) com uma mão (B) com duas mãos.



Fonte: (A) Lansdorp (2004); (B) Trabert (1999).

Assim como na execução do *forehand*, é possível imprimir à bola os mesmos tipos de rotação: *flat*, *underspin* e *topspin*.

2.3 DETECÇÃO DE MOVIMENTOS COM SENSORES INERCIAIS

Nesta seção serão abordadas algumas tecnologias de sensores inerciais vestíveis aplicáveis em sistemas de análise de movimentos de atletas, requisitos para garantir seus usos em aplicações esportivas e os parâmetros providos pelos sensores para avaliação dos movimentos executados.

2.3.1 Acelerômetro, Giroscópio e Sensores Magnéticos

Com o surgimento da tecnologia de Sistemas Microeletromecânicos (MEMS), foi permitida a fabricação de giroscópios, acelerômetros e sensores magnéticos, que são sensores bastante miniaturizados, com um baixo custo e baixo consumo de energia.

Os acelerômetros, giroscópios e sensores magnéticos são sensores capazes de quantificar as acelerações gravitacionais e inerciais, a velocidade angular e os componentes do vetor campo magnético local, em torno de seus eixos, respectivamente. Para o caso em que se

os sensores são de três eixos, os mesmos são ortogonais uns aos outros, sendo as quantidades físicas mencionadas anteriormente medidas em relação a um sistema de coordenadas cartesianas.

As medidas fornecidas pelos três sensores também podem ser usadas para obter informações sobre a orientação de um corpo, porém quando a orientação é fornecida por apenas um tipo de sensor por vez, a precisão da orientação é frequentemente inadequada para determinadas aplicações (SABATINI, 2011). Portanto, para melhorar a precisão e aprimorar a medida da orientação do corpo, giroscópios e acelerômetros são comumente combinados em uma Unidade de Medição Inercial (IMU), sendo ambos sensores usados de forma a explorar o princípio da inércia para fornecer acelerações e/ou velocidades angulares (LOUINGE, 2005). Quando a orientação 3D é necessária, os sensores magnéticos também são incorporados na Unidade de Medição Magnética e Inercial, amplamente conhecida pela sigla MIMU (ou Unidade de Medição Inercial e Magnética, IMMU, ou Sistema de Medição Inercial e Magnética, IMMS). Quando IMUs e MIMUs são usados em conjunto, a orientação da unidade é comumente fornecida como uma saída do dispositivo, conforme estimado pela combinação dos dados medidos por dois ou três sensores, por meio de algoritmos de fusão de sensores (CAMOMILLA et al, 2018).

2.3.2 Requisitos para medição de movimentos em esportes

Diversos giroscópios, acelerômetros e sensores magnéticos, separadamente ou em forma de IMUs e MIMU, vem sendo comercializados nos últimos anos para uso na análise do movimento humano. As características desses dispositivos em termos de escala, frequência de amostragem, consumo de energia e portabilidade variam de acordo com a aplicação e a tecnologia usada para transmissão de dados (cabeadas ou sem fio). Quando sua aplicação está direcionada a alguma atividade esportiva específica, deve-se tomar cuidado para selecionar adequadamente o sistema de medição, considerando as características dos movimentos que serão exercidos (CAMOMILLA et al, 2018). Outras características, como repetibilidade, *feedback* compreensível e comunicável aos atletas, resultados em tempo real e adaptação para análise de múltiplos atletas são requisitos para esse tipo de sistema (LE SAGE et al, 2011). De forma geral, o sistema usado também deve ser pequeno e leve, se tornando portátil para o uso em campo; vestível e sem fio; de fácil e rápida de configuração e análise; se mostrar resistente a água na maioria das aplicações e, em casos como a natação, suportar imersão completa do dispositivo na água; possuir uma boa eficiência energética, principalmente para atividade de

longo período de duração; e ter escala e frequência de amostragem adequada para o tipo de movimento a ser medido, sendo a frequência de amostragem de no mínimo duas vezes a frequência do sinal de interesse, respeitando o teorema de amostragem de Nyquist-Shannon (MANIKANDAN et al., 2021).

Além das características do dispositivo selecionado, devem-se também ser consideradas possíveis fontes de erro, as quais podem em vezes limitar a gama de possíveis aplicações. Um exemplo de possível fonte de erro seriam oscilações entre o corpo do atleta e o dispositivo, podendo ser causado pelo tipo de tecidos utilizados ou até mesmo a técnica de fixação escolhida. A compensação desses erros pode se fazer necessária em muitos casos e, como essas fontes podem ser diferentes para cada usuário, bem como a forma como o sistema é removido/substituído, isso pode vir a se tornar um problema crítico (CAMOMILLA et al, 2018).

2.3.3 Parâmetros obtidos dos sensores

Dados de sensores como giroscópios, acelerômetros e sensores magnéticos podem ser usados para estimar diferentes parâmetros. Estes podem ser baseados na detecção de características nos sinais medidos ou em técnicas de processamento mais sofisticadas, que, por exemplo, permitem que as informações fornecidas por dois ou mais sensores sejam combinadas. Para parâmetros chamados de temporais, tem-se tipicamente os obtidos a partir da detecção da velocidade angular ou aceleração medida, como máximos, mínimos ou mudanças de inclinação, podendo determinar algumas características relacionadas ao movimento, como por exemplo: momento de início ou fim de um movimento, intensidade do movimento e frequência de movimentos cíclicos. Já para parâmetros cinemáticos, que se referem a formas angulares e lineares de posição, velocidade e aceleração, em particular, a estimativa da orientação 3D de um MIMU em relação a um quadro de referência inercial é de importância fundamental, pois é essencial ao obter a orientação absoluta de segmentos corporais ou equipamentos esportivos, ou a orientação de um segmento do corpo em relação a outro (cinemática da junta). Além disso, é necessário separar as componentes gravitacionais e inerciais das acelerações lineares, de acordo com os parâmetros cinemáticos lineares, como a posição MIMU ou a velocidade linear (CAMOMILLA et al, 2018).

2.4 APRENDIZADO DE MÁQUINA

O aprendizado de máquina é composto por um conjunto de técnicas que aprendem com sua experiência, analisando suas observações passadas, utilizando modelos de como suas ações anteriores foram realizadas e a partir dessas avaliações tomar melhores decisões no futuro (HAIGH et al. 2015). Em resumo, o aprendizado de máquina pode ser definido como o campo de estudo que dá aos computadores a habilidade de aprender sem ser explicitamente programado.

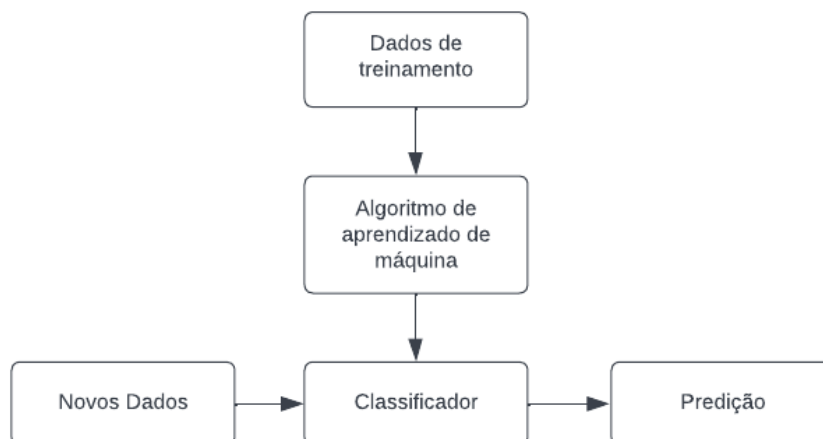
O aprendizado de máquina depende de diferentes algoritmos para resolver problemas de dados. O tipo de algoritmo empregado depende do tipo de problema que se deseja resolver, o número de variáveis, o tipo de modelo que melhor se adequa a ele e assim por diante.

2.4.1 Aprendizado Supervisionado

O aprendizado supervisionado é uma técnica de aprendizado de máquina que gera uma função que mapeia uma entrada para uma saída baseada em exemplos de pares de entrada-saída. Ele infere uma função a partir de dados previamente rotulados, contidos em um conjunto de exemplos de treinamento. Os algoritmos de aprendizado supervisionado são aqueles que necessitam de um auxílio externo. O conjunto de dados de entrada é dividido em dados de treinamento, os quais já possuem suas variáveis de saída definidas, e dados de teste (NASTESKI, 2017).

Todos os algoritmos aprendem alguns padrões da base de dados de treinamento e, então, os aplicam na base de dados de teste para classificação de predição.

Figura 13 - Fluxo de um algoritmo supervisionado de aprendizado de máquina.



Fonte: Adaptado de Nasteski (2017).

A Figura 13 mostra o fluxo básico de funcionamento de um algoritmo de aprendizado de máquina supervisionado, sendo três deles discutidos nos itens 2.2.1.1 a 2.2.1.3.

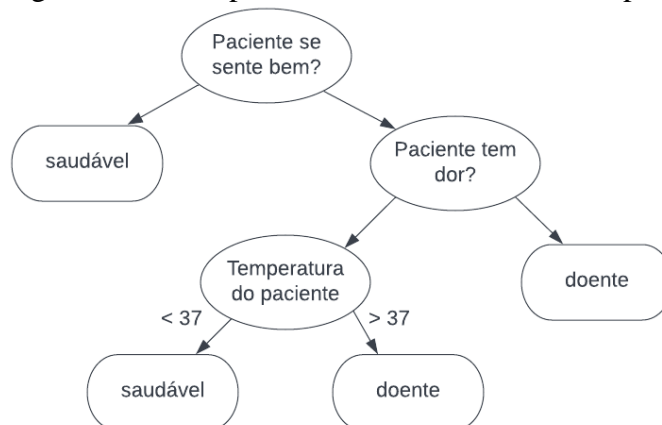
2.4.1.1 Árvore de Decisões

Árvores de decisão são um método de aprendizado supervisionado usado para classificação e regressão. O objetivo é criar um modelo que preveja o valor de uma variável de destino, aprendendo regras de decisão simples que são inferidas a partir de bases de dados previamente fornecidas. Elas podem ser caracterizadas pelo fato de que uma amostra desconhecida é classificada usando uma ou várias funções de decisão de maneira sucessiva (SWAIN e HANS, 1977).

Pode-se citar algumas vantagens das árvores de decisão como: simples de entender e interpretar, podendo ser visualizadas; requer pouca preparação de dados; o custo de utilizar uma árvore é logarítmico em relação ao número de pontos de dados usados para treiná-la; capaz de lidar com múltiplas saídas; capaz de validar um modelo por meio de testes estatísticos, tornando possível verificar a confiabilidade do modelo; apresentam um bom desempenho.

Em relação às desvantagens, notam-se que: árvores de decisão podem se tornar excessivamente complexas, não generalizando bem os dados, o que é chamado de *overfitting*. Mecanismos como poda computacional ou definição de máxima profundidade da árvore podem ser necessários para evitar este tipo de problema; podem ser criadas de forma tendenciosa, caso existam classes dominantes, sendo recomendável equilibrar a base de dados antes treiná-la (MONARD e BARANAUSKAS, 2003)

Figura 14 - Exemplo de Árvore de Decisões simples.



Fonte: Adaptado de Monard e Baranauskas (2003).

Como exemplificado na Figura 14, a árvore de decisão pode ser visualizada graficamente de forma que apresenta as escolhas e seus resultados na forma de uma árvore. Os nós no gráfico representam um evento ou escolha e as arestas do gráfico representam as regras ou condições de decisão. Cada árvore consiste em nós e ramos. Cada nó representa atributos em um grupo que deve ser classificado e cada ramificação representa um valor que o nó pode assumir (MAHESH, 2020).

2.4.1.2 Florestas Aleatórias

As Florestas Aleatórias (*Random forest*) é um método de aprendizado de máquina amplamente utilizado em problemas de classificação e regressão. O método utiliza a geração de diversas árvores de decisão, que juntas formam florestas, com o objetivo de melhorar a predição em relação a uma única árvore. O método de criação das árvores é modificado para que as mesmas se tornem diferentes umas das outras, criando assim árvores não correlacionadas (LOPES, 2018). A ideia é de que a diversidade representada por várias árvores montadas com uma seleção aleatória de atributos e exemplos reduza o sobreajuste (*overfitting*). Quando trata-se de problemas de classificação, a predição é feita através de votação majoritária, já para problema de regressão, a decisão é dada pela média das decisões individuais (PONTE et al. 2020).

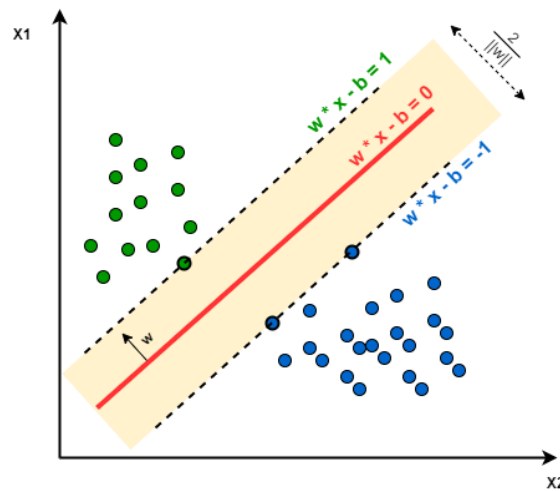
2.4.1.3 *Support Vector Machine*

Outra técnica de aprendizado de máquina muito utilizada é a *Support Vector Machine* (SVM). Em aprendizado de máquina, SVMs são modelos de aprendizagem supervisionado com algoritmos associados de aprendizagem que analisam dados utilizados em análises de classificação e regressão (MAHESH, 2020).

O SVM é um classificador que utiliza o princípio da maximização de margem. O princípio da margem máxima coloca a superfície de decisão entre o limite de duas classes e maximiza a distância do limite das classes, o que reduz a probabilidade de erro na classificação (BONESSO, 2013). O modelo pode ser representado por pontos no espaço, mapeados de maneira que os pontos pertencentes a cada categoria sejam divididos por um espaço claro que seja tão amplo quanto possível. Os novos pontos são, então, mapeados no mesmo espaço e preditos como pertencentes a uma categoria baseados em qual o lado do espaço eles são colocados.

Esse modelo encontra uma linha de separação, chamado de hiperplano, entre dados de duas classes. O hiperplano busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes.

Figura 15 – Representação do modelo de SVM.



Fonte: Aibin (2021).

A distância entre o hiperplano e o primeiro ponto de cada classe é chamada de margem. A SVM define inicialmente a qual classe cada ponto pertence e, em seguida, maximiza a margem a partir de uma função dessa restrição.

2.4.1.3.1 Classificação SVM com Múltiplas Classes

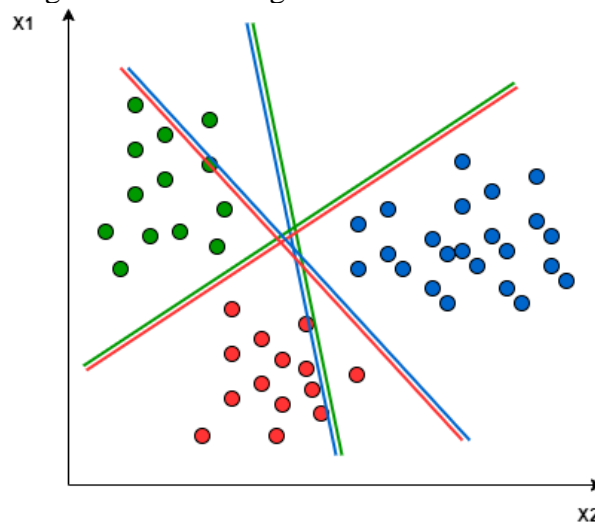
Em sua implementação mais simples, o SVM não oferece suporte à classificação multiclasse. Ele oferece suporte à classificação binária e à separação de pontos de dados em somente duas classes. Para classificação multiclasse, o princípio mais simples é aplicado ao se decompor o problema de multi classificação em vários problemas binários de classificação. Duas abordagens diferentes podem ser utilizadas para resolver esse tipo de problema, *One-to-One* e *One-to-Rest* (AIBIN, 2021).

- *One-to-One*: mapeia os pontos de dados no espaço para obter a separação linear mútua entre cada duas classes, quebrando um problema de múltiplas classes em um classificador binário para cada par de classes. Então, para classificar n classes seriam necessárias $\frac{n(n-1)}{2}$ SVMs.

- *One-to-Rest*: mapeia os pontos de dados no espaço para obter uma separação de cada classe em relação a todas as demais. Assim, para classificar n classes seriam necessárias n SVMs.

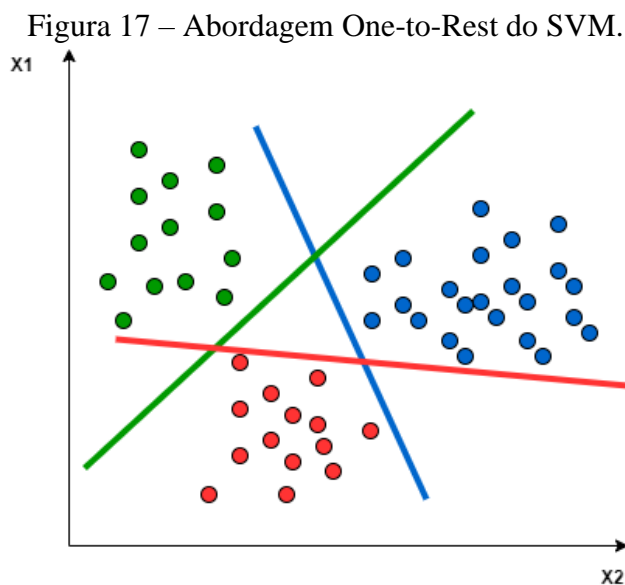
Pensando em um exemplo de classificação de 3 classes (verde, vermelha e azul), ao se utilizar a abordagem *One-to-One*, são necessários hiperplanos que realizem a separação a cada duas classes, não levando em consideração as demais. A Figura 16 mostra os hiperplanos gerados a partir da abordagem *One-to-One* na qual, por exemplo, o hiperplano vermelho-azul maximiza a separação entre as classes vermelho e azul, desconsiderando completamente a classe verde.

Figura 16 – Abordagem One-to-One do SVM.



Fonte: Aibin (2021).

Para a abordagem *One-to-Rest*, os hiperplanos precisam uma classe de todas as demais, o que significa que para cada SVM, todos os pontos serão levados em consideração. Observando a Figura 17, que representa esta abordagem, a linha verde tenta maximizar a separação entre a classe verde e todas as outras duas (vermelho e azul).



2.5 DETECÇÃO DE MOVIMENTOS DE TÊNIS

Ebner e Findling (2019) apontam que existem diferentes abordagens para detectar golpes de tênis fazendo-se uso de amostras de sensores inerciais. Zhao et al. (2019) utilizam para detecção o valor de pico em valores de aceleração através de um valor predefinido. Pei et al. (2017) preferem a abordagem de uma janela deslizante, na qual são gerados dados a partir das janelas obtidas, os quais definem sua volatilidade, para determinar a existência de movimento juntamente com um uso de valores mínimos pré definidos.

Enquanto algumas abordagens utilizam principalmente valores de aceleração, outras técnicas consideram adicionalmente valores calculados de energia a partir de dados fornecidos pelo giroscópio para determinar a existência de movimento. Além disso, a forma da aceleração para o eixo x e os valores de energia dos dados do giroscópio em golpes de tênis possuem uma forma particular, o que torna possível a detecção de golpes a partir de uma DTW (*Dynamic Time Warping*), que é um método utilizado para reconhecimento de sinais a partir da comparação com um sinal pré definido (EBNER e FINDLING, 2019).

2.6 CARACTERÍSTICAS UTILIZADAS PARA CLASSIFICAÇÃO DE MOVIMENTOS

Para segmentação de golpes de tênis, a maioria das abordagens utilizam de uma janela de tempo ao redor do ponto de detecção (EBNER e FINDLING, 2019).

Para extrair *features*, ou seja, as características que descrevem as amostras que representam um determinado movimento, Zhao et al. (2019) utilizam de valores de média,

desvio padrão, assimetria, curtose, mínimos, máximos e a amplitude dos dados de aceleração e do giroscópio para cada um dos eixos, resultando em 38 características usadas durante a classificação. Outros pesquisadores já preferem uma abordagem mais complexa, como, por exemplo, utilizar dos dados gerados por um acelerômetro e um giroscópio e a partir deles obter cerca de 2000 características, para então através do uso de seleção de características baseadas em correlação reduzir para um número de 300 a 500 características distintas (EBNER e FINDLING, 2019).

2.7 ESCOLHA DE CARACTERÍSTICAS

A seleção de características é uma das mais frequentes e importantes técnicas do processamento de dados, se tornando um componente indispensável no processo de aprendizado de máquina. A seleção de características é o processo de detectar as características mais relevantes, removendo as menos relevantes ou redundantes. Características menos relevantes são consideradas aquelas que não carregam informação útil, já as redundantes são aquelas que carregam informações que são fornecidas por outras características já selecionadas (KUMAR e MINZ, 2014). Ao aplicar-se este tipo de técnica, pode-se conseguir uma resposta mais rápida dos algoritmos, aumentar a acurácia na predição, e facilitar a compreensão dos problemas. Em termos de aprendizado supervisionado, a escolha de características dá origem a um novo subconjunto através de uma das seguintes abordagens.

- O tamanho especificado do subconjunto de características que aprimora o processo de avaliação;
- O menor subconjunto que satisfaça uma certa restrição em relação ao processo de avaliação;
- O subconjunto que traz o melhor balanço entre tamanho do subconjunto e avaliação.

O uso correto de algoritmos para subdimensionar o conjunto de características melhora o aprendizado, tanto em relação a capacidade de generalização, velocidade de aprendizado e redução da complexidade dos modelos gerados (KUMAR e MINZ, 2014).

2.7.1 Relevância das Características

De acordo com Kumar e Minz (2014), O melhor subconjunto de características é o subconjunto que contém todas as características relevantes. Na literatura, características são classificadas pela sua relevância a partir de três qualificadores: irrelevante, fracamente relevante e fortemente relevante. É possível definir a relevância de uma característica baseando-se em diferentes critérios. Kumar e Minz (2014) sugerem algumas destas definições: relevância em relação ao rótulo, fortemente relevante em relação às amostras/distribuição, fracamente relevante em relação às amostras/distribuição, relevância como uma medida de complexidade, Inutilidade incremental e Relevância de entropia.

2.8 OTIMIZAÇÃO DE HIPERPARÂMETROS

O uso de hiperparâmetros com valores ótimos possibilita alcançar um modelo com melhor desempenho, evitando-se problemas como o sobre-ajuste (*overfitting*) ou modelos simplistas com uma capacidade inadequada à complexidade do problema. Sendo assim, os hiperparâmetros são usados para configurar vários aspectos de um algoritmo de aprendizado, tendo influência direta sobre o resultado final do modelo (ALVARENGA, 2018).

Geralmente os modelos apresentam no mínimo um hiperparâmetro relacionado à capacidade do modelo em se ajustar aos dados de treinamento e, a depender do modelo, um número mínimo de hiperparâmetros ligados ao algoritmo de treinamento para definir as características relacionadas à convergência. Parâmetros e hiperparâmetros de alguns modelos são apresentados através da Tabela 1.

Tabela 1 - Hiperparâmetros e parâmetros para alguns modelos de aprendizado de máquina.

Modelos	Parâmetros	Hiperparâmetros
Árvore de Decisão	- atributo utilizado em cada nó - valor de corte em cada nó	- profundidade - número de nós folha - número de amostras por divisão de nó - critério de pureza
Floresta Aleatória	- atributo utilizado em cada nó de cada árvore - valor de corte em cada nó de cada árvore	- número de árvores - profundidade das árvores - número de amostras por divisão de nó - critério de pureza
SVM	- vetores de suporte	- tipo de kernel - constante regularização - constante de tolerância

Fonte: Adaptado de Alvarenga (2018).

A otimização dos hiperparâmetros possibilitam a obtenção de modelos com melhor desempenho de predição, porém em alguns casos a otimização é custosa computacionalmente e se faz necessário avaliação do esforço relacionado à estimação dos valores ótimos (ALVARENGA, 2018).

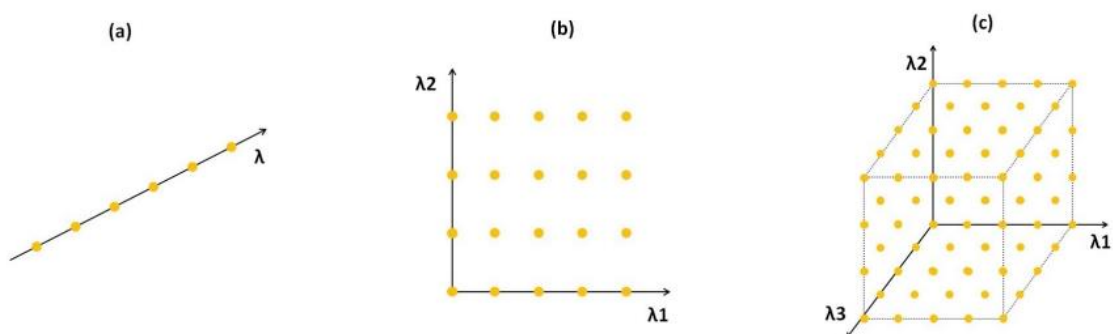
Uma questão extremamente importante na busca dos hiperparâmetros ideias é o custo de se avaliar a função objetivo. Cada avaliação requer um cálculo de desempenho do modelo treinado para cada combinação de hiperparâmetros. A depender dos recursos computacionais disponíveis, da natureza do algoritmo de aprendizagem e da quantidade de dados, cada avaliação pode demorar de minutos até dias (Krizhevsky et al., 2012).

2.8.1 Otimização com Busca de Grade

A Busca de Grade é um método muito utilizado na otimização de hiperparâmetros em modelos com poucos hiperparâmetros. Este método possui uma resolução mais simples e direta, sendo de fácil implementação e paralelização.

Para realizar a otimização é definido um intervalo para cada hiperparâmetro, dentro do qual será realizada uma varredura. Para cada um dos hiperparâmetros, estabelece-se uma resolução de grade que determina a quantidade de pontos a serem considerados em cada intervalo. A execução deste método gera uma estrutura de grade do espaço hiperparamétrico, formada pelo conjunto de vetores que compõem todos os arranjos de candidatos, sendo cada arranjo avaliado nos treinamentos do modelo de aprendizado por meio de uma função de perda (ALVARENGA, 2018). A Figura 18 mostra exemplos de grades geradas para o espaço hiperparamétrico, sendo que cada um dos eixos λ representa o intervalo de valores de um hiperparâmetro.

Figura 18 - Exemplos de grades hiperparamétricas para (A) uma (B) duas (C) três dimensões.



Fonte: Alvarenga (2018).

Após delimitar-se o intervalo dentro do espaço hiperparamétrico e região onde se encontra o valor ótimo de configuração, o método de busca de grade possui grande chance de encontrar um valor ótimo ou muito próximo disso. O problema da utilização deste método se encontra no crescimento exponencial de treinamentos necessários à medida que o número de hiperparâmetros aumenta, tornando a otimização extremamente demorada em certos casos (ALVARENGA, 2018).

2.9 APRENDIZADO DE MÁQUINA EM SISTEMAS EMBARCADOS

A tendência de mover a computação para as bordas (*edge computing*), permitindo que se faça o processamento dos dados de forma local e mais próxima da fonte, está cada vez se tornando mais relevante e permitindo melhorias de desempenho e desenvolvimento de novas aplicações e processamento de dados localmente. A mudança da computação em nuvem para os sistemas embarcados pode trazer vantagens em tempo de resposta, largura de banda, consumo de energia e segurança (SAKR et al, 2020).

A aplicação de técnicas de aprendizado de máquina em sistemas embarcados normalmente está relacionada à etapa de inferência, que é computacionalmente menos intensa do que a etapa prévia de treinamento. Ao executar modelos treinados diretamente nos sistemas embarcados, os dispositivos podem processar localmente os dados coletados de forma imediata, podendo até mesmo filtrá-los, para que os mesmos sejam enviados para algum outro sistema externo ou sirvam de parâmetro para tomada de ações (SAKR et al, 2020).

Modelos de aprendizado de máquina são frequentemente treinados de modo *off-line* em linguagens de programação de mais alto nível como, por exemplo, *Python* e *MATLAB*, com recursos praticamente ilimitados na nuvem. No entanto, aplicar esses modelos em sistemas embarcados, que normalmente operam em linguagem C / C++, com restrição de recursos, pode se tornar bastante complicado. Além disso, para plataformas com grande restrição de memória, especialmente para MCUs de ultra baixa potência, até mesmo a escolha do modelo e de seus parâmetros pode vir a ser um desafio, já que os modelos treinados podem facilmente exceder as capacidades da plataforma de destino e técnicas, como a poda computacional, podem se fazer necessárias (SLIWA et al., 2020).

2.9.1 Poda computacional (*pruning*)

Com o desenvolvimento da inteligência artificial o tamanho dos modelos tornam-se cada vez maiores, sendo necessária uma maior capacidade de computação. Com o objetivo de reduzir o tamanho dos modelos e acelerar a computação em hardware, a poda computacional (*pruning*) tem sido necessária para que as respostas se tornem mais rápidas sem que se perca o mesmo nível de taxa de acerto. Ao realizar o *pruning*, tem-se ganhos no tempo de inferência e diminuição da necessidade de armazenamento, o que é muito útil em dispositivos portáteis (ZHANG, 2021).

3. METODOLOGIA

O sistema proposto para solução do problema de detecção e classificação dos golpes executados durante a prática do tênis é o desenvolvimento de um protótipo de um dispositivo vestível, acoplado ao pulso do jogador, o qual utilizará de modelos de aprendizado de máquina supervisionado para realizar a classificação. Os modelos escolhidos como objeto de estudo são as Árvores de Decisão, Florestas Aleatórias e SVM, os quais serão comparados e avaliados.

3.1 MATERIAIS E FERRAMENTAS

Neste item serão apresentados os recursos de hardware e as ferramentas computacionais utilizadas durante o desenvolvimento do projeto de pesquisa.

3.1.1 Hardware Utilizado

Este item abordará as principais ferramentas de hardware utilizadas durante o desenvolvimento do projeto de pesquisa, sendo elas os módulos ESP32, GY-521 e TP4056. Os módulos são apresentados individualmente nas seções 3.1.1.1 a 3.1.1.3. Além deles, foi utilizado também uma bateria Li-Po 3,7V/850 mAh e um display OLED, de resolução 128x32, 0,91 polegadas, com um driver SSD1306 e interface I2C.

3.1.1.1 ESP32

A placa *DOIT ESP32*, contendo o módulo *ESP-WROOM-32D* com chip D0WDQ6, é considerada um *System-On-Chip (SoC)*. Possui um microcontrolador dual core Tensilica Xtensa 32-bit LX6 de baixo consumo de energia, suporte a wifi, bluetooth e memória flash integrada de 4MiB. Sua arquitetura permite que ele seja programado de forma independente, ou seja, sem a necessidade de outras placas microcontroladoras. Outra vantagem é que o módulo possui integração com a Arduino IDE, o que disponibiliza diversas bibliotecas de código aberto feitas pela comunidade.

3.1.1.2 GY-521

O módulo GY-521 possui na mesma placa um acelerômetro e um giroscópio de alta precisão, com 6 graus de liberdade, ambos controlados pelo circuito integrado MPU6050. As

especificações são dadas para uma tensão de alimentação de $2,37 V - 3,46 V$, tensão lógica de $1,8 V \pm 5\%$ e temperatura a $25^\circ C$.

O acelerômetro contido no módulo é um acelerômetro de tipo capacitivo de 3 eixos. O mesmo usa massas de prova separadas para cada eixo. A aceleração ao longo de cada eixo induz deslocamento na massa de prova correspondente, e sensores capacitivos detectam o deslocamento diferencial. Quando o dispositivo é colocado em uma superfície plana, ele medirá $0 G$ nos eixos X e Y e $+1 G$ no eixo Z. O fator de escala do acelerômetro é calibrado pelo fabricante e é nominalmente independente da tensão de alimentação. Cada eixo possui um conversor ADC sigma-delta dedicado de 16 bits para fornecer saídas digitais. A faixa de escala total da saída digital pode ser ajustada para $\pm 2 G$, $\pm 4 G$, $\pm 8 G$ ou $\pm 16 G$, sendo o fator de escala de sensibilidade $2048 LSB/G$ e tolerância de $\pm 3\%$ ao estar configurado para a escala de $\pm 16 G$. Seu erro de linearidade é dado em $0,5\%$ e sua tolerância inicial de calibração de zero é de $\pm 50 mG$ para os eixos X e Y e $\pm 80 mG$ para o eixo Z. A taxa de amostragem é programável de 4 a 1000 Hz.

O giroscópio contido no módulo também é de 3 eixos. Cada eixo possui um conversor ADC dedicado de 16 bits para fornecer a saída de forma digital. Pode-se também configurar um filtro passa-baixa com algumas frequências de corte pré definidas. A faixa de escala total da saída digital pode ser programada para $\pm 250^\circ/s$, $\pm 500^\circ/s$, $\pm 1000^\circ/s$ ou $\pm 2000^\circ/s$, sendo o fator de escala de sensibilidade $16,4 LSB/(^\circ/s)$ e tolerância de $\pm 3\%$ ao estar configurado para a escala de $\pm 2000^\circ/s$. Seu erro de linearidade é dado em $0,2\%$ e sua tolerância inicial de zero é de $\pm 20^\circ/s$. A taxa de amostragem é programável de 4 a 8000 Hz.

Além dos sensores, o módulo possui um recurso chamado DMP (*Digital Motion Processor*), o qual realiza cálculos complexos com os sensores para que possam ser utilizados em sistemas de reconhecimento de gestos, navegação e diversas outras aplicações. A comunicação com microcontroladores usa a interface I2C e a alimentação pode ser feita dentro da faixa de 3V a 5V.

3.1.1.3 TP4056

O módulo carregador para baterias de lítio TP4056 possibilita que as baterias sejam carregadas sem a necessidade de removê-las do circuito. A tensão de entrada suportada é de até 8V, com uma tensão de saída de $4,2V \pm 1,5\%$ e uma corrente de saída programável de até 1000 mA. O mesmo possui leds indicadores de carga, conexão mini USB e é amplamente utilizado em protótipos e em conjunto com kits de desenvolvimento.

3.1.2 Ferramentas Computacionais

Neste item são apresentados os principais recursos computacionais utilizados para o desenvolvimento do dispositivo proposto e também para criação de ferramentas utilizadas para facilitar a execução de etapas da pesquisa.

A *Plataforma Arduino* foi utilizada para o desenvolvimento do firmware executado no ESP32 e no utilizado no protótipo. Para análise de dados, criação e otimização dos modelos foi utilizado o editor de texto *Visual Studio Code (VSCode)* integrado ao *Jupyter Notebook*, e bibliotecas como *scikit-learn* para linguagem *Python*. Por fim, o ambiente do *Google Firebase* foi utilizado para armazenamento de dados e integração entre o protótipo e as ferramentas executadas na WEB.

3.1.2.1 Plataforma Arduino

A Plataforma Arduino, mais especificamente a IDE fornecida pela plataforma, é gratuita e permite um alto grau de abstração, o que auxilia no desenvolvimento de projetos microcontrolados. A IDE possui uma linguagem própria baseada em C e C++ e uma grande quantidade de bibliotecas que permitem a integração com outros módulos, sensores, dispositivos ou mesmo aplicações.

3.1.2.2 Adafruit MPU6050

Adafruit MPU6050 é uma biblioteca fornecida pela empresa *Adafruit Industries* e disponibilizada na plataforma Arduino. Esta biblioteca permite a integração com o sensor MPU6050 através da interface I2C. Diversas configurações são abstraídas pela biblioteca e acessadas através de funções, o que permite uma maior concentração de tempo no desenvolvimento.

3.1.2.3 Google Firebase

O Google Firebase é uma plataforma digital utilizada para facilitar o desenvolvimento de aplicativos web ou móveis, de uma forma rápida e simples. Uma de suas principais vantagens é sua capacidade de integração com outras aplicações, fornecendo produtos de autenticação, armazenamento de arquivos, analytics, hospedagem, notificações, entre outros.

Para o desenvolvimento do projeto em questão, o recurso utilizado foi o *Firestore Realtime Database*, que é uma tecnologia muito usada para manter estados em tempo real. Seu armazenamento funciona em forma de árvores ou documentos em formato JSON e possui

bibliotecas próprias para, por exemplo, realizar integração com dispositivos IoT, plataforma ARDUINO e aplicações Web via API REST.

3.1.2.4 React

React ou ReactJS é uma biblioteca JavaScript usada para construir interfaces de usuário, permitindo a elaboração de aplicativos baseados em JavaScript. O React permite fragmentar a aplicação em componentes individuais para reutilização e possui uma grande quantidade de bibliotecas disponibilizadas pela comunidade, o que possibilita o desenvolvimento de aplicações complexas de uma forma mais rápida.

Os aplicativos React são executados diretamente no navegador, tendo a capacidade de renderizar componentes que são apresentados na interface individualmente, ao invés de renderizar a página inteira para refletir interações, o que melhora a performance da aplicação.

3.1.2.5 Scikit-learn

Os códigos implementados para os modelos de aprendizado de máquina foram executados utilizando a linguagem *Python* versão 3.7, em conjunto com a biblioteca *scikit-learn* versão 1.1.1, que é uma das bibliotecas do *Python* mais conhecidas e utilizadas atualmente para esta aplicação. A mesma é elaborada em código aberto e é desenvolvida para suportar e possibilitar o treinamento de diversas técnicas de estatística e *Machine Learning*, para aprendizagem supervisionada e não supervisionada.

3.1.2.6 MicroML generator

Para portar o modelo de aprendizado de máquinas implementado através do *scikit-learn* foi utilizado o projeto MicroML, que visa portar algoritmos de machine learning para linguagem C, de forma que possam ser aplicados em microcontroladores. Ele é uma alternativa ao *Tensorflow* para microcontroladores, que é dedicado a redes neurais.

O projeto, na versão 1.1.13, tem suporte a diferentes tipos de classificadores, como: *Decision Tree*, *Random forest*, *XGBoost*, *GaussianNB*, *Support Vector Machines(SVM)*, *Relevant Vector Machines*, *SEFR* e *PCA*.

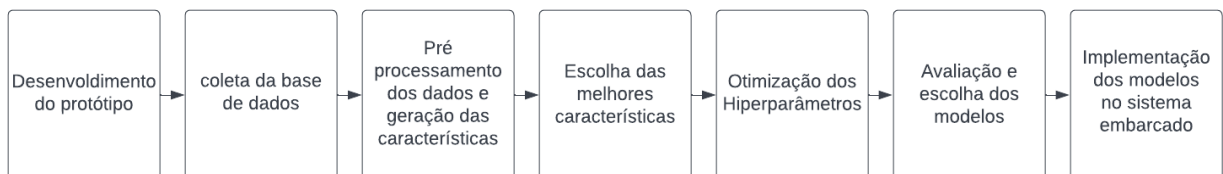
3.2 PROCEDIMENTO

A escolha dos equipamentos é algo bastante particular de cada praticante, já que, por exemplo, a raquete atua como uma extensão do braço do jogador, possui um peso e centro de massa específico, diferentes tensões nas cordas, tamanho e o cabo é geralmente customizado para atender as preferências de cada um (CROSS, 2011). Considerando um sistema mais acessível e de baixo custo, o dispositivo desenvolvido fica localizado no pulso do jogador e fixado através de uma pulseira, similar a um relógio.

A proposta é que sejam detectados e classificados os três golpes mais comuns do tênis: saque, *forehand*, *backhand*. Os movimentos são classificados em tempo real pelo sistema embarcado.

O procedimento geral da metodologia, o qual representa as etapas de execução da pesquisa, é ilustrado na Figura 19 através de um fluxograma. Como um panorama geral das etapas executadas, têm-se inicialmente o desenvolvimento do protótipo, o qual realizará as leituras de aceleração e velocidade angular através do acelerômetro e giroscópio, respectivamente, possibilitando por sua vez a coleta da base de dados.

Figura 19 - Procedimento geral da metodologia.



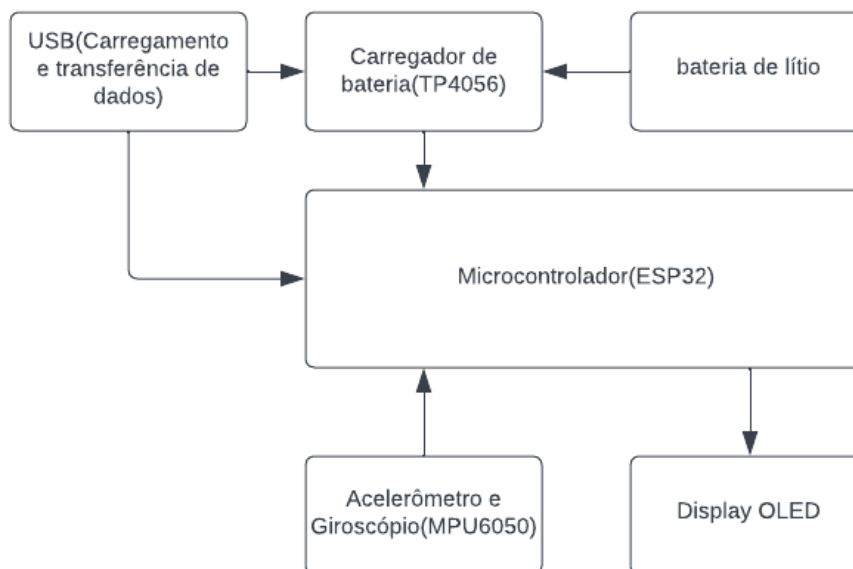
Fonte: O autor.

Após coletados os dados, é realizado o pré processamento dos mesmos, sendo gerados os rótulos e as características (*features*) que os representam. Em seguida é feita escolha das melhores características e otimização dos hiperparâmetros para cada um dos modelos estudados, com o intuito de obter uma melhor capacidade de predição. Por fim, os modelos gerados são avaliados para escolha e implementação do sistema de classificação no sistema embarcado.

3.2.1 Desenvolvimento do Protótipo

Para o desenvolvimento do protótipo, foi montado um kit de desenvolvimento contendo os módulos ESP32, GY-521, TP4056, um display oled 128x32 e uma bateria de lítio para alimentação.

Figura 20 - Diagrama de protótipo.



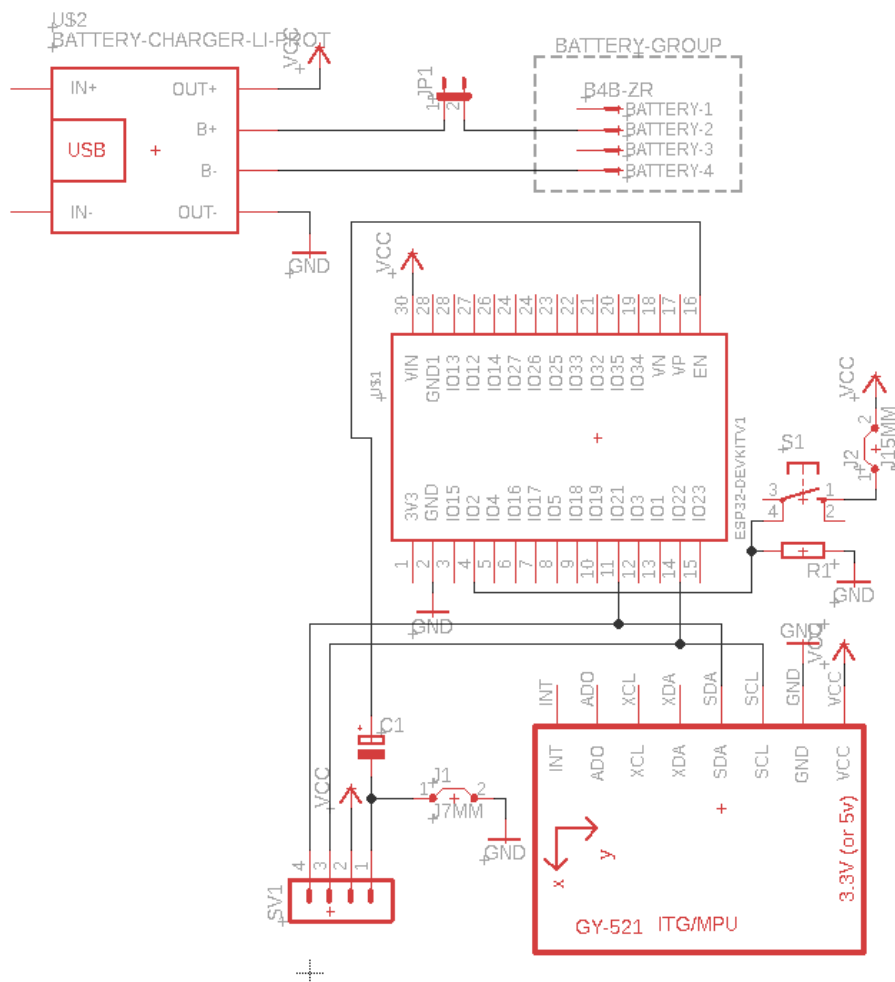
Fonte: O autor.

A Figura 20 mostra o diagrama contendo os principais componentes do dispositivo desenvolvido. O módulo TP4056 e a bateria de lítio são responsáveis por fornecer energia ao sistema. A conexão USB é utilizada para carregamento e também para transferência de dados. O módulo MPU-6050 contém os dois sensores (acelerômetro e giroscópio) que fornecem valores de aceleração e velocidade angular ao módulo ESP32, o qual é responsável por processar os dados, detectar e classificar os movimentos propostos. Para visualização dos dados e realização de testes, é utilizado um display OLED.

3.2.1.2 Projeto do protótipo

Para realizar a integração dos diferentes módulos mostrados no diagrama da Figura 20, foi desenvolvido o projeto do circuito eletrônico através do software *EAGLE 9.6.2*, seguindo o diagrama esquemático da Figura 21.

Figura 21 - Esquemático do circuito.

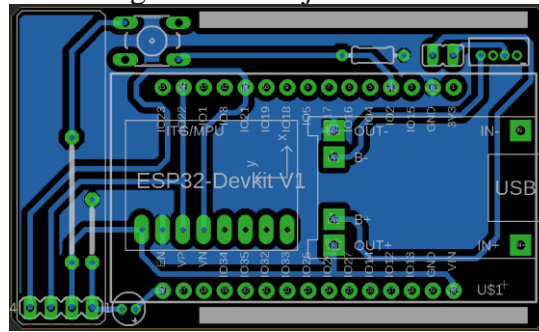


Fonte: O autor.

O circuito conta com os módulos ESP32, GY-521, TP4056, um display OLED 128x32, um botão para acesso a configurações de rotinas de teste e uma conexão para a bateria. Na construção do projeto não se fez necessário o uso de capacitores de desacoplamento, pois os mesmos já estão presentes nos módulos utilizados.

De forma a atender requisitos ergonômicos, as dimensões do protótipo são de extrema importância, já que o mesmo deverá ser fixado no pulso do jogador e não poderá prejudicá-lo durante a prática da atividade. O projeto da PCI, mostrado na Figura 22, possui dimensões de 40 x 65 mm e prevê rasgos laterais para fixação da pulseira.

Figura 22 – Projeto da PCI.

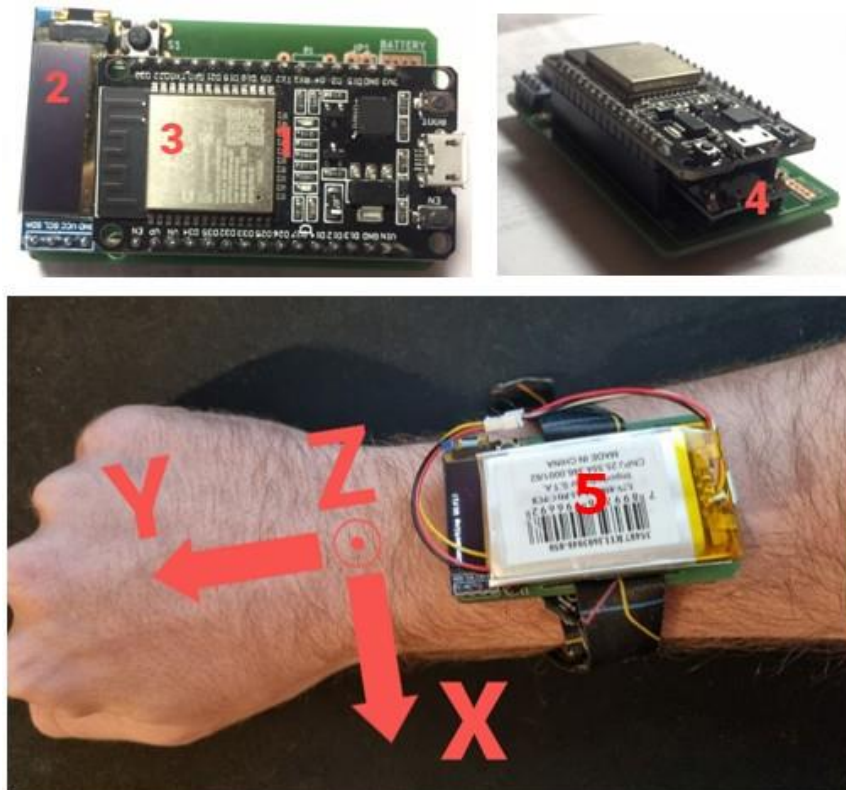


Fonte: O autor.

Após realização do projeto da PCI, foi feita a produção da mesma e a montagem do protótipo. A Figura 23 apresenta o protótipo desenvolvido.

Figura 23 – Protótipo do sistema de aquisição e classificação de sinais inerciais e indicação da orientação dos eixos.

(1) ESP32; (2) Display OLED; (3) GY521 (sob o item 1); (4) TP4056; (5) Bateria.



Fonte: O autor.

3.2.2 Coleta da Base de Dados

Durante o projeto de pesquisa não foram encontradas nenhuma base de dados que atenda às mesmas condições necessárias para o desenvolvimento do presente trabalho. Portanto, a base

de dados utilizada para classificação dos movimentos foi coletada pelo próprio pesquisador com o auxílio de um tenista amador. O tenista é homem, destro, possui 45 anos de idade, 175cm de altura e pesa cerca de 70 kg. O mesmo possui conhecimento da correta execução dos movimentos, tendo se declarado com valor 4 em uma escala de 1 a 5 de familiaridade com o esporte.

As Figuras 24, 25 e 26 representam uma repetição dos movimentos de saque, *backhand* e *forehand*, respectivamente, realizados durante a coleta dos dados.

Figura 24 - Sequência do movimento de saque.



Fonte: O autor.

O local de coleta foi em quadras públicas de tênis no município de Porto Alegre. Para aquisição dos dados, o tenista utiliza o dispositivo vestível acoplado em seu pulso, o qual será o instrumento de coleta das informações do voluntário durante a execução dos movimentos.

Antes da execução dos testes, o mesmo foi orientado sobre o funcionamento do sistema e teve um período de cerca de 15 minutos para se familiarizar.

Figura 25 - Sequência do movimento de *backhand*.



Fonte: O autor.

Os dados coletados para cada um dos movimentos de interesse (saque, *forehand* e *backhand*) são aceleração e velocidade angular nos eixos X , Y e Z , fornecidos, respectivamente, por um acelerômetro e um giroscópio com 3 eixos cada, totalizando 6 graus de liberdade. Os valores de aceleração são medidos em m/s^2 , com uma faixa de escala de $\pm 16 G$ e um fator de sensibilidade de escala de $2048 LSB/G$ com tolerância de $\pm 3\%$. Para a velocidade angular, seus valores são medidos em $^\circ/s$, com uma faixa de escala de $2000 ^\circ/s$ e um fator de sensibilidade de escala de $16,4 LSB/(^\circ/s)$ com tolerância de $\pm 3\%$. Para ambos sensores o conversor ADC utilizado é de 16 bits.

Vale salientar que não foi realizado nenhum processo de caracterização do sistema ou dos sensores isoladamente, não sendo obtida a resolução efetiva do sistema. Desta forma, não se garante a generalização dos padrões observados nos dados e análises para outros dispositivos.

Figura 26 - Sequência do movimento de *forehand*.



Fonte: O autor.

O dispositivo realiza a leitura dos dados a uma taxa de amostragem de 250 Hz, opera por baterias e possui comunicação sem-fio via *wifi*, de forma que o usuário não teve seus movimentos restringidos pela existência de fios. Os dados registrados são enviados ao fim de cada execução (devido a restrições de memória) para uma base de dados provida pelo recurso *Firestore Realtime Database* do *Google Firebase*, através de uma API REST abstraída pela biblioteca de código aberto *FirebaseArduino*.

Para realização de cada um dos tipos de movimento foram executadas 30 repetições, possuindo o voluntário um intervalo de 12 s para cada execução, sendo o movimento e o intervalo indicados através de uma tela de computador. Dentro de cada intervalo foi executado apenas um movimento. Uma ferramenta para auxiliar na realização da coleta de dados foi desenvolvida pelo pesquisador e será apresentada na seção 3.2.2.1.

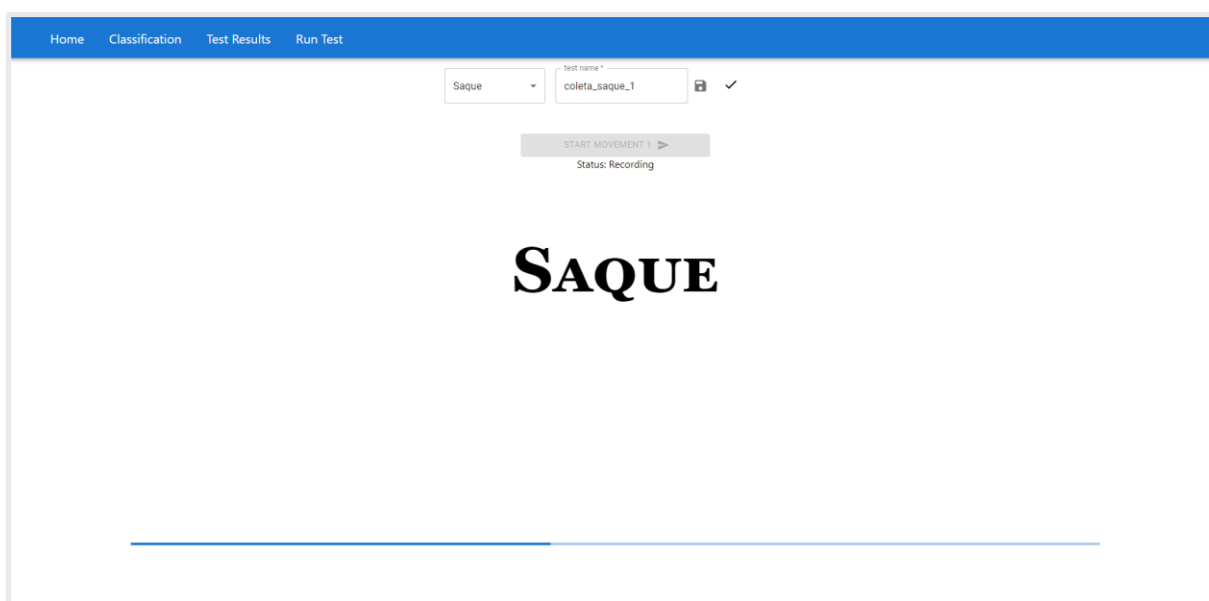
Nas execuções envolvendo o golpe de saque, uma bola de tênis ficava à disposição do tenista. Nos demais golpes, a bola foi lançada até o jogador para que o mesmo realizasse o movimento proposto.

3.2.2.1 Ferramenta para aquisição de dados

Com o intuito de auxiliar o procedimento de coleta da base de dados, foi desenvolvida pelo pesquisador uma aplicação Web com o uso da biblioteca *ReactJS*. Esta aplicação consegue se comunicar com o dispositivo via API REST e configurar o movimento a ser executado e o nome da coleta de dados, o qual é armazenado e utilizado para identificação.

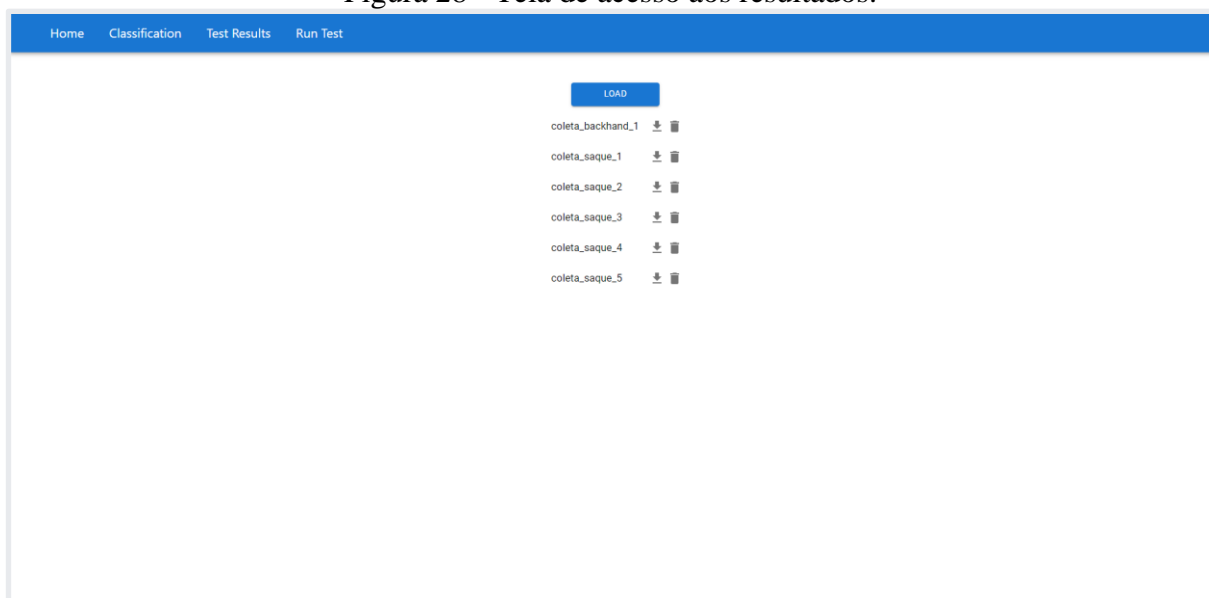
Além disso, a aplicação serve para iniciar o período de coleta, exibir o movimento proposto e possui uma barra de progresso que indica o período disponível para execução de cada movimento. A Figura 27 apresenta a tela que é utilizada durante a coleta dos dados.

Figura 27 - Tela de execução da coleta de dados.



Fonte: O autor.

Figura 28 - Tela de acesso aos resultados.



Fonte: O autor.

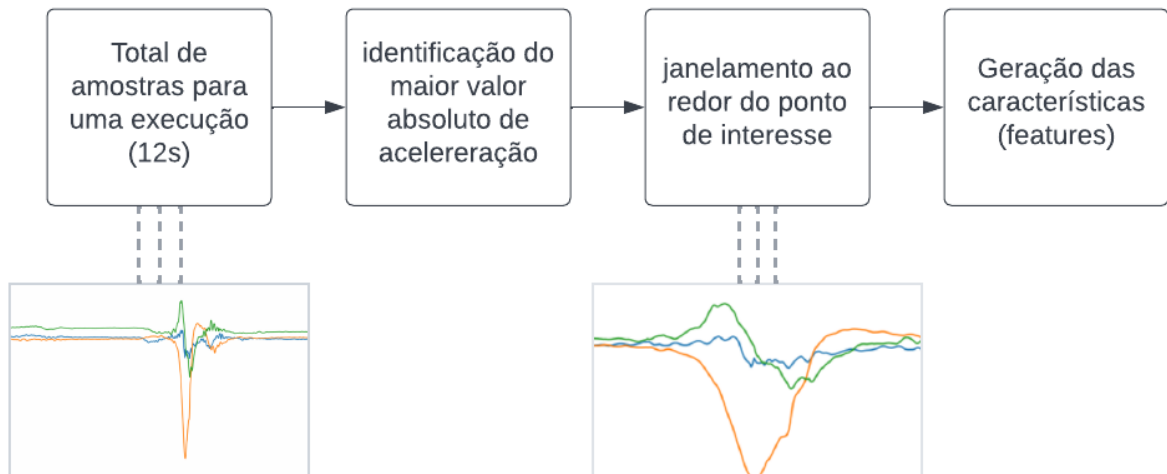
Além disso, a aplicação é capaz de se conectar diretamente à base de dados hospedada no *Google Firebase*, permitindo que os dados coletados sejam convertidos para um formato *.csv* e baixados. Caso se deseje descartar alguma das coletas, a mesma pode ser deletada através da própria aplicação. A Figura 28 representa a tela de acesso aos resultados, a qual possui as funcionalidades descritas anteriormente.

3.2.3 Pré Processamento dos Dados e Geração das Características

A coleta de dados realizada compreende a realização de um movimento proposto dentro de um intervalo de 12s. A taxa de amostragem do sistema utilizado foi de 250 Hz, o que dá um total de 3000 amostras para cada intervalo. Tendo em vista que a execução dos golpes não é representada pela totalidade das amostras e sim por um pequeno segmento deste intervalo, os dados coletados passam por um processo de janelamento ao redor de um ponto de interesse.

A Figura 29 representa o processo de pré-processamento dos dados e geração das características de interesse através de um fluxograma.

Figura 29 - Representação do processo de pré-processamento de geração das características.



Fonte: O autor.

O pesquisador definiu uma janela de 1 segundo para representação de cada golpe. Para cada um dos intervalos, é feita a identificação do ponto com maior valor absoluto de aceleração entre todos os eixos. Ao redor deste ponto, são selecionadas as 125 amostras anteriores e as 124 amostras posteriores, totalizando 250 amostras que representam uma janela de 1 segundo.

Após definidas as janelas, as mesmas são rotuladas com o tipo do movimento executado e as demais amostras são descartadas. A manipulação dos dados foi feita utilizando a linguagem *Python* e bibliotecas como *pandas* e *numpy*.

3.2.3.1 Características Geradas

Após se ter cada golpe rotulado e representado por um conjunto de 250 amostras, as mesmas são utilizadas para geração de características (*features*) que as representam. As características geradas foram.

- Média $\rightarrow \frac{1}{n} \sum_{i=1}^n x_i$
- Desvio Padrão $\rightarrow \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$
- Diferença média absoluta $\rightarrow \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |x_i - y_j|$
- Desvio absoluto da mediana $\rightarrow \frac{1}{n^2} \sum_{i=1}^n |x_i - m(x)|$
- Amplitude interquartil $\rightarrow Q_3 - Q_1$, sendo $Q_i = \frac{i}{4}(n - 1)$
- Assimetria (*skewness*) $\rightarrow \frac{\sum_i^n (x_i - \bar{x})^3}{(n - 1) \cdot \sigma^3}$, sendo σ o desvio padrão

- Curtose (*Kurtosis*) $\rightarrow \frac{\mu_4}{\sigma^4}$, sendo μ_4 o quarto momento central e σ o desvio padrão
- Energia $\rightarrow \sum_{n=-\infty}^{\infty} x^2[n]$
- Mínimo
- Máximo
- Diferença entre máximo e mínimo
- Mediana
- Número de amostras com valor menor que zero
- Número de amostras com valor maior que zero
- Número de valores acima da média

Com 15 características distintas para cada um dos 3 eixos de aceleração e velocidade angular, foram gerados um total de 90 características que representam cada golpe coletado.

3.2.3.2 Separação em Conjuntos Treinamento e Teste

Após se obter as características de cada um dos golpes que representam toda a base de dados, foram divididos dois subconjuntos, treinamento e teste. O subconjunto de treinamento representa 70% do total da base de dados e os demais 30% formam o subconjunto teste.

3.2.4 Escolha das Melhores Características

Com o intuito de se obter modelos com melhor capacidade de predição e performance, foi reduzido o tamanho do conjunto de características. Para realizar a escolha das melhores, foram testados três métodos disponíveis no módulo *sklearn.feature_selection* da biblioteca *scikit-learn* para linguagem *Python*, sendo eles: *SelectKBest*, *RFE(Recursive Feature Elimination)* e *RECV(Recursive Feature Elimination with cross-validation)*.

- ***selectKBest***: seleciona as melhores características baseado em testes estatísticos univariados, avaliando cada característica separadamente. Em geral, é necessário informar o número K de características que se deseja selecionar de um conjunto e o método estatístico a ser utilizado. Para este método foi definido um número de características $k = 5$ e a função de avaliação utilizada foi o *f_classif*;

- **RFE**: realiza a eliminação de características de forma recursiva. O *RFE*, inicialmente, realiza o treinamento de um modelo utilizando o conjunto completo de características providas e define a importância de cada uma delas através de algum atributo específico, como a taxa de acerto, por exemplo. Em seguida, as características são removidas do conjunto original. Este procedimento é repetido recursivamente até que o número desejado seja obtido. Para o método *RFE* foi utilizado como estimador uma Floresta Aleatória e 5 como o número de características desejadas;
- **RFECV**: similar ao *RFE*, porém utiliza de validação cruzada para automaticamente identificar as características que melhor se ajustam ao modelo. Na escolha pelo método *RFECV* foi também usado como estimado uma Floresta Aleatória, com número de partições igual a 5 e o critério de escolha baseado em taxa de acerto.

3.2.5 Otimização dos Hiperparâmetros

Para realizar a otimização dos hiperparâmetros foi utilizada a técnica da busca de grade, que é recomendada para modelos com poucos hiperparâmetros, como é o caso dos modelos estudados (Árvore de Decisão, Floresta Aleatória e SVM). Para implementação da busca de grade foi utilizado o módulo *sklearn.model_selection.GridSearchCV* da biblioteca *scikit-learn*. Os intervalos escolhidos foram definidos visando a construção de modelos mais simples, de forma a serem suportados em um sistema embarcado com restrições de memória. Nesta etapa, as características consideradas já haviam sido restringidas pela etapa descrita na seção 3.2.4.

Para otimização, foram escolhidos três parâmetros para cada um dos três modelos.

- *Árvore de Decisão*: Profundidade máxima da árvore, número máximo de nós folha, número mínimo de amostras para uma divisão de nós;
- *Floresta Aleatória*: Número de árvores, profundidade máxima das árvores e número mínimo de amostras para uma divisão de nós;
- *SVM*: C, gamma e kernel.

Tabela 2 - Intervalos dos hiperparâmetros.

	Hiperparâmetro	Intervalo
Árvore de Decisão	Profundidade máxima da árvore	2 10, passo 2
	Número máximo de nós folha	2 100, passo 1
	Número mínimo de amostras para uma divisão de nós	2 10, passo 1
Floresta Aleatória	Número de árvores	2 20, passo 2
	Profundidade máxima das árvores	2 10, passo 2
	Número mínimo de amostras para uma divisão de nós	2 10, passo 1
SVM	C	0,1 ; 1 ; 10 ; 100 ; 1000
	Gamma	1 ; 0,1 ; 0,01 ; 0,001 ; 0,0001
	Kernel	linear, rbf

Fonte: O autor.

A Tabela 2 apresenta os hiperparâmetros para os três modelos estudados e o intervalo escolhido para otimização através do método de busca de grade.

3.2.6 Método de Detecção dos Movimentos

Para detecção dos movimentos, foi utilizado a média dos valores de aceleração entre cada eixo da derivação numérica de dois pontos, dada por:

$$D[n] = \frac{1}{3} \sum_{i=1}^3 |[A_i[n] - A_i[n - 1]]|$$

Onde, $D[n]$ é o valor da derivação média e A_i é o valor da amostra de aceleração para o correspondente eixo i .

Tanto Kos et al. (2016) quanto Ebner e Findling (2019) recomendam esta abordagem, pois os valores de aceleração mostram uma significativa mudança de valor durante o curto período de tempo no qual a raquete acerta a bola. A aceleração pode alcançar valores elevados durante a movimentação do jogador e de seu punho sem que a bola seja tocada, acarretando em falsas detecções caso seja utilizado apenas um valor mínimo de aceleração para etapa de detecção dos movimentos. Além disso, utilizar da derivação numérica média ao invés da

aceleração, para a detecção dos golpes, permite que a diferença de força entre jogadores não represente uma forte influência na capacidade de detecção.

Para determinar a derivação numérica média foi, inicialmente, calculado a diferença de aceleração entre duas amostras para cada um dos eixos individualmente, sendo então feita a média entre os eixos. Obteve-se assim uma nova sequência de amostras, nas quais os valores máximos representam os pontos de contato entre raquete e bola. Com base nestes pontos, foi buscado em toda a base de dados o menor valor, sendo este definido como o valor mínimo para detecção.

3.2.7 Implementação do Modelo no Sistema Embarcado

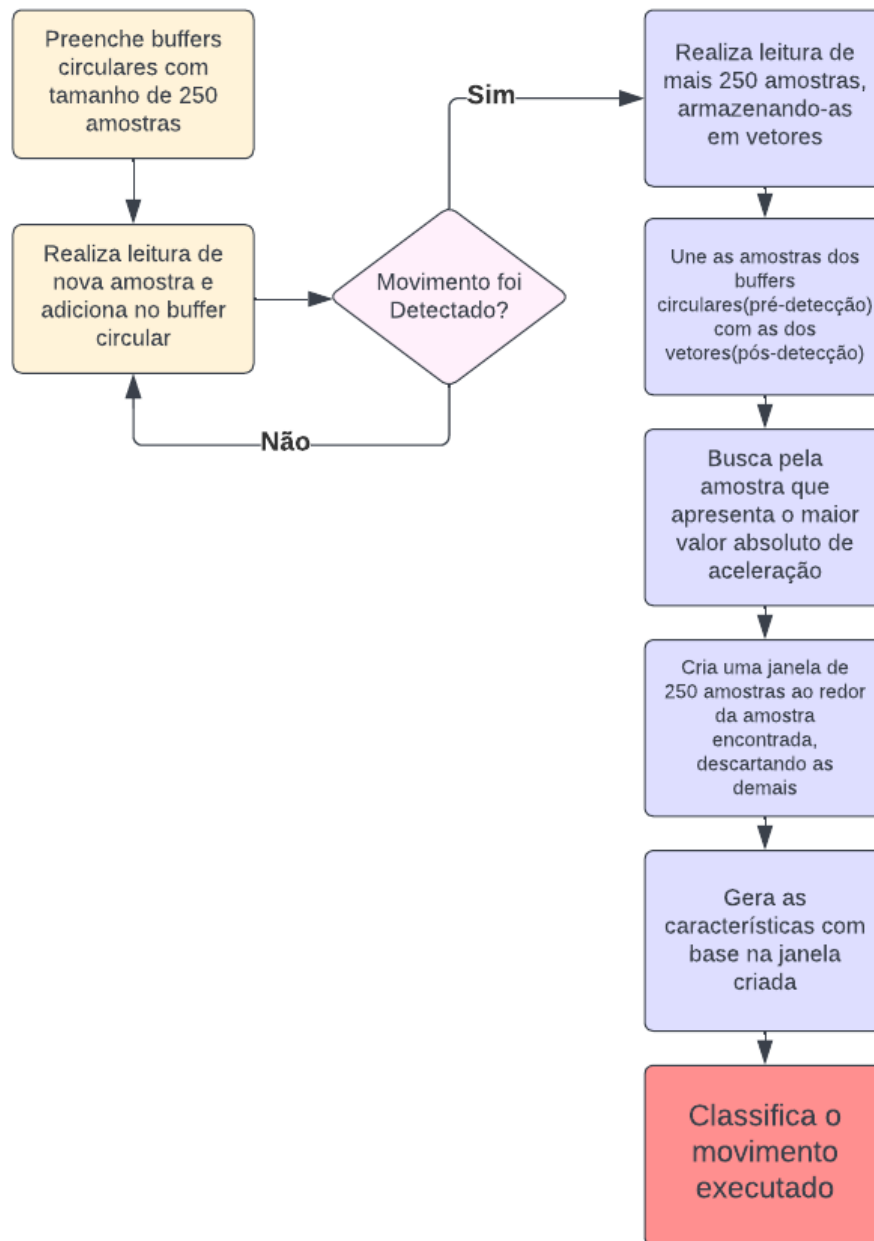
Após escolhido o melhor modelo entre os três estudados, o mesmo foi portado através da biblioteca *micromlgen*, a qual possui suporte para estes três modelos. Para realização do porte, o modelo é previamente treinado com toda a base de dados e, então, passado como parâmetro para uma função *port*, que gera sua representação na linguagem C e permite que seja utilizado na plataforma arduino.

3.2.7.1 Detecção e Classificação dos movimentos.

Para detecção da existência de movimento foi utilizado um valor pré-definido, como descrito anteriormente na seção 3.2.6. A implementação das funcionalidades de detecção e classificação no sistema embarcado partem do princípio de que a taxa de amostragem é a mesma utilizada para coleta dos dados (250 Hz).

A estratégia utilizada para realização da detecção e classificação dos movimentos é representada através do fluxograma da Figura 30. Inicialmente, foi usada uma estrutura de **Buffer Circular**, também chamado de **Fila Circular ou Buffer de Anel**. Um Buffer Circular se mostra uma boa estratégia de implementação para uma fila *FIFO* (*first in, first out*), que opera com um tamanho máximo fixo e deseja armazenar dados sequenciais de forma temporária, como é o caso do algoritmo proposto.

Figura 30 - Fluxograma do processo de detecção e classificação.



Fonte: O autor.

Em um primeiro momento, são realizadas 250 medições de aceleração e velocidade angular em cada um dos eixos, alocando-as no Buffer Circular. Para a taxa de amostragem utilizada, estas amostras representam os movimentos efetuados no último intervalo de 1 segundo. Assim, a cada nova medida realizada, a mesma é alocada na última posição do buffer e a primeira medição é descartada, mantendo-se a representação das últimas 250 medições executadas.

Depois de preenchido o buffer, a cada nova medida realizada é feita uma verificação com base na diferença média entre as duas últimas medidas. Caso esta diferença não supere um

valor predefinido, segue-se realizando novas medidas. Já, caso a média supere o valor predefinido, tem-se a detecção de movimento, a qual considera que há a execução de um golpe em andamento.

Superada a fase de detecção, são registradas mais 250 medições, representando 1 segundo de amostras após a detecção. As 250 amostras pré e pós detecção são unidas, formando assim um conjunto de 500 amostras, que representam 2 segundos de movimento. A partir de então é encontrado o ponto que contém o maior valor absoluto de aceleração entre os 3 eixos e feito um processo de janelamento ao redor deste ponto, obtendo-se assim uma janela de 250 amostras centralizadas no ponto de maior aceleração (similar a etapa de pré processamento dos dados, descrita na seção 3.2.3). Com uma janela já definida, são geradas as mesmas características obtidas a partir do procedimento descrito na seção 3.2.4 (Escolha das melhores características) e, com base nelas, o modelo realiza a classificação do golpe executado e o processo é reiniciado.

3.2.8 Análise de Desempenho de cada Modelo

Os três modelos estudados foram analisados e avaliados seguindo as métricas de taxa de acerto, precisão, sensibilidade e f1-score, latência (tempo para realizar a classificação) e tamanho em disco.² Vale salientar que os testes de taxa de acerto foram realizados com base no conjunto teste.

Para o teste de latência foram utilizadas duas abordagens: execução em *Python* e execução no sistema embarcado.

Execução em Python: para determinar a latência de inferência foi utilizada a função integrada do *Python*, *%timeit*, a qual permite avaliar o tempo de execução de uma instrução de predição. A latência foi medida tanto individualmente quanto em relação a todas as classes, sendo feita uma medida para cada repetição que compõe o conjunto teste e o resultado dado pela média das latências medidas.

Execução no sistema embarcado: para a latência de classificação no sistema embarcado foi utilizada a função *micros*, provida pela própria linguagem de programação Arduino. Esta função fornece o tempo (em micro segundos) passado desde a inicialização do programa. Para determinar a latência, os tempos medidos foram divididos em três diferentes etapas: pré-

² **Taxa de acerto:** percentual de acertos dentre todas as classes; **Precisão:** quantidade de verdadeiros positivos em relação a soma de todos os valores positivos; **Sensibilidade:** capacidade de detectar com sucesso resultados classificados como positivos; **f1-score:** média harmônica entre precisão e sensibilidade

processamento, geração das características e tempo de inferência. Para a etapa de pré-processamento, para cada movimento, são fornecidos dois conjuntos de 250 amostras (que representam os conjuntos pré e pós detecção, citados na seção 3.2.7), sendo então medido o tempo para realizar a união dos conjuntos, buscar a amostra que apresenta o maior valor absoluto de aceleração e realizar o janelamento em torno da mesma. A partir de então, são medidos os tempos para geração das características e de predição. Neste teste foram utilizados a totalidade das amostras que representam a base de dados, sendo medidos os tempos para cada um dos modelos estudados e para cada uma das classes separadamente.

Em relação ao tamanho em disco, nenhum modelo poderá ultrapassar a memória disponível de 650 KB, representando cerca de 50% da memória flash disponível do sistema embarcado.

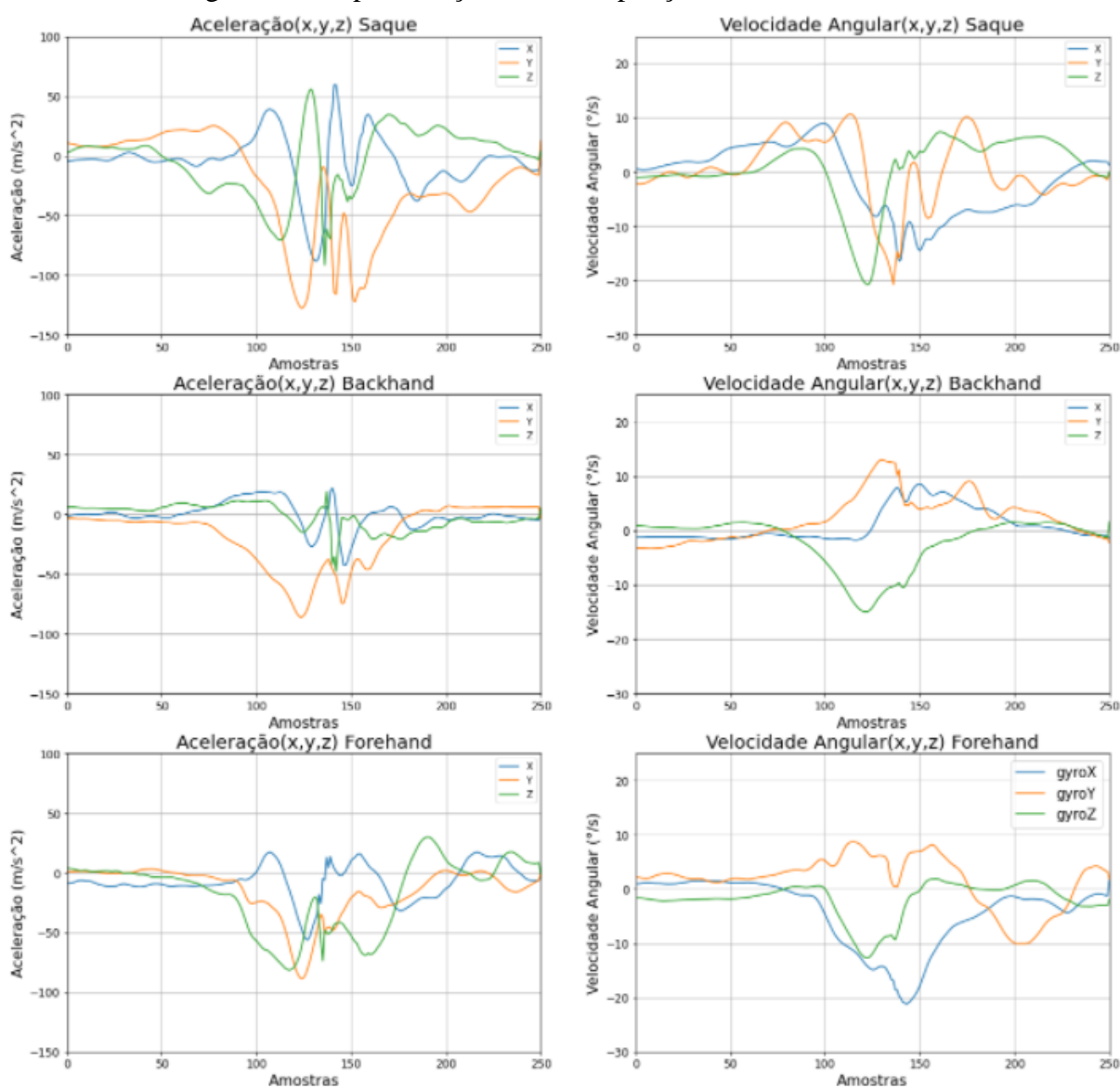
4. ANÁLISE DE RESULTADOS

Nesta etapa serão apresentados e discutidos todos os resultados obtidos durante a coleta dos dados e os testes realizados para validação da pesquisa.

4.1 ESTUDO DA BASE DE DADOS

Com base no experimento descrito na seção 3.2.2, foi feita a coleta de base de dados utilizada durante o projeto de experimentos. A base de dados é composta por conjuntos de 250 amostras de aceleração e velocidade angular nos eixos X, Y, e Z, rotulados entre Saque, *Forehand* e *Backhand*.

Figura 31 - Representação de uma repetição de cada movimento



Fonte: O autor.

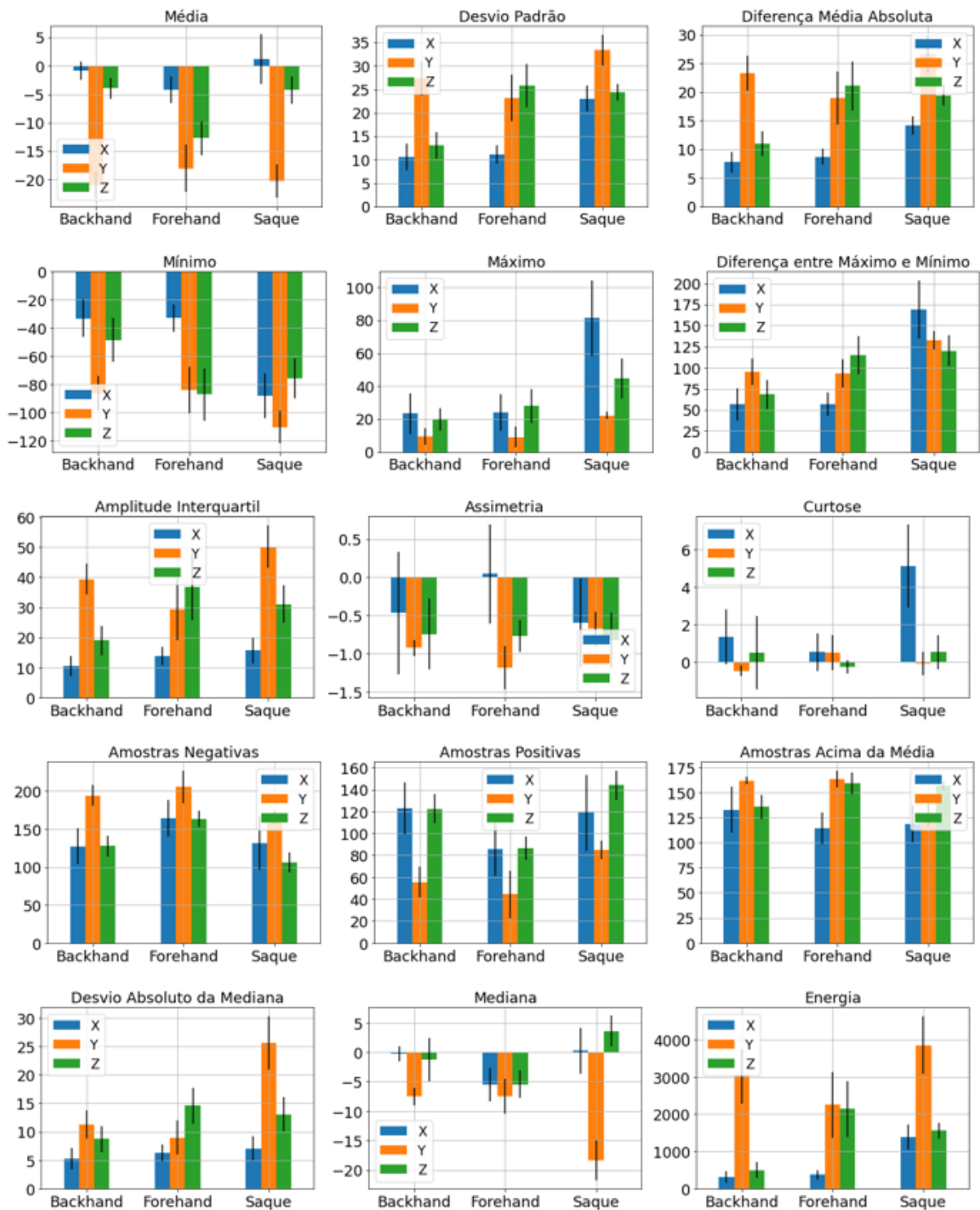
A Figura 31 apresenta a representação de uma repetição de cada um dos movimentos propostos, executados durante a coleta da base de dados. Os valores de aceleração são dados em m/s^2 e os de velocidade angular em $^\circ/s$. Avaliando-se a representação dos movimentos é possível notar visualmente algumas características de cada um dos sinais. As formas de onda para os dados de velocidade angular são mais estáveis e os dados de aceleração possuem uma variação mais abrupta na faixa central, próximo ao ponto de contato entre raquete e bola.

Para o movimento de *Backhand*, a velocidade angular apresenta, em geral, seus valores crescendo negativamente apenas para o eixo Z. Em relação a aceleração, seus maiores valores são registrados no sentido do eixo Y, negativamente. Já para o movimento de *Forehand*, a velocidade angular cresce negativamente nos eixos X e Z e a aceleração cresce negativamente nos eixos Y e Z. Para o movimento de Saque, são encontrados os maiores valores de aceleração entre os três movimentos. A velocidade angular cresce negativamente nos três eixos e também apresenta as maiores oscilações.

Considerando-se todas as 30 repetições registradas para cada movimento, foi possível gerar características que os representam, apresentadas anteriormente na seção 3.2.3.1. As Figuras 32 e 33 apresentam a média e o desvio padrão das características para cada um dos movimentos de interesse, permitindo uma avaliação inicial da capacidade de classificação de cada característica com base no tamanho das barras para cada movimento.

A partir das características de aceleração, mostradas na Figura 32, é possível fazer algumas observações. Nenhuma característica parece, sozinha, diferenciar os três movimentos. A **Média no eixo Z** aparenta possuir boa capacidade de diferenciar o movimento *Forehand* dos demais. Já a **Mediana no eixo Y** indica um potencial de diferenciação do saque, enquanto a **Energia no eixo Z** parece diferenciar o movimento de *Backhand*. Já características como **Amostras positivas, Amostras Acima da Média e Curtose** não parecem diferenciar os movimentos para nenhum dos eixos.

Figura 32 - Média e Desvio Padrão das características para aceleração.

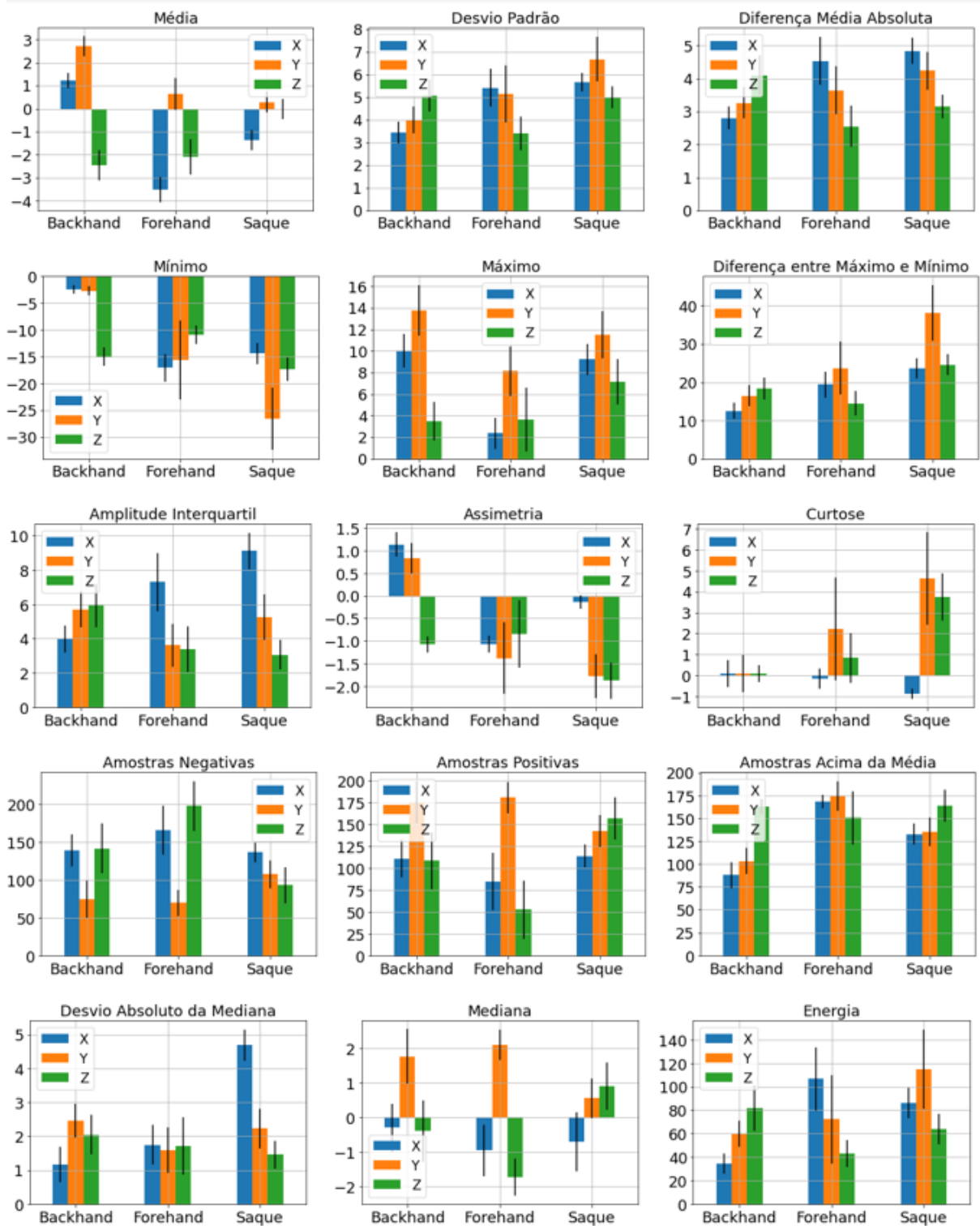


Fonte: O autor.

As características de velocidade angular, mostradas na Figura 33, aparentam possuir uma maior capacidade de diferenciação entre os movimentos. A **Média**, **Assimetria** e **Amostras Acima da Média no eixo X**, sozinhas, parecem possuir boa capacidade de

diferenciação para os três movimentos, já que o tamanho das barras é diferente para todos os movimentos, mesmo que seja considerado o desvio padrão.

Figura 33 - Média e Desvio Padrão das características para velocidade angular.



Fonte: O autor.

Após geradas as características que representarão cada um dos movimentos, é realizada a etapa de escolha das características mais relevantes para a classificação. Para tal escolha, foram testados diferentes métodos, sendo seus resultados apresentados na seção 4.2.

4.2 ESCOLHA DAS CARACTERÍSTICAS

Para definir-se as características com melhor capacidade de predição, foram testados três métodos diferentes disponíveis no módulo *sklearn.feature_selection*, como descrito na seção 3.2.4 (Escolha das melhores características). A Tabela 3 apresenta as características obtidas a partir de cada um dos métodos.

Tabela 3 - Características obtidas para os métodos selectKBest, RFE e RFECV.

Método	Características Escolhidas
<i>selectKBest</i>	<ol style="list-style-type: none"> 1. Média da velocidade angular em X 2. Mínimo da velocidade angular em X 3. Desvio absoluto da mediana da velocidade angular em X 4. Número de valores acima da média da velocidade angular em X 5. Assimetria da velocidade angular em X
<i>RFE</i>	<ol style="list-style-type: none"> 1. Média da velocidade angular em X 2. Desvio Padrão da aceleração em X 3. Mínimo da velocidade angular em X 4. Número de valores acima da média da velocidade angular em X 5. Assimetria da velocidade angular em X
<i>RFECV</i>	<ol style="list-style-type: none"> 1. Média da velocidade angular em X 2. Mínimo da velocidade angular em X 3. Número de valores acima da média da velocidade angular em X 4. Assimetria da velocidade angular em X 5. Energia da aceleração em X

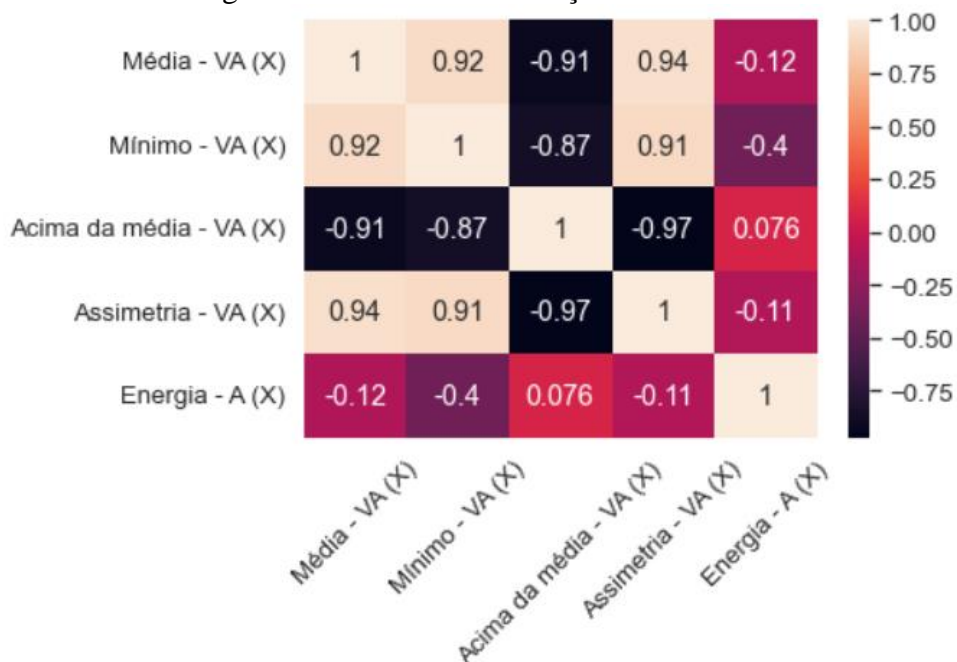
Fonte: O autor.

Nota-se que para todos os métodos utilizados as características escolhidas estão relacionadas ao sentido do eixo X. Além disso, em todos os métodos as características *média*, *mínimo* e *assimetria da velocidade angular* são escolhidas, sendo a primeira a mais relevante para todos os casos. A única característica de aceleração escolhida foi a *energia*, indicando que as características relacionadas à velocidade angular são mais relevantes para a classificação dos movimentos.

Devido ao método *RFECV* utilizar de validação cruzada para encontrar o número ótimo de características, foi decidido seguir para as próximas etapas considerando-se as características

obtidas a partir deste método. A Figura 34 apresenta uma matriz de correlação para as características escolhidas, sendo dados os coeficientes a partir do método de *Pearson*.

Figura 34 - Matriz de correlação das características escolhidas.



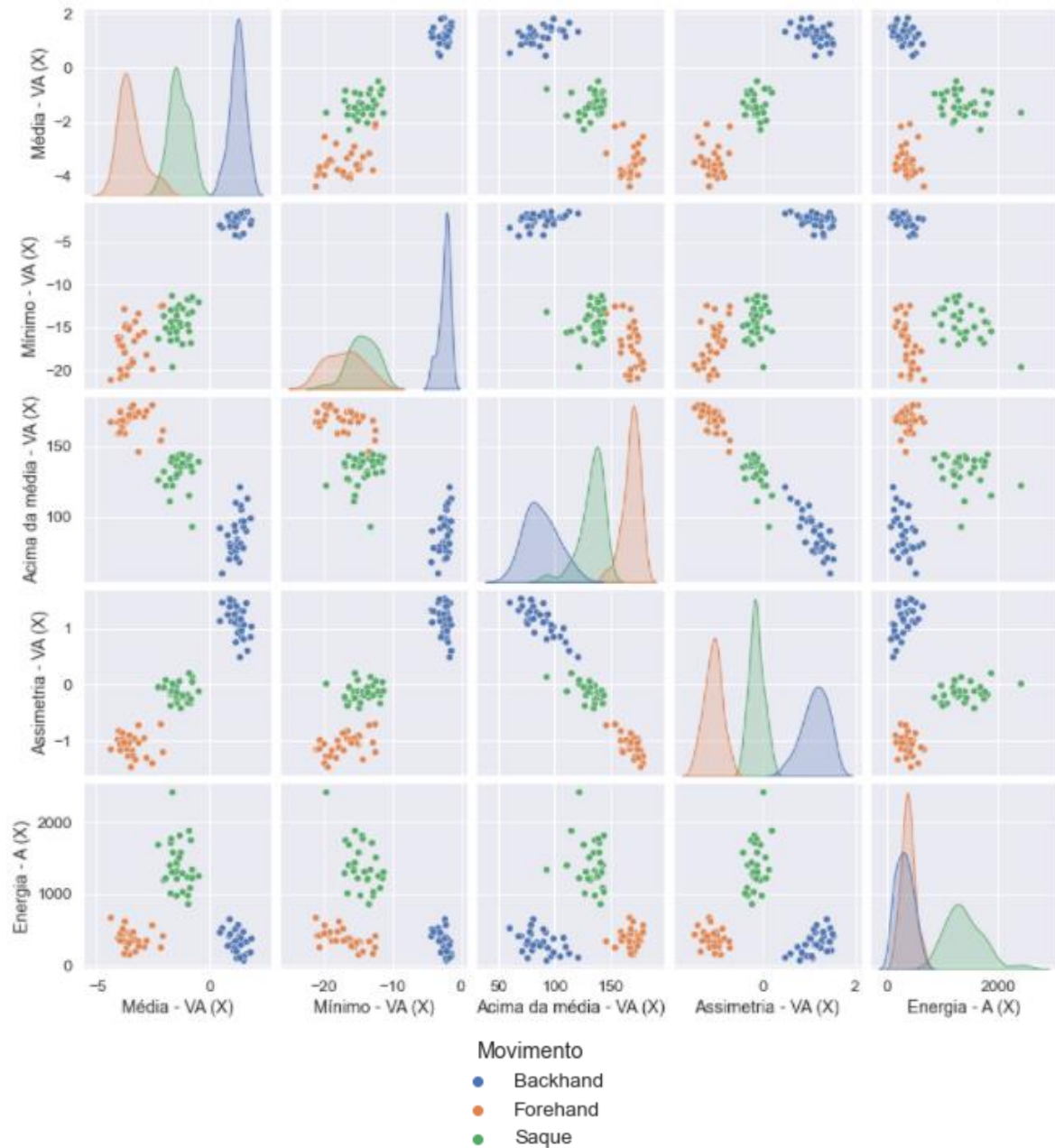
Fonte: O autor.

Com base na matriz de correlação das características, nota-se uma forte correlação positiva entre **Mínimo - VA (X) e Média - VA (X)**. Já **Acima da Média - VA (X) e Média - VA (X)** apresentam uma forte correlação negativa, assim como **Acima da média - VA (X) e Assimetria - VA (X)**.³

Na Figura 35 é possível observar o relacionamento entre as amostras de cada característica e também sua capacidade de diferenciar as classes. Quando não há uma sobreposição entre as amostras de cada classe, é indicado uma boa capacidade de diferenciação. Além disso, é possível identificar fortes correlações entre as características no caso da distribuição das amostras formarem retas de inclinação positiva (correlação positiva) e negativa (correlação negativa).

³ VA: velocidade angular;
A: aceleração

Figura 35 - Relacionamento entre as amostras de cada característica.



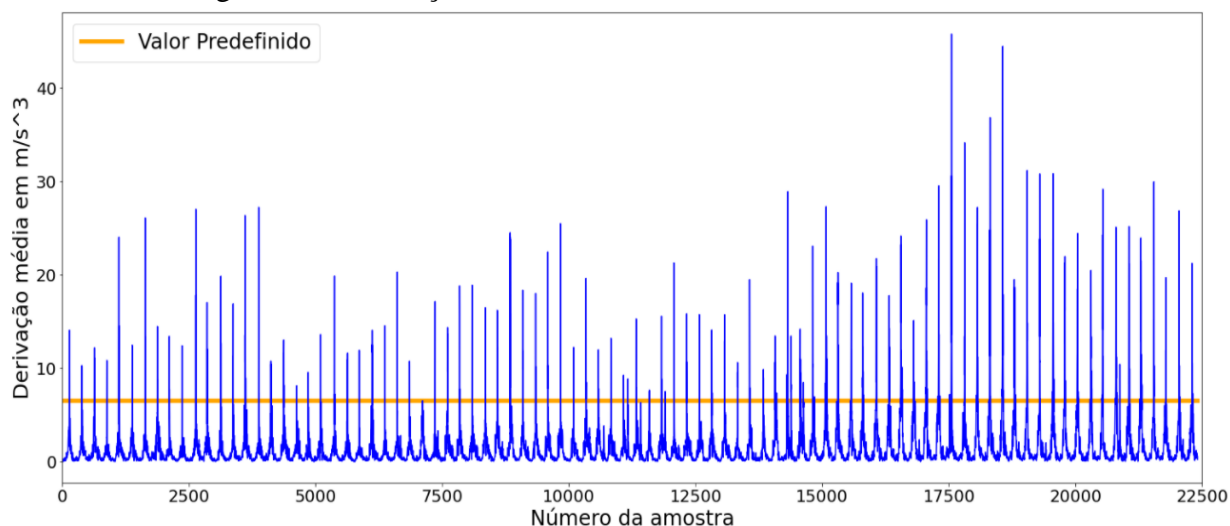
Fonte: O autor.

Nota-se pelos histogramas que os atributos *Média - VA (X)* e *Assimetria - VA (X)* parecem, sozinhas, separar bem os movimentos de interesse, já que quase não ocorre o cruzamento entre as bandas. Além disso, é possível perceber uma forte correlação negativa entre as variáveis *Acima da média - VA (X)* e *Assimetria - VA (X)* e *Acima da média - VA (X)* e *Média - VA (X)*. Uma forte correlação positiva é percebida entre *Mínimo - VA (X)* e *Média - VA (X)* e todos os pares de características parecem possuir boa capacidade de diferenciação dos movimentos.

4.3 DEFINIÇÃO DOS PARÂMETROS DE DETECÇÃO

Conforme descrito na seção 3.2.6 (Método de detecção dos movimentos), foi utilizado o menor valor de derivação média para os pontos de contato entre raquete e bola, considerando-se todos os movimentos que compõem a base de dados coletada. A Figura 36 representa a derivação média entre dois pontos para todos os movimentos adquiridos, indicando também o valor mínimo escolhido para detecção.

Figura 36 - Derivação Média de todos os movimentos coletados.



Fonte: O autor.

A média encontrada entre os valores de pico em cada uma das repetições foi de $19,48 \pm 7,68 m/s^3$, sendo que o valor mínimo, usado como parâmetro para detecção, foi de $6,49 m/s^3$.

4.4 OTIMIZAÇÃO DOS HIPERPARÂMETROS E GERAÇÃO DOS MODELOS

Para otimização dos hiperparâmetros, foi utilizado o método de busca de grade a partir do módulo *sklearn.model_selection.GridSearchCV*. Como descrito na seção 3.2.5, foi realizada a otimização dos hiperparâmetros para três diferentes modelos de aprendizado de máquina supervisionado: Árvore de Decisão, Floresta Aleatória e SVM.

Com as características de cada rótulo já definidas e o conjunto de dados de treinamento, pode-se fazer, inicialmente, uma avaliação dos modelos a partir da função *cross_val_score* da módulo *sklearn.model_selection*. Esta função executa a validação cruzada de acordo com o número de partições informadas via parâmetro *cv*, retornando para cada partição uma métrica

de performance, que por padrão é a taxa de acerto. O *cv* utilizado foi 5 e os modelos foram avaliados com os hiperparâmetros padrão da biblioteca *sklearn*.

Tabela 4 - Média da taxa de acerto das partições para o teste *cross_val_score*.

Modelo	Média da taxa de acerto	Desvio Padrão
Árvore de Decisão	98,5%	3,1%
Floresta Aleatória	100%	0
SVM	82,4%	7,6%

Fonte: O autor.

Observa-se que apenas com a escolha das melhores características já foi possível obter uma taxa de acerto de 100% para o modelo Floresta Aleatória. Já para o modelo Árvore de Decisão, a taxa de acerto média ficou em 98,5% e desvio padrão de 3,1%. No caso do SVM, obteve-se os piores resultados durante a realização do teste, com uma média de 82,4% e desvio padrão de 7,6%.

4.4.1 Otimização e Avaliação dos modelos

Para a Árvore de Decisão, o modelo foi otimizado considerando-se os seguintes hiperparâmetros: Profundidade máxima da árvore, número máximo de nós folha e número mínimo de amostra para uma divisão de nós.

Em relação a Floresta Aleatória, que já havia apresentado uma acurácia de 100% em relação ao conjunto de treinamento, o modelo foi otimizado considerando-se os seguintes hiperparâmetros: número de árvores, profundidade máxima das árvores e número mínimo de amostra para uma divisão de nós.

Para a SVM, que havia apresentado o pior resultado em relação ao conjunto de treinamento, o modelo foi otimizado considerando-se os seguintes hiperparâmetros: C, gamma e kernel.

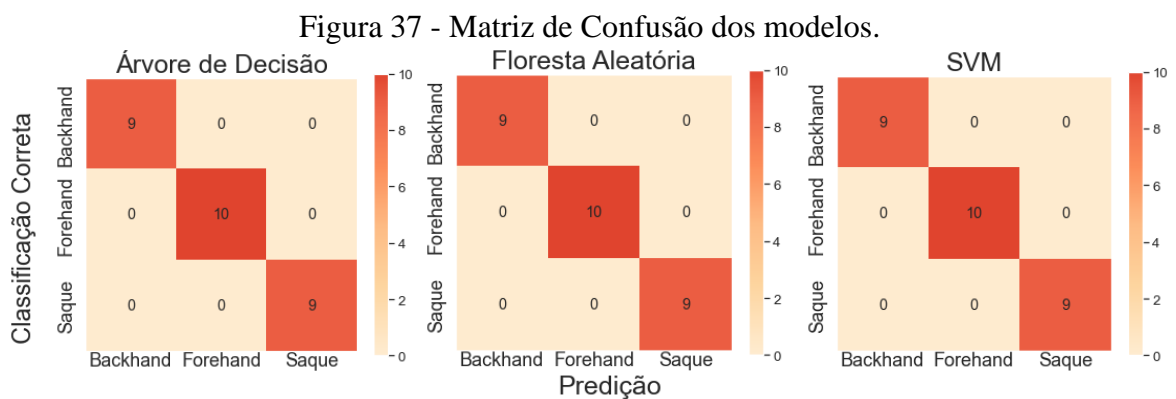
A Tabela 5 apresenta os hiperparâmetros obtidos através da busca de grade para todos os modelos.

Tabela 5 - Hiperparâmetros Escolhidos.

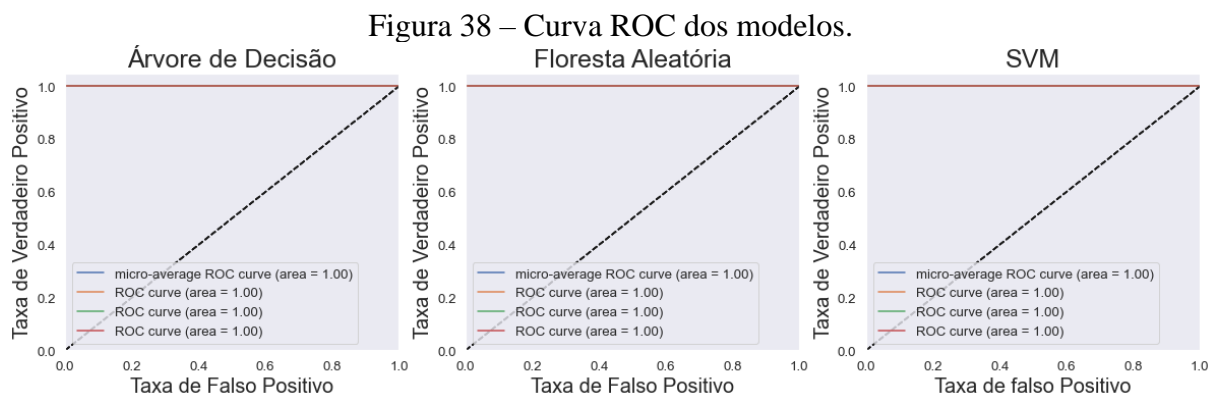
	Hiperparâmetro	Intervalo	Escolhido
Árvore de Decisão	Profundidade máxima da árvore	2 → 10, <i>passo</i> 2	2
	Número máximo de nós folha	2 → 100, <i>passo</i> 1	3
	Número mínimo de amostras para uma divisão de nós	2 → 10, <i>passo</i> 1	2
Floresta Aleatória	Número de árvores	2 → 20, <i>passo</i> 2	2
	Profundidade máxima das árvores	2 → 10, <i>passo</i> 2	2
	Número mínimo de amostras para uma divisão de nós	2 → 10, <i>passo</i> 1	2
SVM	C	0,1 ; 1 ; 10 ; 100 ; 1000	0,1
	Gamma	1 ; 0,1 ; 0,01 ; 0,001 ; 0,0001	1
	Kernel	linear, rbf	linear

Fonte: O Autor.

Os modelos, com os hiperparâmetros otimizados, foram testados frente ao conjunto de dados teste. As matrizes de confusão e curvas ROC, para cada um dos modelos otimizados, são apresentadas nas Figuras 37 e 38.



Fonte: O autor.



Fonte: O autor.

Considerando todos os três modelos otimizados e avaliados em relação ao conjunto de teste, obteve-se um resultado de 100% para todos os modelos nas métricas de taxa de acerto, precisão, sensibilidade e f1-score, indicando também a não ocorrência de sobreajuste⁴ nos modelos. Sendo assim, todos os modelos se mostraram igualmente capazes de realizar a classificação dos movimentos. Além disso, pode-se observar que os melhores hiperparâmetros obtidos, como por exemplo o *número de árvores*, *profundidade das árvores* e *número máximo de nós folha*, possuem um valor considerado pequeno, indicando que as características escolhidas possuem boa capacidade de diferenciar cada movimento e acaba-se não exigindo muita complexidade dos modelos.

É importante salientar que a não generalização da base de dados, ou seja, o fato da coleta ter sido feita apenas com um tenista, inevitavelmente influencia no resultado do processo de classificação, já que os modelos foram treinados e avaliados com movimentos de um mesmo jogador. Outros fatores que podem impactar diretamente nos resultados são o tamanho da base de dados e o número restrito de classes que foram propostas para classificação (*saque*, *forehand* e *backhand*). Além disso, a não caracterização do sistema e dos sensores isoladamente não permite a extrapolação dos dados coletados para o uso em outros sistemas.

Inicialmente não se sabia como os modelos se comportariam em relação às classes escolhidas e se optou por uma restrição das mesmas. Entretanto, o mesmo procedimento de escolha das características e otimização dos hiperparâmetros poderia ser aplicado para classificação de um número maior de classes e de uma base de dados mais generalizada, sendo esperado que a complexidade dos modelos aumente para estes casos.

⁴ Sobreajuste: termo usado para descrever quando um modelo se ajusta muito bem ao conjunto de dados de treinamento, mas se mostra ineficaz para prever novos resultados.

4.4.2 Avaliação da Latência e Tamanho dos Modelos

Para o critério de latência da predição executada em *Python*, foi realizado um teste para cada um dos modelos utilizando a função integrada do *Python*, *%timeit*. As predições foram realizadas sobre o conjunto de dados teste. A Tabela 6 apresenta a latência média e o desvio padrão para todos os movimentos e também para cada movimento, individualmente.

Tabela 6 – Latência média de predição executada em *Python* com conjunto de dados teste.

Movimento	Latência		
Backhand	2,79 ± 0,51 ms	3,34 ± 0,751 ms	2,25 ± 0,46 ms
Forehand	2,52 ± 0,54 ms	2,95 ± 0,207 ms	2,13 ± 0,11 ms
Saque	2,48 ± 0,48 ms	3,54 ± 0,482 ms	2,74 ± 0,97 ms
Todos	2,60 ± 0,50 ms	3,29 ± 0,574 ms	2,38 ± 0,67 ms
-	Árvore de decisão	Floresta aleatória	SVM

Fonte: O autor.

Observa-se que o modelo que apresentou menor latência de predição foi SVM, com uma média de $2,38 \pm 0,67$ ms para todos os movimentos preditos. Já a Floresta Aleatória apresentou o pior resultado entre os demais. Vale ressaltar que, de acordo com as necessidades de projeto, todos os modelos atendem ao critério de latência na predição já que a maior latência obtida ($3,54 \pm 0,48$ ms para o movimento Saque utilizando-se a Floresta Aleatória) é muito menor do que o tempo de execução de cada movimento.

Já, para o teste de latência no sistema embarcado, no qual foram medidos os tempos de pré-processamento, geração das características e predição para cada um dos modelos, os resultados podem ser observados na Tabela 7.

Nota-se, para o teste de latência no sistema embarcado, que o modelo SVM foi o que apresentou melhores resultados no tempo de predição, obtendo um tempo de predição médio de 2384 ± 181 us. Os outros dois modelos, árvore de decisão e floresta aleatória, apresentaram resultados semelhantes em relação a esta métrica, atingindo um tempo médio de predição entre todos os movimentos de 3193 ± 402 us e 3194 ± 401 us, respectivamente.

Ainda, observa-se uma semelhança entre os tempos obtidos para o modelo SVM na execução em *Python* e no sistema embarcado, porém isso se trata apenas de uma coincidência. Para os demais modelos, também atingiu-se pequenas diferenças na comparação entre os dois testes.

Tabela 7 – Latência média de predição no sistema embarcado com toda a base de dados.

	Movimento	Backhand	Forehand	Saque	Todos
Etapas	Pré-processamento	1594 ± 2 us	1629 ± 48 us	1664 ± 38 us	1629 ± 44 us
	Geração de características	3260 ± 39 us	3504 ± 2 us	3574 ± 38 us	3446 ± 142 us
	Predição Árvore de Decisão	2985 ± 712 us	3298 ± 1 us	3297 ± 4 us	3193 ± 402 us
	Predição Floresta Aleatória	2986 ± 703 us	3297 ± 1 us	3297 ± 4 us	3194 ± 401 us
	Predição SVM	2291 ± 310 us	2431 ± 1 us	2430 ± 1 us	2384 ± 181 us

Fonte: O Autor.

Em relação ao tempo de pré-processamento, o qual inclui as etapas de unir as amostras pré e pós detecção, encontrar o máximo valor de aceleração e realizar o janelamento em torno deste ponto, o valor médio obtido para todos os movimentos foi de 1629 ± 44 us. Já para a etapa de geração das características obteve-se uma latência média para todos os movimentos de 3446 ± 142 us.

Em relação ao tamanho dos modelos, a avaliação foi feita utilizando o módulo *Pickle*, que implementa protocolos binários para serialização de estruturas de objetos *Python*. Assim, é possível converter a estrutura do modelo em um fluxo de bytes e medir seu tamanho. A Tabela 8 mostra um comparativo do tamanho entre cada um dos modelos.

Tabela 8 - Tamanho dos modelos.

Modelo	Tamanho do modelo
Árvore de Decisão	4,04 KB
Floresta Aleatória	5,22 KB
SVM	12,39 KB

Fonte: O autor.

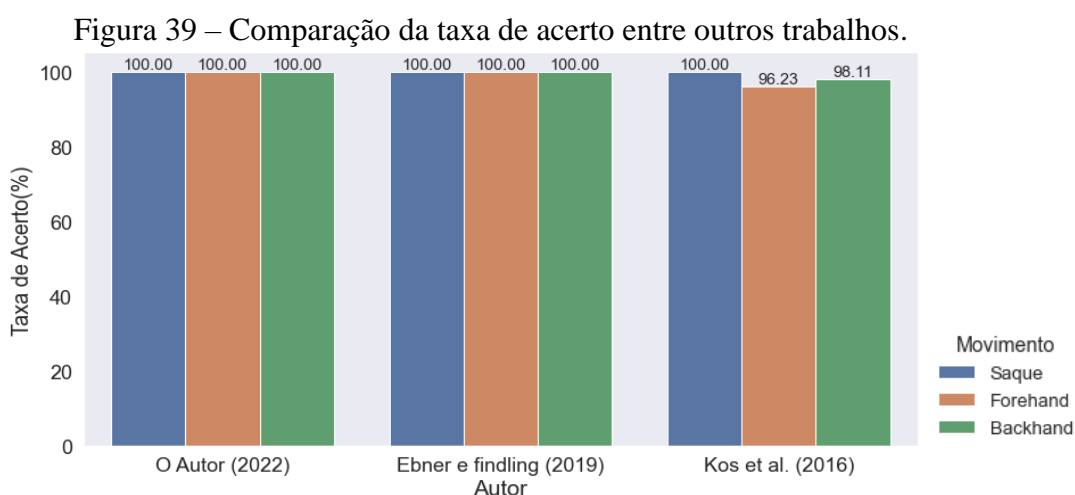
Com relação ao espaço em disco, o menor modelo foi a Árvore de Decisão, com $4,04$ KB e o maior foi o SVM, com $12,39$ KB. Os três modelos são suportados e podem ser implementados no sistema desenvolvido, não sendo seu tamanho um fator impeditivo, e nem necessário o uso de técnicas de poda computacional.

4.5 COMPARAÇÃO COM TRABALHOS DA LITERATURA

Outros trabalhos da literatura se propõem a realizar a classificação de movimentos de tênis através de dispositivos vestíveis. Destes, dois apresentam condições similares ao presente trabalho, baseando-se em um dispositivo acoplado ao pulso do jogador e utilizando de dois sensores, um acelerômetro e um giroscópio, para classificação dos movimentos.

Ebner e Findling (2019) trazem uma abordagem mais complexa, propondo a classificação de 8 movimentos, estando o saque, *forehand* e *backhand* inclusos nestes, e comparando modelos como SVM, KNN e CART para classificação. KOS et al (2016) buscam uma abordagem inicial mais simples, tendo como objetivo a classificação somente dos três movimentos principais do tênis e utilizando de valores de velocidade angular pré-definidos para realizar a classificação. Ambos os trabalhos possuem uma base de dados mais generalizada que a utilizada na presente pesquisa, sendo a do primeiro trabalho coletada com 6 e a do segundo com 3 jogadores.

Diferentemente das outras pesquisas, essa compara e avalia modelos como Árvore de Decisão, Floresta Aleatória e SVM em relação a classificação de movimentos neste esporte. A Figura 39 apresenta uma comparação com os dois trabalhos citados anteriormente.



Fonte: O autor.

Observa-se que para os três trabalhos a taxa de acerto para os movimentos de saque, *forehand* e *backhand*, é bem elevada. Ebner e Findling (2019) conseguem uma taxa de acerto de 100% para estes movimentos. Salienta-se que para a presente pesquisa não há generalização da base de dados, diferentemente dos demais, portanto, não se pode afirmar que os resultados foram superiores em relação aos outros trabalhos.

5. CONCLUSÕES

O presente trabalho apresentou o processo de desenvolvimento de sistema vestível para detecção e classificação dos três principais movimentos do tênis (saque, *forehand* e *backhand*), sendo ele relevante dentro de diversas aplicações dentro do esporte.

Três diferentes modelos de aprendizado de máquina foram comparados e avaliados em relação ao processo de classificação (Árvore de Decisão, Floresta Aleatória e SVM). Para todos os movimentos, obteve-se uma taxa de acerto média de 100 %, sendo a menor latência obtida na predição de 2384 ± 181 us, para o modelo SVM no sistema embarcado. A etapa de otimização dos hiperparâmetros dos modelos, pelos pequenos valores obtidos, e o estudo da base de dados, indicam que as características escolhidas possuem boa capacidade de diferenciar cada um dos movimentos, o que por sua vez diminui a complexidade dos modelos para a classificação.

Além disso, a não generalização da base de dados, sendo composta por movimentos de apenas um tenista, e a avaliação das taxas de acerto com base em um subconjunto teste da mesma base de dados, não permite avaliar a capacidade de generalização do dispositivo desenvolvido. Para tal avaliação, idealmente se faria necessário a coleta dos dados com um número maior de tenistas, sendo assim possível avaliar as taxas de acerto tanto com os tenistas que compõem a base de dados quanto com outros. Outro fator relevante é a não caracterização do sistema a dos sensores isoladamente, não permitindo extrapolar as análises e validar a base de dados para o uso em outros sistemas.

O sistema desenvolvido se mostrou capaz classificar os movimentos propostos para um mesmo tenista e a complexidade dos modelos gerados deixa margem para tentativa de classificação de um maior número de movimentos. Vale ressaltar que tanto a ferramenta desenvolvida para coleta dos dados quanto o processo de geração e otimização dos modelos foram pensados de forma a automatizar parte da execução destas etapas, prevendo, por exemplo, a inclusão de novos movimentos.

5.1 TRABALHOS FUTUROS

Os resultados apresentados na pesquisa demonstram que há possibilidades de melhorias no sistema desenvolvido. Iniciando pela base de dados, o indicado é que a mesma inclua um número maior de tenistas, possibilitando assim uma avaliação melhor com relação a capacidade de generalização dos modelos quanto a classificação. Além disso, há necessidade de

caracterização do sistema de medidas para validação da base de dados, permitindo assim a extrapolação dos dados para uso em outros sistemas.

Como as características escolhidas apresentam boa capacidade de diferenciar as três classes propostas, um maior número de movimentos, como voleio *forehand*, voleio *backhand*, *smash*, e variações como *top spin* e *slice*, poderiam ser incluídas na base de dados e suas classificações avaliadas para os mesmos modelos.

Por fim, o dispositivo poderia ser avaliado durante a realização de partidas, em ambientes com menor controle, para identificação da influência de uma enorme quantidade de movimentos que ocorrem entre e durante cada jogada.

REFERÊNCIAS BIBLIOGRÁFICAS

AIBIN, M. Multiple Classification Using Support Vector Machines, 2021. Disponível em: <<https://www.baeldung.com/cs/svm-multiclass-classification>>. Acessado em: 20 de agosto de 2021.

BENAGES PARDO, Luis; BULDAIN PEREZ, David; ORRITE URUNUELA, Carlos. Detection of tennis activities with wearable sensors. **Sensors**, v. 19, n. 22, p. 5004, 2019.

BONESSO, Diego. Estimação dos parâmetros do kernel em um classificador svm na classificação de imagens hiperespectrais em uma abordagem multiclasse. 2013.

BRAGA NETO, Ludgero. **Características dinâmicas e eletromiográficas do forehand e backhand em tenistas: uma perspectiva biomecânica para avaliar o desempenho**. 2008. Tese de Doutorado. Universidade de São Paulo.

CAMOMILLA, Valentina et al. Trends supporting the in-field use of wearable inertial sensors for sport performance evaluation: A systematic review. **Sensors**, v. 18, n. 3, p. 873, 2018.

CONNAGHAN, Damien et al. Recognition of tennis strokes using key postures. In: **IET Irish Signals and Systems Conference (ISSC 2010)**. IET, 2010. p. 245-248.

CROSS, Rod. A double pendulum model of tennis strokes. **American Journal of Physics**, v. 79, n. 5, p. 470-476, 2011.

DE ALVARENGA JÚNIOR, Wagner José. Métodos de otimização hiperparamétrica: um estudo comparativo utilizando árvores de decisão e florestas aleatórias na classificação binária. 2018.

EBNER, Christopher J.; FINDLING, Rainhard Dieter. Tennis stroke classification: comparing wrist and racket as IMU sensor position. In: **Proceedings of the 17th international conference on advances in mobile computing & multimedia**. 2019. p. 74-83.

ELLIOTT, B. C.; WOOD, G. A. The biomechanics of the foot-up and foot-back tennis service techniques. **Aust J Sports Sci**, v. 3, n. 2, p. 3-6, 1983.

ESPINOSA, Hugo G.; LEE, Jim; JAMES, Daniel A. THE INERTIAL SENSOR: A BASE PLATFORM FOR WIDER ADOPTION IN SPORTS SCIENCE APPLICATIONS. **Journal of Fitness Research**, v. 4, n. 1, 2015.

GREEN, S; IRWIN, P. The Hawk-Eye Tennis System, 2016. Disponível em: <<https://resources.platform.pulselive.com/HawkEye/document/2016/07/29/ec84be34-2375-4b5a-8854-917e0e7021f0/HawkEyeinTennis2016.pdf>>. Acesso em: 23 de novembro de 2021

HAIGH, Karen Zita et al. Machine learning for embedded systems: A case study. **BBN Technologies: Cambridge, MA, USA**, v. 8571, p. 1-12, 2015.

HOFER, D. The open stance forehand, 2004. Disponível em: <<http://www.geocities.com/hofertennis@sbcglobal.net/tiparchive3.html>>. Acessado em: 14 de maio de 2022.

KOS, Marko; KRAMBERGER, Iztok. A wearable device and system for movement and biometric data acquisition for sports applications. **IEEE Access**, v. 5, p. 6411-6420, 2017.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.

KRÜGER, GEOVANE. Estudo Preliminar das Características Eletromiográficas da Articulação do Ombro no Saque Flat no Tênis: Comparação Entre as Técnicas Foot-back e Foot-up. **Monografia (Trabalho de conclusão de curso em Educação Física) Universidade Federal de Santa Catarina**, 2014.

KUMAR, V; MINZ, S. Feature Selection: A literature Review. School of Computer and Systems Sciences, Jawaharlal Nehru University, 2014.

LANSDORT, R. Balancing act. Tennis, v. 1, p. 64-65, 2004.

LE SAGE, T. et al. Embedded programming and real-time signal processing of swimming strokes. **Sports Engineering**, v. 14, n. 1, p. 1-14, 2011.

LOPES, Lucas Pereira. Predição do preço do café Naturais Brasileiro por meio de modelos de statistical machine learning. **Sigmae**, v. 7, n. 1, p. 1-16, 2018.

LUINGE, Henk J.; VELTINK, Peter H. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. **Medical and Biological Engineering and computing**, v. 43, n. 2, p. 273-282, 2005.

MAHESH, Batta. Machine learning algorithms-a review. **International Journal of Science and Research (IJSR)**.**[Internet]**, v. 9, p. 381-386, 2020.

MANIKANDAN, K. Guru et al. Investigations on suitability of MEMS based accelerometer for vibration measurements. **Materials Today: Proceedings**, v. 45, p. 6183-6192, 2021.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Indução de regras e árvores de decisão. **Sistemas Inteligentes-fundamentos e aplicações**, v. 1, p. 115-139, 2003.

NASTESKI, Vladimir. An overview of the supervised machine learning methods. **Horizons**. **b**, v. 4, p. 51-62, 2017.

NETO, Ludgero Braga. **Estudo de características dinâmicas e eletromiográficas do saque no tênis-comparação entre duas técnicas**. 2003. Tese de Doutorado.

PEI, Weiping et al. An embedded 6-axis sensor based recognition for tennis stroke. In: **2017 IEEE International Conference on Consumer Electronics (ICCE)**. IEEE, 2017. p. 55-58.

PONTE, Caio; CAMINHA, Carlos; FURTADO, Vasco. Otimização de florestas aleatórias através de ponderação de folhas em árvore de regressão. In: **Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional**. SBC, 2020. p. 698-708.

SABATINI, Angelo Maria. Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing. **Sensors**, v. 11, n. 2, p. 1489-1525, 2011.

SLIWA, Benjamin; PIATKOWSKI, Nico; WIETFELD, Christian. LIMITS: Lightweight machine learning for IoT systems with resource limitations. In: **ICC 2020-2020 IEEE International Conference on Communications (ICC)**. IEEE, 2020. p. 1-7.

SWAIN, Philip H.; HAUSKA, Hans. The decision tree classifier: Design and potential. **IEEE Transactions on Geoscience Electronics**, v. 15, n. 3, p. 142-147, 1977.

TRABERT, T. **Magnifique! Tennis**, v. 1, p. 68-69, 1999.

WHITESIDE, David et al. Monitoring hitting load in tennis using inertial sensors and machine learning. **International journal of sports physiology and performance**, v. 12, n. 9, p. 1212-1217, 2017.

ZHANG, Zhongpeng. Model Pruning Techniques for Boosting the Inference Efficiency on Embedded Systems. In: **2021 2nd International Conference on Computing and Data Science (CDS)**. IEEE, 2021. p. 119-124.

ZHAO, Hongyang et al. TennisEye: tennis ball speed estimation using a racket-mounted motion sensor. In: **Proceedings of the 18th International Conference on Information Processing in Sensor Networks**. 2019. p. 241-252.