



UNIVERSIDADE FEDERAL DE ITAJUBÁ

Daniel Ferreira Lara
Enzo Yukio Chinen
Matheus Siston Galdino

Table Tennis Motion Classifier

ITAJUBÁ
2023

Daniel Ferreira Lara
Enzo Yukio Chinen
Matheus Siston Galdino

Table Tennis Motion Classifier

Relatório de Projeto Avaliativo
submetido aos prof. Marcelo José Rovai
e José Alberto Ferreira Filho, como
requisito parcial para aprovação na
disciplina de TINYML - Aprendizado
de máquina aplicado para dispositivos
IOT embarcados (IESTI01) do curso
de graduação em Engenharia de
Computação da Universidade Federal
de Itajubá.

ITAJUBÁ
2023

1. Introdução

O tênis de mesa surgiu na Inglaterra durante a segunda metade do século XIX, originalmente chamado *Ping-Pong*. O primeiro campeonato mundial foi realizado em 1926, e desde então, o esporte é praticado ao redor do mundo. ([Confederação Brasileira de Tênis de Mesa, 2023](#))

Com a evolução do esporte, a modalidade passou a ter movimentos básicos praticados pelos jogadores. Dentre estes, estão: saque (*serve*), *forehand*, *backhand* e *chop* (cozinhada). Foi realizado um projeto com o objetivo de identificar cada um destes movimentos através do Arduino Nano integrado a uma raquete. O dispositivo Nano BLE Sense possui um IMU de 9 eixos, dos quais foram utilizados 6 eixos, três do acelerômetro e três do giroscópio. Foram treinados três modelos de aprendizado de Inteligência Artificial (IA) desenvolvidos através da plataforma Edge Impulse. ([Arduino, 2023](#))

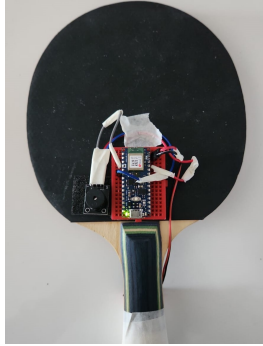
É importante ressaltar que durante o desenvolvimento do projeto, o tema foi modificado, sendo antigamente um classificador de lixo. Tal mudança foi realizada devido ao fato de que o modelo treinado não possuía uma boa acurácia, ou seja, estava abaixo de 60%. Além disso, a câmera do dispositivo não possuía resolução necessária para o teste do modelo.

2. Metodologia

2.1 Obtenção de Dados

Para a captação de dados foi utilizado apenas o Arduino Nano BLE Sense sem o *shield*, acoplado a uma *protoboard* e fixa em uma raquete de tênis de mesa, utilizando o sensor IMU LSM9DS1 integrado na *development board*. A melhor forma encontrada para posicionar a placa foi na face onde o jogador posiciona os dedos da mão, que é contrária àquela na qual a bola colide. Logo a montagem final do dispositivo na raquete é a imagem a seguir:

Figura 1
Montagem para coleta de dados e inferência do modelo



Dentre os movimentos utilizados no esporte foram definidos quatro, os quais são:

1. *Idle*
2. *Backhand*
3. *Cozinhada*
4. *Forehand*
5. *Serve* (Saque)

Além dos movimentos conhecidos, foi definida uma classe *idle*, que representa os momentos em que o jogador aguarda o oponente para fazer seu movimento ou apenas fica com a raquete estática. Após a definição das classes, foi feita a coleta com o dispositivo conectado ao Edge Impulse utilizando o método WebUSB. Os *samples* foram de 19000 ms de frequência de 100 Hz, os quais foram divididos em frações menores entre 1000 ms e 2000 ms de acordo com a duração do movimento. Exemplos dos dados coletados de cada classe são:

Figura 2
Exemplos de dados



Cada um dos movimentos têm suas características próprias representadas pelos sinais recebidos do giroscópio, acelerômetro e magnetômetro do sensor inercial IMU. Para a criação do modelo foram coletados 14 minutos e 55 segundos de dados, com 77% para treino e 23% para teste, divididos do seguinte modo:

Figura 3
Tabela de tempos das classes

Classe	Treino	Teste	Total
Idle	3m 1s	1m 3s	4m 4s
Backhand	1m 57s	29s	2m 28s
Cozinhada	2m 6s	34s	2m 40s
Forehand	2m 15s	38s	2m 53s
Serve	2m 12s	39s	2m 51s

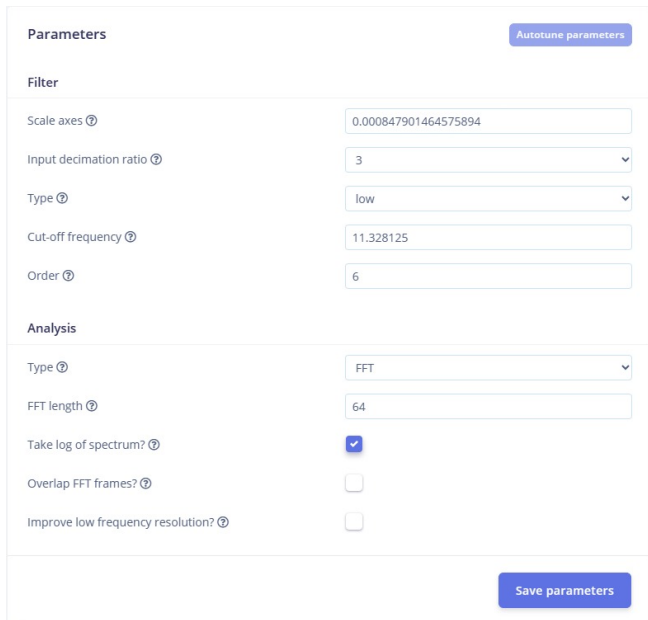
2.2 Definição de Parâmetros

Na criação do impulso foi utilizada uma janela de tamanho de 2000 ms, além de um incremento de janela de 200 ms. A frequência utilizada foi a mesma dos dados obtidos, 100 Hz. O bloco de processamento foi uma análise espectral, onde os eixos de entrada selecionados foram os três eixos do

acelerômetro e os três eixos do giroscópio. Os dados do magnetômetro não foram utilizados. Para o bloco de aprendizado foi selecionado um classificador que tem como saída as cinco classes coletadas: *backhand*, *cozinha*, *forehand*, *serve* e *idle*.

Os três modelos propostos utilizaram diferentes parâmetros para análise espectral dos dados obtidos pelo IMU. Primeiramente foi utilizado a opção de *autotune* dos parâmetros, onde o Edge Impulse retornou a seguinte configuração:

Figura 4
Parâmetros autotune



The image shows a web interface for configuring 'Autotune parameters'. It is divided into two main sections: 'Filter' and 'Analysis'. The 'Filter' section includes settings for 'Scale axes' (0.000847901464575894), 'Input decimation ratio' (3), 'Type' (low), 'Cut-off frequency' (11.328125), and 'Order' (6). The 'Analysis' section includes 'Type' (FFT), 'FFT length' (64), 'Take log of spectrum?' (checked), 'Overlap FFT frames?' (unchecked), and 'Improve low frequency resolution?' (unchecked). A 'Save parameters' button is located at the bottom right.

Parameters	
Autotune parameters	
Filter	
Scale axes ⓘ	0.000847901464575894
Input decimation ratio ⓘ	3
Type ⓘ	low
Cut-off frequency ⓘ	11.328125
Order ⓘ	6
Analysis	
Type ⓘ	FFT
FFT length ⓘ	64
Take log of spectrum? ⓘ	<input checked="" type="checkbox"/>
Overlap FFT frames? ⓘ	<input type="checkbox"/>
Improve low frequency resolution? ⓘ	<input type="checkbox"/>
Save parameters	

Tal configuração resultou em uma visualização dos features da seguinte forma:

Figura 5
Feature Explorer - autotune



As configurações dos parâmetros dos outros dois modelos foram semelhantes, mudando apenas o tamanho do FFT, um sendo 64 e outro 128.

Figura 6
Parâmetros FFT 64 e FFT 128

Parameters
Autotune parameters

Filter

Scale axes ⓘ 1

Input decimation ratio ⓘ 1

Type ⓘ none

Analysis

Type ⓘ FFT

FFT length ⓘ 64

Take log of spectrum? ⓘ ☒

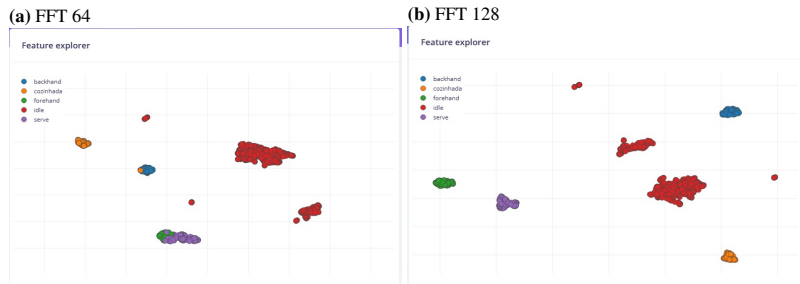
Overlap FFT frames? ⓘ ☒

Improve low frequency resolution? ⓘ ☐

Save parameters

As visualizações de features dessas duas configurações são:

Figura 7
Feature Explorer



Nota-se visualmente uma maior divisão das classes na configuração com tamanho do FFT igual a 128.

2.3 Treinamento do Modelo

Os três modelos treinados utilizaram a mesma configuração de rede neural: com 30 épocas de treinamento, e um *learning rate* de 0,0005. O tamanho de *validation set* utilizado foi de 20%. Foram utilizadas duas camadas intermediárias para a DNN, sendo uma de 20 e outra de 10 neurônios. A camada de entrada de cada um dos modelos possuem tamanhos diferentes, sendo de 162 features a configuração feita pelo autotune, de 222 features a configuração com fft de 64 e de 414 features a configuração com fft de 128.

Os resultados do treinamento dos modelos foram os seguintes:

Figura 8
Treinamento autotune



Figura 9
Treinamento FFT 64



Figura 10
Treinamento FFT 128



Percebe-se uma grande semelhança entre os modelos tanto na acurácia, tempo de inferência e no uso da RAM e flash.

3. Resultados

Primeiramente, como o modelo com os parâmetros determinados pelo autotune, dentre os três modelos treinados, foi o que obteve a menor acurácia, foi decidido não realizar o teste do modelo. Os dois modelos restantes foram testados com os dados separados para teste, 23% do dataset, dessa forma foram obtidos os seguintes resultados:

Figura 11
Teste FFT 64



Figura 12
Teste FFT 128



Nota-se que ambos os resultados foram satisfatórios, atingindo uma acu-

rácia acima de 95%, além disso vale ressaltar que os movimentos que ambos os modelos mais encontraram dificuldades para classificar foram o *idle* e *cozinhada*.

4. Conclusão

Após a finalização do treinamento e testagem do modelo, conclui-se que o projeto funciona, isto é, consegue captar os diferentes movimentos do tênis de mesa. Desta forma, é possível distinguir se o movimento realizado está sendo feito corretamente, o que ajuda na prática do esporte. No entanto, vale destacar que o modelo pode ser melhorado aumentando o dataset, principalmente nas classes *idle* e *cozinhada*, que foram os dois movimentos que mais foram classificados erroneamente, mesmo que o erro tenha sido menor que 5%. Outrossim, pode-se aumentar a quantidade de movimentos distintos, para que tenha mais saídas e mais variedades de movimentos.

Destaca-se que para a utilização do dispositivo numa raquete, é preciso do acoplamento de uma bateria para fornecer energia. Assim, o trabalho realizado teve como objetivo testar a funcionalidade. Portanto, para futuras aplicações, é necessário um estudo para a praticidade da aplicação do dispositivo em uma raquete durante um treino oficial.

Por fim, o projeto no Edge Impulse está disponível para eventuais consultas ([projeto](#)). Além disso, está disponível uma apresentação realizada pelos integrantes, explicando e demonstrando a funcionalidade do trabalho ([vídeo](#)).

Referências

- Arduino (2023). Arduino nano 33 ble sense, <https://docs.arduino.cc/resources/datasheets/ABX00031-datasheet.pdf>. Acesso em: 05 jul. 2023.
- Confederação Brasileira de Tênis de Mesa (2023). História do tênis de mesa, <https://www.cbtm.org.br/conteudo/detalhe/5>. Acesso em: 31 mai. 2023.