

Universidade Federal de Itajubá - UNIFEI
Instituto de Engenharia de Sistemas e Tecnologia da Informação

Ana Paula Bessa Medeiros - 2019004438

Letícia Tempone de Oliveira - 2019004910

Rafael Coelho Paes - 2019000081

Ricardo Magno do Carmo Junior - 2019018649

**TinyML aplicado à Detecção da Doença Pulmonar
Obstrutiva Crônica**

Itajubá, Minas Gerais

2022

Ana Paula Bessa Medeiros - 2019004438
Letícia Tempone de Oliveira - 2019004910
Rafael Coelho Paes - 2019000081
Ricardo Magno do Carmo Junior - 2019018649

TinyML aplicado à Detecção da Doença Pulmonar Obstrutiva Crônica

Relatório da disciplina **IESTI01** - TINYML -
Aprendizado de Máquina Aplicado para Dispo-
sitivos IOT Embarcados, lecionada na Universi-
dade Federal de Itajubá com o objetivo de expor
o projeto final, voltado para detecção da Doença
Pulmonar Obstrutiva Crônica.

Universidade Federal de Itajubá - UNIFEI

Orientador: Marcelo José Rovai

Itajubá, Minas Gerais
2022

Resumo

Este projeto apresenta o estudo e o desenvolvimento de uma aplicação de *TinyML* para detecção e classificação da Doença Pulmonar Obstrutiva Crônica (DPOC). O *TinyML* é uma poderosa área computacional, proporcionando processamentos eficazes em paralelo ao baixo consumo de energia e de memória. Podendo assim, ser inserido em atividades distintas, inclusive do cotidiano do homem, com o fito de auxiliar e aperfeiçoar as tarefas. Nesse viés, o projeto visa utilizar uma rede neural aplicada ao kit de *TinyML* para classificar entre respiração portando a doença, silêncio e respiração saudável. Utilizou-se o *Edge Impulse* desde a coleta de dados, à criação do modelo, ao seu treinamento e testes, para até a otimização do projeto para sua aplicação embarcada. Com isso, embarcou-se o projeto no *Arduino nano 33 BLE Sense*, cujo sensor usado na aplicação foi o microfone da própria placa. De acordo com os resultados obtidos, conclui-se que o modelo, durante a inferência, apresentou boa performance, com acurácia alta e próxima de 90%. Mediante aos áudios reais de DPOC, a acurácia obtida nos testes apresentou uma diminuição, mas a classificação manteve-se correta.

Palavras-chave: DPOC. *TinyML*. Rede neural. Machine Learning. Classificação. *Edge Impulse*.

Sumário

1	Introdução	4
2	Entendendo a Doença	5
2.1	Diagnóstico	5
2.2	Possíveis Tratamentos	5
3	Objetivos	5
4	Descrição do Projeto	6
4.1	Diagrama de Blocos	6
4.2	Hardware utilizado	6
5	Coleta de Dados	7
5.1	Aquisição Manual de Dados	8
5.2	Análise de Dados	8
5.3	Pré-Processamento	9
5.3.1	Reamostragem dos dados	9
5.4	Distribuição final da coleta de dados	11
6	Desenvolvimento do Modelo	11
6.1	Criando o Impulse	11
6.2	Gerando as Features	13
6.3	Treinamento	14
6.4	Resultado do Treinamento	15
6.5	Teste	16
6.6	Deployment	16
6.7	Pós-Processamento	16
6.8	Inferências	17
7	Conclusão	18
	Referências	19

1 Introdução

No ano de 2019, a Doença Pulmonar Obstrutiva Crônica (DPOC) foi a terceira maior causa de morte no mundo, contando com cerca de 3.23 milhões de mortes ([Organização Mundial da Saúde, 2021](#)), como mostra o infográfico da Figura (1).

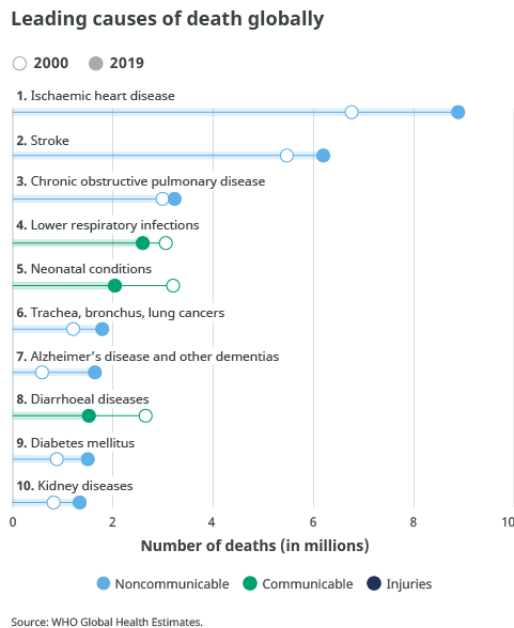


Figura 1 – Principais causas de morte no ano de 2019, comparados ao ano 2000

A DPOC nada mais é do que um conjunto de diversas doenças respiratórias, e está fortemente ligada à prática do tabagismo. Alguns dos sintomas são a **Bronquite Crônica** - responsável pelo estreitamento das vias aéreas e a **Enfisema** - que causa danos irreversíveis nos alvéolos pulmonares, estruturas fundamentais para o funcionamento do aparelho respiratório, causando perda progressiva da função pulmonar.

Como o próprio nome diz, trata-se de uma doença crônica, portanto não há uma cura definitiva. Contudo, existe a possibilidade de tratamento, essencial para seu controle e consequentemente para a longevidade do portador da doença. Portanto, quanto antes o diagnóstico

for feito, mais eficaz o resultado do tratamento será.

Entretanto, apresentar um diagnóstico precoce ainda é um desafio, visto que comumente a DPOC é apenas detectada em estados mais graves, nos quais os sintomas já são marcantes. Tal fato faz os pacientes buscarem ajuda médica tardiamente e, consequentemente, iniciar o tratamento apenas em situações mais avançadas da doença.

Isto posto, uma grande possibilidade de avanço para a área são métodos que auxiliem no diagnóstico, principalmente o tornando mais rápido e eficaz.

Paralelamente, com avanço das análises computacionais e do poder de sensoriamento, técnicas modernas como Inteligência Artificial são utilizadas na saúde, na segurança, na economia, assim como em muitas outras áreas sociais. Não obstante, especificamente o *TinyML* demonstra-se uma poderosa tecnologia a ser agregada no diagnóstico de doenças.

O *TinyML* permite a implementação de algoritmos de aprendizado de máquina em dispositivos na ponta, em sistemas de baixo consumo de energia, com uso de sensores pequenos e de processamento até mesmo em microcontroladores. Sucintamente, essa vertente permite aplicar técnicas computacionais sofisticadas em dispositivos fisicamente pequenos, com baixíssimo consumo de energia e, no geral, em modo *always-on*.

Desse modo, com sua aplicação para investigar DPOC em pacientes, é possível atingir um sensoriamento fácil, prático e eficiente, por meio de um dispositivo sempre ativo no paciente. Em razão do seu baixo uso de memória e de consumo de energia, tal dispositivo pode apresentar dimensões físicas pequenas e peso leve, podendo ser incluído no uso diário dos pacientes. Um exemplo nessa linha foi a tendência das pulseiras/relógios inteligentes, que podem ser utilizados para a medição de batimentos

cardíacos.

2 Entendendo a Doença

Como dito anteriormente, uma das causas da doença é o *tabagismo*, porém ela também pode ser causada por outros meios, como por exemplo a exposição passiva à fumaça e outros gases prejudiciais à saúde humana. A poluição ambiental, a queima de biomassa como as queimadas de lavouras, uso de lenha para cozinha (como no fogão a lenha), poeira e até vapores químicos entram neste grupo ([Ministério da Saúde, 2018](#)).

A DPOC é uma doença com uma instalação lenta. Geralmente, o primeiro sintoma é a falta de ar (*dispneia*) associada a esforços relacionados às ações básicas, como subir escadas, andar depressa ou praticar atividades esportivas. Com o tempo, os problemas respiratórios começam a se intensificar, causando falta de ar até mesmo em pacientes no estado de repouso ([Ministério da Saúde, 2021](#)).

Pacientes com a doença possuem, muitas vezes, outras condições médicas como doenças cardíacas, osteoporose, fadiga crônica, câncer de pulmão e até mesmo depressão e/ou ansiedade ([Organização Mundial da Saúde, 2022](#)).

2.1 Diagnóstico

Para a realização do diagnóstico, inicialmente cinco perguntas são feitas ao paciente:

- Você tem mais de 40 anos?
- Você tem tosse pela manhã?
- Você tem catarro pela manhã?
- Você se cansa mais do que uma pessoa da sua idade?
- Você tem chiado no peito à noite ou ao praticar exercício?

Caso três das respostas respectivas às estas perguntas for positiva, considera-se um rastreamento positivo. Então, é possível prosseguir com os exames, tais como **Espirometria** ou **Raio X de Tórax**. Estes exames serão responsáveis por medir e classificar a gravidade da DPOC no paciente ([Ministério da Saúde, 2010](#)).

2.2 Possíveis Tratamentos

Para um tratamento da doença e consequentemente um prolongamento da sobrevida do paciente, é necessário, em primeiro lugar, tomar medidas educativas de modo a ensinar sobre os fatores de risco e estratégias para minimizar as crises.

Um fator crucial para o tratamento de um paciente é a realização de atividades físicas regulares, unida de uma alimentação reforçada, de modo a manter um estilo de vida saudável ([Ministério da Saúde, 2010](#)). Assim, o paciente conseguira amenizar significativamente os efeitos prejudiciais da doença.

Diversos estudos demonstraram que, nos casos mais graves, o único tratamento médico capaz de aumentar a sobrevida das pessoas com a doença é a **oxigenioterapia** ([Ministério da Saúde, 2021](#)).

3 Objetivos

Este trabalho final tem como objetivo principal desenvolver um modelo de TinyML capaz de detectar a DPOC através do som da respiração de uma pessoa de modo a auxiliar no diagnóstico da doença, evitando tratamentos tardios e a agravamento da doença.

Os objetivos específicos deste trabalho são dados pelos tópicos a seguir:

1. Aplicar os conhecimentos adquiridos na disciplina IESTI01 - TinyML - Aprendizado de Máquina Aplicado para Disposi-

tivos IoT Embarcados para o desenvolvimento de um sistema detector de DPOC.

2. Investigar e compreender técnicas adicionais de *Machine Learning* para implementação em dispositivos de ponta.
3. Criar um *dataset* de áudios de respiração saudável e ajustar suas frequências.

4 Descrição do Projeto

4.1 Diagrama de Blocos

O diagrama de blocos contendo a ordem cronológica das etapas do projeto pode ser visualizado na Figura (2).

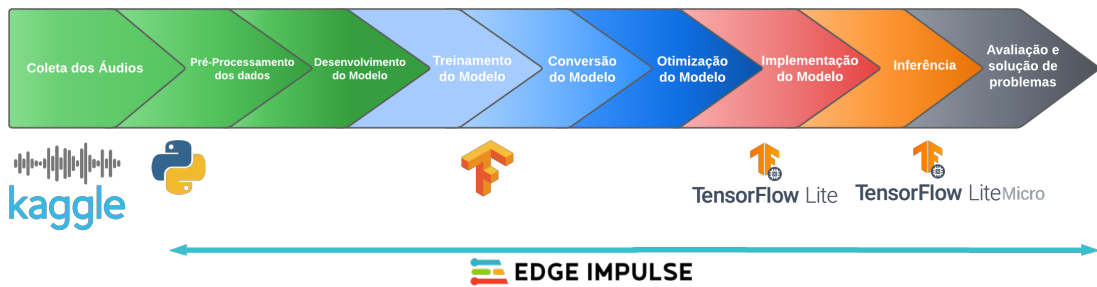


Figura 2 – Diagrama de Blocos

4.2 Hardware utilizado

Para a implementação do modelo a ser elaborado, será necessário utilizar o *Kit Didático Arduino* voltado para *Tiny Machine Learning*, que acompanha os itens:

- Arduino Nano 33 BLE Sense Board;
- Arduino Tiny Machine Learning Shield;

- Cabo USB-A para Micro USB.

A placa *Arduino Nano 33 BLE Sense*, possui, em sua estrutura, um microprocessador ARM Cortex-M4F 32-bit, que opera em até 64 MHz, com 1 MB de memória Flash e 256 KB de memória RAM. Um esquemático contendo suas entradas e saídas pode ser visualizado na Figura (3).

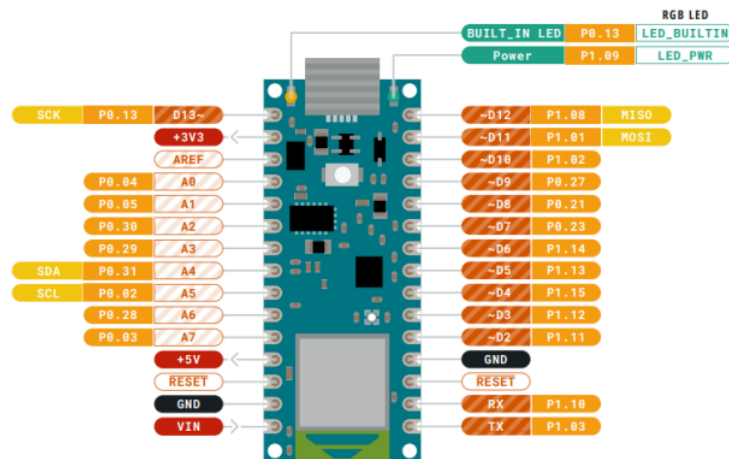


Figura 3 – Conexões Arduino Nano 33 BLE

5 Coleta de Dados

A princípio, os dados utilizados para o desenvolvimento do projeto serão recolhidos do *dataset* [Respiratory Sound Database](#), encontrado no site *Kaggle*, uma comunidade online formada por cientistas de dados e profissionais de *Machine Learning*. Uma breve ilustração do site se encontra na Figura (4).

O arquivo baixado contendo *dataset*, com um tamanho de aproximadamente 4GB, possui um total de 920 arquivos de áudio no formato *.wav*, compreendendo gravações de 126 pacientes de variadas faixas etárias: crianças, adultos e idosos. No total, há uma soma de 5.5 horas de gravação, contendo 6898 ciclos respiratórios, que podem incluir sons saudáveis ou sons contendo chiados (classificados como *wheezes*) ou crepitações (classificados como *crackles*). Alguns dos arquivos de áudios, de modo a simular condições em um ambiente real, possuem ruídos sonoros de fundo.

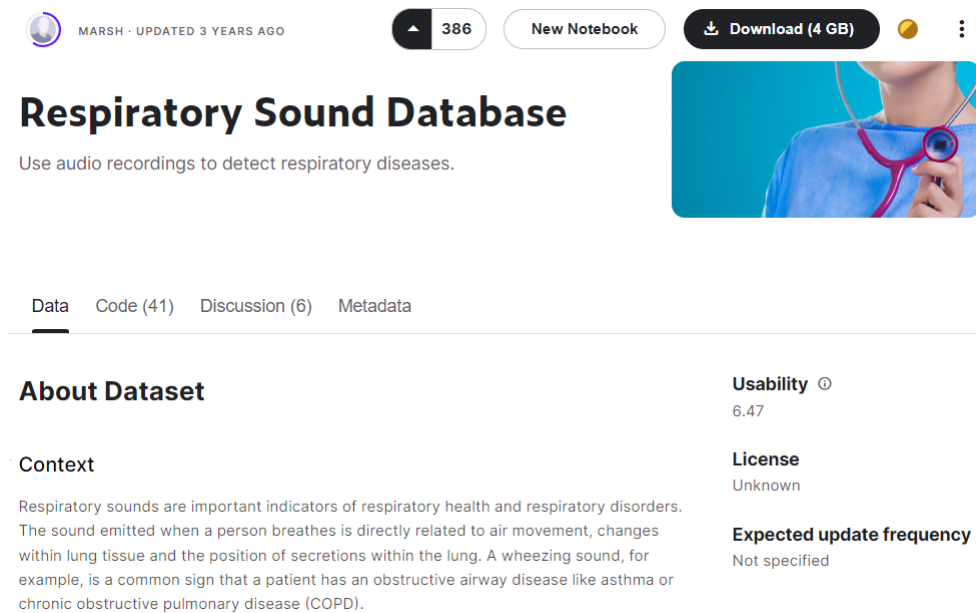
Além dos arquivos de áudio, o *dataset* acompanha outros arquivos que auxiliam na manipulação e análise do mesmo:

- *demographic_info.txt* - Informações sobre o paciente, como número de identificação, idade, sexo, índice de massa corporal e, para caso de pacientes crianças, peso e altura.
- *patient_diagnosis.csv* - Tabela contendo o **Diagnóstico** de cada um dos 126 pacien-

tes, que podem ser: *Saudável (Healthy)*, *DPOC (COPD)*, *Infecção do trato respiratório inferior (LRTI)*, *Infecção do trato respiratório superior (URTI)*.

- *filename_format.txt* - Uma breve explicação sobre o formato de nomeação dos arquivos: O nome de cada arquivo compreende informações como o número do paciente, a região do corpo em que a gravação foi realizada, o modo de aquisição e o dispositivo utilizado para gravar.
- Os arquivos de áudio possuem uma nomenclatura que possibilita identificar o número do paciente, região de gravação e o dispositivo de gravação:
 - Microfone AKGC417L (AKGC417L);
 - Estetoscópio 3M Littmann Classic II SE (LittC2SE);
 - Estetoscópio 3M Litmmann 3200 Electronic (Litt3200).
- Cada arquivo de áudio *.wav* compreende um arquivo *.txt* de mesmo nome, onde está registrado a quantidade de chiados e crepitações do respectivo áudio, juntamente com o momento temporizado em que estes ocorreram.

Além deste *dataset*, para contribuir com a qualidade e robustez do modelo, serão gravados, pela equipe, uma maior quantidade de minutos de áudio saudáveis e também áudios de silêncio.



Respiratory Sound Database
Use audio recordings to detect respiratory diseases.

About Dataset

Context

Respiratory sounds are important indicators of respiratory health and respiratory disorders. The sound emitted when a person breathes is directly related to air movement, changes within lung tissue and the position of secretions within the lung. A wheezing sound, for example, is a common sign that a patient has an obstructive airway disease like asthma or chronic obstructive pulmonary disease (COPD).

Usability 6.47

License Unknown

Expected update frequency Not specified

Figura 4 – Página do *dataset* utilizado no site Kaggle

5.1 Aquisição Manual de Dados

Diante da necessidade de aperfeiçoamento dos dados a serem utilizados no modelo, a coleta manual dos áudios saudáveis e áudios de silêncio foram realizados utilizando o microfone digital **MP34DT05** presente na placa *Arduino Nano 33 BLE Sense*.

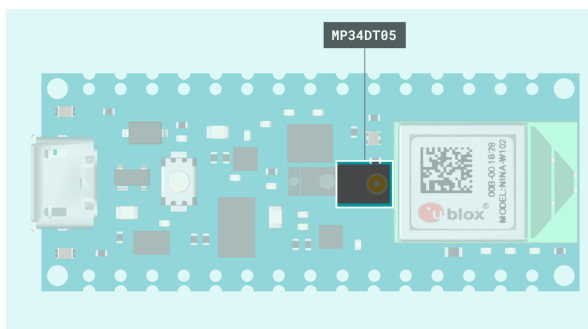


Figura 5 – Microfone Digital MP34DT05

Os áudios foram capturados na frequência de 16kHz e em tamanhos de 20 segundos para manter um padrão no *dataset* inteiro.

5.2 Análise de Dados

Com a finalização da coleta do *dataset*, os dados crus obtidos de uma respiração saudável, apresentaram comportamento similares entre si. Pela representação do áudio saudável visto na figura (6), observa-se que entre cada respiração, há tempos de silêncio, e o conjunto *silêncio + áudio* consiste, portanto, em aproximadamente 4000ms.

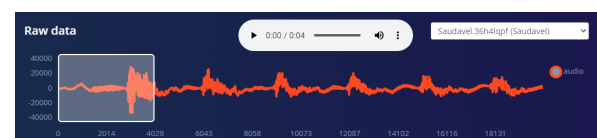


Figura 6 – Dado Crú - Respiração Saudável

Dessa maneira, pensando no melhor funcionamento possível do modelo desenvolvido, tendo em vista que ao uma pessoa saudável coletar seus dados para fazer uma inferência, espera-se que também haja o espaço de silêncio entre as respirações propriamente ditas. Com isso, escolheu-se trabalhar com uma janela de 4000ms no Edge Impulse e que tenha um incremento durante o seu deslizamento pelo dado, de 4000 ms também.

Assim, espera-se garantir que durante a coleta de dados, o modelo não corra o risco de pegar somente partes de silêncio de um áudio gravado por uma pessoa e não saiba afirmar se trata-se de uma respiração saudável ou com DPOC, que é o principal objetivo do trabalho.

5.3 Pré-Processamento

Tendo em vista os arquivos presentes no *dataset*, é nítido que será necessário efetuar uma pré-seleção nos dados. No caso, a doença a ser identificada no projeto será restrita à DPOC, portanto se faz essencial recolher apenas os dados que correspondem à esta doença para a primeira parte do *dataset*.

Para a aplicação desejada, de modo a utilizar os recursos disponíveis da placa (Microfone), se torna interessante lidar apenas com os dados gravados com o **Microfone AKGC417L** Figura (7), uma vez que os sons captados por este, diferente de um estetoscópio profissional, podem se assemelhar muito mais aos sons captados pelo microfone da placa, tornando a implementação mais precisa.

Também faz-se necessário converter a

frequência dos arquivos de áudio do *dataset*, uma vez que a gravação dos arquivos de áudio utilizando a placa será feita em 16kHz, portanto, os arquivos do *dataset* serão convertidos para essa mesma frequência para evitar erros no treinamento do modelo.



Figura 7 – Microfone Profissional AKGC417L

5.3.1 Reamostragem dos dados

Para executar a conversão dos arquivos de áudio do *dataset* coletado no site [Kaggle](#), foi utilizada, além da biblioteca *Librosa*, uma série de outras bibliotecas que estão disponível para a linguagem *Python*, como *numpy*, *pandas*, *matplotlib*, *ffmpeg*, *soundfile*, etc. Na Figura (8) é possível visualizar o código da função criada executar a reamostragem, de modo a transformar os áudios de 44.1kHz em 16kHz.

```

[1] 1 !pip install ffmpeg

[2] 1 import numpy as np
   2 import pandas as pd
   3 import librosa
   4 import matplotlib.pyplot as plt
   5 from scipy.io import wavfile
   6 import soundfile
   7 import ffmpeg
   8 import librosa
   9 import os

[3] 1 def sr_converter(base_dir):
   2     for files in os.listdir(base_dir):
   3         if os.path.isfile(os.path.join(base_dir, files)):
   4             pathname = base_dir + files # gets the file path in a string
   5             y, sr = librosa.load(pathname)
   6             resample = librosa.resample(y, orig_sr=sr, target_sr=16000) # resamples .wav file to 16kHz
   7             soundfile.write(pathname, resample, 16000, subtype='PCM_16') # overwrites the old file with the resampled one

[4] 1 # Directory for the two parts of the dataset
   2 base_dir_healthy = '/content/Healthy/'
   3 base_dir_COPD = '/content/COPD/'

[5] 1 # Converting the sr
   2 sr_converter(base_dir_healthy)
   3 sr_converter(base_dir_COPD)

[6] 1 # Creating a Zip file to be able to download:
   2 !zip -r /content/file_copd.zip /content/COPD

[7] 1 # Creating a Zip file to be able to download:
   2 !zip -r /content/file_healthy.zip /content/Healthy

```

Figura 8 – Código em Python para o pré-processamento dos dados

Para executar o código, é necessário primeiramente fazer o *upload* dos arquivos de áudio desejados, separando-os em pastas diferentes, de preferência.

Primeiramente, faz-se a inclusão das bibliotecas necessárias, como mostra os blocos 1 e 2 da Figura (8). Após isso, a função de conversão *sr_converter* é definida no bloco 3. Esta função recebe como argumento um diretório base, contendo os arquivos de áudio que foram carregados no *Colab*. Em seguida um laço *for* é executado, e uma condição *if* verifica a presença de cada arquivo presente nesta pasta. Caso a condição seja cumprida, o processo de conversão é iniciado.

Então, é atribuído à variável *pathname* o diretório completo do arquivo (pasta + nome do arquivo). Logo em seguida, utilizando a função

da biblioteca *Librosa*, é extraída as informações deste arquivo de áudio, de modo a armazená-las nas variáveis *y* e *sr* (Sample Rate).

Com isso, uma nova variável é criada, denominada *resample*, que receberá o novo arquivo *.wav* com a frequência desejada, de acordo com os parâmetros da função *librosa.resample*. Por fim, este novo arquivo sobre-escreve o arquivo antigo de modo a economizar armazenamento.

Nos blocos 4 e 5 da Figura (8) a função de conversão é executada. Os blocos 6 e 7 servem para criar um arquivo *.zip* com os novos dados. A Figura (9) mostra uma comparação entre dois arquivos do mesmo áudio, sendo um não convertido e outro após o processo de conversão. As informações foram retiradas com auxílio do *software VLC Media Player*. O código pode ser acessado pelo [Google Colab](#).

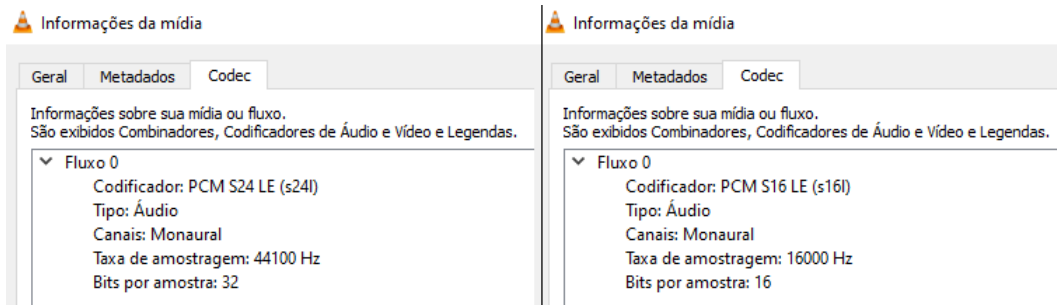


Figura 9 – Taxa de Amostragem dos arquivos de áudio após a conversão

5.4 Distribuição final da coleta de dados

Ao juntar os dados coletados manualmente e os dados do *dataset* no Edge Impulse, os dados

são separados na taxa de 80/20 para *training* e *testing* respectivamente, como apresentado na figura a seguir:

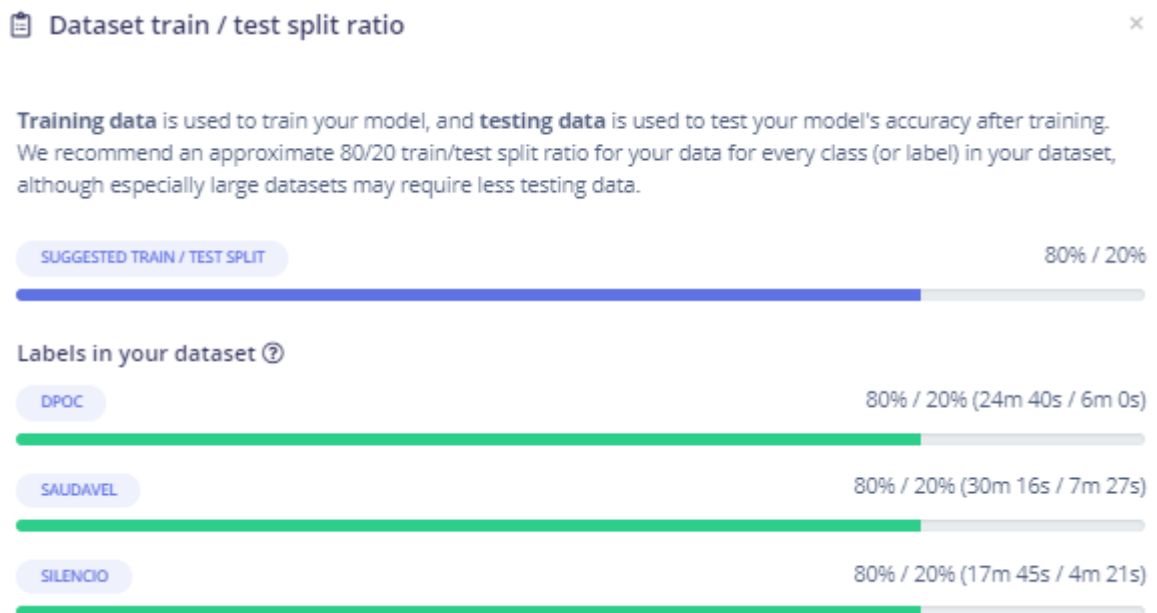


Figura 10 – Distribuição entre *training* e *testing* na plataforma Edge Impulse AI

6 Desenvolvimento do Modelo

Visa-se testar diferentes arquiteturas de redes neurais, variando a quantidade de camadas, de neurônios e de técnicas de otimização do modelo. Para definir em si o modelo a ser usado, a análise será baseada entre um equilíbrio de boa performance de predição, ou seja, baixo erro. Mas também, mantendo a otimização de tempo e de uso da memória. Para assim, tentar garantir uma boa inferência, mas sem comprometer

as características necessárias para sua implementação em dispositivos de hardware focados em *TinyML*.

6.1 Criando o Impulse

Ao criar o impulso, tem-se em mente o espectrograma dos dados coletados e como serão realizadas as inferências, como o objetivo é captar um ciclo de respiração, é razoável utilizar uma janela de 4 segundos com um acréscimo

também de 4 segundos para evitar do modelo se equivocar na classificação dos dados, pois, como um ciclo de respiração possui inspiração e expiração, o primeiro é identificado como silêncio pelo microfone.

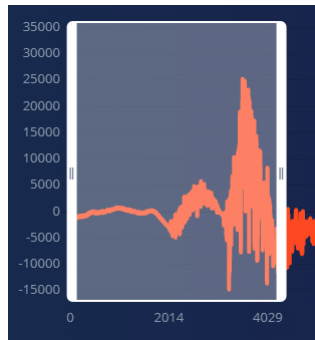


Figura 11 – Espectrograma de um ciclo de respiração saudável

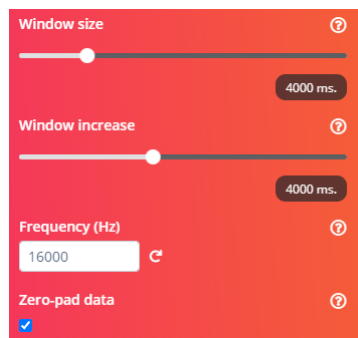


Figura 12 – Janelas usadas na geração do impulso

Para o processamento de dados do modelo, dois caminhos poderiam ser seguidos, poderia ser usado um processamento por **MFE** (*Mel-filterbank energy*) ou por **Spectrogram**. Neste modelo, será usado **Spectrogram** pois apresentou melhores resultados no final do treinamento, assim como também evidenciado pelo uso do **EON Tuner** que será discutido mais adiante.

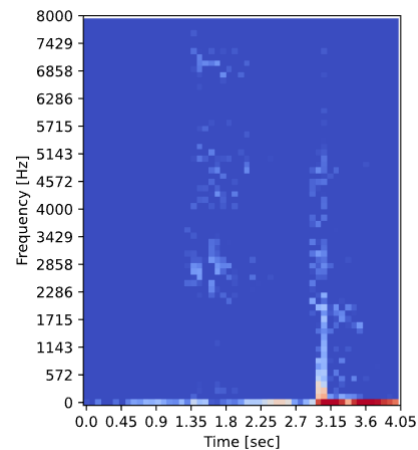


Figura 13 – Visualização do espectrograma de um dado saudável

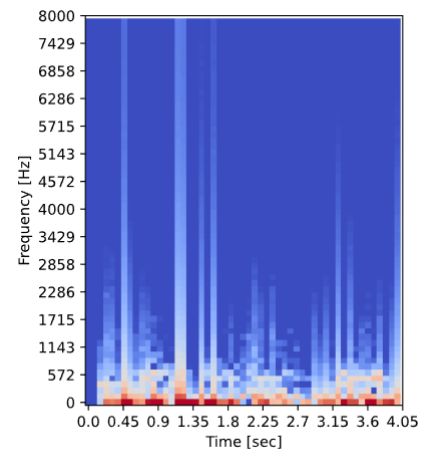


Figura 14 – Visualização do espectrograma de um dado com DPOC

Como modo de aprendizado no treinamento, o melhor método que pode ser empregado nesse tipo de rede neural é o *Classification (Keras)*, portanto, é o que será utilizado.

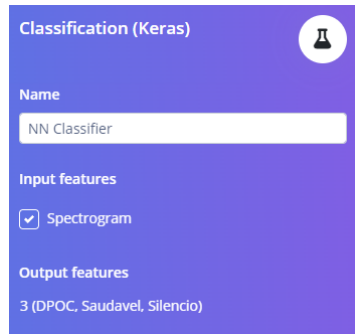


Figura 15 – Classification (Keras) learning block

esse projeto, serão utilizados os seguintes:

Spectrogram	
Frame Length	0.075
Frame Stride	0.075
Frequency Bands	128

Normalization	
Noise Floor (dB)	-32

Tabela 1 – Configuração do *Spectrogram*

6.2 Gerando as Features

Para gerar as *features*, primeiro é necessário estabelecer os parâmetros do *Spectrogram*. Para

Portanto, a partir desses parâmetros, as *features* do projeto podem ser geradas como a figura a seguir mostra:

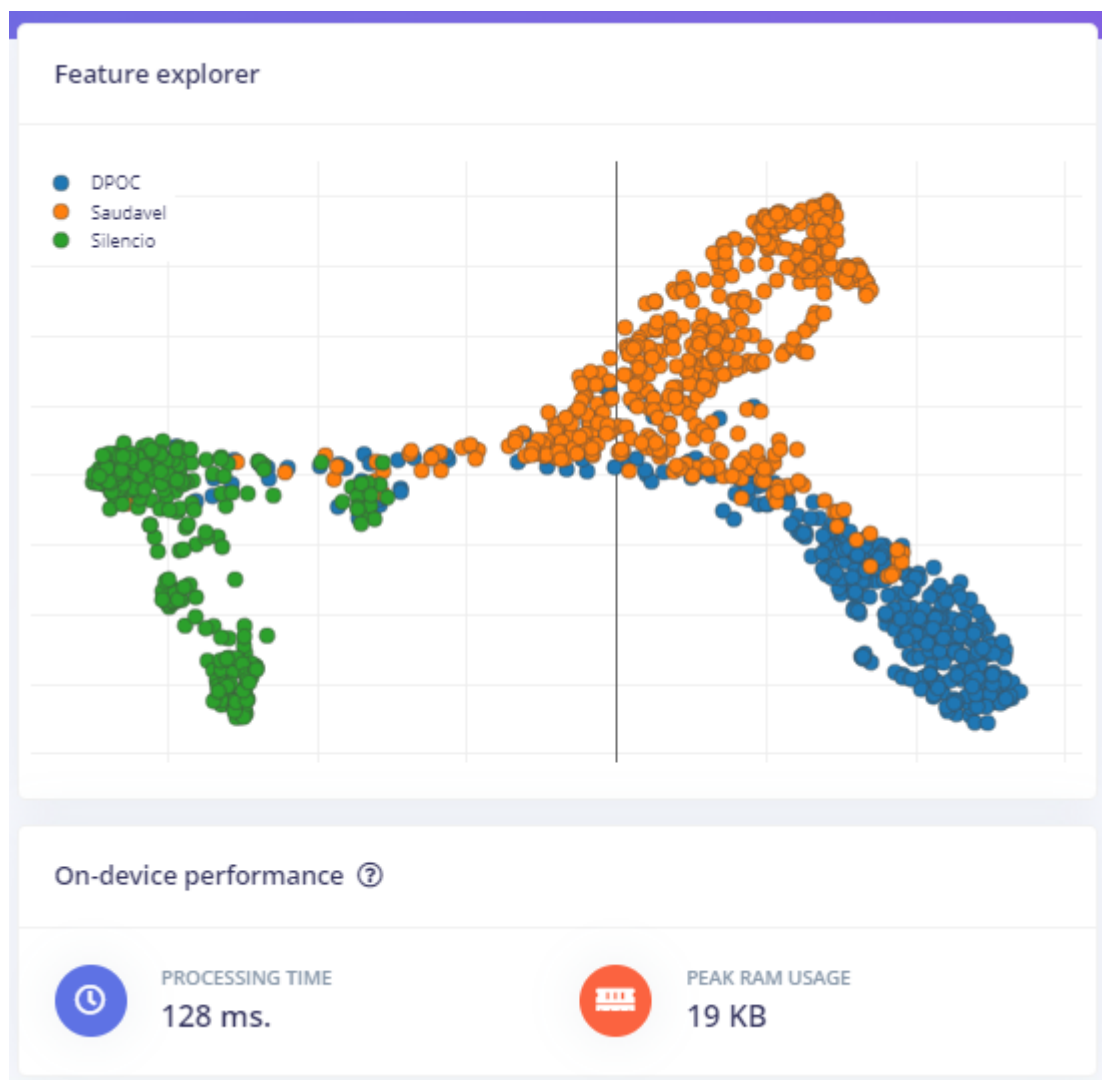


Figura 16 – *Features* geradas através do **Spectrogram**

Ao analisar este gráfico, é possível notar uma boa distinção entre os dados de entrada processados, apesar dos dados de *DPOC* e dos dados saudáveis possuírem alguma sobreposição no centro do gráfico, a maior concentração de ambos está em extremos opostos.

Importante também é ressaltar que os dados de *silêncio* são muito distintos e parecem quase não apresentar sobreposição com outros dados.

6.3 Treinamento

Para o treinamento do modelo, faz-se necessário estabelecer os parâmetros da *Neural Network*, serão configurados o treinamento, as opções de áudio e a arquitetura, respectiva-

mente.

Training Settings	
Number of training cycles	100
Learning Rate	0.005
Validation Set Size	20

Tabela 2 – Configuração do treinamento da *Neural Network*

Audio Training Options	
Data Augmentation	✓
Add Noise	High
Mask Time Bands	Low
Mask Frequency Bands	None
Warp Time Axis	✓

Tabela 3 – Configuração das opções de áudio da *Neural Network*

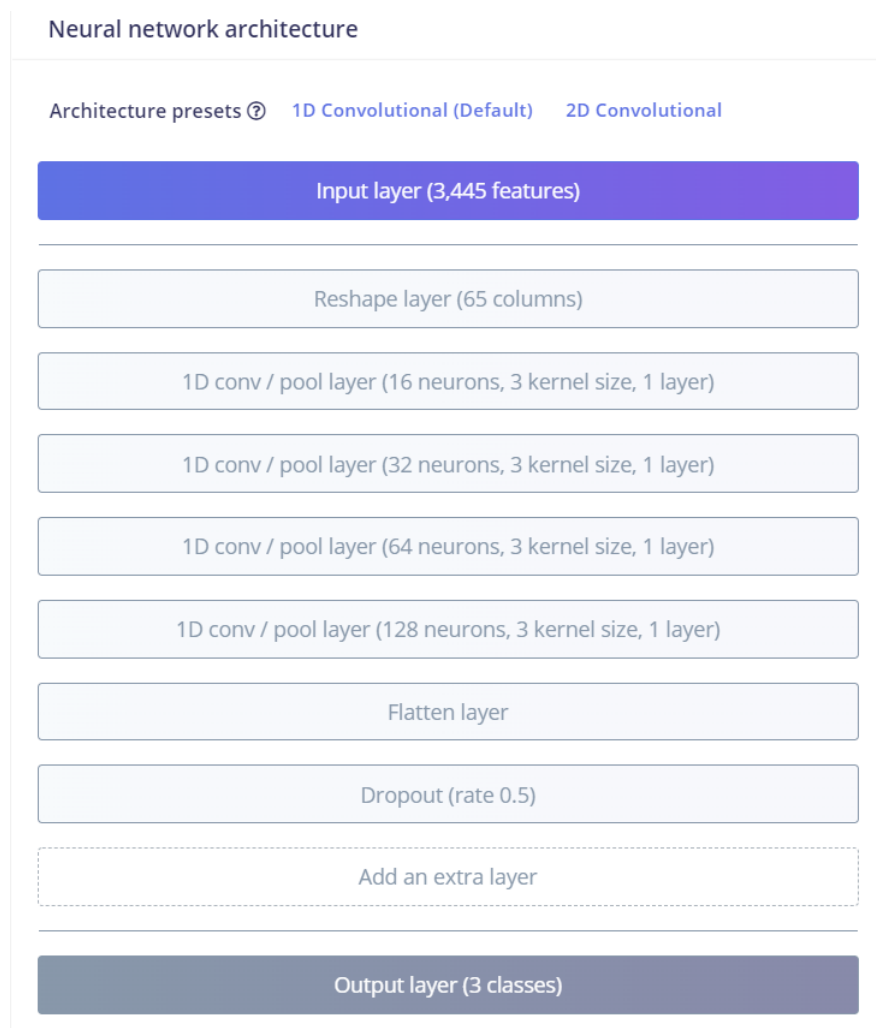


Figura 17 – Configuração da arquitetura da *Neural Network*

6.4 Resultado do Treinamento



Figura 18 – Resultados do treinamento do modelo.

Como pode ser analisado nesta figura, o treinamento apresenta ótimos resultados, uma acurácia alta de 95,4% e baixo loss.

Pelo *Data explorer*, é possível ver uma clara distinção entre os dados, refletindo a boa acurácia do modelo treinado.

Outro ponto interessante a se destacar é o

uso de 16K da memória *RAM*, um valor razoável e dentro dos limites, considerando que o *Arduino Nano 33 BLE Sense* possui 256K.

O *Inferencing time* não possui grande importância no desenvolvimento desse projeto por se tratar de um projeto de processamento de áudio.

6.5 Teste

Para efetuar o teste do modelo, serão usados os dados alocados para teste citados anteriormente (figura 10).

O resultado do teste é apresentado na figura a seguir:

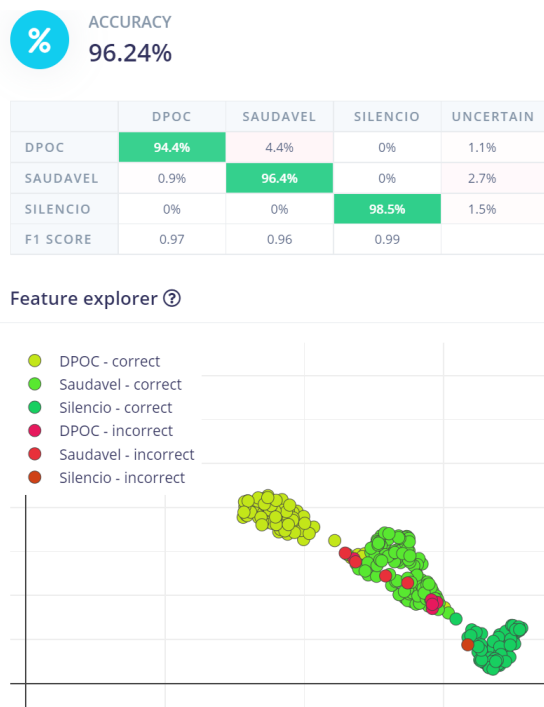


Figura 19 – Resultados do teste do modelo.

O teste confirma o resultado obtido no treinamento, o modelo apresenta alta acurácia, exibe qualidade e eficiência.

6.6 Deployment

O *deployment* do será realizado utilizando o modelo quantizado, pois como pode ser visto pela figura a seguir, possui desempenho semelhante ao não-quantizado e apresenta uma redução significativa no gasto de memórias *RAM* e *flash*.

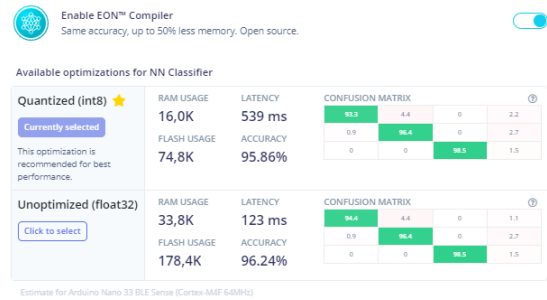


Figura 20 – Otimizações do modelo pelo **EON TUNER**

Porém, também há a possibilidade de usar o modelo não-quantizado, pois ele respeita os limites de memória do *Arduino Nano 33 BLE Sense*.

6.7 Pós-Processamento

Enfim, nesta seção são descritos os passos para a realização das inferências.

Primeiro é necessário preparar o *Arduino* a fim de mostrar os resultados sem a necessidade de um monitor serial, já que o objetivo do projeto é apresentar um resultado rápido, prático e portátil, sem a dependência de um eletrônico maior como um computador ou notebook.

Quando o microcontrolador está em repouso, ou seja, sem realizar o processo de inferência, o *led* azul estará ligado.

Para sinalizar ao microcontrolador que ele deve iniciar o processo de coleta de dados para uma inferência, será utilizado o botão do *shield* presente no *kit* do *Arduino*. Enquanto esse processo de coleta estiver ocorrendo, o microcontrolador manterá os *leds* acesos na cor roxa.

Após a coleta do dado e processamento, o resultado será exibido através dos *leds* do *Arduino Nano 33 BLE Sense* no seguinte esquema de cores:

- DPOC: *led* vermelho;
- Saudável: *led* verde;
- Silêncio: *led* branco.

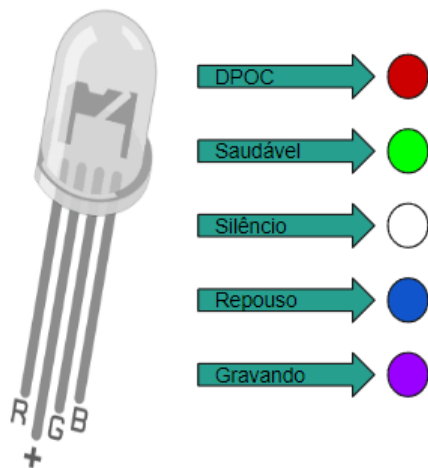


Figura 21 – Representação da funcionalidade dos *leds*

6.8 Inferências

Enfim, para testar a placa, pode ser realizado as inferências principais de silêncio, áudios saudáveis e áudios de DPOC.

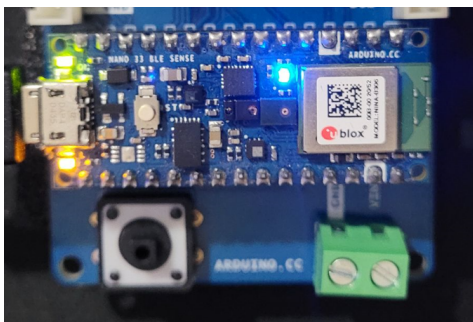


Figura 22 – Placa em repouso

Ao pressionar o botão, o *Arduino* entra em modo de gravação:

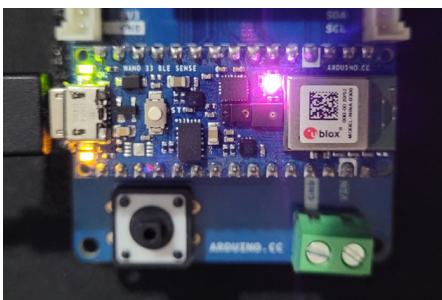


Figura 23 – Placa em gravação

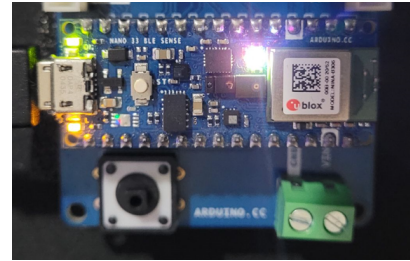


Figura 24 – Placa exibindo resultado de silêncio

Listing 1 – *Output* do monitor serial

```
*****
Diagnostico ==> Silencio com
probabilidade 0.87
*****
DPOC: 0.10547
Saudavel: 0.02344
Silencio: 0.87109
```

Agora, realizando a inferência de um áudio saudável:

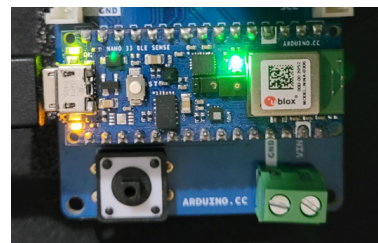


Figura 25 – Placa exibindo resultado de um áudio saudável

Listing 2 – *Output* do monitor serial

```
*****
Diagnostico ==> Saudavel com
probabilidade 1.00
*****
DPOC: 0.00000
Saudavel: 0.99609
Silencio: 0.00000
```

E, finalmente, a inferência de um áudio com Primeiro, realizando a inferência de silêncio. **DPOC.**

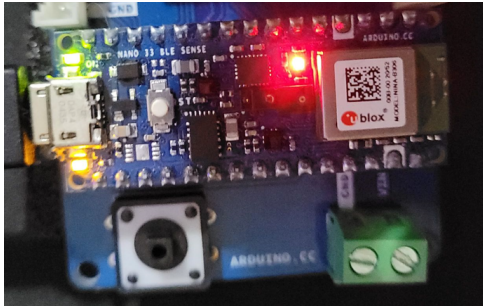


Figura 26 – Placa exibindo resultado de um áudio com DPOC

Listing 3 – *Output* do monitor serial

```
*****
Diagnostico ==> DPOC com
probabilidade 1.00
*****
DPOC: 0.99609
Saudavel: 0.00000
Silencio: 0.00000
```

7 Conclusão

Após a realização de todo o estudo e desenvolvimento do modelo, seguido dos testes de inferência com a aplicação na placa *Arduino*, foi possível observar que o modelo criado opera com uma boa precisão, conseguindo identificar todas as classes criadas mantendo uma boa acurácia.

Apesar dos resultados satisfatórios, existem alguns pontos no desenvolvimento que poderiam ser aprimorados em uma aplicação futura. Por exemplo, a utilização de um microfone mais potente, de modo a captar com mais perfeição as ondas sonoras da respiração, agregaria muito no projeto, desde que o mesmo não fosse muito custoso, pois a ideia é oferecer uma solução acessível para substituir equipamentos médicos muito caros.

Outra maneira de improvisar os resultados seria a criação de um *dataset* próprio mais variado, de modo a garantir uma melhor robustez do mesmo. Os dados deveriam ser gravados de

diversos pacientes, sejam portadores da DPOC ou saudáveis. Feito isso, o modelo responderia melhor, conseguindo auxiliar com mais clareza no diagnóstico dos pacientes.

No decorrer do projeto, foi necessário se atentar com o conceito de equilibrar o consumo de energia com a performance do sistema. A área do *TinyML* trabalha com dispositivos embarcados que possuem grandes limitações de recursos, uma vez que a proposta é manter um tamanho pequeno, uma boa economia de energia e consequentemente o menor custo possível. Com isso, será sempre necessário saber ponderar, (*trade-off*), durante a criação de um modelo, as questões de eficiência energética e desempenho.

Alguns modelos treinados como teste muitas vezes forneceram bons resultados, mantendo uma boa precisão de treinamento e de teste. Porém, devido ao limite de *Memória RAM* da placa utilizada no Kit *Arduino*, não foi possível realizar as inferências na mesma.

Por fim, é possível, após a realização do trabalho, observar a importância do *TinyML* para as mais diversas áreas do conhecimento. Ao longo do semestre, as aulas ministradas mostraram inúmeras aplicações de *Machine Learning* com dispositivos embarcados *Tiny*, o que mostra seu grande escopo de aplicabilidade. A filosofia do *TinyML* é apresentar soluções eficientes, com o menor consumo possível de energia e com a maior portabilidade possível, gerando, por fim, um meio mais acessível no quesito custo para a resolução de um problema.

No mundo, ainda existem diversos países e regiões onde a medicina ainda não possui uma grande acessibilidade, fato que pode ser causado pelos altos custos de equipamentos médicos profissionais. A solução apresentada no projeto, que representa apenas uma fração daquilo que pode ser feito pelo *TinyML*, mostra como o desenvolvimento na área de *Machine Learning* pode auxiliar milhões de pessoas globalmente.

O projeto está disponível no **Edge Impulse** contendo todos os arquivos usados no pelo link: <https://studio.edgeimpulse.com/public/113809/latest> projeto: <https://github.com/Rafael-CP/IESTI01-Final-Project>

Um repositório no **Github** foi criado

Referências

Ministério da Saúde. Caderno de atenção básica - doenças respiratórias crônicas. 2010. Disponível em: https://bvsms.saude.gov.br/bvs/publicacoes/doencas_respiratorias_cronicas.pdf. 5

Ministério da Saúde. *Dia Mundial da Doença Pulmonar Obstrutiva Crônica – DPOC*. [S.l.], 2018. Disponível em: <https://bvsms.saude.gov.br/21-11-dia-mundial-da-doenca-pulmonar-obstrutiva-cronica-dpoc/>. 5

Ministério da Saúde. *3ª quarta-feira do mês (17/11) – Dia Mundial da Doença Pulmonar Obstrutiva Crônica (DPOC): “Pulmões Saudáveis – Mais Importante do que Nunca”*. [S.l.], 2021. Disponível em: <https://encurtador.com.br/xCPTV>. 5

Organização Mundial da Saúde. *Chronic obstructive pulmonary disease (COPD)*. [S.l.], 2021. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>. 4

Organização Mundial da Saúde. *Chronic obstructive pulmonary disease (COPD)*. [S.l.], 2022. Disponível em: [https://www.who.int/news-room/fact-sheets/detail/chronic-obstructive-pulmonary-disease-\(copd\)](https://www.who.int/news-room/fact-sheets/detail/chronic-obstructive-pulmonary-disease-(copd)). 5