

UNIFEI - Universidade Federal de Itajubá

Trabalho de Conclusão de Disciplina

IESTI01



Matheus Souza Diego - 2019015128
Luis Felipe Buzo - 2019005023
Adriano Carvalho Maretti - 2020009562

Itajubá
2021

1. Introdução:

Nos últimos anos o Brasil presenciou por duas vezes desastres ambientais e perdas humanas decorrentes de rompimento de barragens. Recentemente um relatório divulgado apresentou dados de abalos sísmicos que ocorreram na região de Mariana antes dos eventos de ruptura. Mas a verdade é que, hoje, ainda existem barragens que correm o risco de se romper devido a atividades sismológicas, e o problema vai além, barragens estão sendo construídas em áreas já consolidadas como sendo de alto risco devido a tais fenômenos. O caso supracitado se refere a barragem de Pedreira - SP, em área de cisalhamento, mantendo toda a população da região em permanente estado de atenção.

Por mais que existam sensores na estrutura que detectam possíveis danos, eles só alertam quando o dano à estrutura já aconteceu, diminuindo o tempo de evacuação da população local.

A ideia do projeto consiste então em um sistema capaz de detectar variações na atividade sismológica da região, podendo assim, prever possíveis danos às estruturas dessas edificações, antes mesmo de ocorrer, alertando em tempo real caso haja qualquer tipo de abalo e a sua intensidade.

Com a implementação do projeto, a defesa civil e os órgãos responsáveis pelas barragens poderiam organizar evacuações com antecedência, minimizando perdas humanas e danos materiais

1.1. Materiais e Métodos

Para a execução perfeita do projeto seria necessário um sensor sísmico que além de caro é difícil de ser encontrado, com isso em mente optamos por usar o acelerômetro presente no IMU do arduino nano 33 BLE, esse por sua vez simularia diferentes vibrações. A ideia inicial seria fixar o sensor em um aparelho elétrico que se movimenta com o acionamento e, alterando a potência do aparelho seriam simulados abalos de diferentes intensidades.

Nos primeiros testes nos deparamos que os aparelhos como batedeira e secador geravam ondas contínuas bem parecidas e, mesmo que fossem alteradas as potências durante a coleta de dados, essas variações pouco se assemelhavam com sismógrafos reais. A ideia que surgiu então foi utilizar o microfone do mesmo arduino, dessa vez o fixamos no alto falante de um mini system phillips e nesse executamos um som capturado durante abalos sísmicos, dessa forma obtivemos ondas bem parecidas com a dos sismógrafos. Para simular abalos de diferentes intensidades ajustamos o volume de saída, assim quando o volume estava no máximo as ondas eram maiores e, quando no nível médio, as ondas eram menores.

2. Edge impulse

2.1. Coleta dos Dados

A coleta se deu por meio do microfone presente no Arduino Nano 33 BLE, fixado na saída de um alto falante que emitia sons capturados durante abalos sísmicos. A primeira captura apresentada foi falha, visto que a vibração máxima era uma onda que basicamente atingia a amplitude máxima a todo momento, por isso optamos por refazer a coleta do dataset com um volume mais baixo já na vibração máxima, obtendo um dado mais próximo de um sismógrafo real.

Ainda assim, houveram muitas colisões no modelo, visto que muitos pontos da vibração mínima e máxima são iguais, alterando apenas a amplitude da onda em certos momentos.

NoVib

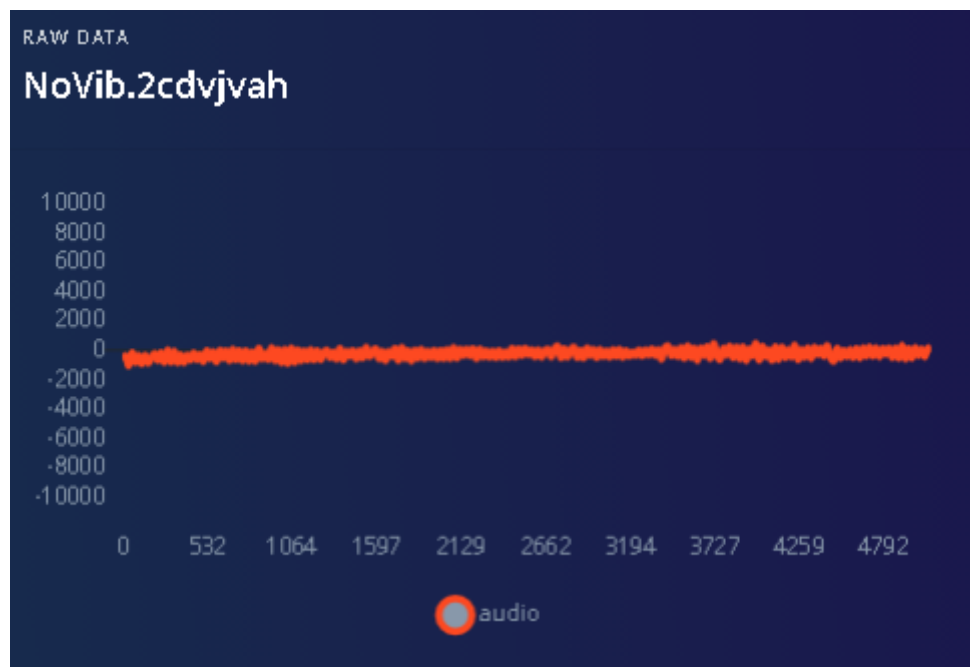


Imagem 1: Gráfico da label sem vibração

VibMax:

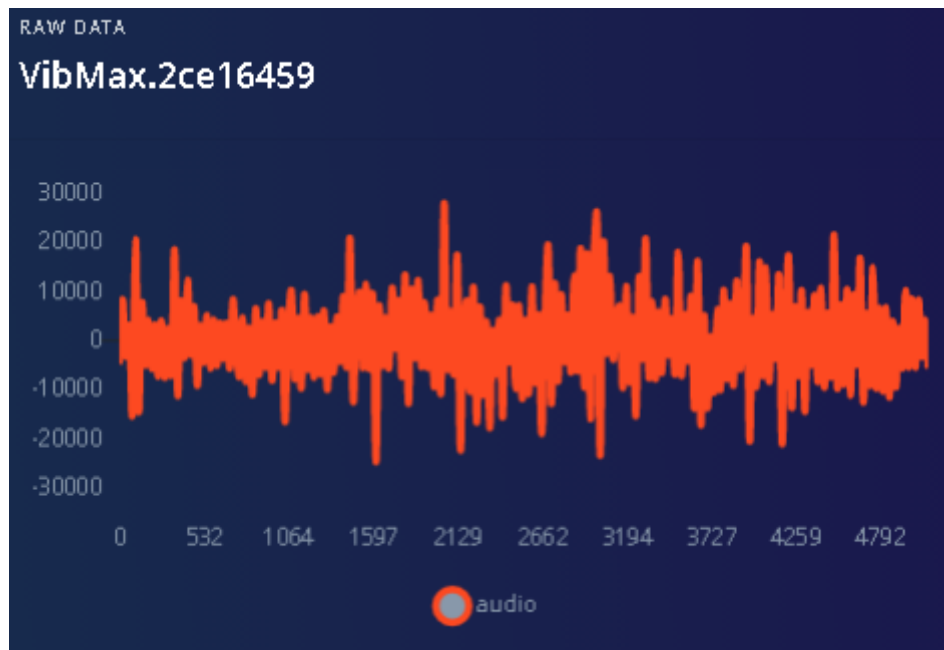


Imagem 2: Gráfico da label com vibração máxima

VibMin:

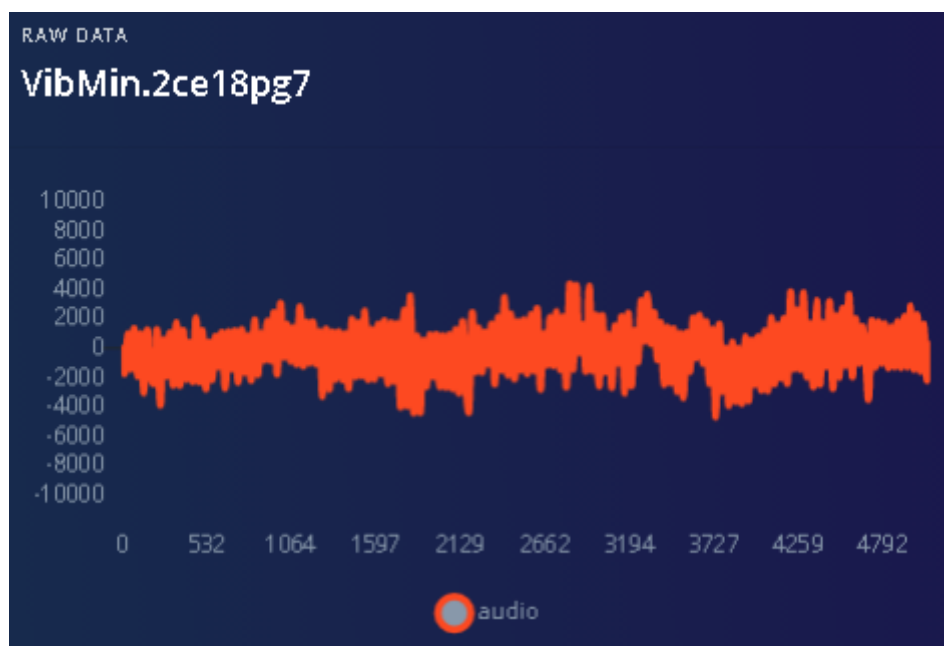


Imagem 3: Gráfico da label com vibração mínima

2.2. Treinamento da IA

Inicialmente usamos um modelo apenas com layers densas, durante a apresentação foi alertado que o melhor desempenho se daria com a utilização de convoluções. Refizemos todos os hiperparâmetros da rede e a treinamos novamente, obtendo resultados diferentes dos anteriores, porém mais coniventes com a realidade.

As configurações escolhidas foram a sequência de 1D convolucional com 8 e 16 neurônios, uma layer flatten, dropout e uma layer densa de 64 neurônios. A rede foi treinada mais uma vez, porém ainda não apresentava o accuracy desejado, além disso não era possível fazer o deploy para o Arduino. Finalmente, fizemos a análise do EON Turner e o parâmetros seguintes foram obtidos:

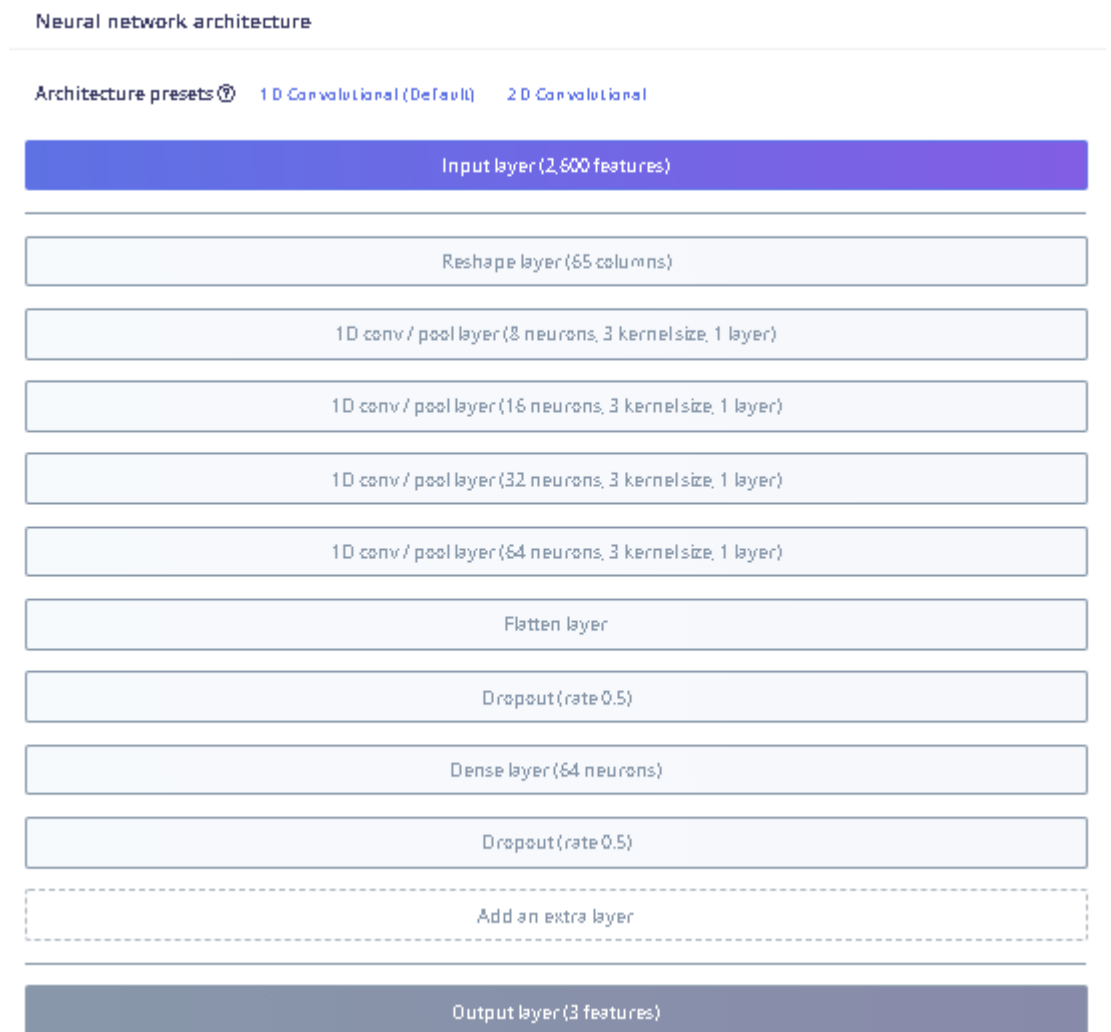


Imagem 4: Hiper parâmetros da rede neural

Dito isso, treinamos, retreinamos e o resultado foi esse:

Last training performance (validation set)

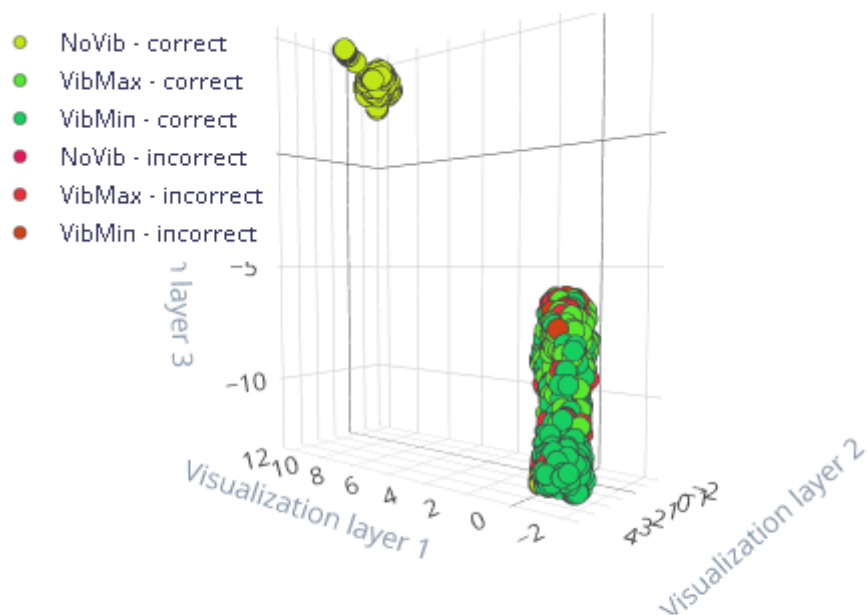


Confusion matrix (validation set)

	NOVIB	VIBMAX	VIBMIN
NOVIB	100%	0%	0%
VIBMAX	0%	62.0%	38.0%
VIBMIN	0%	10.3%	89.7%
F1 SCORE	1.00	0.70	0.82

Imagem 5: Resultados obtidos após o treinamento

Feature explorer (full training set) ?



On-device performance ?



Imagem 6: Gráfico resultante da rede neural e uso de hardware.

2.3. Teste do Modelo

A seguir temos o resultado do teste do modelo, apresentando resultados relativamente bons, considerando a similaridade das labels do dataset

Model testing results

ACCURACY
72.41%



	NOVIB	VIBMAX	VIBMIN	UNCERT...
NOVIB	93.2%	0%	6.8%	0%
VIBMAX	0%	34.6%	26.1%	39.2%
VIBMIN	0%	8.4%	82.4%	9.2%
F1 SCORE	0.96	0.49	0.72	

Imagem 7: Acurácia no teste do modelo.

Feature explorer ?

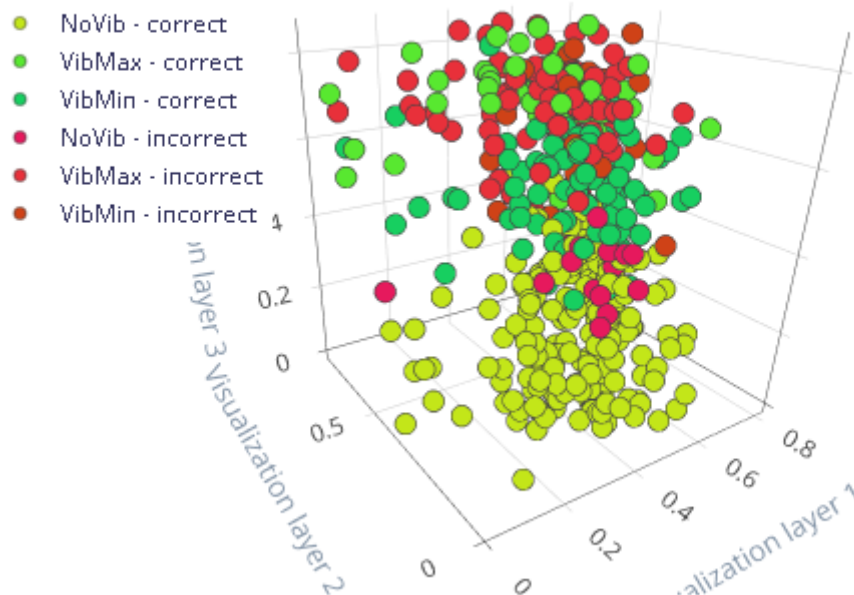


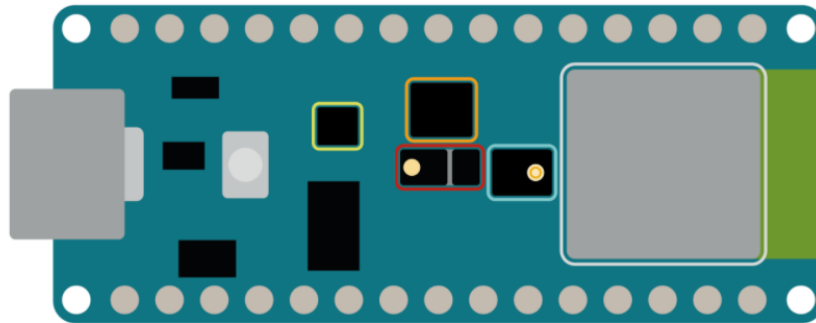
Imagem 8: Gráfico 3D dos testes.

3. Arduino Nano-33 BLE:

A placa de desenvolvimento utilizada, tanto na captura do dataset quanto para a execução final do código da rede neural foi o Arduino Nano 33 BLE. Um circuito desenvolvido propriamente para inteligência artificial, porém com baixo consumo de energia, o que o configura como um dispositivo próprio para tiny machine learning. Em vista de seu baixo consumo de energia e relativo baixo desempenho, o tamanho

total do algoritmo deve ser levado em consideração, bem como a sua complexidade de execução, além disso deve ser observado o hardware que está sendo utilizado no momento, para que haja a certeza de que o sistema consegue fornecer memória o suficiente.

A placa é equipada com sensor inercial de 9 eixos, sensor de temperatura e umidade, barômetro, microphone, sensor de gestos, de proximidade, de cor e de luz.



- ◆ Color, brightness, proximity and gesture sensor
- ◆ Digital microphone
- ◆ Motion, vibration and orientation sensor
- ◆ Temperature, humidity and pressure sensor
- ◆ Arm Cortex-M4 microcontroller and BLE module

Imagem 9: Sensores Arduino nano 33 ble

NoVib

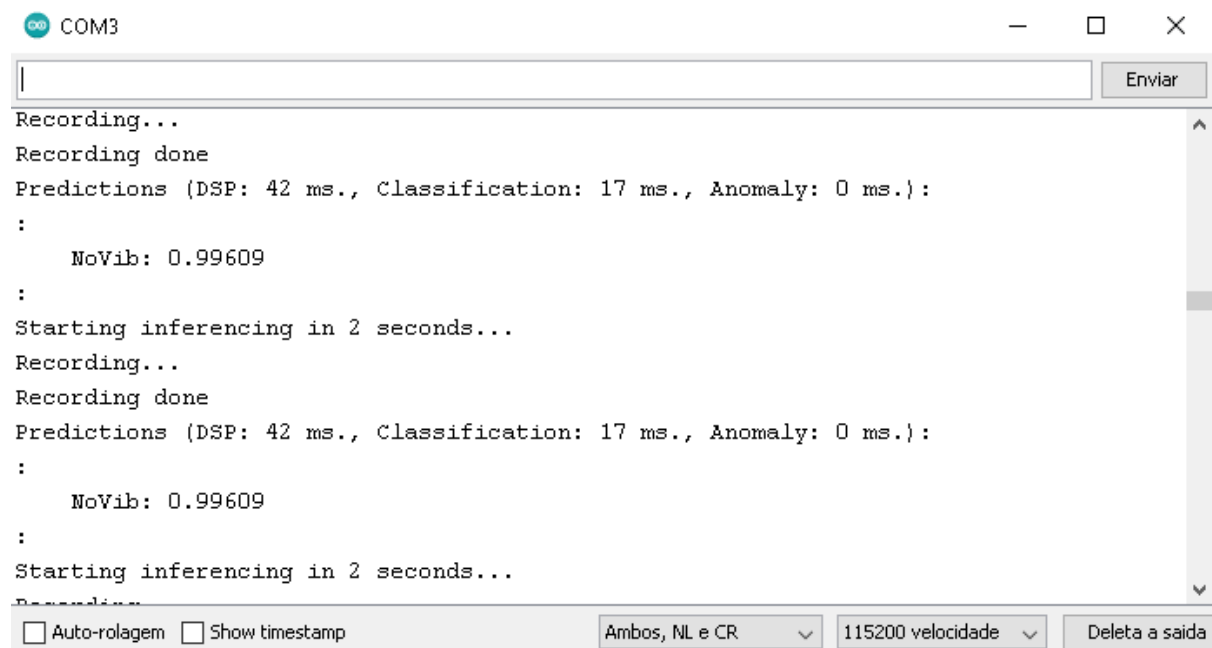


Imagem 10: Resultado de teste com o sistema sem abalos.

VibMax:

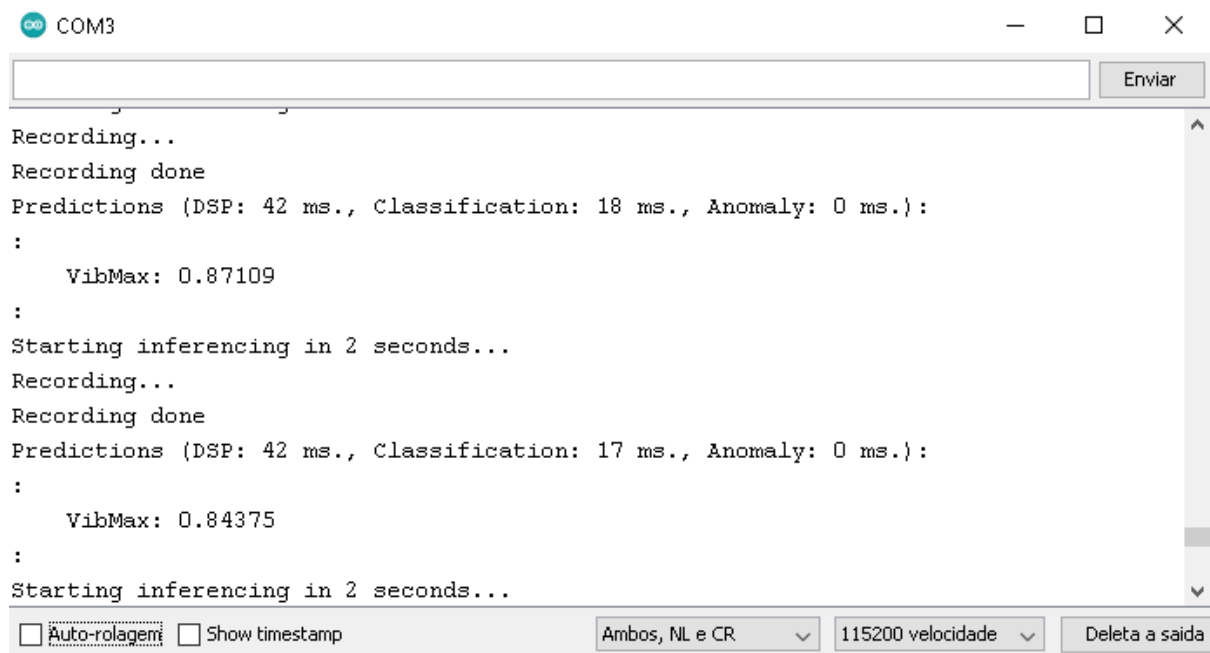


Imagem 11: Resultado de teste com o sistema com abalos intensos.

VibMin:

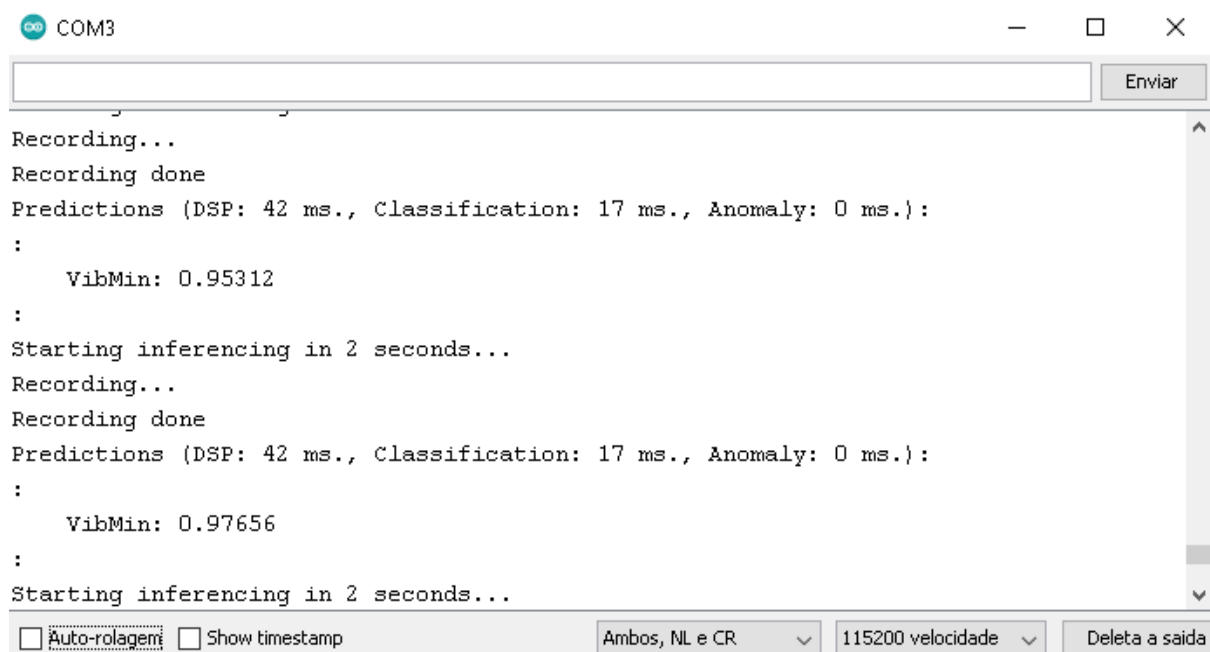


Imagem 12: Resultado de teste com o sistema com abalos leves.

O vídeo com o acendimento dos LEDs será incluído no conjunto da tarefa no SIGAA.

4. Conclusão

Infelizmente não foi possível seguir a ideia original que seria utilizar sensor sísmico e dataset pré existente obtido na barragem. Empecilhos esses, resultantes do elevado

custo do sensor e da dificuldade de acesso, tanto ao canteiro de obras quanto ao material de estudo dos engenheiros.

Porém o resultado final foi satisfatório, sendo possível simular de forma fidedigna os abalos sísmicos, além do sistema detectar com exatidão o tipo de abalo ocorrendo no momento.

O projeto pode ser encontrado através do link:

<<https://studio.edgeimpulse.com/public/44276/latest>>.

O link a seguir se refere ao drive com o código do arduino e o vídeo do projeto em funcionamento:

<<https://drive.google.com/drive/folders/1wEPyuYeBpaj7ZK4ZASVJ4eIk9-LF52jc?usp=sharing>>