Aplicação de Árvore de Decisão para Classificação de Sinais de Vibração em Ponte

Wander P. Jesus, Evandro J.G. Lima, Douglas A. Rocha e Zélia M. A. Peixoto, Membro *IEEE*

***Resumo*— Este trabalho apresenta um sistema de geração e treinamento de uma árvore de decisão para classificação de sinais de vibração em pontes. Para o controle do sistema foi utilizado o microcontrolador ESP32-WROOM-32, e o módulo acelerômetro MPU5060 para medições das vibrações, ambos responsáveis pela geração do *dataset* usado na etapa de treinamento. A acurácia dos resultados confirma o bom desempenho da ferramenta e indica possibilidades promissoras para classificação de sinais de vibração em estruturas que requerem monitoramento de saúde estrutural.**

Palavras-Chave*— *Árvore de decisão, Acelerômetro, Vibração, Ponte.

***Abstract*— This paper presents a system of generating and training a decision tree to classify vibration levels in bridges. The ESP32-WROOM-32 microcontroller was used to control the system, and the MPU5060 accelerometer module was used to measure vibrations, both responsible for generating the dataset used in the training stage. The acuracy of the results confirms the good performance of the tool and indicates the promises for classifying the vibration signals in the structures that monitor structural health.**

Keywords — Decision tree, Accelerometer, Vibration, Bridge.

I. INTRODUÇÃO

Pontes são estruturas civis importantes devido à sua possibilidade de melhoramento de trânsito e a conexão entre locais. Como resultado do uso permanente, é essencial que essas estruturas sejam confiáveis e seguras para os usuários, evitando danos materiais e acidentes graves, como perda de vidas humanas, o que torna importante que quaisquer alterações nessas estruturas sejam monitoradas [1].

As vibrações são naturais em qualquer sistema físico, e as pontes estão sujeitas a constantes efeitos mecânicos como, a passagem de veículos e pedestres. Os problemas acontecem quando ocorrem danos que produzem variações significativas de rigidez e massa dessas estruturas, de modo que a intensidade das vibrações se eleve. Nesta situação, as estruturas podem vibrar em frequências com amplitudes crescentes, e sob o efeito de ressonância, causar rupturas irreversíveis e ruir [2].

Este trabalho consiste no treinamento de uma árvore de decisão para classificar vibrações em pontes e viadutos. A geração do *dataset* foi realizada por um protótipo, de confecção autoral, em *software* EAGLE®, e construído em uma máquina prototipadora LPKF Protomat® S63. Os dados foram usados para treinamento e geração de uma árvore de decisão desenvolvida em ambiente MATLAB®.

W.P. Jesus (wander.pjesus@gmail.com), E.J.G Lima (evandrojgl@gmail.com), D.A. Rocha (douglasabreudarocha@gmail.com) e Z.M.A. Peixoto (assiszmp@pucminas.br), Departamento de Engenharia Eletrônica e de Telecomunicação, Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte-MG-Brasil.

A. Árvore de Decisão

Árvore de decisão é um recurso de *machine learning* que classifica dados através da divisão do conjunto de domínios do problema. A técnica consiste em uma árvore binária em que cada nó é uma estrutura do tipo *if-then-else* contendo um teste sobre algum atributo. O algoritmo de aprendizado utiliza amostras de casos, nos quais se conhecem as classificações verdadeiras, para aplicar condições pré-definidas. Para diferenciar as categorias, um resultado esperado deve ser designado para cada padrão de dados [3].

Um dos meios utilizados para treinar uma árvore de decisão é o algoritmo CART (*Classification and Regression Trees*), que consiste em uma técnica não paramétrica para induzir árvores de classificação e regressão, dependendo se o atributo classe é categórico (classificação) ou contínuo (regressão). As árvores geradas pelo CART são binárias, isto é, respondem apenas questões simples do tipo "sim" ou "não", desde a raiz (*root node*) até as folhas (*leaf nodes*) [4]. Um exemplo de utilização deste algoritmo, é a função *Fitctree*, disponibilizada no MATLAB®.

II. MÉTODO

Para a implementação deste trabalho, foi construído um protótipo de monitoramento de vibração e posição [5]. O protótipo é composto de um sensor *Microelectromechanical Systems* (MEMS) modelo MPU6050, da InvenSense®. Este módulo contém um acelerômetro e um giroscópio, ambos com 3 eixos, caracterizando 6 graus de liberdade, DOF (*Degrees of Freedom*), controlados por um microcontrolador ESP32-WROOM-32 da Espressif®. Neste trabalho, o protótipo foi usado para a geração do *dataset* a partir da leitura dos sinais de vibração de uma ponte em escala reduzida, vibrada mecanicamente por um *shaker*. Os dados gerados disponibilizaram três condições específicas de vibração, identificadas com os rótulos normal, alterado e sobrecarga, aos quais indicam, respectivamente, uma situação de vibração para condições de estrutura sem danos, uma condição cujos valores de vibração estariam alterados por um possível dano na estrutura e uma condição de níveis de vibração crítico. Os danos foram simulados com a remoção manual de peças da estrutura. Os sinais de vibração foram lidos pelo protótipo via comunicação USB, através de um algoritmo programado em ambiente de desenvolvimento integrado Arduino, e armazenados em uma planilha no formato CSV (*Comma Separated Values*). Essa planilha foi alimentada com 1500 valores, sendo 500 para cada uma das condições supracitadas. A Figura 1 mostra o protótipo usado para geração dos dados.

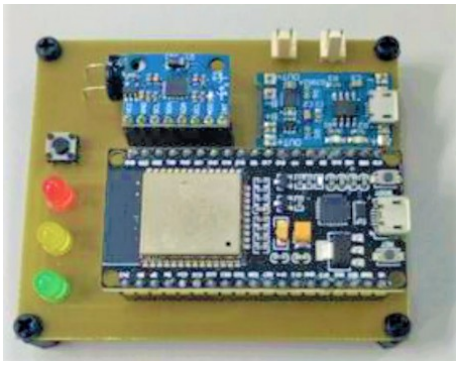

Fig.1. Protótipo com microcontrolador ESP32-WROOM-32 (Espressif) e MPU6050 (InvenSense).

As Figuras 2 a 4 mostram os três gráficos dos três valores de amplitude das vibrações lidos do módulo MPU6050, consistindo dos sinais AccX (vermelho), AccY (verde) e AccZ (Azul). Os valores médios de vibração foram usados para as *features* de treinamento.

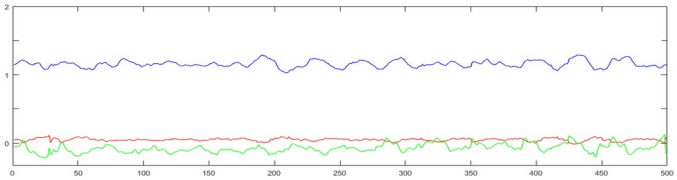

Fig.2. Valores lidos do MPU6050 em situação normal.

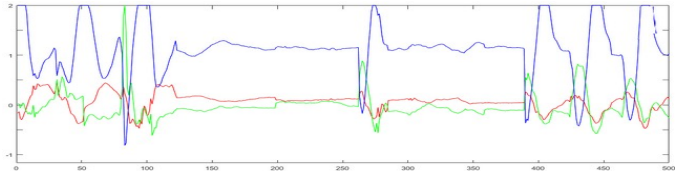

Fig.3. Valores lidos do MPU6050 em situação alterado.

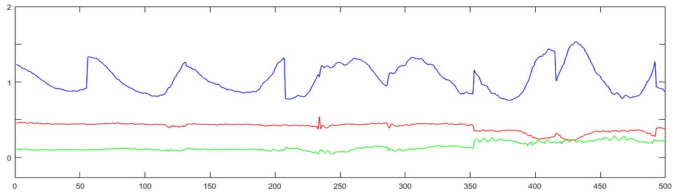

Fig.4. Valores lidos do MPU6050 em situação de sobrecarga.

Para a geração da árvore de decisão, foi programado em três etapas, um algoritmo com a utilização da ferramenta *Classification Learner* do MATLAB®. Na primeira, foi feito o carregamento da planilha com o *dataset*. Na segunda etapa, foi criada uma seção de classificação utilizando-se como preditores (*Predictors*), os 3 valores lidos do acelerômetro. Finalmente, foram gerados os três modelos de árvores de decisão *Coarse Tree, Medium Tree* e *Fine Tree* disponibilizados no *software*, a partir da função *Fitctree*.

III. RESULTADOS

Ao gerar os modelos, foi observado que o modelo *Fine Tree* obteve maior acurácia, alcançando o valor de 97,2%, contra 96% do modelo *Medium Tree* e 89,3% do modelo *Coarse Tree*. A matriz de confusão gerada para o modelo *Fine Tree* é mostrada na Figura 5, com índices de acerto iguais a 94%, 97% e 99% para os *status* Alterado, Normal e Sobrecarga, respectivamente.

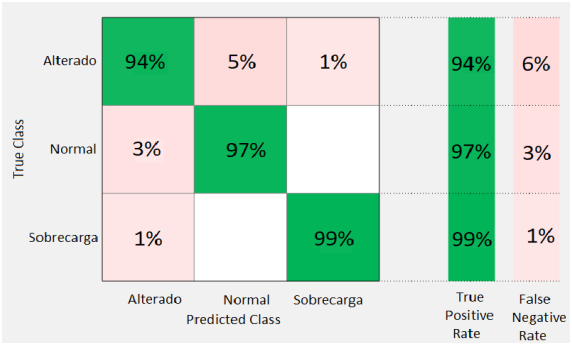

Fig.5. Matriz de confusão do modelo *Fine Tree*.

A curva ROC (*Receiver Operating Characteristic*) para o modelo *Fine Tree* é apresentada na Figura 6. Pode ser observado que a área sob a curva, AUC (*Area Under The Curve*) é igual 0,98, denotando uma probabilidade de 98% de predição correta para o estado de vibração normal, com 3% de falsos positivos e 97% de verdadeiros positivos. Para os sinais de vibração em alerta e sobrecarga, os valores da AUC para a ROC foram de 98% e 99%, respectivamente.

Podemos perceber que em ambas as métricas os valores obtidos demostraram uma boa performance do modelo treinado, validando o desempenho da árvore de decisão.

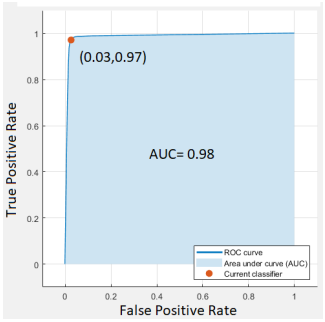

Fig.6. Curva ROC para os níveis de vibração normal.

IV. CONCLUSÕES

Neste trabalho foi apresentado um método para treinamento de uma árvore de decisão em ambiente MATLAB®. Os resultados apresentados demonstraram uma promissora possibilidade de implementação da técnica em um *hardware* para aplicações de *machine learning*. A continuidade deste estudo, já em andamento, tem condições propícias para projetar um sistema real de prevenção de acidentes e intervenções que requerem monitoramento de saúde estrutural em pontes, com intuito de evitar danos patrimoniais e perdas de vidas humanas.

V. REFERÊNCIAS

[1] H. Bachmann, W. Ammann e F. Deischl, Vibration Problems in Structures Practical Guidelines, *Birkhauser Verlag*, 1994.

[2] N. Subramanian, Design of structured concrete structures, *Oxford*, 2013.

[3] J. Quinlan, C4.5: Programs for Machine Learning, *London – Cassell*, 2014.

[4] M. Hasanipanah, R. Faradonbeh and H. Amnieh, Forecasting blast-induced ground vibration developing a CART model. *Engineering with Computers 33*, 2017: 307-316.

[5] W. P. Jesus, E. J. G. Lima, R. G. Alvares, J. R. Silveira and Z. M. A. Peixoto, Vibration and position monitoring in civil structures using Wi-Fi systems, *Set international journal of broadcast engineering*, 2019: 93-98.