

Detecção de danos em dormentes ferroviários baseada em aprendizado de máquina usando FFT

1st Bruno Moreira da Silva
*Pós-graduação em Instrumentação,
Controle e Automação de Processos de
Mineração*
Universidade Federal de Ouro Preto
Ouro Preto, Brasil
bruno.moreira1@aluno.ufop.edu.br

2nd Alan Kardek Rêgo Segundo
Departamento de Controle e
Automação
Universidade Federal de Ouro Preto
Ouro Preto, Brasil
alankardek@ufop.edu.br

3rd Eleir Mundim Bortoleto
Instituto Tecnológico Vale
Universidade Federal de Ouro Preto
Ouro Preto, Brasil
eleir.bortoleto@itv.org

Resumo— Este artigo apresenta uma abordagem de aprendizado de máquina supervisionado para classificar danos estruturais em dormentes de aço ferroviários usando sinais de vibração. Um sistema embarcado de baixo custo foi desenvolvido para coletar dados de vibração com o uso de acelerômetros MEMS instalados em dormentes. Os sinais adquiridos foram processados com a Transformada Rápida de Fourier (FFT) para extrair características no domínio da frequência. Essas características foram usadas para treinar e avaliar diferentes classificadores, incluindo Árvores de Decisão, *Random Forest* e Redes Neurais MLP. Os modelos foram avaliados por meio de métricas de desempenho, validação cruzada e testados em dois conjuntos adicionais para verificar sua capacidade de generalização. O classificador MLP apresentou o melhor desempenho, com acurácia de 91,7% no conjunto de teste, além de boa robustez frente a novos dados. Os resultados demonstraram a viabilidade da implementação de modelos de baixa complexidade computacional em dispositivos de computação de borda, como o ESP32, para a detecção automática de danos em dormentes de aço ferroviários pro.

Palavras chaves—Dormentes ferroviários, Monitoramento de vibração, Aprendizado de máquina, FFT, Redes neurais

I. INTRODUÇÃO

As estruturas de engenharia estão constantemente expostas a esforços excessivos, intempéries e ações dinâmicas, o que pode levar à sua deterioração, ruptura ou até colapso. Esse cenário reforça a importância do monitoramento contínuo dessas estruturas para prevenir incidentes de gravidade moderada a fatal [1].

No contexto ferroviário, os dormentes — elementos fundamentais para a estabilidade da via — estão sujeitos ao estresse mecânico cíclico e à degradação ambiental no decorrer do tempo. Métodos tradicionais de inspeção, como a avaliação visual, são frequentemente morosos, subjetivos e incapazes de detectar falhas incipientes. A integridade estrutural das ferrovias é essencial para evitar problemas como rompimento de trilhos, desnivelamento da via e descarrilamentos. Além de aumentar a segurança operacional, o monitoramento contínuo reduz custos com a manutenção corretiva e previne possíveis acidentes [2].

O Monitoramento de Saúde Estrutural (*Structural Health Monitoring* - SHM) é um processo que além de estimar a integridade de estruturas, também identifica falhas desde os estágios iniciais até a manifestação de danos significativos [3]. Esse campo tem evoluído significativamente com o uso de tecnologias diversas — térmicas, elétricas, ópticas, magnéticas e computacionais — conforme afirmam Das e Saha [4], o SHM compreende as etapas de detecção,

aquisição, transmissão e processamento de sinais, culminando na apresentação de dados para o diagnóstico estrutural.

Nesse contexto, o monitoramento de vibrações tem se destacado como uma técnica eficiente, pois permite, por meio do processamento dos sinais — utilizando métodos como a Transformada Rápida de Fourier (*Fast Fourier Transform* - FFT) — identificar as frequências naturais da estrutura [5]. Alterações nas frequências modais, ao longo do tempo, podem indicar degradações ou danos, possibilitando intervenções preventivas e corretivas [6]. Além disso, as informações extraídas podem ser utilizadas como entrada para algoritmos de aprendizado de máquina, permitindo a construção de modelos capazes de classificar automaticamente estruturas danificadas e não danificadas com base em seus padrões vibracionais [7].

Técnicas baseadas em inteligência artificial vêm ganhando relevância nesse cenário, devido à sua capacidade de identificar padrões complexos em sinais de vibração, oferecendo soluções autônomas e escaláveis para a classificação de falhas estruturais. Com os avanços da computação embarcada e da computação de borda (*edge computing*), tornou-se viável a execução de algoritmos leves de aprendizado de máquina diretamente em dispositivos microcontroladores. Modelos como Árvores de Decisão, *Random Forest* e Redes Neurais Artificiais têm sido amplamente aplicados para esse fim, permitindo diagnósticos mais rápidos e confiáveis [8].

Neste artigo, propõe-se a investigação e a comparação de diferentes algoritmos de aprendizado de máquina supervisionado aplicados à classificação de danos em dormentes ferroviários, com o objetivo de identificar o modelo mais adequado para implementação embarcada. Para isso, sinais de vibração são adquiridos por sensores acelerômetros, processados por meio da FFT e utilizados como entrada para modelos dos tipos Árvore de Decisão, *Random Forest* e Rede Neural MLP (*Multi-Layer Perceptron*). A avaliação do desempenho dos modelos é realizada com base nas métricas extraídas das matrizes de confusão obtidas nos conjuntos de treinamento e de teste. Além disso, emprega-se a validação cruzada do tipo *k-fold* e testes com dois conjuntos de dados adicionais para verificar a capacidade de generalização dos modelos.

II. TRABALHOS RELACIONADOS

O monitoramento de vibrações estruturais com sensores acelerômetros e sistemas embarcados tem sido amplamente estudado em diversas aplicações, como ferrovias, pontes, barragens e material rodante. No contexto ferroviário, Stenström *et al.* [9] e Milne *et al.* [10], [11] propuseram

sistemas embarcados com acelerômetros MEMS (*Micro-Electro-Mechanical Systems*) de baixo custo instalados diretamente nos dormentes e com dados armazenados localmente em cartão de memória. Seus estudos mostram que é possível medir deslocamentos e rigidez do lastro com precisão comparável a sensores de alta precisão, porém de maior custo.

Kaewunruen e Remennikov [5], [12] utilizaram análise modal via FFT para detectar trincas e vazios nos dormentes, relacionando variações nas frequências modais com o estado estrutural. Esses trabalhos indicam que uma taxa de amostragem de 512 Hz de um acelerômetro é adequada para capturar modos de vibração relevantes para a compreensão da dinâmica de dormentes.

Além do setor ferroviário, Zanelli *et al.* [13] aplicaram sensores MEMS e comunicação BLE (*Bluetooth Low Energy*) para análise modal em pontes ferroviárias, enquanto Fernandes *et al.* [6] propuseram um sistema com protocolo LoRaWAN (*Long Range Wide Area Network*) para monitorar vibrações em estruturas de barragens. Grimmelsman e Zolghadri [14] compararam acelerômetros de baixo e alto custo, validando o desempenho de sensores MEMS em SHM.

Além das aplicações tradicionais em ferrovias, o SHM vem se beneficiando da integração com tecnologias emergentes, como a inteligência artificial embarcada na borda. Segundo Angeline [15], a combinação de sensores de aceleração e giroscópio com algoritmos de aprendizado de máquina, como *Random Forest*, *Gradient Boosting* e Redes Neurais, permite detectar anomalias em tempo real. O estudo de Angeline [15] destaca o uso do SMOTE (*Synthetic Minority Over-sampling Technique*) para balanceamento de classes e demonstra que modelos treinados em dados reduzidos, mas pré-processados de maneira satisfatória, podem alcançar recuperação superior a 93%, reforçando o potencial do uso de modelos de aprendizado de máquina processados na borda aplicado no SHM.

No campo da inteligência embarcada, Katsidimas *et al.* [16] propuseram um sistema de monitoramento extremo, capaz de realizar localização de impactos em tempo real diretamente no dispositivo. Utilizando sensores piezoelétricos e modelos de *Random Forest* e redes neurais de baixa complexidade implementados com *TinyML*, os autores alcançaram precisão superior a 90%, com latência inferior a 400 ms, demonstrando a viabilidade de SHM embarcado com baixo consumo.

Zhang *et al.* [17] apresentaram um modelo leve de Rede Neural convolucional (*CNN*) para identificação de danos em concreto via sinais de emissão acústica. O modelo foi embarcado em um microcontrolador ARM Cortex-M4, alcançando 99,6% de acurácia com apenas 30,5 kB e inferência em 166 ms. Esse estudo reforça que mesmo arquiteturas mais complexas como *CNNs* de menor número de camadas e neurônios podem ser otimizadas para SHM em dispositivos de baixo recurso computacional.

Conforme o estudo de Zhao e Wang [18] foi desenvolvido um sistema embarcado com base em *edge computing* para o monitoramento de condições de motores industriais, fazendo uso de sensores MEMS, microcontroladores e modelos *TinyML* integrados. Nesse sistema, a FFT é processada localmente para extrair características das vibrações, e a inferência é realizada diretamente no dispositivo por meio dos pesos e vieses obtidos de uma Rede Neural MLP treinada,

alcançando elevada acurácia e comprovando sua viabilidade mesmo em ambientes industriais desafiadores.

Esses trabalhos evidenciam o avanço das abordagens embarcadas para detecção de danos em estruturas, e corroboram com o propósito deste estudo, que busca identificar o algoritmo de aprendizado supervisionado mais viável para embarque em microcontroladores de baixo custo com o objetivo da detecção automática de danos em dormentes ferroviários.

III. METODOLOGIA

A. Arquitetura do sistema desenvolvido

O sistema proposto é composto por dois módulos principais: um nó sensor e um nó centralizador, ambos baseados na placa Heltec Wi-Fi LoRa 32 V2 (ESP32 embutido). O nó sensor integra um acelerômetro MEMS MPU-6050, responsável pela medição das vibrações nos dormentes ferroviários, cujas leituras de aceleração foram realizadas a uma taxa de amostragem de 512 Hz.

Após essa aquisição, o sinal foi processado localmente no próprio nó sensor, onde aplicou-se a FFT para extração das componentes no domínio da frequência. Em seguida, o vetor resultante da FFT foi transmitido via comunicação LoRa (*Long Range*) para o nó centralizador.

Conectado à internet via Wi-Fi, o nó centralizador, que atua como *gateway*, encaminha os dados recebidos ao ambiente de aplicação em tempo real. Essa comunicação foi realizada por meio do protocolo MQTT (*Message Queuing Telemetry Transport*) com integração à plataforma Node-RED, que também foi responsável pela visualização das informações recebidas por *dashboards*. Os dados foram automaticamente armazenados em um banco de dados MySQL, gerenciado pela interface phpMyAdmin, permitindo sua posterior exportação em formato CSV para o treinamento dos modelos de aprendizado de máquina. Na Figura 1, pode ser observado o sistema desenvolvido.

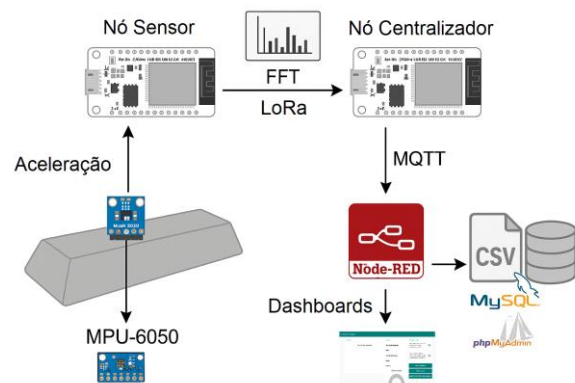


Fig. 1. Arquitetura do sistema desenvolvido para aquisição dos dados.

B. Setup experimental e coleta de dados em laboratório

Para a coleta de dados, foram utilizados dez dormentes de aço em ambiente controlado de laboratório, sendo seis em boas condições estruturais e quatro com sinais visíveis de danos (trincas ou degradação do material). Cada dormente foi excitado manualmente com um martelo de borracha em seis diferentes regiões, e para cada região foram realizadas três excitações independentes, totalizando 18 excitações por dormente.

O nó sensor foi fixado diretamente sobre os dormentes durante os testes. As Figuras 2 e 3 exemplificam os dormentes utilizados, com e sem danos, e a configuração física do sensor embarcado.



Fig. 2. Dormente de aço com dano estrutural.



Fig. 3. Dormente de aço sem dano estrutural e nó sensor instalado.

C. Filtragem dos dados

Durante cada excitação com o martelo de borracha, o sinal capturado pelo acelerômetro apresentou uma resposta vibracional prolongada. Assim, uma única excitação mecânica permitiu a extração de múltiplos segmentos válidos do sinal, uma vez que a vibração reverberou pelo dormente por alguns segundos. Para cada segmento extraído, calculou-se a FFT, resultando um vetor de 256 componentes espectrais de amplitude, o qual representa uma amostra de dados a ser utilizada nos modelos de aprendizado de máquina.

Os múltiplos vetores obtidos em cada tentativa de excitação foram transmitidos para o banco de dados MySQL. Posteriormente, esses dados foram exportados em formato CSV, organizados e pré-processados para treinamento dos modelos. Como etapa de filtragem, mantiveram-se apenas os vetores, cuja amplitude máxima fosse superior a 1, tendo como objetivo descartar amostras que podem apresentar ruídos. Essa etapa assegurou que o conjunto de dados fosse composto apenas por amostras representativas de eventos vibracionais relevantes.

D. Preparação do conjunto de dados

Antes do treinamento dos modelos de aprendizado de máquina, o conjunto de dados filtrados, contendo os vetores de amplitude da FFT, foram normalizados utilizando a técnica de *Z-score*. Essa técnica transforma os dados para uma distribuição com média zero e desvio padrão igual a um. Essa padronização foi realizada com o método *StandardScaler*, da biblioteca *Scikit-learn* [19], conforme a Equação:

$$Z = (x - \mu) / \sigma \quad (1)$$

Em que x representa o valor original, μ a média da variável e σ o seu desvio padrão. Essa etapa é fundamental para garantir que todas as características tenham a mesma escala, evitando o viés de variáveis com magnitudes maiores.

As estatísticas de normalização (média e desvio padrão) foram calculadas a partir do conjunto completo de amostras do treinamento e armazenadas com o objetivo de utilizar posteriormente em novos conjuntos de dados e para um possível embarque do modelo com melhor desempenho no microcontrolador ESP32.

E. Modelos de aprendizado de máquina

O aprendizado de máquina supervisionado é amplamente utilizado no SHM para classificar estados da estrutura com base em dados de sensores. O método de classificação utiliza um conjunto de dados rotulados, em que as entradas possuem saídas conhecidas, com o objetivo de treinar o modelo para realizar previsões sobre novos dados [20]. Neste trabalho, foram avaliados três modelos supervisionados de classificação: Árvore de Decisão, *Random Forest* e Rede Neural MLP.

A Árvore de Decisão organiza os dados (*features*) de forma hierárquica, a qual toma decisões com base em condições sucessivas até atingir uma classificação final [21]. Já o modelo *Random Forest* combina múltiplas Árvore de Decisão para aumentar a robustez e reduzir o risco de *overfitting* [20]. Enquanto o MLP é uma rede neural composta por camadas de neurônios interligados. Esse tipo de rede neural é capaz de modelar relações não lineares complexas entre entradas e saídas. Cada neurônio realiza operações matemáticas baseadas em somas ponderadas e funções de ativação [22].

Os modelos de Árvore de Decisão, *Random Forest* e MLP foram escolhidos por aliar bom desempenho preditivo e viabilizar o embarque no ESP32. Os modelos baseados em árvores (Árvore de Decisão e *Random Forest*) apresentam baixa complexidade computacional na inferência e podem ser facilmente implementados em C/C++. O modelo MLP com menor número de camadas e neurônios permite conversão para *TensorFlow Lite*, possibilitando execução eficiente em microcontroladores com recursos limitados. No caso de modelos como o SVM (*Support Vector Machine*), que é muito utilizado para aplicações em conjuntos de dados de FFT, além de gerar maiores números de vetores de suporte, possui maior custo computacional e não conta com suporte nativo no *TensorFlow Lite*, o que dificulta sua aplicação embarcada [8].

Todos os modelos foram treinados com validação cruzada *k-fold* de 5 dobras, permitindo uma análise mais confiável do desempenho. A avaliação do desempenho dos modelos foi realizada com base na matriz de confusão e nas métricas derivadas, como acurácia, precisão, *recall* e *F1-score*. Além disso, foi utilizada a curva ROC (*Receiver Operating Characteristic*) para analisar a capacidade discriminativa dos modelos. A curva ROC utiliza a taxa de verdadeiros positivos em relação à taxa de falsos positivos, com o cálculo da AUC (*Area Under the Curve*), como medida complementar de desempenho [23].

F. Características dos modelos de aprendizado de máquina para treinando e testes

Para a tarefa de classificação binária entre dormentes com e sem danos estruturais, foram avaliados três modelos supervisionados amplamente utilizados em problemas de aprendizado de máquina: Árvore de Decisão, *Random Forest* e Rede Neural MLP. O conjunto de dados foi dividido em 70% para treinamento e 30% para teste, garantindo que os modelos fossem avaliados com dados não vistos durante o aprendizado.

Para minimizar o desbalanceamento entre as classes foi utilizado o SMOTE, quando necessário maximizar o desempenho do modelo, que gera novas instâncias sintéticas da classe minoritária a partir da interpolação de amostras existentes, tornando as classes mais equilibradas e permitindo que os modelos aprendam padrões relevantes da classe menos representada. Os hiperparâmetros de cada modelo foram ajustados conforme as tabelas a seguir.

1) Árvore de Decisão

A Árvore de Decisão foi escolhida por sua simplicidade e capacidade de interpretação e utilizados o critério *Gini* e o balanceamento SMOTE. A validação cruzada com 5 dobras (*k-fold*) foi utilizada também com SMOTE integrado ao *pipeline*. Os hiperparâmetros, para maximizar o desempenho do modelo, foram escolhidos conforme a Tabela I.

TABELA I
Hiperparâmetros Árvore de Decisão

Parâmetros	Valor
Critério	<i>Gini</i>
Profundidade máxima	3

2) Random Forest

O modelo *Random Forest* foi utilizado com o objetivo de aumentar a robustez do modelo e minimizar variações nos dados de entrada. Foram utilizadas 50 árvores com profundidade máxima controlada, bem como um número mínimo de amostras por divisão e por folha. Os hiperparâmetros foram escolhidos conforme a Tabela II. A ponderação entre classes foi mantida balanceada para compensar o desbalanceamento original da base e também utilizada a técnica de SMOTE.

TABELA II
Hiperparâmetros *Random Forest*

Parâmetros	Valor
Critério	<i>Gini</i>
Número de árvores	50
Profundidade máxima	6
class_weight	<i>balanced</i>

3) Rede Neural MLP

O MLP foi adotado por sua capacidade de modelar relações não lineares nos dados espectrais da FFT. A arquitetura consistiu em duas camadas ocultas com 4 neurônios cada, utilizando a função de ativação tangente hiperbólica e saída *sigmoid*, conforme a Tabela III. O modelo foi treinado inicialmente com o *Scikit-learn* e sem a técnica de SMOTE, depois replicado em *Keras* com os mesmos pesos e vieses, permitindo sua posterior conversão para *TensorFlow Lite*. Desse modo, é possível o embarque do modelo MLP no

microcontrolador ESP32 contido na placa Heltec WiFi LoRa V2.

TABELA III
Hiperparâmetros MLP

Parâmetros	Valor
Camadas ocultas	(4, 4)
Função de ativação	<i>tanh</i>
Função de saída	<i>sigmoide</i>
Função de perda	<i>binary cross-entropy</i>
Otimizador	<i>adam</i>
Regularização (alpha)	0.5
Iterações máximas	2000

Todos os hiperparâmetros utilizados foram definidos com base em uma análise comparativa das métricas de avaliação (acurácia, precisão, *recall* e *F1-score*) e na interpretação das matrizes de confusão obtidas durante os testes. Essa abordagem permitiu maximizar a eficiência do modelo e ajustar o desempenho com foco na detecção correta dos dormentes danificados.

A validação cruzada com *k-fold* ($k=5$) foi aplicada utilizando os mesmos hiperparâmetros previamente definidos para cada modelo. O objetivo foi avaliar a robustez dos modelos sem realizar ajuste adicional durante a validação, permitindo uma análise mais fiel do desempenho com os parâmetros fixos. Assim, a validação *k-fold* foi utilizada como método de estimativa da generalização dos modelos configurados.

Todo o *pipeline* de processamento, treinamento e avaliação dos modelos de aprendizado de máquina foi desenvolvido na linguagem Python 3.10 e implementado localmente com o uso do software VS Code, utilizando os seguintes pacotes:

- *Scikit-learn* para criação e avaliação dos modelos de classificação;
- *Imbalanced-learn* – para aplicação da técnica de balanceamento SMOTE;
- *Pandas* e *NumPy* – para manipulação e estruturação dos dados;
- *Matplotlib* – para visualização de métricas e curvas ROC;
- *TensorFlow/Keras* – para construção do modelo MLP equivalente em *Keras* e conversão para o formato *.tflite*.

A organização dos dados, treinamento e exportação dos modelos seguiu princípios de reprodutibilidade, com controle do estado aleatório (*random_state* = 42) nos principais algoritmos.

G. Conjunto de dados para validação

Após o treinamento e o ajuste dos modelos supervisionados — Árvore de Decisão, *Random Forest* e MLP — com o conjunto de dados principal, foi realizada uma etapa adicional de validação para verificar a capacidade de generalização dos modelos. Essa etapa consistiu na aplicação de cada modelo treinado em dois novos conjuntos de dados

independentes, contendo amostras de dormentes não utilizadas durante o treinamento.

Os conjuntos adicionais foram obtidos sob as mesmas condições experimentais, mantendo-se o protocolo de aquisição, a taxa de amostragem e o processo de normalização dos dados. As métricas utilizadas para essa validação incluíram acurácia, precisão, *recall* e *F1-score*, possibilitando uma análise comparativa detalhada do desempenho de cada modelo. Essa abordagem permite avaliar o comportamento dos algoritmos diante de variações nos dados e confirmar sua efetividade na detecção de danos estruturais em dormentes ferroviários em cenários não vistos previamente.

IV. RESULTADOS E DISCUSSÃO

Um conjunto de 439 amostras foi obtido a partir dos ensaios laboratoriais em dormentes de aço após a filtragem. Cada amostra foi rotulada como 0 (estrutura saudável) ou 1 (estrutura danificada). Esse conjunto de dados foi então normalizado e armazenado para o treinamento e testes dos modelos de aprendizado de máquina supervisionados para a classificação em relação a saúde do dormente.

Como o conjunto de dados apresentava desbalanceamento entre as classes (aproximadamente 40% de amostras de dormentes com dano e 60% de dormentes saudáveis), foi aplicada a técnica SMOTE, quando necessário, exclusivamente no conjunto de treinamento para o balanceamento das classes. Após, os hiperparâmetros de cada modelo (mencionados na metodologia) foram ajustados visando maximizar o desempenho preditivo e a capacidade de generalização.

Todos os resultados foram obtidos por meio do ambiente de desenvolvimento em Python, utilizando a biblioteca *scikit-learn*. A avaliação dos modelos foi conduzida por meio de quatro principais abordagens: (i) análise de desempenho a partir dos conjuntos de treino e teste; (ii) gráfico ROC; (iii) avaliação da validação cruzada *k-fold* (5 dobras) e (iv) desempenho em conjuntos de dados inéditos, o qual permite avaliar a capacidade de generalização dos modelos de forma mais robusta. Os resultados foram comparados com base em métricas, como acurácia, precisão, *recall*, *F1-score* e área sob a curva (AUC) obtida pelo gráfico ROC.

A. Treinamento

A Tabela IV apresenta os resultados de acurácia, precisão, *recall* e *F1-score* obtidos durante o treinamento dos modelos. Nota-se que tanto o MLP quanto a *Random Forest* obtiveram desempenho superior, atingindo valores acima de 97% de acurácia, o que evidencia boa capacidade de aprendizado. A *Árvore de Decisão*, apesar de mais simples e interpretável, teve desempenho inferior, o que é esperado para modelos menos complexos.

TABELA IV
Métricas do conjunto de treinamento

Modelo	Acurácia	Precisão (Dano)	Recall (Dano)	F1-score (Dano)
Árvore de Decisão	84,8%	86%	89%	87%
Random Forest	97,5%	95%	100%	98%
MLP (Keras)	98,7%	98%	98%	98%

B. Avaliação nos conjuntos de teste

A Tabela V apresenta as métricas do conjunto de testes para cada modelo, sendo extraídas de suas matrizes de confusão. Pode-se observar que tanto o modelo *Random Forest* quanto o MLP alcançaram resultados semelhantes. A *Árvore de Decisão*, embora se mostre um modelo interessante, teve um baixo desempenho comparado aos demais, por exemplo, uma acurácia de 84,8%.

TABELA V
Métricas do conjunto de testes

Modelo	Acurácia	Precisão (Dano)	Recall (Dano)	F1-score (Dano)
Árvore de Decisão	84,8%	77%	80%	78%
Random Forest	91,7%	81%	98%	89%
MLP (Keras)	91,7%	81%	98%	89%

As Figuras a seguir, ilustram as matrizes de confusão de cada modelo, evidenciando o comportamento de cada classificador em termos de acertos e erros nas predições. No caso da Figura 4 é evidenciado que a *Árvore de Decisão* obteve 76 classificações corretas da classe “Sem Dano” e 36 da classe “Com Dano”, com 11 falsos positivos e 9 falsos negativos. Já as Figuras 5 e 6 mostram que tanto o modelo *Random Forest* quanto o MLP erraram apenas uma amostra danificada, com *recall* de 97,8% para a classe “Com Dano”, o que indica excelente desempenho para aplicações onde a sensibilidade a falhas é crítica.

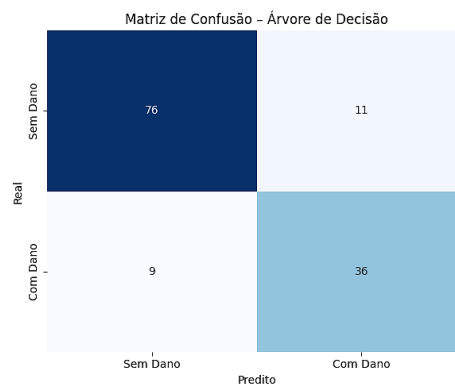


Fig. 4. Matriz de confusão Arvore de Decisão.

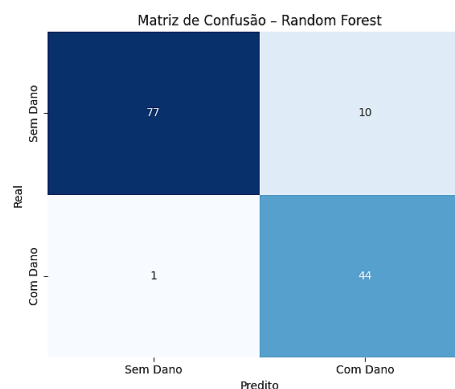


Fig. 5. Matriz de confusão Random Forest.

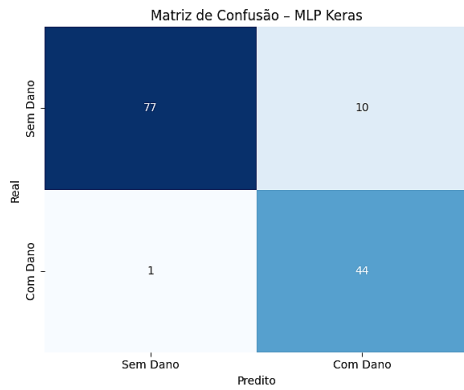


Fig. 6. Matriz de confusão MLP Keras.

C. Curvas ROC

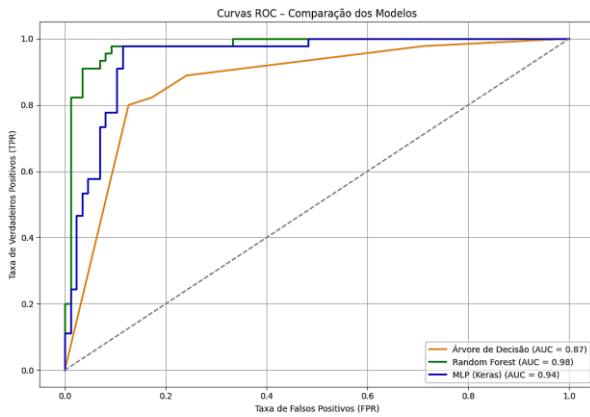


Fig. 7. Curvas ROC para análise dos modelos.

As curvas ROC dos modelos, Figura 7, demonstram que o modelo *Random Forest* foi o mais robusto neste cenário, apresentando a maior área sob a curva (AUC = 0,98), muito próxima do ideal (1,0). Isso indica que o modelo alcança uma alta taxa de verdadeiros positivos (TPR) com uma baixa taxa de falsos positivos (FPR), o que evidencia sua excelente capacidade discriminativa.

O modelo MLP (*Keras*) também apresentou desempenho muito confiável, com AUC de 0,94. Sua curva ROC é bastante elevada, indicando boa capacidade de generalização, mesmo frente a dados complexos, por exemplo, à aqueles extraídos por meio da FFT. Já a Árvore de Decisão apresentou a curva mais distante do canto superior esquerdo, sugerindo menor sensibilidade.

D. Avaliação da validação cruzada (5-Folds)

Para avaliar a capacidade de generalização dos modelos, foi utilizada a validação cruzada do tipo *k-fold* com $k=5$, em que o conjunto de dados foi particionado em cinco subconjuntos aproximadamente iguais. Em cada iteração, quatro subconjuntos foram usados para treinamento e um para teste, garantindo que cada instância fosse avaliada uma única vez. A Tabela VI apresenta as médias e os desvios padrão das métricas de desempenho obtidas ao longo dos cinco *folds*, com ênfase na classe de maior interesse: dormentes com dano.

TABELA VI
Métricas para avaliação da validação cruzada

Modelo	Acurácia ($\pm\sigma$)	Precisão (Dano) ($\pm\sigma$)	Recall (Dano) ($\pm\sigma$)	F1-score (Dano) ($\pm\sigma$)
Árvore de Decisão	79,77% $\pm 5,63\%$	66,22% $\pm 2,66\%$	84,81% $\pm 11,25\%$	74,11% $\pm 5,60\%$
Random Forest	88,18% $\pm 6,12\%$	80,27% $\pm 10,77\%$	88,67% $\pm 4,99\%$	84,00% $\pm 7,51\%$
MLP (Keras)	88,86% $\pm 1,51\%$	80,89% $\pm 3,68\%$	87,79% $\pm 3,66\%$	84,12% $\pm 2,58\%$

O modelo MLP apresentou menor desvio padrão nas métricas, indicando maior estabilidade entre os *folds*. O modelo *Random Forest*, embora com ótimo desempenho médio, demonstrou maior variação, tendo leve sensibilidade à divisão dos dados e sugerindo um possível *overfitting*. O modelo de Árvore de Decisão, além de menor desempenho, teve a maior variação do desvio padrão, evidenciando sua limitação em generalizar de forma consistente.

E. Avaliação de desempenho com conjunto de dados inéditos

Para a avaliação do desempenho dos modelos de aprendizado de máquina, foram utilizados dois conjuntos de dados inéditos: o conjunto de teste A, que foi composto por 111 amostras, e o conjunto de teste B que continha 260. Os dois conjuntos foram filtrados e normalizados conforme as estatísticas da normalização do conjunto de dados que treinou os modelos.

Ao aplicar os modelos nos dois conjuntos de dados inéditos, observou-se uma queda significativa no desempenho dos modelos *Random Forest* e Árvore de Decisão. Por outro lado, o MLP manteve desempenho robusto, com alta acurácia, precisão e *F1-score* para a classe “com dano”, ou seja, isso reforça sua capacidade de generalização diante à inferência de novos dados. Na Tabela VII, pode-se observar o alto desempenho da Rede Neural MLP em classificar novos dados comparados aos outros métodos abordados.

TABELA VII
Desempenho dos modelos nos conjuntos de dados inéditos para a classe “com dano”

Conjunto de dados A	Acurácia	Precisão	Recall	F1-score (com dano)
Árvore Decisão	56%	49%	83%	61%
Random Forest	65%	56%	74%	64%
MLP	88%	97%	74%	84%
Conjunto de dados B				
Árvore Decisão	28%	56%	26%	35%
Random Forest	45%	75%	41%	53%
MLP	87%	99%	83%	91%

De maneira geral, os resultados reforçam a estabilidade do MLP frente a novos dados, sendo um indicativo de sua viabilidade para aplicações reais em campo. Devido ao seu melhor desempenho dentre os demais, sua generalização e ser

um algoritmo de baixa complexidade computacional, o modelo MLP foi o selecionado para o embarque e a inferência na borda do ESP32.

Por outro lado, o modelo *Random Forest*, embora potente, demonstrou maior variabilidade e, com a utilização de novos dados, apresentou baixo desempenho que é o caso de um *overfitting*. A Árvore de Decisão é uma alternativa simples e leve para embarque, no entanto, teve um baixo desempenho em todas as avaliações.

É importante destacar que a limitação de 439 amostras está relacionada ao número de dormentes disponíveis para a aquisição dos dados. Por se tratar de um experimento realizado em laboratório, local controlado, o armazenamento dos dormentes de aço foi restrito devido ao peso e suas dimensões, o que dificultou a sua estocagem.

V. CONCLUSÃO

Este trabalho apresentou uma abordagem para detecção de danos em dormentes ferroviários por meio da análise de vibrações e aprendizado de máquina, utilizando vetores de FFT como entrada para classificadores supervisionados. A solução foi desenvolvida com foco em aplicação embarcada de baixo custo, explorando os modelos de aprendizado de máquina que requerem recursos computacionais limitados, como os disponíveis no microcontrolador ESP32 que integra a placa Heltec Wi-Fi LoRa 32 V2.

Foram avaliados três modelos de classificação — Árvore de Decisão, *Random Forest* e Rede Neural MLP — com base em métricas como acurácia, precisão, *recall*, *F1-score* e AUC, considerando tanto o desempenho em treino e teste quanto a validação cruzada e a aplicação em novos conjuntos de dados.

Dentre os modelos avaliados, o MLP se destacou por seu desempenho equilibrado, alta capacidade de generalização e menor variação entre os *folds*, demonstrando robustez mesmo diante de dados não vistos anteriormente. Além disso, pode ser embarcado na borda de microcontroladores, como o ESP32, visto que tem duas camadas ocultas com 4 neurônios sendo um modelo de Rede Neural com baixa complexidade.

O modelo *Random Forest* também apresentou bom desempenho, especialmente em termos de separabilidade (AUC), mas demonstrou maior sensibilidade à divisão dos dados e queda de desempenho em novos cenários, sugerindo risco de *overfitting*. A Árvore de Decisão, por sua vez, teve o desempenho mais modesto, com baixa estabilidade e acurácia reduzida, além de alta variância na validação cruzada.

Como perspectiva futura, destaca-se a possibilidade de embarcar o modelo MLP em microcontroladores de baixo consumo energético, como o ESP32, viabilizando uma solução embarcada de baixo custo para utilização em ambiente controlado. Além disso, a realização de coletas de dados em campo, com variações ambientais e operacionais, corrobora para a utilização do sistema proposto que visa à detecção automática de danos em dormentes da via ferroviária permanente.

Por fim, os resultados obtidos demonstraram a viabilidade do uso de modelos de baixa complexidade de aprendizado de máquina para a detecção de danos em dormentes ferroviários, contribuindo para soluções mais eficientes, autônomas e acessíveis no contexto das manutenções preditiva e preventiva.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ), do Instituto Tecnológico Vale (ITV) e da Universidade Federal de Ouro Preto (UFOP).

REFERÊNCIAS

- [1] Chang, Hung-Fu, and Mohammad Shokrolah Shirazi. "Integration with 3D visualization and IoT-based sensors for real-time structural health monitoring." *Sensors* 21.21 (2021): 6988.
- [2] Kouroussis, Georges, David P. Connolly, and Olivier Verlinden. "Railway-induced ground vibrations—a review of vehicle effects." *International Journal of Rail Transportation* 2.2 (2014): 69-110.
- [3] Lawal, Omobolaji, *et al.* "Development and validation of a framework for smart wireless strain and acceleration sensing." *Sensors* 22.5 (2022): 1998.
- [4] Das, Swagato, and Purnachandra Saha. "A review of some advanced sensors used for health diagnosis of civil engineering structures." *Measurement* 129 (2018): 68-90.
- [5] Kaewunruen, Sakdirat, and Alex M. Remennikov. "Field trials for dynamic characteristics of railway track and its components using impact excitation technique." *Ndt & E International* 40.7 (2007): 510-519.
- [6] Fernandes, Fernando dos Santos Alves, *et al.* "Desenvolvimento de Nó Sensor para Monitoramento de Vibrações com IoT." *Simpósio Brasileiro de Automação Inteligente-SBAI*. Vol. 1. No. 1. 2021.
- [7] Sofi, A., *et al.* "Structural health monitoring using wireless smart sensor network—An overview." *Mechanical Systems and Signal Processing* 163 (2022): 108113.
- [8] Abadade, Youssef, *et al.* "A comprehensive survey on tinyml." *IEEE Access* 11 (2023): 96892-96922.
- [9] Stenström, C., J. Lindqvist, and F. Andersson. "Condition based maintenance using MEMS accelerometers: For faster development of IoT in railways." *Infra Sweden* 2030 (2017).
- [10] Milne, David, *et al.* "Proving MEMS technologies for smarter railway infrastructure." *Procedia engineering* 143 (2016): 1077-1084.
- [11] Milne, David, *et al.* "Monitoring and repair of isolated trackbed defects on a ballasted railway." *Transportation Geotechnics* 17 (2018): 61-68.
- [12] Kaewunruen, Sakdirat, and Alexander Remennikov. "Application of vibration measurements and finite element model updating for structural health monitoring of ballasted railtrack sleepers with voids and pockets." *Mechanical Vibrations: Measurement, Effects and Control* (2009).
- [13] Zanelli, Federico, *et al.* "Development and field validation of wireless sensors for railway bridge modal identification." *Applied Sciences* 13.6 (2023): 3620.
- [14] Grimmelsman, Kirk A., and Navid Zolghadri. "Experimental evaluation of low-cost accelerometers for dynamic characterization of bridges." *Dynamics of Civil Structures, Volume 2: Proceedings of the 37th IMAC, A Conference and Exposition on Structural Dynamics 2019*. Cham: Springer International Publishing, 2019.
- [15] Angeline, Lorita, *et al.* "AloT-Driven Machine Learning for Anomaly Detection in Structural Health Monitoring." *2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*. IEEE, 2024.
- [16] Katsidimas, Ioannis, *et al.* "An impact localization solution using embedded intelligence—Methodology and experimental verification via a resource-constrained IoT device." *Sensors* 23.2 (2023): 896.
- [17] Zhang, Yuxuan, *et al.* "Leveraging acoustic emission and machine learning for concrete materials damage classification on embedded devices." *IEEE Transactions on Instrumentation and Measurement* 72 (2023): 1-8.
- [18] Zhao, Xijia, and Peng Wang. "A deployable edge computing solution for machine condition monitoring." *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2024.
- [19] Pedregosa, Fabian, *et al.* "Scikit-learn: Machine learning in Python." *The Journal of machine Learning research* 12 (2011): 2825-2830.

- [20] Chaudhary, Archana, Savita Kolhe, and Raj Kamal. "An improved random forest classifier for multi-class classification." *Information Processing in Agriculture* 3.4 (2016): 215-222.
- [21] Mariniello, Giulio, *et al.* "Structural damage detection and localization using decision tree ensemble and vibration data." *Computer - Aided Civil and Infrastructure Engineering* 36.9 (2021): 1129-1149.
- [22] Farias, Thiago S., Rafael G. Rossi, and Mato Grosso do Sul–MS–Brazil. *Estudo Comparativo de Arquiteturas de Redes Neurais em Análise de Sentimentos*. Diss. Dissertação (Mestrado)—Universidade Federal do Mato Grosso do Sul, 2021. Acesso em: 09 dez. 2024.
- [23] Yadav, Sanjay, and Sanyam Shukla. "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification." *2016 IEEE 6th International conference on advanced computing (IACC)*. IEEE, 2016.