

Aplicação de Machine Learning para detecção de falhas e obtenção de parâmetros estatísticos em mancais de elementos rolantes

Palavras-Chave: MANCAL, MACHINE LEARNING, ATRIBUTOS, FALHAS.

Autores(as):

PEDRO FERRI PEREIRA, LAMAR -FEM - UNICAMP

Prof^(a). Dr^(a). KATIA LUCCHESI CAVALCA DEDINI (orientador(a)), LAMAR -FEM - UNICAMP

INTRODUÇÃO:

Os mancais de elementos rolantes são componentes essenciais em aplicações industriais, pois fornecem suporte e facilitam o movimento de eixos rotativos. No entanto, esses componentes são suscetíveis a diversas falhas, como desgaste, fadiga superficial (pitting e spalling) e lubrificação inadequada, que podem comprometer sua funcionalidade e causar danos significativos aos sistemas em que são utilizados (NORTON, 2014). Essas falhas não apenas alteram a vibração do sistema, mas também reduzem a vida útil dos elementos mecânicos, exigindo uma abordagem robusta para a interpretação dos parâmetros de comportamento do sistema e dos componentes mecânicos. Nesse contexto, a análise de dados coletados torna-se crucial para a diagnose e prevenção de falhas, bem como para a otimização do tempo de uso e funcionalidade dos componentes.

A implementação de ferramentas eficientes e programáveis, como o Machine Learning, permite análises superiores às limitadas pela programação clássica, utilizando métodos matemáticos avançados e uma vasta quantidade de dados. Este projeto visa continuar o trabalho de graduação de Luís Otávio Garavaso, que abordou outros tipos de falhas em elementos rotativos, e aplicar o algoritmo LightGBM de classificação para a análise de falhas e obtenção de parâmetros estatísticos. Os resultados e análises serão compartilhados com o Laboratório de Máquinas Rotativas (LAMAR, FEM-UNICAMP) para apoiar as pesquisas do convênio Unicamp Petrobrás “Ferramentas de Diagnóstico para Vorecon – Fase 2 – Monitoramento Online”. Ao focar no estudo de falhas precoces em mancais, esta pesquisa acadêmica busca contribuir significativamente para o avanço do conhecimento e das práticas de manutenção preditiva em sistemas industriais.

METODOLOGIA:

A pesquisa inicial para selecionar um algoritmo de gradient boosting ideal foi fundamental para o projeto, já que este algoritmo de Machine Learning é a base para a predição de falhas dos dados analisados. Embora o algoritmo LightGBM da Microsoft tenha sido considerado, o Hist Gradient Boosting do Scikit-

Learn foi escolhido devido à sua vasta biblioteca de módulos e manuais disponíveis, apresentando uma construção sequencial e desempenho similar ao LightGBM.

Após selecionar e compreender o algoritmo preditivo, foi necessário definir os parâmetros e conjuntos de dados adequados. A extração de características de assinatura do sinal de vibração, conhecidas como "Features Estatísticas" (imagem 1), é crucial para representar aspectos relevantes dos dados de forma compacta e informativa. As features, derivadas de dados brutos ou transformações destes, são essenciais para melhorar a precisão e eficiência dos modelos de Machine Learning, permitindo a detecção de padrões importantes.

No domínio do tempo, características como médias e variâncias são extraídas diretamente dos dados ao longo do tempo. No domínio da frequência, técnicas como a Transformada de Fourier ajudam a revelar padrões não visíveis no tempo. A engenharia de features, por sua vez, envolve criar novas características a partir das já existentes, utilizando conhecimento específico do domínio. No projeto, foram utilizadas features no domínio do tempo e frequência, conforme referenciado em estudos anteriores, para garantir uma base consolidada na literatura.

Os dados foram processados e divididos em conjuntos de treino e teste, com a implementação inicial do algoritmo Hist Gradient Boosting. A validação cruzada foi aplicada para evitar vieses e maximizar o desempenho do algoritmo. O Grid Search (imagem 2) foi usado para encontrar a melhor combinação de hiper parâmetros, o que ajudou a otimizar o modelo. No entanto, os resultados iniciais mostraram precisão de treino e teste abaixo de 70%, indicando a necessidade de reestruturar o modelo.

A reestruturação envolveu a importação de bibliotecas essenciais e a organização dos dados em um Data Frame. A validação cruzada estratificada com agrupamento foi utilizada para garantir a integridade dos dados de treino e teste. Um grid de hiper parâmetros foi definido para o classificador Random Forest, com a busca pelos melhores parâmetros realizada através do GridSearchCV. Esse processo ajudou a identificar as features mais importantes, melhorando a robustez do modelo.

A análise de correlação (imagem 3) foi usada para identificar redundâncias entre as features. A matriz de correlação e o mapa de calor ajudaram a visualizar a intensidade e direção das relações entre diferentes features, permitindo a eliminação de redundâncias e a seleção das características mais representativas. Com base nessa análise, foram selecionadas 3 features de cada domínio, evitando sobreposição de informações.

Para determinar a relevância das features, foi utilizada a importância das características fornecida por modelos de aprendizado de máquina. A função `model.feature_importances_` do Random Forest ajudou a identificar as features mais importantes, que foram "Root Mean Square", "Valor de Pico", "PSD" e "coeficiente de energia de detalhe de wavelet 3". A combinação de features que apresentou os melhores resultados foi selecionada para o modelo final.

A otimização do modelo foi realizada utilizando a Otimização Bayesiana, que ajusta os hiper parâmetros para melhorar a performance. A aplicação prática foi feita com a biblioteca `scikit-optimize`, definindo um espaço de hiper parâmetros e utilizando um processo gaussiano como modelo de substituto. A função de aquisição Expected Improvement ajudou a selecionar os hiper parâmetros a serem avaliados, iterativamente melhorando o modelo até alcançar um critério de parada.

Após a implementação da Otimização Bayesiana (imagem 4), a acurácia de treino e teste do modelo melhorou significativamente, passando de 76,97% para 83% no treino e de 69,59% para 74% no teste. Isso garantiu um modelo robusto, bem ajustado aos dados e com features que representavam adequadamente as características dos sinais, melhorando a capacidade de predição e a confiabilidade dos resultados.

RESULTADOS E DISCUSSÃO:

O projeto focou na aplicação de técnicas avançadas de machine learning, especificamente o algoritmo Hist Gradient Boosting, para detecção de falhas em mancais de rolamento. Inicialmente, os dados de vibração foram processados para extrair características relevantes, chamadas features. O modelo foi treinado e avaliado usando validação cruzada, mas os resultados preliminares indicaram problemas de underfitting e precisão insatisfatória.

Para resolver esses problemas, o modelo foi reestruturado com uma nova seleção de features e aplicação de técnicas de otimização, incluindo a Otimização Bayesiana. Essa técnica ajustou os hiper parâmetros do modelo de maneira eficiente, resultando em uma melhora significativa na precisão das predições. A acurácia média do modelo de treino aumentou para 82,45%, enquanto a do modelo de teste foi para 74,32%.

O projeto concluiu com sucesso a construção de um modelo robusto e eficaz, destacando a importância da seleção adequada de features e da otimização de hiper parâmetros. Além de cumprir os objetivos iniciais, o trabalho explorou técnicas avançadas, superando as expectativas de planejamento.

ANEXOS

FEATURES NO DOMÍNIO DO TEMPO	
1. RMS $RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$	7. ASSIMETRIA $Sk = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2\right)^{3/2}}$
2. VARIÂNCIA $var = \sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$	8. ENTROPIA DE SHANNON $Ent = - \sum_{i=1}^n x_i^2 \log(x_i^2)$
3. VALOR DE PICO $PvT = \max(x_i)$	9. FATOR DE FORMA $SF = \frac{RMS}{\frac{1}{n} \sum_{i=1}^n x_i }$
4. FATOR DE CRISTA $CF = \frac{PvT}{RMS}$	10. FATOR DE IMPULSO $IF = \frac{PvT}{\frac{1}{n} \sum_{i=1}^n x_i }$
5. CURTOSE $Kurt = \frac{\sum_{i=1}^n (x_i - \mu)^4}{n \times var^2} - 3$	11. FATOR DE DESOBSTRUÇÃO $Clf = \frac{PvT}{\frac{1}{n} \sum_{i=1}^n (\sqrt{ x_i })^2}$
6. VALOR PICO A PICO $PP = \max(x_i) - \min(x_i)$	12. INTEGRAL DE LINHA $LI = \sum_{i=0}^n x_{i+1} - x_i $
FEATURES NO DOMÍNIO DA FREQUÊNCIA	
13. Pvf $PvF = \max(r_k)$	15. PSD $PSD = \left \sum 2 \times FFT(x(t)) \times 1/np \right ^2$
14. En $En = \sum_{k=1}^N r_k$	

Imagem 1: Atributos Estatísticos

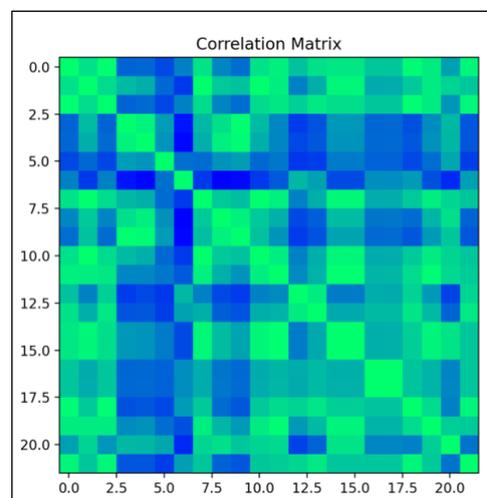


Imagem 3: Matriz de Correlação

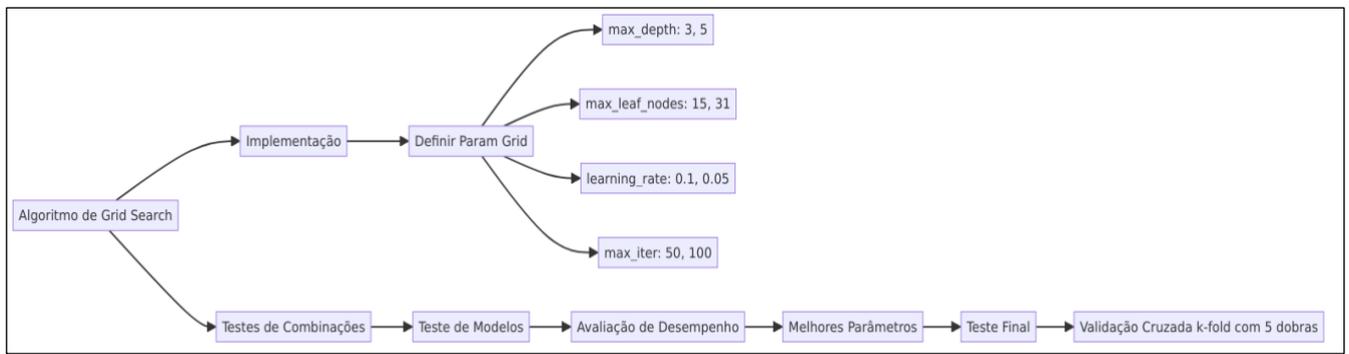


Imagem 2: Algoritmo de Grid Search

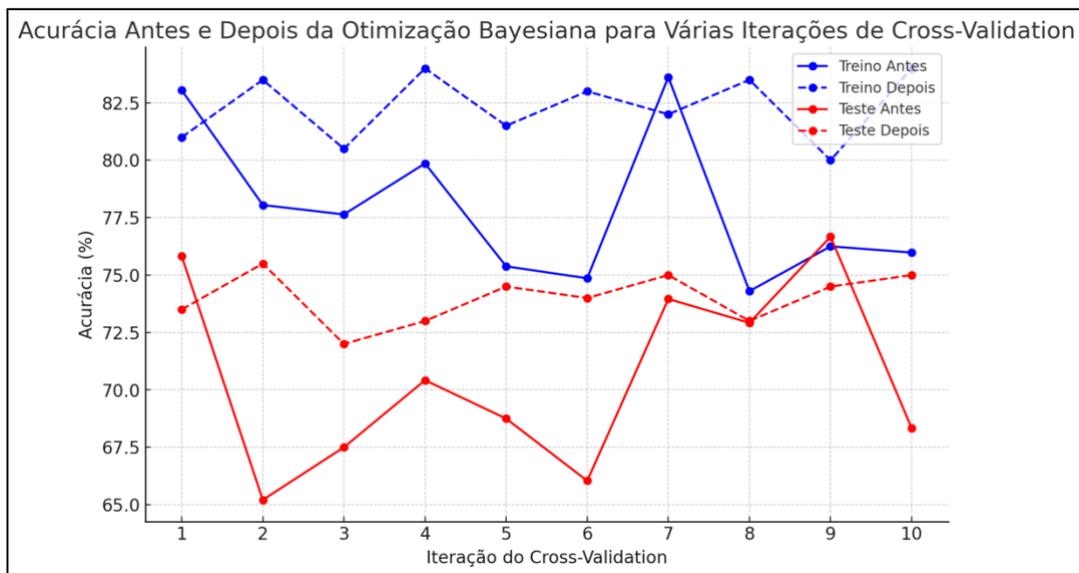


Imagem 4: Resultados Treino e teste antes e após Otimização Bayesiana para cada iteração

BIBLIOGRAFIA

- Kimotho, J.K., & Sextro, W. 2014. "An approach for feature extraction and selection from non-trending data for machinery prognosis." University of Paderborn, Pohlweg 47-49, 33098 Paderborn, Alemanha.
- CARRER, L.; GARAVASO, L. O. ; DANIEL, G. B. ; MACHADO, T. H. ; CAVALCA, K. L. . Study on localized defect model and identification in Elastohydrodynamic lubricated ball bearings using Support Vector Machine. In: 15th SIRM European Conference on Rotordynamics, 2023, Darmstadt. 15th SIRM European Conference on Rotordynamics, 2023
- SCIKIT-LEARN. scikit-learn: machine learning in Python. Disponível em: <<https://scikit-learn.org/stable/>>.
- Konstruktions- und Antriebstechnik (KAt) - Data Sets and Download (Universität Paderborn). Disponível em: <<https://mb.uni-paderborn.de/kat/forschung/kat-datacenter/bearing-datacenter/data-sets-and-download>>.
- Rasmussen, C. E., & Williams, C. K. I. (2006). "Gaussian Processes for Machine Learning." The MIT Press.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, 12, 2825-2830.