

# Fault detection for rotating machinery based on vibration data using machine learning

Lucas de T. Barreto<sup>1</sup>, Rodrigo K. Rosa<sup>2</sup>, Danilo Silva<sup>2</sup>, Danilo Braga<sup>3</sup>

<sup>1</sup> Departamento de Engenharia Mecânica  
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC - Brazil

<sup>2</sup> Departamento de Engenharia Elétrica e Eletrônica  
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC - Brazil

<sup>3</sup> Dynamox – Florianópolis, SC – Brazil

{lucasdetoledo4, rodrigokrosa}@gmail.com, danilo.silva@ufsc.br,  
danilo@dynamox.net

**Abstract.** *This paper addresses the detection of mechanical faults in rotating machinery using machine learning techniques. Vibration signals were collected from machines in operation in the industry, and features of these signals were extracted, ranging from harmonics of a motor's rotation speed to specialized features typically considered by vibration analysts. After data cleaning, preprocessing, and the construction of the training pipeline and hyperparameter optimization, machine learning models such as logistic regression, support vector machines, random forests, neural networks, and gradient boosting (XGBoost) were explored. The results showed that the XGBoost model performed the best, achieving an ROC AUC metric of 91%.*

**Resumo.** *Este artigo aborda a detecção de falhas mecânicas em máquinas rotativas usando técnicas de aprendizado de máquina. Foram coletados sinais de vibração das máquinas em operação na indústria e extraídas features desses sinais, desde harmônicas da velocidade de rotação do motor até features especializadas tipicamente consideradas por analistas de vibração. Após limpeza e pré-processamento dos dados, foi construída uma pipeline de treinamento e otimização de hiperparâmetros. Foram explorados modelos como regressão logística, máquinas de vetores de suporte, florestas aleatórias, redes neurais e gradient boosting (XGBoost). Os resultados mostraram que o modelo XGBoost obteve o melhor desempenho, alcançando uma métrica de ROC AUC de 91%.*

## 1. Introdução

A ocorrência de falhas em máquinas industriais representa um obstáculo para a operação eficiente e ininterrupta da produção, tornando crucial a detecção precoce e a aplicação da manutenção preditiva. Essas medidas são essenciais para prevenir danos graves, diminuir os custos operacionais e minimizar o tempo de parada.

O transportador de correia desempenha um papel fundamental nos sistemas de transporte da indústria. Trata-se de um equipamento mecânico utilizado para o movimento contínuo de materiais ao longo de curtas ou longas distâncias. Seus componentes principais incluem o motor, roletes, tambores, correia, chute de carregamento e polias [Zimroz and Król 2009].

Na indústria de mineração, o monitoramento de condição apresenta desafios adicionais em comparação a outros setores. Isso se deve ao ambiente severo e à dispersão dos ativos, que muitas vezes abrangem extensões consideráveis, chegando a vários quilômetros [Zimroz and Król 2009]. Essa complexidade torna arriscado para os técnicos realizarem o monitoramento em campo, tornando as técnicas automatizadas de detecção e diagnóstico de falhas, com base em dados coletados por sensores de vibração, uma ferramenta promissora [Alharbi et al. 2023]. O objetivo é assegurar a eficiência operacional e a confiabilidade das máquinas.

A literatura científica aborda o problema de detecção e diagnóstico de falhas há décadas, abordando diferentes técnicas, desde métodos tradicionais baseados em análise de dados de vibração como a análise de envelope [Randall 2021], métodos baseados em dados utilizando técnicas de aprendizado de máquina clássicos baseados em extração de atributos [Gao et al. 2015, Ahmed and Nandi 2020] e até mais recentemente, o uso de redes neurais profundas [Janssens et al. 2016, Lei et al. 2020].

O conjunto de dados utilizado neste projeto consiste de dados de vibração e relatórios com laudos de analistas fornecidos pela empresa parceira. Os dados de vibração são adquiridos por *dataloggers* sem fio (com acelerômetro e sensor de temperatura embarcado) produzidos pela empresa, os quais encontram aplicação em diversos setores industriais, permitindo o monitoramento das condições dos ativos. Um dos principais ativos monitorados pela empresa são os transportadores de correia e portanto, buscou-se resolver o problema de detecção e classificação de falhas para este tipo de ativo.

Este artigo tem como objetivo desenvolver um modelo de aprendizado de máquina a partir de *features* de tempo e frequência extraídas dos sinais de vibração. Para atingir esse objetivo, são realizadas as seguintes etapas: (1) preparação dos dados, que envolve a limpeza e organização dos dados de vibração e dos rótulos obtidos por meio dos relatórios; (2) determinação do desempenho ótimo que pode ser alcançado por técnicas de aprendizado de máquina a partir de uma extração de *features* simples, o que por sua vez envolve a estruturação de uma metodologia de otimização de hiperparâmetros para cada técnica considerada; e (3) extração de *features* mais especializadas, seguida de nova otimização de hiperparâmetros e treinamento dos modelos.

No artigo, serão descritos os passos adotados para a preparação dos dados, treinamento dos modelos e os resultados obtidos com os seguintes modelos de aprendizado de máquina: Regressão Logística, Máquina de Vetores de Suporte (SVM), Floresta Aleatória (*Random Forest*), XGBoost (*Extreme Gradient Boosting*) e Perceptron Multicamadas (*MLP*).

## 2. Metodologia

### 2.1. Conjunto de dados

O conjunto de dados disponibilizado pela empresa Dynamox é composto de duas fontes principais de dados: sinais de vibração e relatórios de diagnóstico feitos por analistas.

O conjunto de dados abrange uma extensa quantidade de medições de sinais de vibração, realizadas em diferentes motores de transportadores de correia, operando em condições reais na indústria de mineração. Foram coletados dados ao longo de um período de dois anos, com medições diárias. Paralelamente, foram elaborados relatórios por ana-

listas de vibração da empresa, os quais incluem informações sobre os modos de falhas identificados nas medições, a data de criação do relatório, seu diagnóstico, dentre outros detalhes referentes a cada componente monitorado. Além do mais, tais relatórios foram elaborados em intervalos regulares de 15 dias para cada ponto de medição.

## **2.2. Pré-processamento dos dados**

O pré-processamento dos sinais de vibração envolve a aplicação de técnicas para limpar os dados que podem afetar negativamente a tarefa de aprendizado, como a falta de dados ou a presença de dados inconsistentes e incoerentes. Neste último caso incluem-se sinais de vibração provenientes de máquinas paradas e sinais de vibração ceifados (*clipping*).

### **2.2.1. Limpeza dos sinais de vibração de máquinas paradas**

A detecção de sinais de vibração em máquinas inativas ocorre devido a coleta dos dados durante o processo de desligamento para manutenção ou inspeção de ativos. Nesse contexto, a presença contínua de um sensor IoT, responsável por monitorar o equipamento, realiza a captura de tais sinais mesmo quando as máquinas estão paradas, embora esses sinais muitas vezes não forneçam informações necessariamente relevantes, devendo ser excluídos da análise.

Conseqüentemente, a limpeza dos sinais de vibração de máquinas paradas foi realizada utilizando duas abordagens: a Planicidade Espectral, medida pela entropia de Wiener, e o Valor Eficaz (*RMS*).

De acordo com [Boashash 2015], a entropia de Wiener é calculada com base no espectro de densidade de potência do sinal no domínio da frequência. Essa medida estima a uniformidade da distribuição de energia do sinal em diferentes componentes de frequência e situa-se numa faixa entre 0 e 1. Uma alta entropia no domínio da frequência implica em uma maior uniformidade na distribuição de energia do sinal, enquanto uma baixa entropia indica menor uniformidade. Neste estudo, foi estabelecido um limiar de 0,85 para caracterizar os sinais de vibração de máquinas paradas. Todos os sinais de vibração com um limiar igual ou superior a esse valor foram considerados indicativos de máquinas paradas e, portanto, descartados.

Em conjunto com a entropia de Wiener, para caracterizar os sinais de vibração de máquinas paradas, também foi adotado um limiar baseado no Valor Eficaz (*RMS*). Experimentalmente, foi definido um limiar de referência, em que qualquer sinal de vibração com um valor *RMS* igual ou inferior a esse limiar foi considerado indicativo de máquina parada. Neste estudo, o limiar escolhido foi de 0,1 mm/s<sup>2</sup> e todos os sinais de vibração que possuíam um valor *RMS* igual ou inferior a esse limiar foram descartados, juntamente com os sinais identificados pela entropia de Wiener.

Ambos os limiares escolhidos foram recomendados por analistas de vibração e validados experimentalmente com os dados disponíveis.

## 2.2.2. Limpeza dos sinais de vibração com *clipping*

A ocorrência da condição de *clipping* em sinais de vibração é um fenômeno no qual os valores do sinal ultrapassam a faixa superior ou inferior de medição do sensor. Tal fenômeno pode ser causado por inúmeros fatores, sendo os principais deles: ressonâncias mecânicas e saturação do sensor. Portanto, a condição de *clipping* proporciona uma distorção no sinal e, conseqüentemente, perda de precisão na representação deste. Por esse motivo, optamos por excluir tais sinais do conjunto de dados.

Sinais de vibração podem sofrer de dois tipos de *clipping*: *clipping* rígido e *clipping* suave. O rígido ocorre quando os valores excedentes do sinal são ceifados abruptamente até o limite de *clipping*, enquanto o suave reduz gradualmente os valores excedentes em direção ao limite [Laguna and Lerch 2016].

A identificação e descarte dos sinais com *clipping* suave ocorre por meio da análise dos histogramas de cada sinal. Ao verificar se um sinal ultrapassa os limites de uma faixa dinâmica estabelecida a partir de 70% dos valores (limiar escolhido experimentalmente), é possível detectar a presença de *clipping* suave, como indicado na Figura 1. Essa detecção é realizada através do cálculo dos quantis do sinal sem a componente de corrente contínua e da comparação desses quantis com os valores mínimo e máximo do sinal. O sinal é considerado com *clipping* suave e pode ser descartado ou tratado adequadamente, caso a razão entre os quantis e os valores extremos seja maior que 0,7. Caso contrário, o sinal é considerado sem *clipping* suave e pode ser utilizado para análises posteriores.

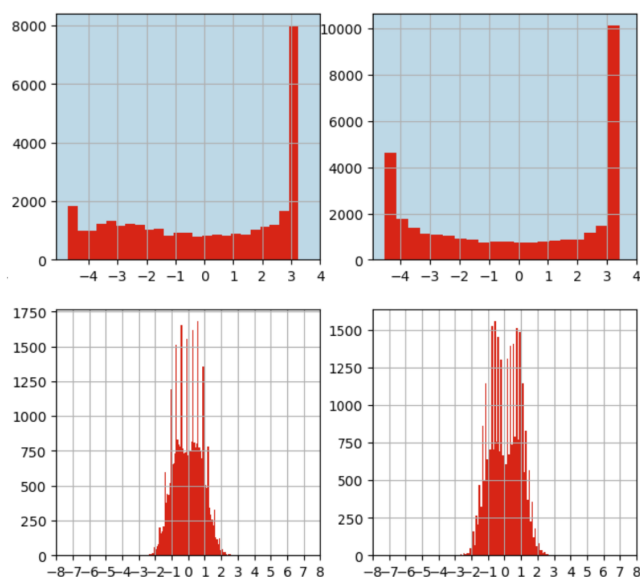


Figura 1. Sinais de vibração com *clipping* suave (fundo azul) versus sinais sem *clipping* (fundo branco).

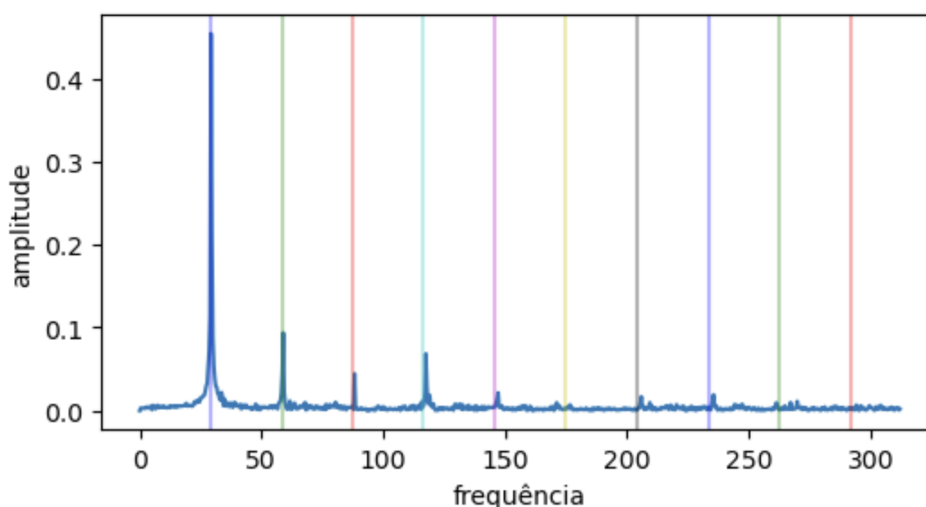
## 2.3. Extração de features

### 2.3.1. Features baseline

Visando estabelecer um *baseline* para o processo de extração de *features*, escolheu-se *features* derivadas das 10 primeiras harmônicas da velocidade de rotação dos sinais de

vibração, as quais podem ser facilmente obtidas e possuem características específicas para os modos de falha. Segundo [Randall 2021], falhas como desalinhamento e desbalanceamento manifestam-se nos primeiros harmônicos (múltiplos) da velocidade do eixo.

Estas harmônicas representam as bandas espectrais de maior energia do espectro (calculado pela transformada discreta de Fourier) (Figura 2) e densidade espectral de potência (*power spectral density, PSD*) estimada pelo método de Welch [Welch 1967] e possuem características distintas para diferentes modos de falha. As características extraídas incluíram a amplitude máxima para cada harmônica para o espectro e a PSD dos sinais.



**Figura 2. Amplitudes máximas das 10 primeiras harmônicas do espectro para um sinal com desbalanceamento.**

Detalhadamente, a *pipeline* de extração das *features* incluiu as seguintes etapas: (1) cálculo do espectro e PSD do sinal, utilizando a transformada discreta de Fourier e o método de Welch, respectivamente; (2) cálculo da primeira harmônica e, em seguida, das dez primeiras harmônicas, utilizando o metadado do número de rotações por minuto das máquinas referentes aos sinais de vibração; (3) definição de um intervalo de frequências para cada harmônico, determinado pelos índices correspondentes às posições dos valores de frequência, obtidos pela divisão da frequência das harmônicas pela resolução espectral (em que o limite inferior/superior do intervalo é o valor inteiro arredondado para baixo/cima desse índice, ambos acrescidos de um número de pontos especificado); (4) extração do valor de amplitude máxima encontrado no intervalo pré-definido.

Essas 20 amplitudes máximas (*features*), juntamente com as falhas (*targets*) identificadas nos relatórios, foram utilizadas para compor os conjuntos de treinamento e teste do modelo.

### 2.3.2. *Features* especializadas

Com o objetivo de obter características que aprimorem a descrição dos modos de falha e da condição normal, uma abordagem de *features* especializados foi adotada, na qual

incorporou-se uma diversa variedade de *features* as *features baseline*, fornecidas e validadas experimentalmente por analistas de vibração.

O conjunto de *features* especializadas pode ser categorizada como *features* no domínio do tempo e frequência, dividindo em *features* de aceleração, velocidade, deslocamento e envelope. As métricas estatísticas no domínio do tempo utilizadas foram: raiz do valor quadrático médio (RMS), pico a pico, curtose, fator de crista e fator de crista "plus" (média ponderada entre RMS, amplitude máxima do sinal e fator de crista, com pesos, 3, 3 e 4, respectivamente). Dentre as métricas estatísticas no domínio da frequência, temos: RMS, RMS carpete (RMS espectral por meio da média móvel de múltiplas janelas no domínio da frequência), pico a pico, curtose, fator de crista e fator de crista "plus". Por fim, o conjunto de dados foi estabelecido com um total 121 *features*.

Todas as *features*, tanto no tempo quanto na frequência, foram calculadas globalmente e para diferentes faixas de frequência, a depender da métrica específica. Os intervalos de frequência e harmônicos utilizados para a extração das *features* especializadas encontram-se na Tabela 1.

## 2.4. Divisão dos conjuntos de treinamento e teste

Para a criação do conjunto de dados de treinamento e teste, o processo envolveu a associação de cada relatório de diagnóstico ao seu respectivo sinal de vibração específico. A seleção desses sinais foi feita com base na data mais próxima de cada relatório, em um intervalo de 7 dias anteriores a este.

A fim de evitar vazamento de dados, ou seja, que o modelo apenas memorize a classificação de sinais muito semelhantes entre si, por estarem temporalmente próximos, a divisão entre os conjuntos de treinamento e teste foi realizada de forma temporal, utilizando a coluna "rodada", a qual denota períodos temporais quinzenais de coletas (Figura 3).

O conjunto de treinamento foi composto por 72,8% dos dados, enquanto o conjunto de teste foi composto pelo restante. Essa abordagem é especialmente relevante, uma vez que cada acelerômetro coleta periodicamente informações de uma mesma máquina, que pode evoluir em seus modos de falha ao longo do tempo.

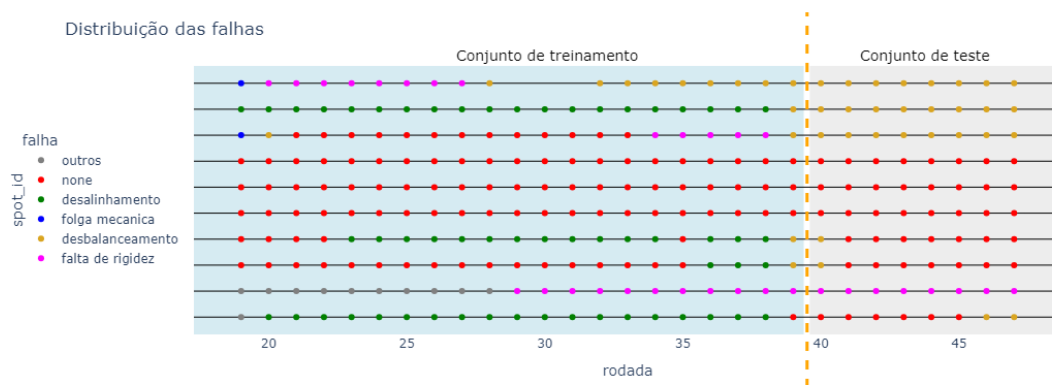


Figura 3. Divisão do conjunto de treinamento (azul) e teste (cinza).

**Tabela 1. Features especializadas extraídas.**

	Métricas	Faixas de frequência (Hz) / Harmônicas (x)
Aceleração	RMS Carpete	[100, 1000], [1000, 6395]
	RMS	Global, [5, 20], [20, 200], [5, 200], [200, 500], [500, 1000], [5, 1000], [1000, 3000], [3000, 6395]
	Pico espectral	Global, 0.5x, 1x, 1.5x, 2x, 2.5x, 3x, 4x, 5x, 6x, 7x, 8x, 9x, 10x
	Pico a pico	
	Curtose	
	Fator de crista Fator de crista plus	Global, [1000, 6395]
Velocidade	RMS	Global, [5, 200], [200-500], [500, 1000], [5, 50], [50, 110], [110, 160], [160, 200], [5, 1000]
	Pico a pico	Global
	Pico espectral	Global, 0.5x, 1x, 1.5x, 2x, 2.5x, 3x, 4x, 5x, 6x, 7x, 8x, 9x, 10x
Deslocamento	RMS	Global, [5, 200], [200, 500], [500, 1000], [5, 50], [50, 110], [110, 160], [160, 200], [5, 1000]
	Pico a pico	Global
Envelope	RMS Carpete	[800, 2000], [2000, 3000], [2000, 4000], [500, 5000], [1000, 5000], [2000, 6200]
	RMS	
	Pico a pico	[300, 795], [300, 1500], [800, 2000],
	Curtose	[2000, 3000], [2000, 4000], [500, 5000],
	Fator de crista	[1000, 5000], [2000, 6200]
	Fator de crista plus	

## 2.5. Metodologia de otimização de hiperparâmetros

Para realizar a busca de hiperparâmetros de forma eficiente, é crucial dividir o conjunto de dados em conjuntos de treinamento, validação e teste. Nesse estudo, adotamos uma estratégia de busca para a otimização de hiperparâmetros, que utiliza a validação cruzada estratificada *K-Fold* com repetições. Esta abordagem consiste em, primeiramente, dividir o conjunto de validação em  $K = 5$  *folds* estratificados, mantendo a distribuição de defeitos e normalidade dos dados. Em seguida, 4 dos  $K = 5$  *folds* são utilizados para o treinamento do modelo, enquanto o quinto *fold* é utilizado para avaliação. Este processo é repetido para cada *fold* para obter uma otimização mais robusta, na qual a acurácia final é definida pela média das acurácias dos  $K$  treinamentos, reduzindo a influência da aleatoriedade na divisão dos dados em cada *fold* [Bishop 2006]. Finalmente, o processo de validação cruzada completo é repetido por 5 vezes com diferentes randomizações, resul-

tando em uma robustez ainda maior para a acurácia de validação.

No presente estudo, foram exploradas diferentes combinações de hiperparâmetros para cada *fold* do conjunto de validação cruzada. Para encontrar a melhor combinação, foi empregado uma estratégia de Busca Aleatória (*Random Search*) com 100 iterações, para as quais buscou-se otimizar a métrica ROC AUC (*Receiver Operating Characteristic Area Under the Curve*), cuja definição está descrita na seção 2.7.

## 2.6. Modelos de aprendizado de máquina utilizados

Neste estudo, utilizando a biblioteca scikit-learn, foram treinados os seguintes modelos de aprendizado de máquina: Regressão Logística, SVC (*Support Vector Classifier*), Floresta Aleatória, MLP e XGBoost.

A Regressão Logística é um algoritmo simples e interpretável de classificação linear que calcula a probabilidade de uma instância pertencer a uma classe por meio do uso da função logística [Géron 2019]. No presente trabalho, para o modelo de regressão logística, os parâmetros otimizados foram: *penalty* e *C*.

O SVC é um algoritmo de aprendizado supervisionado comumente usado para tarefas de classificação. Tem como objetivo encontrar a melhor fronteira de decisão que separa maximamente as diferentes classes. Isto é alcançado por meio do mapeamento dos dados em um espaço de dimensão mais alta, usando uma função de kernel, como por exemplo o kernel gaussiano (*radial basis function*, RBF) [Cervantes et al. 2020]. No presente trabalho, foi utilizado o *kernel RBF* e os parâmetros otimizados, para o modelo de SVC, foram: *gamma* e *C*.

A Floresta Aleatória é um algoritmo de aprendizado por *ensemble* que combina várias árvores de decisão para realizar previsões. Durante a inferência, as árvores fornecem probabilidades para cada uma das classes e a previsão final é baseada numa votação suave (*soft voting*), a qual corresponde a calcular, para cada uma das classes, a média das probabilidades preditas. Esse processo proporciona robustez ao modelo, permitindo lidar com conjuntos de dados de maior dimensionalidade e analisar a importância das variáveis [Géron 2019]. No presente trabalho, para o modelo de floresta aleatória, os parâmetros otimizados foram: *n\_estimators*, *criterion*, *class\_weight*, *oob\_score*, *bootstrap*, *min\_weight\_fraction\_leaf*, *min\_impurity\_decrease*, *min\_samples\_split*, *min\_samples\_leaf*, *max\_samples* e *ccp\_alpha*.

O MLP é um algoritmo de aprendizado de máquina que consiste em uma rede neural artificial (ANN), com uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Esta arquitetura é capaz de aprender representações complexas dos dados e realizar tarefas de classificação e regressão [Géron 2019]. No presente trabalho, foi utilizado uma arquitetura MLP com somente 1 camada oculta, constituída por 512 neurônios e os parâmetros otimizados foram: *solver*, *learning\_rate* e *batch\_size*.

O XGBoost é um algoritmo avançado de aprendizado de máquina que utiliza uma combinação de modelos fracos, como árvores de decisão de baixa profundidade, para criar um modelo preditivo poderoso. O XGBoost emprega uma técnica de *boosting* que adiciona iterativamente novos modelos ao conjunto, focando nas instâncias que foram previamente classificadas incorretamente [Chen and Guestrin 2016]. No presente trabalho, para o modelo de XGBoost, os parâmetros otimizados foram: *booster*, *objective*, *class\_weight*,



*n\_estimators, eta, max\_depth, subsample, scale\_pos\_weight, colsample\_bytree* e *lambda*.

## 2.7. Métricas de avaliação

Como evidenciado na Tabela 2, o conjunto de dados exibe um desequilíbrio substancial entre as classes de defeitos. Nesse contexto, foram empregadas métricas que não são afetadas pela prevalência das classes, ou seja, métricas que não variam conforme a proporção das classes. Portanto, para avaliar a capacidade preditiva dos modelos, foram utilizadas duas métricas: ROC AUC e taxa de falsos positivos para uma dada taxa de verdadeiro positivo.

**Tabela 2. Distribuição das classes de falha.**

<b>Tipo de falha</b>	<b>Classes</b>	<b>N° de amostras</b>
<b>Falhas Individuais</b>	normalidade	2711
	falta de rigidez	487
	desalinhamento	396
	desbalanceamento	291
	outros	138
	folga mecânica	44
<b>Falhas Agrupadas</b>	normalidade	2711
	falhas	1356

Para a ROC AUC (*Area Under the Receiver Operating Characteristic Curve*) (1), tem-se que esta é uma métrica que avalia a habilidade de um modelo de classificação em distinguir corretamente entre classes positivas e negativas. A curva ROC é construída traçando a taxa de verdadeiros positivos (TPR) em relação à taxa de falsos positivos (FPR). A área sob essa curva (ROC AUC) é então calculada, conforme

$$\text{ROC AUC} = \int_0^1 \text{TPR}(thr) d\text{FPR}(thr) \quad (1)$$

em que um valor maior indica um melhor desempenho do modelo na classificação das classes.

## 2.8. Matriz de confusão

As matrizes de confusão constituem uma métrica amplamente utilizada para avaliar o desempenho de modelos de aprendizado de máquina. Estas consistem em tabelas que nos permitem analisar as predições do modelo, classificando-as como verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN). Tal conceito é ilustrado na Figura 4.

Além disso, a partir das métricas previamente estabelecidas (TP, TN, FP, FN), derivam-se outras métricas, sendo estas: taxa de verdadeiro positivo  $\text{TPR} = \text{TP}/(\text{TP} + \text{FN})$ , taxa de verdadeiro negativo  $\text{TNR} = \text{TN}/(\text{TN} + \text{FP})$ , taxa de falso positivo  $\text{FPR} = 1 - \text{TNR}$  e taxa de falso negativo  $\text{FNR} = 1 - \text{TPR}$ .

		Previsto	
		Não	Sim
Real	Não	Verdadeiro Negativo (TN)	Falso Positivo (FP)
	Sim	Falso Negativo (FN)	Verdadeiro Positivo (TP)

Figura 4. Matriz de confusão de um classificador binário.

### 3. Resultados

#### 3.1. Features baseline

Para o conjunto de *features baseline*, foram realizados experimentos utilizando os modelos de Regressão Logística, SVC com kernel RBF, Floresta Aleatória, MLP com 1 camada oculta com 512 neurônios e XGBoost, para os quais os resultados na validação cruzada e no conjunto de teste podem ser observados na Tabela 3.

Tabela 3. Resultados dos modelos para as *features baseline* na validação cruzada e conjunto de teste.

Modelo	Validação cruzada	Conjunto de teste	
	ROC AUC	ROC AUC	FPR (90% de TPR)
Regressão Logística	73,9%	67,0%	87,7%
SVC (kernel RBF)	82,9%	77,41%	70,1%
Floresta Aleatória	85,5%	81,2%	51,6%
MLP	83,8%	80,1%	54,1%
XGBoost	88,5%	85,5%	41,0%

A partir da Tabela 3, podemos observar que o modelo XGBoost apresentou o melhor desempenho em termos de ROC AUC na validação cruzada e conjunto de teste, alcançando uma performance de 85,5% em ROC AUC no conjunto de teste.

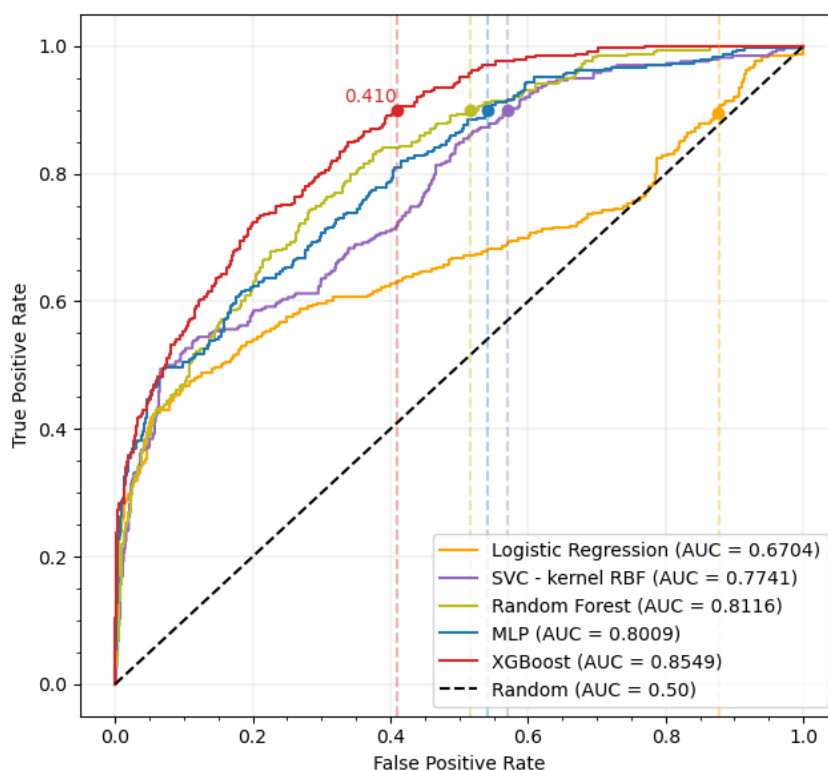
Com base nos resultados obtidos pelos modelos, é possível aprimorá-los por meio de um ponto de operação na curva ROC, que nos permite controlar os limiares de probabilidade de falha dos modelos através das TPR e FPR.

A curva ROC representa a TPR em relação à FPR em diferentes pontos de operação do classificador. Um modelo com uma curva ROC mais próxima do canto superior esquerdo do gráfico, indica um melhor desempenho, pois possui um TPR mais alto e um FPR mais baixo. A área sob a curva ROC (AUC) é frequentemente utilizada como uma métrica resumida para comparar diferentes modelos, em que um valor de ROC AUC de 0,5 representa um classificador aleatório, enquanto um valor de ROC AUC de 1 representa um classificador perfeito. Assim, a curva ROC permite avaliar o *trade-off* entre TPR e FPR, fornecendo informações sobre como o desempenho do modelo varia em diferentes pontos de operação.

Neste estudo, para o problema de detecção de falhas, é desejável maximizar a identificação da falha e minimizar a classificação incorreta de amostras com falha como

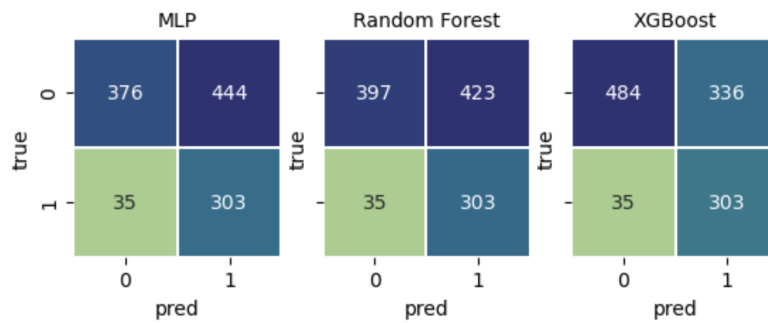
normalidade. Em outras palavras, buscamos aumentar a TPR e reduzir a FNR, mesmo que isso resulte em um aumento na ocorrência de falsos positivos, ou seja, de "alarmes falsos". Portanto, para alcançar este objetivo, optamos por trabalhar com uma TPR de 90%, o que corresponde a uma taxa de FNR de 10%.

Ao selecionar essa taxa, como mostrado na Figura 5, observamos um alto número de falsos positivos, como esperado. Sendo assim, arquiteturas mais simples, como Regressão Logística e SVC, sofreram mais com o *trade-off* entre TPR e FPR, alcançando FPRs elevadas de 87,7% e 70,1%, respectivamente. Contudo, para arquiteturas mais sofisticadas, as FPRs permaneceram-se entre 40-50%, em que o XGBoost alcançou melhores resultados, com 41% de FPR.



**Figura 5. Curvas ROC dos modelos treinados com *features baseline* com FPR destacados para uma TPR de 0,9 (90%).**

Além disso, os resultados apresentados anteriormente, podem ser observados através da matriz de confusão de cada classificador binário. Dado isso, as matrizes de confusão dos modelos que apresentaram melhor performance podem ser analisadas por meio da Figura 6.



**Figura 6. Matrizes de confusão dos modelos MLP, Floresta Aleatória e XGBoost, treinados com *features baseline* (TPR de 0.9).**

### 3.2. *Features* especializadas

Para o conjunto de *features* especializadas, também foram realizados experimentos utilizando os modelos de Regressão Logística, SVC com kernel RBF, Floresta Aleatória, MLP e XGBoost, para os quais os resultados na validação cruzada e no conjunto de teste podem ser observados na Tabela 4.

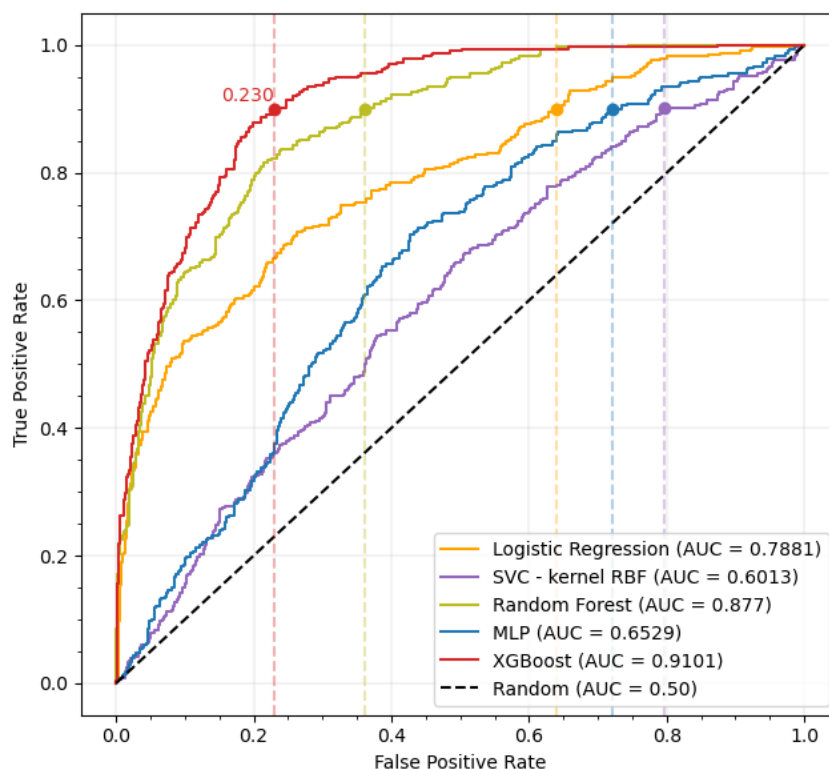
**Tabela 4. Resultados dos modelos para as *features* especializadas na validação cruzada e conjunto de teste.**

Modelo	Validação cruzada		Conjunto de teste
	ROC AUC	ROC AUC	FPR (90% de TPR)
<b>Regressão Logística</b>	85,7%	78,8%	64,0%
<b>SVC (kernel RBF)</b>	61,5%	60,1%	79,7%
<b>Floresta Aleatória</b>	92,7%	87,7%	36,1%
<b>MLP</b>	68,7%	65,3%	72,2%
<b>XGBoost</b>	95,1%	91,0%	23,0%

Com base nos resultados apresentados na Tabela 4, fica evidente que o modelo XGBoost continua a superar todos os outros modelos treinados em termos de ROC AUC, tanto durante a validação cruzada quanto no conjunto de teste. Destacadamente, o modelo XGBoost alcança 91% de ROC AUC no conjunto de teste, solidificando sua posição como o modelo de melhor desempenho.

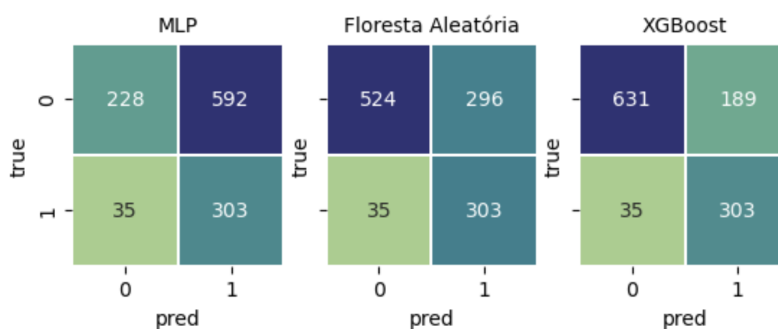
Além disso, uma observação importante surge ao compararmos os resultados obtidos com as *features baseline* e especializadas, conforme apresentados na Tabela 3 e Tabela 4, respectivamente. Tanto as arquiteturas de Floresta Aleatória quanto XGBoost apresentaram melhorias substanciais em suas métricas de desempenho, demonstrando que a adição de *features* mais especializadas desempenhou um papel fundamental na melhoria do desempenho destes modelos.

A partir da figura 7, observa-se que modelos como SVC e MLP sofreram mais com o *trade-off* TPR e FPR, alcançando FPRs elevadas de 79,7% e 72,2%, respectivamente. Contudo, para a arquitetura XGBoost, obtivemos resultados ótimos, alcançando somente 23% de FPR.



**Figura 7. Curvas ROC dos modelos treinados com *features* especializadas com FPR destacados para uma TPR de 0,9 (90%).**

Além disso, os resultados observados previamente por meio da Figura 7 podem ser reafirmados por meio das matrizes de confusão de cada classificador binário, encontradas na Figura 8.



**Figura 8. Matrizes de confusão dos modelos MLP, Floresta Aleatória e XGBoost, treinados com *features* especializadas (TPR de 0.9).**

Por fim, para o modelo que apresentou maior desempenho (XGBoost), foi analisado o ranqueamento das *features* utilizando a própria biblioteca do XGBoost, a fim de analisar quais *features* exerceram maior influência no desempenho alcançado. Tal ranqueamento apontou que as 3 *features* mais importantes para o modelo foram: RMS Global, RMS na faixa de frequência de 3000Hz a 6395Hz e Curtose Global, respectivamente. Além disso, mais informações sobre o ranqueamento das *features* podem ser encontradas na Figura 9.

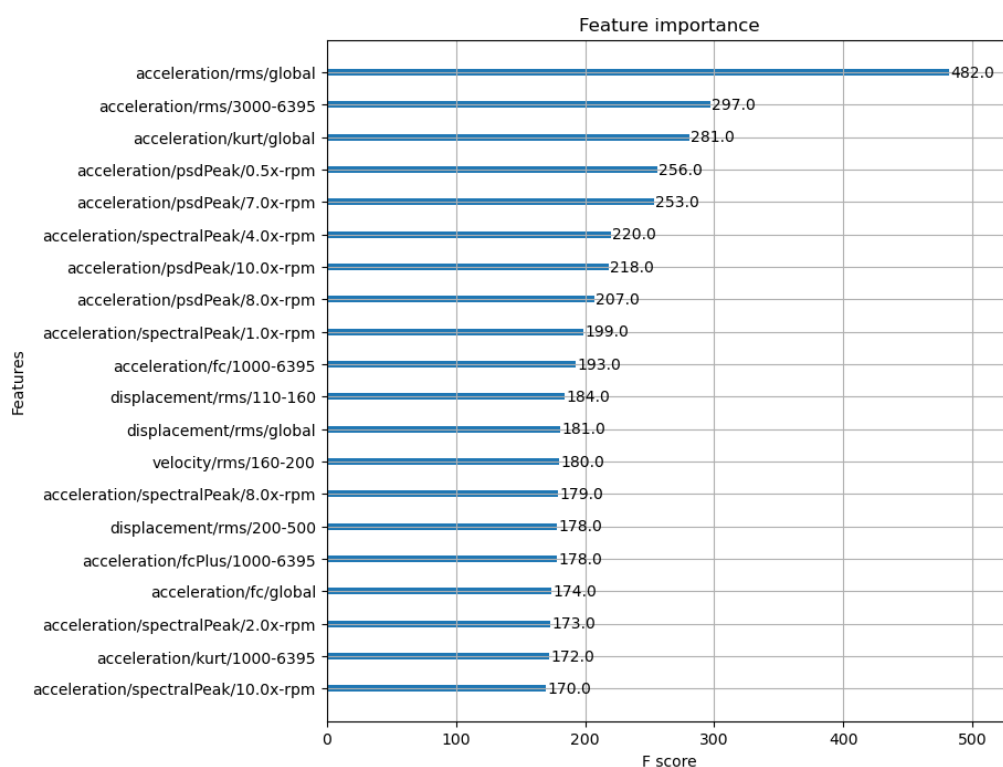


Figura 9. Ranking das 20 *features* mais importantes para o modelo XGBoost.

#### 4. Conclusões e Trabalhos Futuros

Neste artigo, foi apresentado o desenvolvimento de modelos de aprendizado de máquina para detecção de falhas em máquinas rotativas, mais especificamente os transportadores de correia. Devido a origem dos dados coletados serem de máquinas em operação industrial, foram realizados esforços significativos para limpar e pré-processar os dados, garantindo assim um conjunto adequado para treinamento e teste dos experimentos.

Pela natureza do problema e para a aplicação como suporte à tomada de decisão em relação à manutenção de cada ativo, foi essencial a escolha da métrica a ser utilizada e a definição do ponto de operação do modelo de classificação binária. A métrica selecionada foi a ROC AUC, considerada a mais apropriada para otimizar o modelo, e o ponto de operação foi fixado em 90% de TPR a fim de minimizar os falsos negativos, mesmo que isso resulte em um aumento na quantidade de falsos positivos.

Os resultados obtidos demonstraram que o modelo XGBoost utilizando as *features* especializadas obteve o melhor desempenho, alcançando uma ROC AUC de 91% e uma FPR de 23%. O modelo mostrou uma razoável capacidade de detecção de falhas mecânicas mesmo com *features* simples, como as 10 primeiras harmônicas da velocidade de rotação da máquina, mas as *features* especializadas proporcionaram ainda um ganho significativo de desempenho.

No mais, observa-se que as *features* extraídas da aceleração do sinal apresentam importância maior no treinamento do melhor modelo em comparação ao restante das *features*, sendo o RMS Global a *feature* mais importante dentre as 127 utilizadas.

A próxima etapa do projeto envolverá a extração de novas *features* no domínio do tempo, frequência e tempo-frequência, como por exemplo *features* obtidas usando a transformada de Wavelet [Ahmed and Nandi 2020].

## Referências

- Ahmed, H. and Nandi, A. K. (2020). *Condition monitoring with vibration signals: Compressive sampling and learning algorithms for rotating machines*. John Wiley & Sons.
- Alharbi, F., Luo, S., Zhang, H., Shaukat, K., Yang, G., Wheeler, C. A., and Chen, Z. (2023). A brief review of acoustic and vibration signal-based fault detection for belt conveyor idlers using machine learning models. *Sensors*, 23(4):1902.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.
- Boashash, B. (2015). *Time-frequency signal analysis and processing: a comprehensive reference*. Academic press.
- Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., and Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215.
- Chen, T. and Guestrin, C. (2016). XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.
- Gao, Z., Cecati, C., and Ding, S. X. (2015). A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches. *IEEE transactions on industrial electronics*, 62(6):3757–3767.
- Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Incorporated.
- Janssens, O., Slavkovikj, V., Vervisch, B., Stockman, K., Loccufer, M., Verstockt, S., Van de Walle, R., and Van Hoecke, S. (2016). Convolutional neural network based fault detection for rotating machinery. *Journal of Sound and Vibration*, 377:331–345.
- Laguna, C. and Lerch, A. (2016). An efficient algorithm for clipping detection and declipping audio. In *Audio Engineering Society Convention 141*. Audio Engineering Society.
- Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., and Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138:106587.
- Randall, R. B. (2021). *Vibration-based condition monitoring: industrial, automotive and aerospace applications*. John Wiley & Sons.
- Welch, P. D. (1967). The use of fast fourier transforms for the estimation of power spectra: A method based on time averaging over short modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15:70–73.
- Zimroz, R. and Król, R. (2009). Failure analysis of belt conveyor systems for condition monitoring purposes. *Mining Science*, 128(36):255.