

Detecção de falhas em motores elétricos utilizando Aprendizado de Máquina

Leonardo Rezende Novais
Departamento de Engenharia Elétrica
Universidade Federal de São Carlos
São Carlos, Brasil
leonardo.novais@estudante.ufscar.br

Robson Barcellos
Departamento de Engenharia Elétrica
Universidade Federal de São Carlos
São Carlos, Brasil
rbarcellos@ufscar.br

Abstract - Este trabalho propõe um método de detecção de falhas em motores elétricos utilizando técnicas de Aprendizado de Máquina. Para treinamento e teste do modelo, foi utilizado um banco de dados contendo 1951 séries temporais de 4 diferentes sensores, que simulam 6 estados de funcionamento: normal, desbalanceamento, desalinhamento horizontal, desalinhamento vertical, falha no rolamento interno, e falha no rolamento externo.

Keywords - Aprendizado de Máquina, motores elétricos, detecção de falhas

1. INTRODUÇÃO

Dada a larga utilização de máquinas rotativas na Indústria, sua disponibilidade e eficiência são pontos fundamentais a serem observados. Além dos evidentes impactos financeiros que uma interrupção no processo industrial pode causar, existem impactos secundários que podem ser ocasionados por uma falha nesse tipo de máquina, como na segurança dos operadores, ou na reputação da empresa.

A manutenção preditiva surge como uma alternativa para a redução dos altos custos associados às manutenções periódicas ou corretivas. Trata-se de uma abordagem proativa, que visa identificar uma falha, antes que esta cause a interrupção do processo industrial de forma inesperada. Através do acompanhamento do estado de funcionamento das máquinas rotativas, é possível identificar uma falha antes da interrupção do processo, e realizar a manutenção no momento mais oportuno, contribuindo para a confiabilidade e eficiência do processo industrial [1].

Neste contexto, métodos de Aprendizado de Máquina podem ser utilizados para a identificação e classificação dessas falhas. Algoritmos de Aprendizado de Máquina como o SVM [2] e Redes Neurais [3] têm sido utilizados nesse tipo de problema, e alcançado bons resultados.

Este trabalho propõe um método de Aprendizado de Máquina para identificação de determinadas falhas comumente verificadas em máquinas rotativas. A partir de dados coletados de um simulador de falhas, realiza-se um processo de extração de características dos sinais, e um subsequente treinamento do modelo. Os resultados foram por fim, comparados à abordagem apresentada em [4], que além de utilizar um modelo mais complexo, utiliza um conjunto maior de atributos.

2. DESCRIÇÃO DO SISTEMA

2.1. Machinery Fault Simulator (MFS)

O simulador de falhas utilizado na geração do banco de dados [5] foi o SpectraQuest's Machinery Fault Simulator Alignment-Balance-Vibration, conforme Figura 1. Ele faz parte de uma família de simuladores de falhas da empresa

SpectraQuest [6], que são desenvolvidos para emular e estudar a assinatura das falhas mais comumente encontradas em máquinas rotativas. As especificações do MFS, disponibilizadas em [5], são dadas pela Tabela 1.

Tabela 1 – Especificações do motor [5]

Especificação	Valor
Motor	1/4 CV DC
Intervalo de Frequência	700-3600 RPM
Peso do sistema	22 kg
Diâmetro do eixo	16 mm
Comprimento do eixo	520 mm
Rotor	15,24 cm
Distância entre rolamentos	390 mm

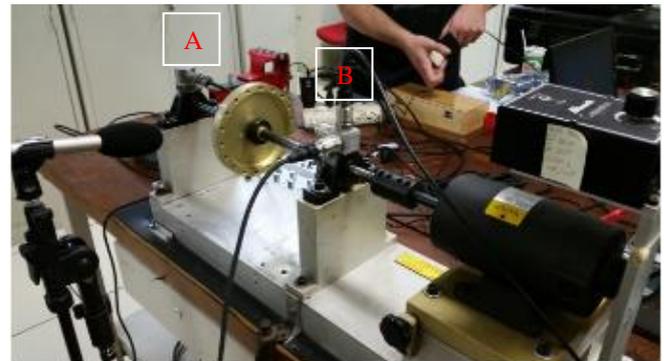


Fig. 1. Simulador de falhas da SpectraQuest utilizado na geração do banco de dados. A e B indicam as posições dos rolamentos externos e internos, respectivamente, assim como seus respectivos acelerômetros. Fonte: [4]

Para a construção do banco de dados, além do funcionamento sem falhas, foram simulados 3 tipos de falhas, quais sejam:

- Desalinhamento do sistema: Essa falha simula o desalinhamento vertical, ou horizontal, entre o eixo do rotor e o eixo de rotação da carga, conforme indicado na Figura 2.
- Desbalanceamento do sistema: Ocasionalmente por um desbalanceamento de massa ao redor do eixo de rotação, essa falha é simulada com a utilização de um disco ao redor do eixo, conforme a Figura 3.
- Falhas nos rolamentos: No simulador de falhas utilizado, os rolamentos estão posicionados nas extremidades do eixo de carga, apoiados sobre a estrutura metálica. As posições dos rolamentos externo e interno, são indicadas por A e B na Figura

2. Para simular as falhas desses elementos, foram utilizados rolamentos com 3 tipos de defeitos: falha na pista do anel externo, falha na pista do anel interno, e falha nos elementos rolantes. As falhas dos rolamentos foram introduzidas uma de cada vez, ora no rolamento externo, ora no rolamento interno. No entanto, dada a dificuldade desse tipo de falha ser detectada de maneira isolada, os rolamentos defeituosos foram introduzidos em conjunto com o desbalanceamento do sistema.

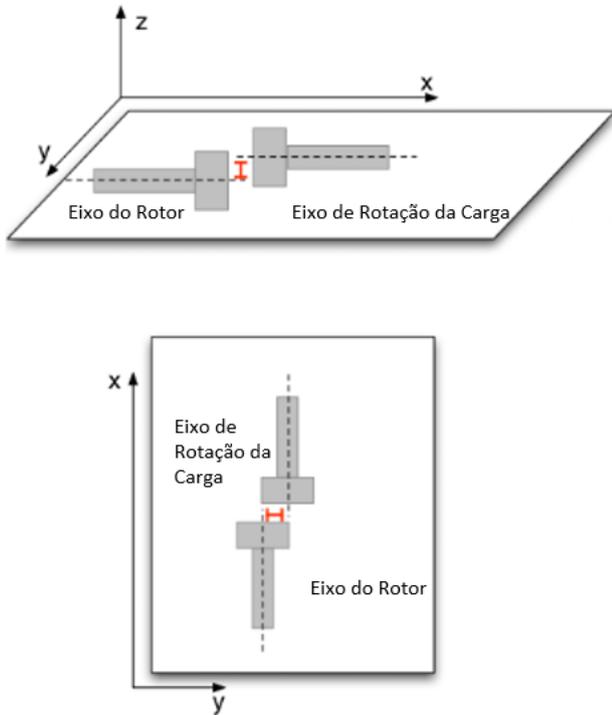


Fig. 2. Desalinhamento vertical e horizontal exemplificados na imagem superior e inferior, respectivamente. Fonte: [3]

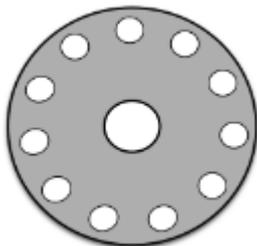


Fig. 3. Disco utilizado para simular o desbalanceamento ao redor do eixo. O eixo é inserido no orifício central, enquanto os orifícios externos são utilizados para a inserção de pesos. Fonte: [3]

2.2 Banco de Dados - MAFAULDA

O banco de dados utilizado neste estudo foi o Machinery Fault Database (MAFAULDA). O MAFAULDA é composto por 1.951 séries temporais multivariadas, contendo cada uma das falhas listadas anteriormente, além do estado normal de funcionamento [5]. Cada uma das 1.951 séries temporais tem duração de 5 segundos, e é composta por 250.000 amostras de quatro sensores: um tacômetro, dois acelerômetros (posicionados um em cada rolamento, que medem a aceleração nos eixos radial, axial e tangencial), e um microfone, totalizando oito sinais. Uma amostra de 0,2 segundos dos sinais do tacômetro e do microfone, está

representada na Figura 4, e uma amostra de mesma duração dos sinais dos acelerômetros está representada na Figura 5.

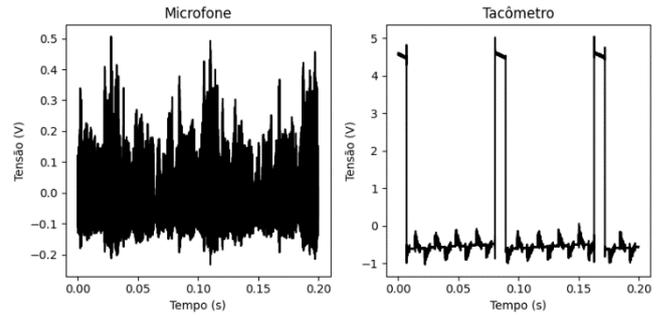


Fig. 4. Amostra de 0,2 segundos dos sinais do Tacômetro e Microfone

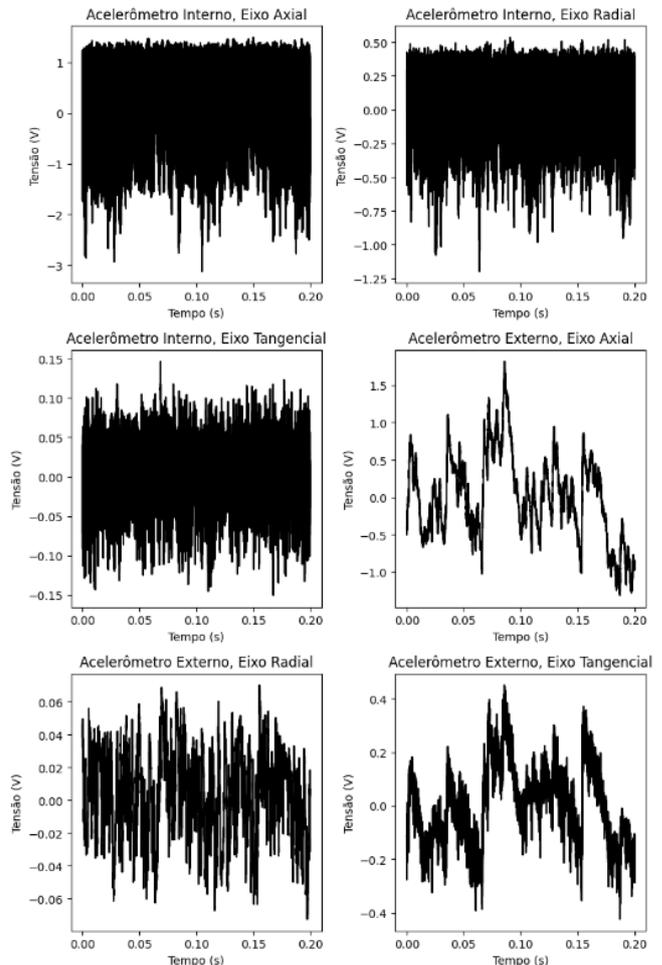


Fig. 5. Amostra de 0,2 segundos dos sinais dos acelerômetros

Apesar do autor informar em [5] que a frequência de aquisição é de 50 kHz, as frequências de rotação do motor só coincidiram com as informadas pelo autor, quando a frequência de aquisição assumida foi de 51,2 kHz (valor informado pelo fabricante do módulo de aquisição), conforme detalhado na seção seguinte.

Para cada falha, foram registradas as séries temporais multivariadas dos quatro sensores com diferentes frequências de rotação R_f , no intervalo $720 \leq R_f \leq 3600$ RPM, com incrementos de aproximadamente 60 RPM. No entanto, em alguns casos, essa velocidade máxima não foi atingida, devido à vibração induzida pela respectiva falha. Os registros das séries temporais foram divididos por tipo de falha, conforme a Tabela 2.

Tabela 2 – Número de Séries Temporais registradas por tipo de falha [5]

Tipo de Falha	Número de Séries Temporais
Normal	49
Desbalanceamento	333
Desalinhamento	498
Falha nos rolamentos	1071
Total	1951

Para cada um dos tipos de falha indicados na Tabela 2, as séries temporais foram construídas da seguinte maneira:

- Normal: 49 séries temporais com diferentes frequências de rotação no intervalo estipulado, sem a indução de falhas.
- Desbalanceamento: O cenário de desbalanceamento do sistema foi simulado com 7 cargas distintas que variavam de 6 g a 35g. Devido à vibração induzida pelo desbalanceamento, não foi possível alcançar velocidades maiores que 3300 RPM com todas as cargas, limitando assim, o número de séries temporais, totalizando 333.
- Desalinhamento: Foram simulados cenários de desalinhamento horizontal com magnitudes entre 0.50 mm e 2.00 mm, e cenários de desalinhamento vertical com magnitudes de 0.51 mm a 1.90 mm, sendo 197 séries temporais para o deslocamento horizontal, e 301 séries temporais para o deslocamento vertical, totalizando 498 séries temporais.
- Falhas no rolamento: Foram simulados 3 tipos de falhas em cada um dos rolamentos, sendo elas: falha na pista de rolamento externo, falha nos elementos rolantes, e falha na pista de rolamento interno, conforme indicado na Figura 6. As falhas nos rolamentos são praticamente imperceptíveis quando apresentadas de maneira isolada, e, por esse motivo, foram introduzidas cargas entre 6 g e 35 g para desbalancear o sistema. Foram registradas 188 séries temporais com a falha na pista de rolamento externa, para cada um dos rolamentos. Para as falhas nos elementos rolantes, registrou-se 184 e 188 séries temporais para os rolamentos interno e externo, respectivamente. Por fim, para a falha na pista de rolamento interna, registrou-se 186 e 137 séries temporais para os rolamentos interno e externo, respectivamente.

O desbalanceamento entre as classes, com destaque para a classe “Normal”, é notório. Essa característica do conjunto de dados é um ponto importante que pode impactar negativamente no desempenho do modelo, e deve ser analisado durante a avaliação dos resultados.



Fig. 6. Componentes de um rolamento. Fonte: [8]

3. EXTRAÇÃO DE ATRIBUTOS

Em Aprendizado de Máquina, as *features*, ou, atributos, são informações obtidas a partir de dados brutos, que são utilizadas como entrada do modelo. Fazendo uma analogia com modelagem estatística, os atributos seriam as variáveis independentes, e a variável que se deseja prever com o modelo de Aprendizado de Máquina, seria a variável dependente.

A escolha dos atributos é uma etapa fundamental do processo de modelagem, dado que nem todos os atributos disponíveis possuem relação com a variável que se deseja prever. Além disso, um número elevado de atributos pode resultar em um *overfitting*, ou, sobreajuste no modelo, quando este possui um desempenho elevado no treinamento, mas não é capaz de generalizar para dados novos.

O trabalho apresentado em [4], utilizou a técnica de SBM (Similarity-based modeling) para realizar a classificação das falhas simuladas em [5]. Os atributos utilizados como entrada do modelo, em diferentes combinações, foram divididos em três categorias:

- Frequência de rotação do motor: determinada a partir do sinal do tacômetro, como detalhado em [3].
- Atributos espectrais: obtidos a partir da análise no domínio da frequência dos sinais presentes no conjunto de dados (com exceção do tacômetro)
- Atributos estatísticos: obtidos por meio de estatísticas descritivas dos sinais presentes no conjunto de dados

Neste trabalho, avaliou-se se seria possível obter um resultado semelhante ao obtido em [4] utilizando um subconjunto destes atributos, conforme descrito abaixo.

3.1. Frequência de rotação

De acordo com [3], a frequência de rotação do motor pode ser estimada a partir da Transformada discreta de Fourier $S(k)$ obtida a partir do sinal $s(t)$ do tacômetro.

Para realizar essa estimativa, toma-se um índice k_a , tal que $|S(k_a)| \geq |S(k)|, \forall \{k \in \mathbb{Z} | 0 \leq k \leq N\}$, em que N é o número de pontos em $S(k)$. Desta forma, com a frequência F_S de amostragem do sinal $S(k)$, calcula-se uma frequência f_a pela Equação 1.

$$f_a = \frac{k_a F_s}{N} \text{ Hz} \quad (1)$$

Em seguida, um novo índice k_b é obtido, de forma que $|S(k_b)| \geq |S(k)|$, $\forall k \in \mathbb{Z} \setminus [k_a - 3, k_a + 3] \cap [0, N]$, de modo a se obter uma nova frequência f_b . O processo é repetido para um novo índice k_c , de modo que $|S(k_c)| \geq |S(k)|$, $\forall k \in \mathbb{Z} \setminus ([k_a - 3, k_a + 3] \cup [k_b - 3, k_b + 3]) \cap [0, N]$, para o cálculo de uma frequência f_c . Por fim, esse processo de obtenção do índice, de modo que se obtenha o valor de pico do sinal em módulo, e que o referido índice não pertença à união dos intervalos $[k_\theta - 3, k_\theta + 3]$, em que $k_\theta = \{k_a, k_b, k_c\}$ é repetido para um índice k_d , e uma frequência f_d é calculada. Assim, a estimativa da frequência de rotação do motor R_f pode ser finalmente determinada por:

$$R_f = \min\{f_a, f_b, f_c, f_d\} \quad (2)$$

Este método foi então utilizado para calcular a velocidade de rotação das 49 séries temporais do motor em funcionamento normal. Utilizando $F_s = 50$ kHz, as velocidades de rotação mínima e máxima calculadas foram de 720 e 3600 RPM, respectivamente. Para o cenário em que $F_s = 51,2$ kHz, as velocidades de rotação mínima e máxima calculadas foram de 737,28 e 3686,40 RPM, respectivamente, valores estes, que coincidem com o intervalo de velocidades informado em [5]. No entanto, conforme Tabela 1, 3686,40 RPM extrapola a frequência de rotação máxima informada nas especificações do motor. Para corrigir essa inconsistência, considerou-se que o intervalo de frequências de rotação informado (737 a 3686 RPM) está incorreto. Assim, utilizou-se a frequência F_s de 50 kHz, resultando em um intervalo de rotação entre 720 e 3600 RPM.

3.2. Atributos Estatísticos

- Média Simples: As médias simples de cada um dos oito sinais (tacômetro, 3 eixos de cada acelerômetro, e microfone) foram utilizadas como atributos do modelo.
- Curtose: A curtose é uma medida estatística adimensional que indica a deformação da Função Densidade de Probabilidade do sinal, em relação à uma distribuição normal. [7]

4. RESULTADOS EXPERIMENTAIS

A técnica de Aprendizado de Máquina escolhida para realizar a classificação da falha do motor elétrico, foi o modelo *Random Forest* [9]. Baseado em árvores de decisão, o modelo possui algumas características que o fazem se destacar no desempenho de tarefas de classificação similares à apresentada neste trabalho. São elas:

- Boa velocidade de treinamento
- Boa acurácia em problemas de classificação multiclasse
- Pouco suscetível à sobreajuste

Para avaliação do modelo, utilizou-se a estratégia k-dobras Estratificadas de validação cruzada. Nessa abordagem, o conjunto de dados original é dividido em um número k de partições (também chamadas de dobras), em que o conjunto de treino e teste correspondem a $1 - 1/k$ e $1/k$ do número de

elementos do conjunto de dados, respectivamente. Para que cada dobra seja representativa da distribuição dos dados originais, cada uma delas possui a mesma proporção de falhas apresentada no conjunto de dados completo. Em cada rodada de validação cruzada, realiza-se uma divisão dos dados, em que um número k-1 de dobras é utilizado como conjunto de treino, e uma dobra é utilizada como conjunto de teste. Esse processo é realizado um número k de vezes, gerando, portanto, k divisões distintas, e, ao fim, o desempenho do modelo é calculado pela média dos resultados obtidos em cada uma das divisões. Na Figura 7, apresenta-se como o conjunto de dados seria dividido em um cenário em que se utiliza k=5.

Dataset					
1ª divisão	Teste	Treino	Treino	Treino	Treino
2ª divisão	Treino	Teste	Treino	Treino	Treino
3ª divisão	Treino	Treino	Teste	Treino	Treino
4ª divisão	Treino	Treino	Treino	Teste	Treino
5ª divisão	Treino	Treino	Treino	Treino	Teste
	Dobra 1	Dobra 2	Dobra 3	Dobra 4	Dobra 5

Fig. 7. Ilustração da estratégia de validação cruzada k-dobras, em que se utiliza k=5

Neste trabalho, o número k de dobras adotado foi 5, resultando em um conjunto de treino e teste de 80% e 20% do conjunto de dados, respectivamente.

3.3. Random Forest

Para contextualizar o funcionamento do *Random Forest*, antes se faz necessário o entendimento do modelo de classificação por Árvore de Decisão. Como o nome indica, Árvore de Decisão é uma técnica de Aprendizado de Máquina que utiliza estruturas semelhantes à uma árvore, que dividem o conjunto de dados recursivamente até que se obtenha o melhor resultado de classificação possível.

Supondo-se um conjunto de dados com 10 elementos, divididos em 2 classes quaisquer, utilizando-se os atributos x_1 e x_2 , seria possível obter a árvore de decisão da Figura 8, em que $\{a, b, c\} \in \mathbb{R}$, são determinados de modo que se divida as classes da melhor maneira possível. Existem métodos para se determinar qual o melhor atributo para determinado nó, mas que não serão abordados neste trabalho.

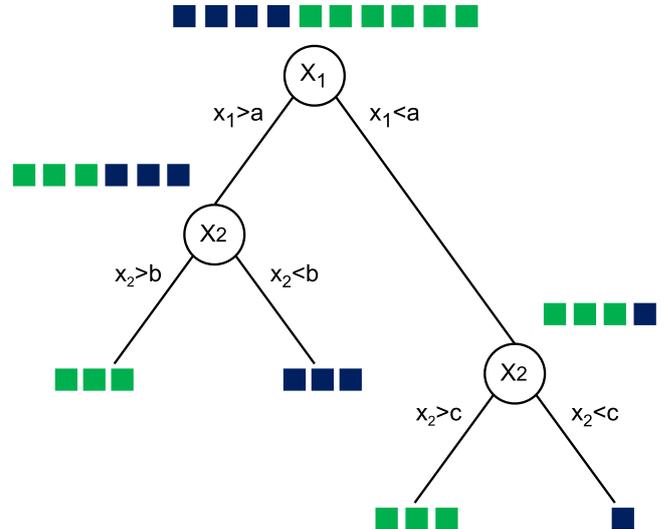


Fig. 8. Ilustração de uma árvore de decisão para um dado conjunto de dados qualquer de classificação binária

Assim, cada nó interno possui um atributo, cujo valor utilizado é determinado de modo que se obtenha a melhor separação entre as classes do subconjunto de elementos presente no nó. O objetivo é que sucessivos nós dividam o conjunto de dados até que se obtenham “folhas” em que se tenha uma classe com grande predominância, ou, idealmente, somente uma classe, tal qual ilustrado na Figura 8.

Desta forma, o modelo de *Random Forest* utiliza uma “floresta” de Árvores de Decisão, treinadas com subconjuntos do conjunto de dados original. Como cada árvore é treinada com um subconjunto diferente, utilizam também distintos atributos, resultando, conseqüentemente, em distintas classificações. Assim, para classificar um novo elemento, o modelo utiliza um sistema de votação, em que a moda das classificações da floresta determina a classificação derradeira.

É importante notar, que modelos de Aprendizado de Máquina baseados em Árvores de Decisão, não exigem que os atributos utilizados sejam normalizados, e, neste trabalho, tampouco foi realizado aumento dos dados. Deste modo, a etapa que antecede ao treinamento do modelo, é a definição das grandezas que definem as características ou estrutura deste. Essas grandezas são denominadas de hiperparâmetros, e variam de acordo com o modelo utilizado, incluindo, por exemplo, o número de camadas, ou a taxa de aprendizagem em uma rede neural.

Um método comumente utilizado para a escolha dessas grandezas, é definir um subconjunto dos hiperparâmetros possíveis para determinado modelo, assim como um conjunto de valores a serem adotados por cada um deles. Em seguida, realiza-se uma exploração exaustiva de cada combinação de hiperparâmetros possível, avaliando-se o desempenho do modelo em cada uma delas. A combinação de valores que resultar no melhor desempenho, é então utilizada, e a esse processo de escolha, dá-se o nome de busca em grade. Essa busca, no entanto, pode representar um custo computacional elevado. Neste trabalho, para reduzir o universo de busca, utilizou-se somente dois dos principais hiperparâmetros do *Random Forest*:

- Profundidade máxima de cada árvore – Define a complexidade de cada árvore de decisão da floresta, isto é, o número de níveis máximo desde o nó inicial, até as folhas. Se esse parâmetro for muito alto, o modelo pode experimentar sobreajuste
- Número de árvores na floresta – um incremento do número de árvores, geralmente é acompanhado de um incremento na performance do modelo, assim como no custo computacional

Para definir a profundidade máxima de cada árvore, a busca em grade percorreu os valores 5, 7, 9, 11, 15 e 20, enquanto para o número de árvores na floresta, a busca se limitou aos valores de 100, 500 e 1000. A definição desses intervalos de busca não seguiu um critério objetivo pré-estabelecido, mas sim, o conhecimento empírico associado à experimentação. Dado o desbalanceamento do banco de dados, a métrica de avaliação utilizada foi o F1-Score Macro. As 5 melhores combinações de hiperparâmetros estão definidas na Tabela 3.

Tabela 3 – 5 melhores modelos da Busca em Grade

Profundidade Máxima	Número de árvores	F1-Score Macro (Teste)	F1-Score Macro (Treino)
20	1000	87,19% ± 2,74%	100% ± 0 %
20	500	86,61% ± 2,88%	100% ± 0 %
20	100	86,37% ± 3,07%	100% ± 0 %
15	1000	86,32% ± 1,63%	99,95% ± 0,06 %
15	500	86,14% ± 2,62%	99,96% ± 0,06 %

A melhor combinação de hiperparâmetros, portanto, foi com a profundidade máxima de 20, e utilizando 1000 árvores em cada floresta. Considerando a margem de erro, no entanto, os resultados são estatisticamente iguais. Para comparar este modelo com uma simples Árvore de Decisão, com profundidade máxima de 20, obteve-se um F1-Score Macro de 65,39% ± 4,52%, denotando a superioridade do *Random Forest*. O ganho no F1-Score Macro ao aumentar o número de árvores da floresta, foi marginal, assim como o ganho ao aumentar a profundidade máxima de cada árvore. Desta forma, para evitar o custo de processamento em um universo maior de busca, esta foi limitada aos valores citados anteriormente.

Para compreender o desempenho do modelo para cada tipo de falha presente no conjunto de dados, utilizou-se uma matriz de confusão, conforme Figura 9. Nesta matriz, cada linha representa uma falha, enquanto as colunas representam a classificação dada pelo modelo. Assim, dado um elemento a_{ij} da matriz, seu valor é definido pelo número de vezes que o modelo classificou a falha da linha i , como sendo a falha da coluna j . Em um modelo ideal, portanto, todos os elementos fora da diagonal principal, seriam zero. Essa matriz nos permite analisar o desempenho do modelo para cada uma das classes, como avaliar se alguma falha está muito frequentemente sendo classificada como outra falha, ou se alguma classe está afetando a acurácia de maneira desproporcional. Nesses casos, seria possível explorar uma solução, como realizar um melhor balanceamento das classes, utilizando aumento de dados, por exemplo. Ademais, como complemento à matriz de confusão, as métricas de avaliação [10] presentes na Tabela 4, foram analisadas para cada uma das classes.

Falha	Desal. Horizontal	Desal. Vertical	Desbalanceamento	Elem. Rol. do Rolam. Ext.	Elem. Rol. do Rolam. Int.	Normal	Pris. de Rol. Ext. do Rolam. Ext.	Pris. de Rol. Ext. do Rolam. Int.	Pris. de Rol. Int. do Rolam. Ext.	Pris. de Rol. Int. do Rolam. Int.
Desal. Horizontal	35	4	0	0	0	0	0	0	0	0
Desal. Vertical	2	57	0	0	0	0	1	0	0	0
Desbalanceamento	0	4	61	0	0	0	0	0	1	1
Elem. Rol. do Rolam. Ext.	0	0	0	27	0	0	0	0	0	0
Elem. Rol. do Rolam. Int.	0	1	0	0	35	0	0	0	1	0
Normal	0	1	2	0	0	7	0	0	0	0
Pris. de Rol. Ext. do Rolam. Ext.	0	0	1	0	0	0	34	0	3	0
Pris. de Rol. Ext. do Rolam. Int.	0	0	0	0	0	0	0	37	0	0
Pris. de Rol. Int. do Rolam. Ext.	1	0	0	0	1	0	0	0	36	0
Pris. de Rol. Int. do Rolam. Int.	0	0	3	0	0	0	1	0	0	34

Fig. 9. Matriz de confusão para o modelo Random Forest

Tabela 4 – Métricas de Avaliação do modelo Random Forest

Falha	Precisão	Sensibilidade	F1-Score	Amostras de teste
Desalinamento Horizontal	0,92	0,90	0,91	39
Desalinamento Vertical	0,85	0,95	0,90	60
Desbalanceamento	0,91	0,91	0,91	67

Elementos rolante do rolamento externo	1,00	1,00	1,00	27
Elementos rolantes do rolamento interno	0,97	0,95	0,96	37
Normal	0,88	0,70	0,78	10
Pista de rolamento externa do rolamento externo	0,97	0,89	0,93	38
Pista de rolamento externa do rolamento interno	1,00	1,00	1,00	37
Pista de rolamento interna do rolamento externo	0,88	0,95	0,91	38
Pista de rolamento interna do rolamento interno	0,97	0,89	0,93	38

Tanto pela Figura 9, quanto pela Tabela 4, é possível inferir que a eficiência da classificação do estado Normal de funcionamento se destacou negativamente das outras classes. Uma hipótese provável para explicar esse comportamento, é a significativa sub-representação desta classe no conjunto de dados original.

Por fim, uma comparação com os resultados obtidos em [4] foi realizada. O artigo citado realizou a comparação de diversos cenários utilizando modelos e conjuntos de atributos distintos, dentre eles, um modelo de *Random Forest* que utiliza 46 atributos como entrada do modelo. Para uma comparação justa, as condições de avaliação foram iguais. O artigo citado utiliza a acurácia média para avaliação do modelo, e no caso do modelo que utiliza somente o *Random Forest*, o intervalo de confiança não foi apresentado. Além disso, a validação cruzada foi feita com 10 dobras (k=10). Assim, conforme Tabela 5, o modelo deste trabalho obteve uma acurácia média de $91,39\% \pm 2,09\%$, enquanto o modelo de *Random Forest* do artigo estudado atingiu 92,70%, dando uma desvantagem pouco significativa ao modelo deste artigo. É importante notar, no entanto, que este trabalho utilizou um subconjunto dos atributos utilizados em [4] (17 e 46 atributos, respectivamente).

Tabela 5 – Comparativo entre o modelo apresentado neste trabalho, e o modelo apresentado em [4]

Modelo	Número de atributos	Acurácia Média
Modelo proposto	17	$91,39\% \pm 2,09\%$
Modelo apresentado em [4]	46	92,70%

É possível, portanto, resumir o processo de obtenção do modelo por meio do fluxograma da Figura 10.



Fig. 10. Fluxograma do modelo proposto

4. CONCLUSÃO

Os resultados apresentados neste trabalho, obtiveram uma desvantagem ligeira, e pouco significativa, considerando a margem de erro, na acurácia obtida, quando comparados aos resultados do modelo que utiliza somente *Random Forest* em [4]. No entanto, estes foram obtidos utilizando-se um subconjunto dos atributos do trabalho citado.

Outro ponto de destaque, é que ainda há espaço para melhora na performance. Como destacado na matriz de confusão da Figura 9, alguns estados de funcionamento, em especial, o “Normal”, estão sub-representados no conjunto de dados utilizado. Desta forma, utilizar uma abordagem de aumento de dados, pode resultar em uma melhor acurácia, observando para que a estratégia não incorra em vazamento de dados no treinamento do modelo. Por fim, também é possível que se obtenha uma acurácia melhor, ao aumentar o número de hiperparâmetros na busca em grade, ou ainda investigando a utilização de outros atributos.

REFERÊNCIAS

- [1] BAZAN, G. H.. *et al.* **Multi-Fault Diagnosis in Three-Phase Induction Motors Using Data Optimization and Machine Learning Techniques**. Electronics, 2021, 10, 1462.
- [2] PANDARAKONE, S. E. *et al.* **Distinct analysis of induction motor bearing using frequency spectrum determination and support vector machine**. IEEE Trans. Ind. Applicat., 2017, 53, p. 3049-3056.
- [3] LIMA, A. A. *et al.* **On fault classification in rotating machines using fourier domain features and neural networks**. IEEE 4th Latin American Symposium On Circuits and Systems (LASCAS), 2013, p.1-4.
- [4] MARINS, M. *et al.* **Improved similarity-based modeling for the classification of rotating-machine failures**. Journal of The Franklin Institute, 2018, Vol.355 (4), p. 1913-1930.
- [5] Mafaulda. **Machinery fault database**. Disponível em: <https://www02.smt.ufrj.br/offshore/mfs/>. Acesso em 14 Outubro de 2022.
- [6] Spectraquest. **Machinery Fault Simulators**. Disponível em: <https://spectraquest.com/products/simulators/machinery-fault-simulators/>. Acesso em 14 de Outubro de 2022.
- [7] LIMA, A. A. *et al.* **The influence of feature vector on the classification of mechanical faults using neural networks**. IEEE 7th Latin American Symposium on Circuits and Systems (LASCAS), 2016, p. 115-118 .
- [8] Dicronite. **How to avoid bearing failure**. Disponível em: <https://www.dicronite.com/blog/bearings/>. Acesso em 2 de Outubro de 2022.
- [9] BREIMAN, L. **Random Forests**. Journal of Machine Learning, 2001, p. 5-32.
- [10] SAMMUT, C; WEBB, G. **Encyclopedia of Machine Learning**. Springer, 2011.