

Uma Abordagem Eficiente para Classificação de Falhas em Rolamentos de Motores Elétricos

Rafael Fernando Silva e Souza* Mailson Ribeiro Santos**
Luiz Affonso Guedes**

* Pós-graduação em Engenharia Elétrica e de Computação,
Universidade Federal do Rio grande do Norte, RN, (e-mail:
rafael.souza.080@ufrn.edu.br).

** Departamento de Engenharia de Computação e Automação,
Universidade Federal do Rio Grande do Norte, RN.
(e-mail: mailson.ribeiro.102@ufrn.edu.br)
(e-mail: affonso@dca.ufrn.br)

Abstract: This paper aims to analyze the influence of input variables on the performance of electric motor bearing fault classifiers. For this, the decision tree algorithm is used to determine and order the relevance of each input variable in the classification process. In this case, the relevance of seven time-domain variables associated with motor shaft vibration signals are analyzed (RMS, peak, minimum, maximum, standard deviation, kurtosis and crest factor). Then, a performance analysis is performed between the number of fault classifier inputs by the accuracy obtained by it. With this, one can observe the best relationship between efficiency and effectiveness of the classifier. The results were obtained using data from the benchmark of Case Western University.

Resumo: Este artigo se propõe a analisar a influência das variáveis de entrada no desempenho de classificador de falhas em rolamentos de motores elétricos. Para isto se utiliza o algoritmo de árvore de decisão para se determinar e ordenar a relevância de cada variáveis de entrada no processo de classificação. No caso são analisadas a relevância de sete variáveis no domínio do tempo associadas a sinais de vibração do eixo do motor (valor RMS, de pico, mínimo, máximo, desvio padrão, curtose e fator de crista). Então é realizada uma análise de desempenho entre número de entradas do classificador de falhas pela acurácia obtida pelo mesmo. Com isto, pode-se observar a melhor relação entre eficiência e eficácia do classificador. Os resultados foram obtidos utilizando-se dados do *benchmark* da Case Western University.

Keywords: Electric motor bearings; Decision Tree; Machine Learning; Artificial Intelligence.

Palavras-chaves: Rolamentos em motores elétricos; Árvore de Decisão; Aprendizagem de Máquinas; Inteligência Artificial.

1. INTRODUÇÃO

Motores elétricos são equipamentos centrais nas diversas tipo de indústrias. Um motor elétrico é um sistema composto de diversos elementos mecânicos e elétricos que trabalham para o funcionamento normal do conjunto. A parte mecânica é composta basicamente de enrolamento do estator, barra do rotor, eixo e rolamentos; sendo este composto por um conjunto de anéis, internos e externos, e esferas que giram, formando uma pista entre os anéis. Em geral, o mau funcionamento do motor elétrico está associado a danos nos seus componentes mecânicos, principalmente nos rolamentos. Por sua vez, o dano causado pela vibração é o fator que mais influencia no funcionamento do motor Cerrada et al. (2018), ocasionando vibrações anormais no seu funcionamento, com consequentemente perda de eficiência.

Os rolamentos de motores elétricos possuem basicamente três partes: *Inner-Race*, *Outer-Race* e *Ball*. Esses componentes dos rolamentos são susceptíveis a danos pois quando

em funcionamento do motor elétrico esses componentes giram no sentido do eixo do motor e surgem forças mecânicas no sentido oposto ao sentido de seu funcionamento. Como isto há desgaste natural do material desses componentes, gerando imperfeições em suas geometrias. Imperfeições na geometria desses componentes dos rolamentos, aliado ao desalinhamento dos mesmos em relação ao eixo do motor, geram vibrações em alta frequência do eixo do motor, que por sua vez tendem a agravar ainda mais esse fenômeno. Dependendo da gravidade dessas imperfeições pode levar à operação ineficiente do motor (operação em falha) até a parada de seu funcionamento (defeito). Na Figura 1 é ilustrado esse fenômeno.

Na literatura há várias abordagens relacionadas à detecção e classificação de falhas de motores elétricos que utilizam técnicas de aprendizagem de máquina (em inglês, *Machine Learning* (ML)). Muitas dessas abordagens utilizam como entrada de seus modelos características (em inglês *features*) dos sinais de vibração do eixo do motor. Essas características são obtidas através operações no domínio

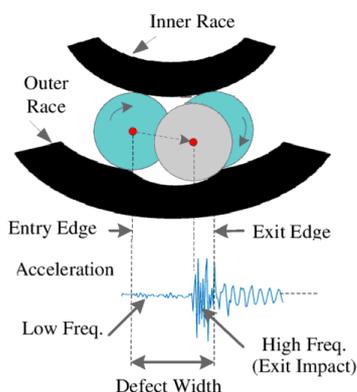


Figura 1. Detecção de falhas em rolamentos. Fonte: (Ismail et al., 2022)

do tempo (medidas estatísticas, por exemplo), frequência (componentes da Transformada Rápida de Fourier - FFT, por exemplo) ou tempo-frequência (componentes da Transformada Wavelet Discreta - DWT, por exemplo) sobre amostras de sinais de vibração do eixo do motor. As características no domínio do tempo mais utilizadas são os valores mínimo, máximo, RMS, Curtose, desvio-padrão, de pico e de fator de crista do sinal de vibração num dado intervalo de tempo (janela de tempo) (Haj Mohamad and Nataraj, 2021; Du et al., 2017; Niu et al., 2005; Honarvar and Martin, 1997; Prieto et al., 2013; SAMANTA and AL-BALUSHI, 2003; Sreejith et al., 2008). Neste trabalho utilizaremos apenas características no domínio do tempo, devido à boa relação entre simplicidade computacional e desempenho, além delas serem mais intuitivas.

Um problema relevante encontrado na área de classificação de falhas é justamente se determinar adequadamente as *features* que serão adotadas como as entradas do modelo de ML. Isto tem impacto direto não somente no desempenho do modelo, como também na complexidade computacional do mesmo. Assim, escolhas adequadas de *features* têm impacto na eficácia e na eficiência dos modelos de classificação de falhas baseados em técnicas de ML. Diante deste relevante aspecto, neste artigo propomos analisar o impacto das *features* sobre o desempenho de classificadores de falhas em rolamentos de motores elétricos. Para isto, apresentamos uma abordagem que inicialmente obtém a relevância de cada *feature* para o processo de classificação de falhas e, então, se analisa o desempenho do classificador de falhas para diversas configurações de número de *features*, seguindo a ordem de importância destas. Neste trabalho utilizamos um modelo de árvore de decisão para o classificador de falhas e as *features* são obtidas com operações no domínio do tempo.

O restante deste artigo possui mais quatro seções. Na Seção 2 é apresentada a fundamentação teórica para compreensão da proposta, contendo conceituação sobre modelo

de árvore de decisão e estratégia de seleção de *features*. Na Seção 3 é apresentada a nossa abordagem para realizar a análise e classificação de falhas de forma eficaz e eficiente. Na Seção 4 são apresentadas as configurações experimentais (*dataset* e modelos e parâmetros utilizados) e os resultados obtidos. Na Seção 5 são apresentadas as principais conclusões obtidas e indicações de trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção iremos apresentar alguns conceitos necessários para o melhor entendimento deste trabalho como árvore de decisão e abordagens para seleção de características do modelo de ML.

2.1 Árvore de Decisão

Em computação, uma árvore é uma estrutura de dados formada por um conjunto de elementos que armazenam informações chamadas nodos ou nós. É similar a uma árvore encontrada na natureza, mas invertida de cabeça para baixo, ou seja, a raiz é o elemento mais acima da estrutura que possui ligações para outros elementos denominados ramos ou filhos que podem estar ligados a outros ramos e o elemento mais abaixo, que não possui ramos, é conhecido como nó folha, nó terminal ou nó externo. Na Figura 2, temos um exemplo de uma árvore.

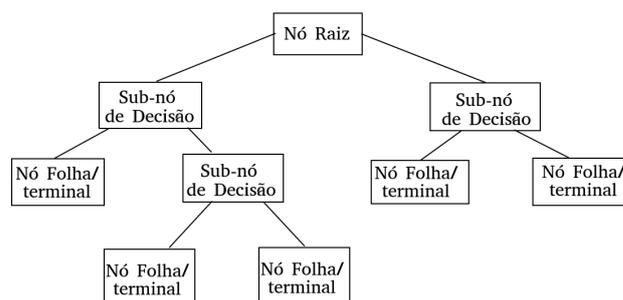


Figura 2. Estrutura de uma Árvore de Decisão.

Árvore de Decisão (em inglês *Decision Tree* - DT) é uma técnica de aprendizagem supervisionada de máquina, cujo conhecimento é armazenado em uma estrutura de árvore. Nos casos em que a árvore de decisão é usada para classificação, os critérios de partição dos dados mais conhecidos para se obter a estrutura da árvore são baseados na entropia e no índice Gini. Esses índices medem o grau de heterogeneidade dos dados com relação à estrutura de árvore obtida. No caso do índice de Gini, mede-se o grau de pureza dos nós da árvore.

Ao se construir a estrutura da árvore em função da profundidade da mesma, podem surgir dois tipos de erros: *underfit* e o *overfit* Allamy (2014). O primeiro ocorre quando, devido à pouca quantidade de amostras de treino, a arquitetura não permite a criação de regras específicas para tomar decisão, assim, o modelo não consegue criar regras adequadas de comportamento de dados. Enquanto o segundo caso ocorre devido à obtenção de uma árvore super-especializada aos dados de treino, mas com pouca

capacidade de generalização. No primeiro caso a árvore tende a ser pouco profunda, enquanto que no segundo a estrutura da árvore tende a ser bastante profunda. Deste modo, a profundidade máxima da árvore é um de seus hiperparâmetros, além do critério utilizado para se determinar a sua estrutura.

O algoritmo de árvore de decisão pertence à família de algoritmos denominada de *Top Down Induction of Decision Trees* - TDIDT - uma vez que a árvore pode ser representada como um conjunto de regras para tomar uma decisão. Cada regra, por sua vez, começa na raiz da árvore e finaliza numa folha.

De modo geral, o algoritmo de aprendizagem de árvore de decisão corresponde a se obter a sua estrutura de árvore. Para isto, de posse de um conjunto de **DADO** - que contém exemplos de aprendizado (A), uma condição de parada $t(A)$ e uma função de avaliação $Av(A, \text{atributo})$ - determina-se as regras do tipo: **SE** em A for satisfeita a condição de término $t(A)$, **ENTÃO** retorne o valor da classe, **CASO CONTRÁRIO** particione os exemplos de A em subconjuntos e, de acordo com o valor, continua para o próximo nó.

Além de obter bom desempenho na tarefa de classificação de dados, as árvores de decisão são consideradas como modelos de ML do tipo caixa-branca, pois permitem que se interpretem facilmente os seus resultados e tomadas de decisões. No caso, as folhas da árvore são as saídas do modelo e quanto mais próximas da raiz da árvore as folhas estiverem, mais relevantes elas são para a tomada de decisão do modelo. Assim, árvores de decisão se mostram bastante adequadas para serem utilizadas na nossa abordagem de classificação de falhas de rolamentos de motores elétricos.

2.2 Seleção de Características

A seleção de características (*features*) da ML é uma importante fase para conseguir um modelo de aprendizagem eficiente, visto que determinadas variáveis de entrada do modelo apresentam um maior grau de informação. Com isto é possível, em tese, em certas condições é possível se obter modelos com número de entradas reduzidas, cujos desempenhos são similares a modelos com maior quantidade de de entradas (*features*). Assim, obter-se o número adequado de entradas do modelo de ML diminui a complexidade do mesmo, impactando diretamente nos seus custos computacionais para treinamento e execução.

Há disponível na literatura várias técnicas para redução de número de *features* de modelos de ML. Como exemplo podemos citar análise de colinearidade entre variáveis, para eliminar variáveis com alto grau de redundância. Também é possível técnicas de redução de dimensionalidade, como PCA (*Principal Component Analysis*) para se reduzir o número de *features* do modelo de ML, mas nesse caso é introduzido uma etapa a mais de pré-processamento, tanto para treinamento quanto para execução da ML.

Como neste artigo estamos utilizando a árvore de decisão como modelo de ML, para seleção das *features* do modelo de classificação de falhas iremos utilizar o grau de importância da *feature* na árvore de decisão, que está associado

à sua distância até a raiz da árvore. Lembrando que as classes são as folhas da árvore.

3. ABORDAGEM PROPOSTA

Nesta seção iremos apresentar o passo-a-passo da nossa abordagem para obter um modelo eficiente de classificação de falhas em rolamentos de motores elétricos.

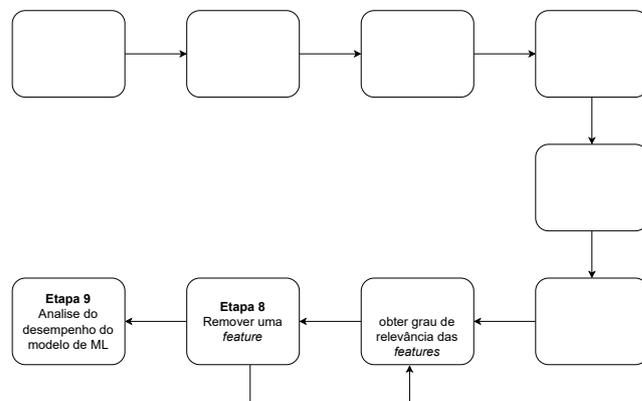


Figura 3. Procedimento para obtenção de classificador de falhas eficiente.

O procedimento proposto para se obter um classificador de falhas eficiente é ilustrado na Figura 3, onde as suas etapas têm as seguintes atividades:

- **Etapa 1** - Obtenção dos dados de série temporal provenientes dos sensores de vibração do eixo do motor. Caso necessário, deverá ser realizado procedimento de filtragem para remoção de ruídos.
- **Etapa 2** - Obtenção dos valores das características (*features*) dos sinais de vibração. Esses valores são obtidos dentro de um intervalo de tempo pré-determinado (janela de tempo). Neste trabalho utilizamos as seguintes sete *features* no domínio do tempo: valor mínimo, valor máximo, valor RMS (*root mean square*), valor de curtose, valor de pico, fator de crista e desvio-padrão do sinal de vibração no intervalo de tempo analisado. Essas *features* foram escolhidas baseando-se nos argumentos apresentados em (Haj Mohamad and Nataraj, 2021).
- **Etapa 3** - Normalização dos valores das *features*, para se evitar enviesamento dos resultados de classificação de falhas.
- **Etapa 4** - Determinação das classes de falhas, com rotulagem da classe de falha para cada intervalo de tempo do sinal analisado.
- **Etapa 5** - Escolha de dados para treino e teste dos modelos de MLs.
- **Etapa 6** - Escolha do modelo de ML a ser adotado, configuração dos seus hiperparâmetros e definição da métrica de desempenho.
- **Etapa 7** - Realizar o treinamento e validação do modelo de ML e obter grau de relevância de cada *feature* para a classificação das falhas.
- **Etapa 8** - Remover a *feature* menos e retornar à **Etapa 7**, até que haja apenas uma *feature*.
- **Etapa 9** - Analisar a relação entre desempenho do modelo de ML em função da quantidade de *features*, selecionadas a partir de suas relevâncias.

4. RESULTADOS

Iremos apresentar o *dataset* utilizado nos experimentos, a configuração dos dos hiperparâmetros utilizados para a obtenção dos modelos de árvore de decisão e os resultados alcançados. Os resultados mostram a relação de desempenho dos modelos de ML com relação ao número de *features*, as quais são selecionadas segundo seu grau de relevância. A métrica de desempenho utilizada foi a acurácia.

4.1 Configuração Experimental

Neste trabalho utilizamos o *benchmark* da Case Western Reserve University (CWRU) para falhas de rolamentos em motores elétricos (Neupane and Seok, 2020). Esse *benchmark* disponibiliza dados de dois sensores de vibração localizados no eixo do motor para operação normal e com falhas no rolamento do tipo *inner-race* e *ball*. Na Figura 4 é mostrada uma foto da plataforma experimental do *benchmark* CWRU.

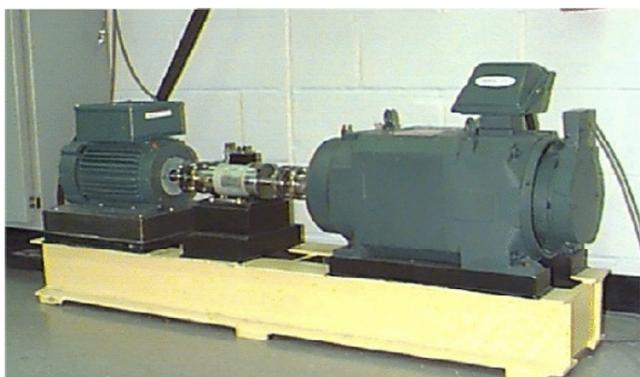


Figura 4. Foto da plataforma geradora do *dataset* do *Benchmark* de falhas em rolamentos de motores elétricos da CWRU.

Neste trabalho utilizamos dados referente ao motor com potência de 1hp e velocidade de 1772 r/min, levando em consideração apenas um sensor de vibração, localizado no final do eixo do motor, e coletando 12.000 amostras/s. Nós utilizamos uma janela com 500 amostras para o cálculo das 7 *features*.

O algoritmo de árvore de decisão utiliza a aprendizagem supervisionada é necessário realizar a normalização dos dados para evitar problema enviesamento do classificador devido a diferença de escalas. A normalização é uma técnica que transforma os valores numéricos em uma escala comum e sem perda de informação que, neste caso, foi transformado os valores das *features* em números binários: 0 (zero), para falha *Inner-race*, e 1 (um) para falha *ball*.

Para o treinamento e teste dos modelos das árvores de decisão, os dados foram divididos em três classes: normal (sem falhas), com falha do tipo *inner-race* ou com falha do tipo *ball*. Foram utilizados 70% dos dados para treino e 30% para teste das árvores de decisão.

Os experimentos foram realizados utilizando-se biblioteca *sklearn* (Pedregosa et al., 2011) da linguagem *Python*. Sendo que para se determinar o grau de relevância de cada *feature* se utilizou a função *features_importance()*,

que retorna em porcentagem o grau de relevância de cada *feature* para o classificação de falhas.

Os hiperparâmetros configurados para geração das árvores de decisão foram o *critério* de Gini para pedir a qualidade (pureza) da árvore obtida e o *splitter*, que é utilizado para elaborar a estratégia de divisão em cada nó. No caso se utilizou a estratégia gulosa de se escolher a melhor divisão dos nós. Para os demais hiperparâmetros, foram utilizados os valores *defaults* do pacote (Pedregosa et al., 2011).

Então foi seguido o procedimento descrito na Seção 3, que consiste em obter a relação entre desempenho do modelo de ML (utilizando a métrica de acurácia) em função da quantidade de *features* utilizada, seguindo a ordem decrescente de relevância das mesmas.

4.2 Resultados Obtidos

Os resultados apresentados a seguir visam demonstrar a pertinência do procedimento apresentado neste trabalho para obtenção de classificador de falhas eficiente. Assim, primeiro são obtidos os graus de relevância de cada uma das sete características para o modelo de classificação de falhas baseado em árvores de decisão, correspondendo à **Etapa 7** do procedimento, então, em seguida é apresentado o resultado obtido da relação entre desempenho do modelo (utilizando acurácia como métrica de desempenho) e o número de características utilizado para obtenção do mesmo. Por fim, são apresentadas as árvores de decisão obtidas utilizando-se sete características e a que obteve o melhor desempenho.

A Tabela 1 contém os valores, em porcentagem, dos grau de relevância das sete características para o processo de classificação de falhas de rolamento do tipo *inner-race* e *ball*. Na qual observamos que as características mais relevantes são respectivamente mínimo, valor de pico, máximo e RMS.

Tabela 1. Relevância das sete características para o processo de classificação de falhas.

<i>Features</i>	Porcentagem (%)
Mínimo	20,33
Valor de Pico	7,45
Máximo	4,22
RMS	1,34
Desvio Padrão	0,01
Curtose	0,01
Fator de Crista	0,01

Na Tabela 2 são apresentados os resultados obtidos ao aplicar a metodologia discutida na Seção 3, nesta tabela temos as colunas do número de características e acurácia obtida. Destacamos em negrito o melhor resultado que foi obtido ao utilizar quatro entradas no modelo de ML, neste resultado conseguimos acurácia de 95%. Também é importante destacar que ao utilizar as três característica menos relevantes (desvio padrão, Curtose, fator de crista) o resultado a métrica de avaliação diminui, ou seja, obtivemos um modelo de classificação de falhas com eficiência e eficácia.

A Figura 5 mostra a árvore de decisão obtida quando utilizou apenas as quatro características mais relevantes

Tabela 2. Acurácia e Quantidade de características para o processo de classificação de falhas.

Quantidade de <i>features</i>	Acurácia (%)
1	0,66
2	0,88
3	0,87
4	0,95
5	0,90
6	0,89
7	0,90

para o processo de classificação de falhas. Enquanto que a Figura 6 apresenta a árvore de decisão obtida quando se utilizou as sete características. Ao analisar estes resultados identificamos que a árvore com todas as características é maior que a árvore apenas com quatro características, logo esse modelo menor também tem o custo computacional menor.

5. CONCLUSÃO

Neste trabalho foi realizada uma análise da influência das variáveis de entradas (características ou *features*) no desempenho modelo baseado em árvore de decisão para classificação de falhas em rolamentos de motores elétricos. O algoritmo de árvore de decisão foi escolhido para fazer a classificação de falhas por sua simplicidade e permitir inerentemente identificar a ordem de relevância das entradas do classificador. Inicialmente utilizamos um total de sete características no domínio do tempo (valores de RMS, pico, mínimo, máximo, desvio-padrão, curtose e fator de crista) do sinal de vibração no eixo do motor para realizar a classificação e identificar a ordem de relevância dessas características e acurácia do modelo. Então, o procedimento adotado foi remover a característica menos relevante para o modelo. Então o modelo de árvore de decisão é retreinado e obtém-se a sua acurácia. Esse procedimento de remoção de características e avaliação do modelo foi realizado até o número de características ser igual a um. Então, ao final do procedimento pode-se escolher a melhor relação entre número de características e acurácia do modelo de classificação de falhas, buscando uma solução eficiente em termos computacionais. Observa-se que era esperado que o aumento do número de características do modelo leva ao aumento da sua acurácia. Porém, no caso particular analisado neste artigo (classificação de falhas em rolamentos de motores elétricos com características no domínio do tempo), observou-se que o aumento do número de características não necessariamente implica no aumento do desempenho do modelo. Para o *dataset* analisado neste artigo, o melhor desempenho do modelo foi obtido com quatro características e um conjunto de oito. Assim, aqui foi possível conciliar eficiência com eficácia do modelo de classificação de falhas.

Como trabalho futuro se planeja investigar a viabilidade de se desenvolver classificadores específicos para cada tipo de falha. No caso, para N falhas teríamos N classificadores binários projetados com características específicas, ou invés de se ter apenas um classificador com N classes.

REFERÊNCIAS

Allamy, H. (2014). Methods to avoid over-fitting and under-fitting in supervised machine learning (compara-

tive study).

- Cerrada, M., Sánchez, R.V., Li, C., Pacheco, F., Cabrera, D., Valente de Oliveira, J., and Vásquez, R.E. (2018). A review on data-driven fault severity assessment in rolling bearings. *Mechanical Systems and Signal Processing*, 99, 169–196. doi:<https://doi.org/10.1016/j.ymssp.2017.06.012>. URL <https://www.sciencedirect.com/science/article/pii/S0888327017303242>.
- Du, S.C., Liu, T., Huang, D.L., and Li, G.L. (2017). An optimal ensemble empirical mode decomposition method for vibration signal decomposition. *Journal of Vibration and Acoustics*, 139(3). doi:10.1115/1.4035480. URL <https://doi.org/10.1115/1.4035480>.
- Haj Mohamad, T. and Nataraj, C. (2021). Fault identification and severity analysis of rolling element bearings using phase space topology. *Journal of Vibration and Control*, 27(3-4), 295–310.
- Honarvar, F. and Martin, H.R. (1997). New statistical moments for diagnostics of rolling element bearings. *Journal of Manufacturing Science and Engineering*, 119(3), 425–432. doi:10.1115/1.2831123. URL <https://doi.org/10.1115/1.2831123>.
- Ismail, M.A.A., Balaban, E., and Windelberg, J. (2022). Spall fault quantification method for flight control electromechanical actuator. *Actuators*, 11(2). doi:10.3390/act11020029. URL <https://www.mdpi.com/2076-0825/11/2/29>.
- Neupane, D. and Seok, J. (2020). Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *IEEE Access*, 8, 93155–93178. doi:10.1109/ACCESS.2020.2990528.
- Niu, X., Zhu, L., and Ding, H. (2005). New statistical moments for the detection of defects in rolling element bearings. *International Journal of Advanced Manufacturing Technology*, 26, 1268–1274. doi:10.1007/s00170-004-2109-4.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Prieto, M.D., Cirrincione, G., Espinosa, A.G., Ortega, J.A., and Henao, H. (2013). Bearing fault detection by a novel condition-monitoring scheme based on statistical-time features and neural networks. *IEEE Transactions on Industrial Electronics*, 60(8), 3398–3407. doi:10.1109/TIE.2012.2219838.
- SAMANTA, B. and AL-BALUSHI, K. (2003). Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mechanical Systems and Signal Processing*, 17(2), 317–328. doi:<https://doi.org/10.1006/mssp.2001.1462>. URL <https://www.sciencedirect.com/science/article/pii/S0888327001914621>.
- Sreejith, B., Verma, A.K., and Srividya, A. (2008). Fault diagnosis of rolling element bearing using time-domain features and neural networks. In *2008 IEEE Region 10 and the Third international Conference on Industrial and Information Systems*, 1–6. doi:10.1109/ICIINFS.2008.4798444.

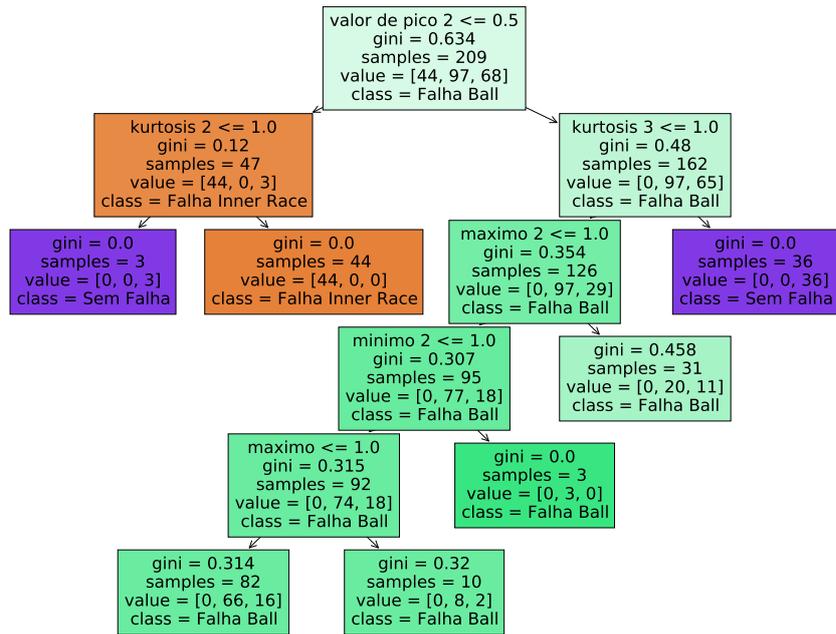


Figura 5. Árvore de Decisão com melhor desempenho (com somente as quatro características relevantes).

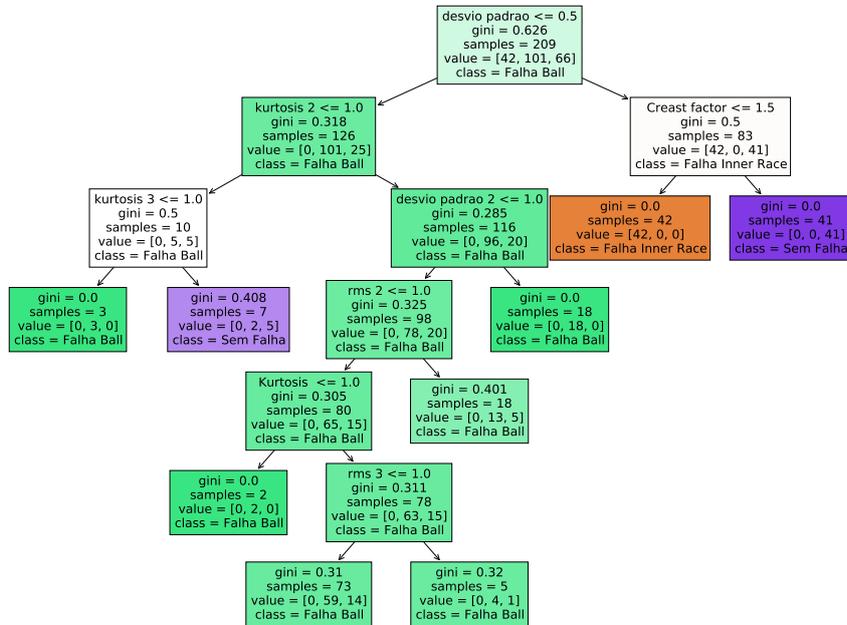


Figura 6. Árvore de Decisão com sete características.