

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
ENGENHARIA ELÉTRICA

RAVI HELON DE MELO SERAFIM FERREIRA

**Sistema de Manutenção Preditiva para Motores Elétricos Utilizando de
Sinais de Vibração e Aprendizado de Máquina**

CAMPINA GRANDE
12 de dezembro de 2019

RAVI HELON DE MELO SERAFIM FERREIRA

Sistema de Manutenção Preditiva para Motores Elétricos Utilizando de Sinais de Vibração e Aprendizado de Máquina

Trabalho de conclusão de curso submetido à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Orientador: Rafael Bezerra Correia Lima

CAMPINA GRANDE

12 de dezembro de 2019

RAVI HELON DE MELO SERAFIM FERREIRA

Sistema de Manutenção Preditiva para Motores Elétricos Utilizando de Sinais de Vibração e Aprendizado de Máquina

Trabalho de conclusão de curso submetido à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

APROVADO EM: 10/12/2019

Rafael Bezerra Correia Lima

Orientador
UFCG

George Acioli Junior

UFCG

CAMPINA GRANDE
12 de dezembro de 2019

Agradecimientos

“I’ve worked all kinds jobs in all kinds of places. Never met a good person, or a bad one. It’s just folks, friend. Everyone’s just folks.”
(Nakak, from Warframe)

Resumo

A IIoT é um dos paradigmas mais modernos do setor e tem a manutenção preditiva como um dos seus princípios fundamentais e, portanto, este artigo apresenta uma implementação prática de todo o processo, desde o hardware de aquisição de dados até o relatório final que pode ser enviado para qualquer trabalhador em campo notificando sobre a condição do equipamento monitorado.

Palavras-chaves: IIoT, Manutenção Preditiva, Aprendizado de Máquina, Computação em Nuvem

Abstract

IIoT is one of the most modern paradigms of the industry and has predictive maintenance as one of its core principles and, thus, this paper presents an practical implementation of the entire process, from the data acquisition hardware to the final report that can be sent to any worker on the field notifying about the condition of the monitored equipment.

Key-words: IIoT, Predictive Maintenance, Machine Learning, Cloud Computing

Lista de abreviaturas e siglas

SVM Máquina de Vetor de Suporte

IoT Internet das Coisas

IIoT Internet das Coisas Industriais

ANN Redes Neurais Artificiais

Lista de símbolos

Σ	Somatório
∇	Gradiente
$F(x) _{x=a}$	Função de x com x avaliado no ponto a
\vec{X}	Vetor X
\hat{y}	Valor estimado de y
\tanh	Tangente hiperbólica

Sumário

1	Introdução	10
1.1	Objetivo Geral	10
1.2	Metodologia	11
1.3	Referencial Teórico	11
2	Aprendizado de Máquina	13
2.1	Redes Neurais Artificiais	13
2.2	Árvores de Decisão	16
2.2.1	Floresta de Decisão	17
2.2.2	Gradient Boost	18
2.3	Máquina de Vetor de Suporte	18
3	Manutenção Preditiva e Análise de Vibração	22
3.1	Estratégias de Manutenção	22
3.2	Indicadores para Monitoramento de Condições	23
3.3	Sinais de Vibração Emitidos por Máquinas Girantes	24
3.3.1	Desbalanceamento	24
3.3.2	Desalinhamento	25
3.3.3	Rolamentos	27
4	Materiais e Métodos	28
4.1	Desbalanceamento	28
4.2	Desalinhamento	29
4.3	Defeitos em Rolamentos	29
4.4	Seleção de Parâmetros	30
4.5	Metodologia de Análise	30
4.6	Relatório Final e PowerBI	31
5	Resultados	32
6	Conclusão	34
	Referências	35
	ANEXO A	39

1 Introdução

IoT é um conceito relativamente novo que está crescendo em popularidade, sendo o próprio termo cunhado apenas em 1999 por [1] com um significado relacionado à capacidade dos dispositivos, coisas, para conectar, comunicar e trabalhar juntos sem a necessidade de uma constante supervisão humana [2]. Assuntos relacionados essenciais são, entre outros, aprendizado de máquina, computação em nuvem, sistemas embarcados, alguns dos quais já existiam, mas ainda assim são muito importantes [3].

Este conceito aplicado à indústria recebe o apelido IIoT e tem manutenção preditiva enquanto um dos seus principais focos uma vez que qualquer equipamento, com o tempo, tende a perder eficiência, se desgaste ou até quebre, e é do interesse de qualquer industrial conhecer antecipadamente e evitar mais danos à linha de produção [4]. A manutenção preditiva consiste na abordagem desses problema, sendo seu objetivo detectar e corrigir falhas antes que qualquer efeito danoso se realize [5].

É neste contexto que o presente trabalho foi proposto, ao estudar algumas das ferramentas disponíveis relacionadas ao tema, na esperança de auxiliar o desenvolvimento desse campo, com foco na análise de algumas das falhas mais comuns que ocorrem em máquinas elétricas, amplamente utilizadas nas indústrias [6], usando dados de vibração. Sendo assim, os primeiros passos previstos são fazer com que o diagnóstico possa ser feito pelo equipamento, usando aprendizado de máquina, e no campo, usando computação em nuvem.

1.1 Objetivo Geral

O objetivo do trabalho desenvolvido é então o desenvolvimento de um software capaz de identificar problemas de balanceamento, desalinhamento e rolamentos danificados remotamente por meio de sinais de vibração analisados em se utilizando de métodos de aprendizado de máquina.

Para facilitar a progressão foram definidas algumas metas intermediárias, enumeradas abaixo, de forma a garantir um progresso gradual com tarefas bem especificadas, sem sobrecarga.

1. Aquisição de dados;
2. Estudos referentes à manutenção preditiva e análise de vibração;
3. Elaboração de código capaz de discriminar as características que permitem a análise;
4. Estudos referentes ao métodos de aprendizado de máquina adequados ao problema;

5. Escolha de plataforma em nuvem com suporte para as ferramentas que se deseja utilizar;
6. Treinamento do modelo de aprendizado de máquina;
7. Composição do sistema de análise completo capaz de realizar diagnóstico;
8. Interface humano-máquina por meio do PowerBI;
9. Automatização completa do processo;
10. Documentação.

1.2 Metodologia

Dito isso na próxima Seção serão descritos os trabalhos anteriores na área e que foram de grande influência para a execução deste. Nos Capítulos 2 e 3 são descritos, respectivamente, os principais métodos de aprendizado de máquina e alguns conceitos de monitoramento preditivo utilizando sinais de vibração para motores elétricos. Por fim o Capítulo 4 descreve a realização dos experimentos, no que diz respeito aos materiais e métodos utilizados, no Capítulo 5 os resultados obtidos e no Capítulo 6 o que se pode concluir dos resultados assim como propostas para expansão do trabalho realizado.

1.3 Referencial Teórico

Existem alguns estudos que analisam a aplicação do monitoramento preditivo na análise de vibrações de máquinas elétricas, como [7] e [8], e até mesmo alguns usando algoritmos de aprendizado de máquina, como visto em [9], [10], [11] e [12].

Também existem artigos que estudam o uso de algoritmos de aprendizado de máquina em ambientes de computação em nuvem e IoT, como [13], [14] e [15] assim como estudos sobre soluções de IoT para análise de vibração como [16].

Contudo, não foi encontrado um artigo sobre a análise de dados de vibração para o processamento desses dados por uma ferramenta de aprendizado de máquina na nuvem. Ainda assim existem alguns trabalhos que foram extremamente úteis na execução do presente trabalho.

No que diz respeito à análise de vibração foi utilizado como embasamento principalmente os trabalhos de [17], que se propõe a estudar a fundo os desgastes nos rolamentos, um dos problemas mais comuns em máquina giratórias, e [8], que se propõe a fazer um estudo detalhado sobre a aplicabilidade da manutenção preditiva em máquinas assíncronas, assim como o livro [18], que é inteiramente focado e é um dos materiais mais completos e bem fundamentados no assunto de manutenção preditiva por meio de sinais de vibração.

Quanto a aprendizado de máquina alguns dos trabalhos mais influentes foram o livro [19], que descreve a aplicação e bases para diferentes métodos de aprendizado de máquina aplicados em python, os tutoriais disponíveis pela microsoft, como [20], e alguns artigos referentes a métodos mais específicos, tais como [21].

A inspiração do trabalho e o trabalho sobre o qual se esperava obter uma melhora foram [22] e [23], que descrevem a implementação de um software com interface facilitadora para que técnicos ou engenheiros responsáveis em campo pudessem realizar o diagnóstico preditivo.

2 Aprendizado de Máquina

Aprendizado de máquina é um assunto que vem sendo amplamente discutido em várias mídias, oferecendo por vezes perspectivas que podem parecer tão irreais que escapam à capacidade de serem concebidas. Desde coisas que são apenas impressionantes, como prever, a partir de sinais de vibração, a condição de um motor elétrico até coisas que pensávamos serem exclusividades da espécie humana, como argumentar[24], mentir[25],[26] e contar histórias originais[27][28].

Uma das primeiras definições para aprendizado de máquina foi dada por Arthur Samuel na qual é dito que é o "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados"[29], contudo essa definição é muito generalista e ajuda pouco no que diz respeito ao entendimento e das técnicas que se utilizam e portanto sejamos mais precisos. Outra definição foi cunhada por Tom M. Michell na qual "diz-se que um programa de computador aprende pela experiência E, com respeito a algum tipo de tarefa T e performance P, se sua performance P nas tarefas em T, na forma medida por P, melhoram com a experiência E"[30].

Pode-se dizer que se tratam de métodos capazes de transcender aquilo que foi explicitamente programado e exercer funções mais complexas. Alguns dos principais métodos que foram aplicados no decorrer deste trabalho foram as redes neurais, as florestas de decisão e as máquinas de vetor de suporte, que serão explicadas a seguir.

2.1 Redes Neurais Artificiais

As ANN foram introduzidas em 1943 por Warren McCulloch e Walter Pitts[31], um neurofisiologista e um matemático, respectivamente. Obteve relativo sucesso até os anos 60 o que levou à crença de que estaríamos lidando com máquinas realmente inteligentes em um futuro próximo[32].

Porém, devido à capacidade computacional da época, isso se mostrou impraticável naquele primeiro momento e os investimentos não se mantiveram o que as levou a cerca de duas décadas no esquecimento. Nos anos 80 o interesse foi renovado, devido a novas técnicas para se realizar o treino, mas caiu novamente devido ao advento das Máquinas de Vetor de Suporte, que pareciam mais precisas e ter bases teóricas mais sólidas[32].

Recentemente voltaram a crescer em popularidade devido a alguns fatores tais quais a disponibilidade de grande quantidade de dados, computadores com capacidade de processamento muito maior, outras melhorias nos algoritmos de treino e o fato de que algumas das limitações teóricas se mostraram inofensivas na prática[32].

As Redes Neurais são muito versáteis e podem ser utilizadas nas mais variadas aplicações, mas pode-se descrever o problema a ser resolvido de forma a, dadas variáveis de saída, discretas ou contínuas y , categóricas ou não, obter parâmetros x que sejam capazes de minimizar, por meio de uma relação $F(\mathbf{x})$, uma função custo $L(y, F(x))$.

A inspiração do método é no modelo de transmissão de informação no cérebro humano, daí a denominação de Redes **Neurais**. Onde as entradas podem ser visto como os estímulos externos, as quais seriam repassadas a neurônios que enviam informação de como o corpo deve reagir.

Para facilitar o entendimento é comum o emprego de diagramas tais quais o presente na Figura 2.1. A primeira camada, com circunferências amarelas, são as entradas, X_1, X_2, X_m , conectados à seguinte por meio das ponderações, W_1, W_2, W_m . Em verde tem-se a denominada camada oculta e pode ser composta por mais de um nós, ou neurônio, os quais atuam na combinação de suas entradas de acordo com a função de ativação ϕ . Por fim \hat{y} é a saída estimada e y a saída real, se disponível.

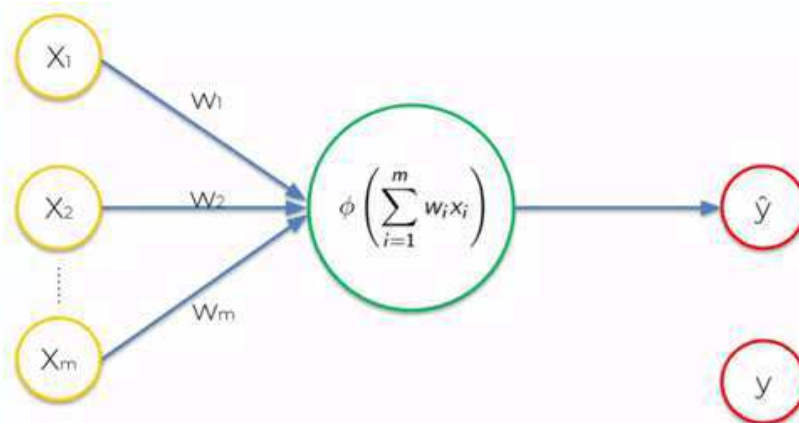


Figura 2.1 – Diagrama Representativo de uma ANN. Fonte: Adaptado de [33]

O funcionamento do método é dividido em três etapas, Treino, Validação e Implantação.

Na etapa de treino objetiva-se determinar primeiramente hiperparâmetros, como quantidade de camadas ocultas, nós por camada oculta, funções de ativação, dentre outros e, posteriormente, os valores dos pesos. O resultado final deve ser um modelo capaz de aproximar a saída por meio das entradas.

Essa etapa se baseia em uma publicação de Donald Hebb na qual é proposto um modelo para o neurônio biológico em que a conexão entre neurônios fica mais forte quando é ativada repetidas vezes. Sendo assim esse treinamento se trata de uma mímica desse processo ao ajustar os pesos iterativamente de forma às interações entre nós que levaram a respostas corretas serem reforçadas em contraponto às que levaram a respostas errôneas.

Este foi por anos um gargalo para o campo até que foi desenvolvido o método do gradiente descendente [34], relevante até hoje, no qual a atualização a cada iteração é baseada no gradiente, ponderado pela taxa de aprendizado η , da função custo C .

Sabe-se que, para uma função $C(\vec{x})$ diferenciável ao redor de um ponto A , a direção para sua mais rápida diminuição é a do negativo do gradiente naquele ponto, ou $-\nabla C(\vec{x})|_{\vec{x}=A}$. Portanto pode-se escrever a seguinte expressão para atualização dos valores de peso a cada iteração t :

$$w_i^{(t)} = w_i^{(t-1)} - \eta^{(t)} \nabla C(\hat{y} = F(\vec{x}), y) \quad (2.1)$$

Cabe aqui explicitar-se a importância das funções de ativação, as quais representam o processamento da informação pelo cérebro e determinação de o que será repassado às camadas seguintes. São por meio destas que há adição das não linearidades que diferenciam as redes neurais de regressões lineares. Normalmente são escolhidas funções sigmóides ou ReLu cujas saídas estão sempre entre 0 e 1.

Quando uma rede neural como a representada pelo diagrama da Figura 2.1 contém apenas um neurônio com função de ativação do tipo limiar ela recebe o nome de Perceptron. Este por sua vez é um classificador simples e, em geral, não é utilizado por si só.

Voted Perceptron é uma combinação de vários perceptrons onde se pondera a saída de cada um e se realiza uma votação de forma a determinar a saída predita pelo modelo completo. A equação para a predição é descrita da seguinte forma:

$$\hat{y} = \text{sign}\left(\sum_{m=1}^M \text{sign}(c_m(w_m \vec{x}_m + b_m))\right) \quad (2.2)$$

Uma alternativa mais prática é a utilização do *Averaged Perceptron*, o qual faz a ponderação nos próprios perceptrons e somente depois é aplicada a função de ativação, reduzindo a equação em 2.2 a:

$$\hat{y} = \text{sign}\left(\sum_{m=1}^M \text{sign}(c_m(w_m \vec{x}_m + b_m))\right) \quad (2.3)$$

Configurações com maior quantidade de nós e de camadas podem ser chamadas de perceptron multicamadas ou apenas rede neural e, se houver um número significativo de camadas ocultas, pode-se denominar de rede neural de aprendizado profundo, contudo os números específicos dependem da aplicação.

Após o treino o modelo obtido deve passar por uma avaliação em se utilizando dados de entrada diferentes daqueles usados para o treino, com saídas conhecidas, para verificar os

resultados são satisfatórios para a aplicação em questão. Se não forem deve-se repetir a etapa de treino, se forem pode-se partir para a implantação.

2.2 Árvores de Decisão

As árvores de decisão são uma ferramenta muito útil para a análise de decisão [35], que é um fluxograma para que a decisão final, que é referenciada como folhas da árvore, possa ser alcançada pela análise das decisões explicitada por cada nó. Segue um exemplo da Figura 2.2 de uma delas sendo aplicada à recomendação de um livro. Observe que a resposta à questão dos nós leva à escolha de um ramo e a partir daí é atingida a decisão final ou outra escolha deve ser feita [21].

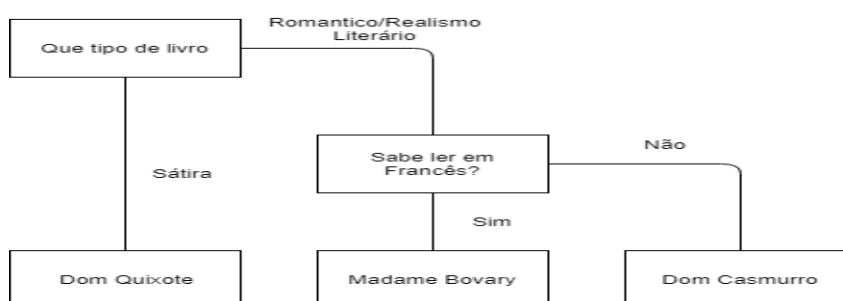


Figura 2.2 – Exemplo de Árvore de Decisão. Fonte: Autoria Própria

Na ciência da computação, esse conceito é estendido para que, com base nas entradas e no treinamento apropriado, se possa extrair conclusões.

Este se trata de um algoritmo guloso, significando que buscasse tomar a melhor decisão a cada iteração e uma decisão é dita melhor de que outra dependendo de parâmetro avaliado. Os parâmetros mais comuns são pela taxa de Impureza de Gini ou pelo ganho de informação, ou Entropia.

O primeiro é definido como a taxa de classificações errôneas para um elemento qualquer se o mesmo fosse classificado aleatoriamente de acordo com a distribuição do subconjunto pós-decisão, o que equivale matematicamente a:

$$I_G = \sum_{\forall i} p_i * (1 - p_i) = 1 - \sum_{\forall i} p_i^2 \quad (2.4)$$

Onde p_i é a probabilidade de ser escolhido a classificação i .

Para o segundo, em teoria da informação, a informação é dada por:

$$H(T) = I_E(p_1, p_2, \dots, p_J) = - \sum_{i=1}^J p_i \log_2 p_i \quad (2.5)$$

Portanto o ganho de informação é dado por:

$$I_r = H(T) - \sum_{\forall a} H(T|a) \quad (2.6)$$

Em geral o resultado da utilização dessas duas técnicas é equivalente a Impureza de Gini demanda menos da máquina e é normalmente utilizado para minimizar os erros de classificação e a Entropia por sua vez é mais utilizado em mineração de dados, para obter informações previamente desconhecidas sobre o conjunto de dados.

Dentre as árvores de decisão se destacam alguns algoritmos a exemplo do ID3 e do CART. No ID3 será escolhida aleatoriamente uma janela da base de dados para a qual será gerada uma árvore de decisão que será utilizada para avaliar o resto da base de dados. Os elementos que forem classificados erroneamente, se existirem, são adicionadas à janela para a próxima iteração. O processo se repete até que não ocorram mais classificações erradas. Já no CART é gerada uma árvore grande que é então podada de forma a encontrar menores ótimas.

Apesar de ser um método de fácil implementação, não é um método muito utilizado por si só devido à relativa falta de robustez, porém algumas técnicas podem ser utilizadas para melhorar os resultados obtidos, a exemplo da Agregação de Bootstrap, ou Bagging, e do Aumento gradiente.

2.2.1 Floresta de Decisão

A aplicação da técnica de Bagging a uma árvore de decisão resulta no método conhecido como Floresta de Decisão, já que se utiliza de diversas árvores.

Suponhamos então um preditor $\psi(x, L)$, onde x são as variáveis de entrada e L o conjunto de treino da forma $\{(x_n, y_n); n = 1, \dots, N\}$. Supondo ainda a disponibilidade de um número K de conjuntos de treino L_k .

Pode-se melhorar os preditores separados ao agregar seus resultados dos K preditores $\psi(x, L_k)$ por meio de uma média ou votação dependendo da natureza da saída [36].

Contudo, em geral, não se dispõe de multiplas bases de dados e é aí que é introduzida a técnica de geração de amostras *bootstrap* pela geração de repetidos $L_k^{(B)}$ onde cada um é formado pela escolha aleatória com reposição de elementos em L . A utilização da técnica de agregação nesses conjuntos de amostras é chamado de agregação *bootstrap* ou *bagging*.

Por ser aleatório com reposição os conjuntos de amostras novos provavelmente conterão elementos repetidos enquanto outros não aparecerão.

2.2.2 Gradient Boost

Uma maneira de atualizar a árvore de decisão é pelo que é chamado de *Boosting*, que por sua vez pode ser feito de várias maneiras.

O *Boosting* por si só é uma abordagem meta-heurística na qual se propõe encontrar um forte preditor combinando uma infinidade de indicadores mais fracos treinados no mesmo conjunto de dados [37]. Uma de suas variantes, e a utilizada neste trabalho, é o chamado *Gradient Boosting*, pois emprega uma técnica de gradiente descendente, semelhante à usada em outros métodos como redes neurais, para melhorar iterativamente a previsão geral.

Sendo y a saída real do sistema e $F_m(\vec{x})$ a estimativa gerada pelo modelo em sua m -ésima iteração pode-se escrever para algoritmos de *boosting*:

$$F_{m+1}(x) = F_m(x) + h(x) \quad (2.7)$$

Onde a cada iteração se tenta corrigir o erro da iteração anterior fazendo $h(x) = y - F_m(x)$.

Pode-se buscar a melhoria nessas estimativas utilizando-se de uma função custo, semelhante à abordagem por gradiente descendente. É comum a escolha do erro quadrático, $\frac{1}{2}(y - F_m(x))^2$, mas pode-se utilizar outras funções.

2.3 Máquina de Vetor de Suporte

As Máquinas de Vetor de Suporte foram originalmente concebidas por [38]. Em 1992 é proposto em [39] a aplicação de kerneis a hiperplanos para problemas não lineares e mais recentemente o padrão mais utilizado atualmente foi o proposto em [40]. O método consiste na tentativa de escolher a melhor reta a ser utilizada como classificador ótimo, separando pontos de diferentes classes.

Na representação da Figura 2.3 pode-se distinguir 3 classificadores, contudo apenas um pode ser denominado ótimo e para todos os efeitos neste método será considerado a melhor aquela reta que estiver à maior distância relativa aos pontos mais próximos de cada uma das classificações possíveis.

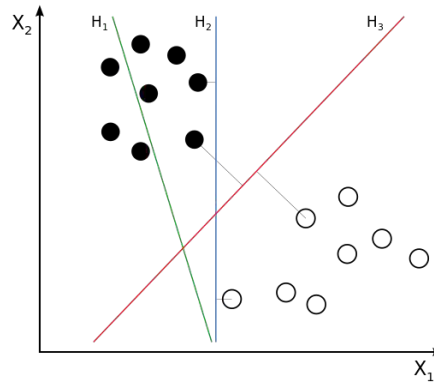


Figura 2.3 – Possíveis retas classificadoras

Escolhendo um determinado classificador genérico tal que \vec{w} é um vetor de suporte perpendicular ao mesmo, \vec{I} o ponto a ser analisado e b o coeficiente linear da reta a seguinte Lei de Decisão pode ser escolhida para distinguir os pontos pretos:

$$\vec{w} \cdot \vec{I} - b \geq 0 \quad (2.8)$$

Esta solução pode ser melhor visualizada na Figura 2.4. Importante mencionar que apesar de estar sendo exemplificado em duas dimensões a solução pode ser facilmente extrapolada para dimensões superiores.

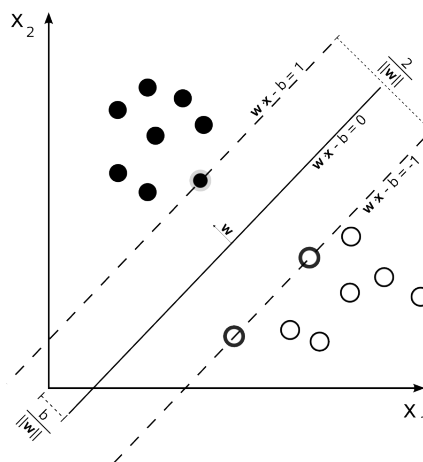


Figura 2.4 – Classificador ideal

Contudo na equação 2.8 a única variável conhecida é o dado analisado, \vec{I} , levando à necessidade de obtenção das características da reta, \vec{w} e b , por meio do treino.

Para encontrar os parâmetros de dita reta aplica-se uma lei de decisão mais restritiva para os pontos na base de dados de treino, a reta deve então ser tal que os pontos de grupos diferentes fiquem na maior distância possível desta.

Esta nova lei está presente na equação 2.10 onde \vec{x}_p e \vec{x}_b são os pontos sabidamente pretos e brancos, respectivamente.

$$\vec{w} \cdot \vec{x}_p - b \geq 1 \quad (2.9)$$

$$\vec{w} \cdot \vec{x}_b - b \leq -1 \quad (2.10)$$

Alternativamente, após certo algebrismo, com a introdução da variável y para facilitar futuro equacionamento ao evitar o trabalho com multiplas funções simultaneamente.

$$y(\vec{w} \cdot \vec{x}_i - b) \geq 1 \quad (2.11)$$

$$y = \begin{cases} 1, & X = X_p \\ -1, & X = X_b \end{cases} \quad (2.12)$$

Deseja-se então determinar a maior distância D entre os pontos dos dois diferentes grupos e a reta a ser escolhida. Esta distância que separa as regiões das diferentes classes pode ser dada por

$$D = (\vec{x}_p - \vec{x}_b) \cdot \frac{\vec{w}}{\|\vec{w}\|} \quad (2.13)$$

$$D = \frac{\vec{x}_p \cdot \vec{w} - \vec{x}_b \cdot \vec{w}}{\|\vec{w}\|} \quad (2.14)$$

A equação da condição, 2.11, para os pontos na borda, resulta em

$$\vec{w} \cdot \vec{x}_i = b + y \quad (2.15)$$

Substituindo em 2.14 obtem-se a equação final para a distância em relação à norma do vetor \vec{w}

$$D = \frac{2}{\|\vec{w}\|} \quad (2.16)$$

Obter o máximo D equivale portanto a minimizar $\|\vec{w}\|$. Diversas funções custo podem ser utilizadas para resolver este problema, mas a mais comum é $\frac{1}{2}\|\vec{w}\|^2$. Também é importante levar em consideração a restrição, $y(\vec{w} \cdot \vec{x}_i - b) = 1$, que nos leva a um problema de minimização sob a restrição de uma função.

A resolução deste gênero de problema é normalmente por meio da técnica dos multiplicadores de Lagrange que oferece uma nova função a ser minimizada, mas sem restrições, dada por:

$$L = f(\vec{v}) - \sum_{\forall k} \lambda_k g_k(\vec{v}) \quad (2.17)$$

Onde $f(\vec{v})$ é a função que deseja-se minimizar, $\vec{v} = [\|\vec{w}\| \ b]$, $g_k(\vec{v})$ as restrições e λ_k os multiplicadores de Lagrange

Ao aplicar o gradiente para minimizar a equação em 2.17 obtém-se as seguintes equações, A primeira sendo um resultado muito importante, significando que pode-se determinar o vetor \vec{w} por meio de soma ponderada dos dados de treino:

$$\vec{w} = \sum_{\forall i} \lambda_i y_i \vec{x}_i \quad (2.18)$$

$$0 = \sum_{\forall i} \lambda_i y_i \quad (2.19)$$

Ao substituir as equações 2.18 e 2.19 nas equações 2.17 e 2.8 se obtém as expressões finais, do valor a ser minimizado no treinamento e da forma de classificar pontos, respectivamente.

$$L = \sum_{\forall i} \lambda_i - \frac{1}{2} \sum_{\forall i} \sum_{\forall j} \lambda_j \lambda_i y_j y_i (\vec{x}_j \cdot \vec{x}_i) \sum_{\forall i} \lambda_i y_i \vec{x}_i \cdot \vec{I} - b \geq 0 \implies \text{Ponto preto} \quad (2.20)$$

$$(2.21)$$

Algumas características importantes são as de que para grupos linearmente inseparáveis há a possibilidade de utilização de diferentes Kerneis, que substituem transformações $\phi(\vec{x})$ entre espaços vetoriais de forma a não ser necessário conhecê-las, bastando conhecer os produtos, definidos pelos Kerneis, $K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$ e com isso há a garantia de que não haverá problemas com mínimos locais na hora do treino.

3 Manutenção Preditiva e Análise de Vibração

Uma das áreas mais negligenciadas na indústria é a manutenção, como dito em [6] em 2012 menos de 60% das indústrias se davam ao trabalho de implementar técnicas de manutenção preditiva, em alguns casos mesmo tendo a tecnologia disponível, contudo isto pode estar mudando devido à recente popularização de conceitos tais como IoT.

Com novas tecnologias e o reconhecimento de que pode-se melhorar significativamente a eficiência nas indústrias espera-se que esta área seja um dos focos de investimento no futuro próximo e neste Capítulo serão explorados alguns dos seus paradigmas, principalmente associados àquela que é uma das mais difundidas, a manutenção preditiva por meio de sinais de vibração.

No diagrama presente na Figura 3.1 estão explicitadas as principais etapas referentes à análise preditiva de condições.

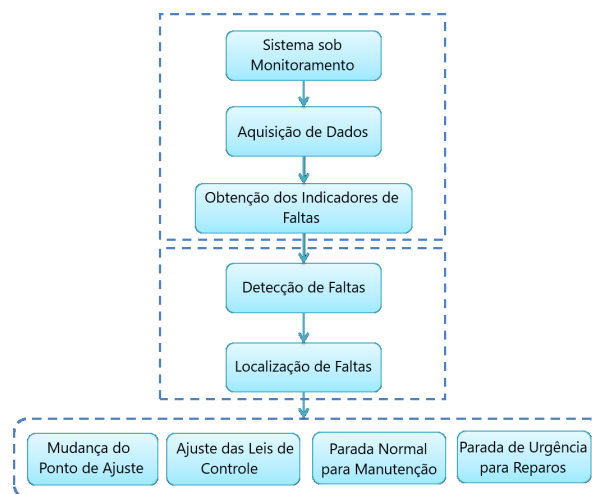


Figura 3.1 – Etapas do Diagnóstico de um Sistema. Fonte: Adaptado de [41]

Os focos das atividades deste projeto é nas atividades do segundo bloco, detecção e localização de faltas.

3.1 Estratégias de Manutenção

Antes de tudo existem diferentes estratégias para realização da manutenção dos equipamentos de dada indústria.

A manutenção corretiva, conhecida também como *run-to-break* é uma das mais comuns, já que não exige nenhum trabalho extra. Apesar de garantir o máximo de tempo entre manuten-

ções só é recomendável para equipamentos motores de baixo custo e de baixa relevância para o processo produtivo como um todo, uma vez que o equipamento terá de ser substituído.

Outra técnica relativamente comum é a manutenção corretiva, ou preventiva sistemática. Nesta técnica são estabelecidos intervalos regulares para a inspeção das máquinas. Apesar de ser menos custoso, em termos de compra de novos equipamentos, há excesso na quantidade de manutenções que devem ser realizadas e não há garantias de que o equipamento não quebrará entre as manutenções.

Isto pode levar o responsável com um dilema, manutenção mais frequentes podem garantir que o equipamento não quebrará entre verificações, mas aumenta o número de manutenções desnecessárias.

A última estratégia é a manutenção preditiva que se utiliza de modelos para a máquina de forma a determinar o tempo de vida útil restante sendo capaz de determinar o tempo ideal entre manutenções de forma a torná-las menos custosas, contudo faz-se necessário profissionais preparados para utilizar tais tecnologias e a utilização de modelos eficientes para a detecção dos defeitos.

A manutenção preditiva ainda pode ser classificada entre intermitente e permanente, onde a primeira apesar de mais barata e permitir a utilização de parâmetros e métodos mais robustos é incapaz de alertar para problemas repentinos.

Já a manutenção permanente não pode utilizar de métodos tão robustos por ter que realizar análises em tempo real e sem grandes capacidades de armazenamento e pode ter custos adicionais para implementação já que poderá comprometer a integridade dos equipamentos analisados e haverá a necessidade da implantação de sensores fixos, porém há a possibilidade de identificar problemas a qualquer momento.

3.2 Indicadores para Monitoramento de Condições

Diferentes indicadores podem ser utilizados para a manutenção preditiva em motores elétricos, algumas das mais conhecidas são utilizando-se de dados de vibração, de termografia e do líquido lubrificante.

Uma análise comparativa das vantagens associadas a cada uma dessas abordagens pode ser visualizada na Tabela 1.

	Principais Vantagens	Principais Limitações	Campos de Aplicação
Análise de Vibrações	<ul style="list-style-type: none"> - Detecção de Falhas em estados precoces - Possibilidade de Realizar Diagnóstico Aprofundado - Permite Fiscalização Contínua e a Distância 	<ul style="list-style-type: none"> - Espectro Muitas vezes Difícil de se analisar - No caso de Fiscalização a Distância Pode Ser Muito Caro 	<ul style="list-style-type: none"> - Detecção de Falhas em todas as partes cinemáticas da máquina e de sua estrutura.
Análise do Lubrificante	<ul style="list-style-type: none"> - Detecção de Poluentes no Lubrificante Antes de Causarem Danos - Possibilidade de Identificar a Origem da Anomalia por Análise das Partículas 	<ul style="list-style-type: none"> - Não Permite Localização Precisa de Falhas - Necessidade de precauções na Coleta dos Fluidos 	<ul style="list-style-type: none"> - Controle de Propriedades Físico-Químicas do Lubrificante - Detecção de Falta de Lubrificante
Termografia	<ul style="list-style-type: none"> - Permite Realizar o Controle Rápido da Instalação - Interpretação frequentemente Instantânea dos Resultados 	<ul style="list-style-type: none"> - Detecção de Falhas em Estados Menos Precoces de que Análise de Vibrações - Controle Limitado Àquilo que está à "Vista" da câmera (Aquecimento da Superfície) - Não permite Diagnóstico Aprofundado 	<ul style="list-style-type: none"> - Detecção de Todos as Falhas que Causadoras de Aquecimento (Falta de Lubrificante Especialmente)

Tabela 1 – Comparativo Qualitativo entre Técnicas de Monitoramento Preditivo. Fonte: Adaptado de [17]

Apesar de ter sido escolhido para estudo neste trabalho a análise por vibrações, por ser mais bem difundido e ser mais completa na quantidade de problemas que podem ser identificados é importante ter em mente que esses métodos não são mutuamente excludentes e podem ser empregados simultaneamente.

3.3 Sinais de Vibração Emitidos por Máquinas Girantes

A análise por vibrações se baseia no fato de que uma máquina elétrica, em seus movimentos rotóricos periódicos, naturalmente emite sinais de vibração, devido a vários fatores de fábrica associados às perdas de energia.

Quando o equipamento está com defeito, por sua vez, esses sinais de vibração vão sofrer influência também do problema em questão, que amplificará as vibrações de formas que diferem de acordo com o problema em mãos.

Neste trabalho foram estudados principalmente os sinais associados aos defeitos de desbalanceamento, desalinhamento e desgaste nos rolamentos. Cada qual com suas próprias características.

3.3.1 Desbalanceamento

O primeiro problema avaliado é o desbalanceamento. Este problema é associado à irregularidade no que diz respeito à distribuição de massa no eixo da máquina girante.

Uma ilustração do problema está apresentada na Figura 3.2, onde estão representadas diferentes seções do rotor uma sem problemas e duas outras onde o momento de inércia não coincide com o do eixo, um por diferença radial e outra formando um ângulo.

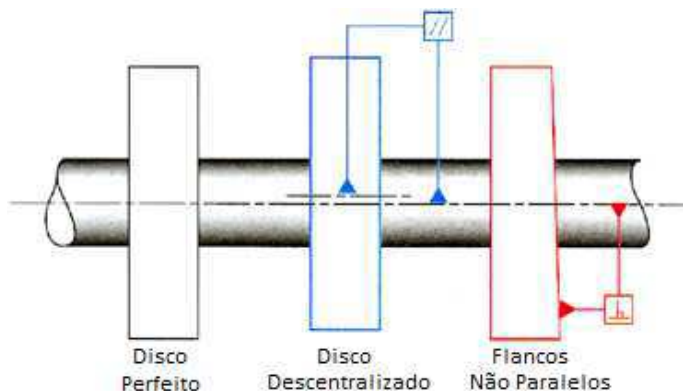


Figura 3.2 – Ilustração da Falta de Balanceamento em Motores. Fonte: Adaptado de [17]

Em ambos os casos sabe-se que os efeitos podem ser verificados ao se comparar a frequência fundamental com a de uma máquina semelhante, como pode ser visto na Figura 3.3.

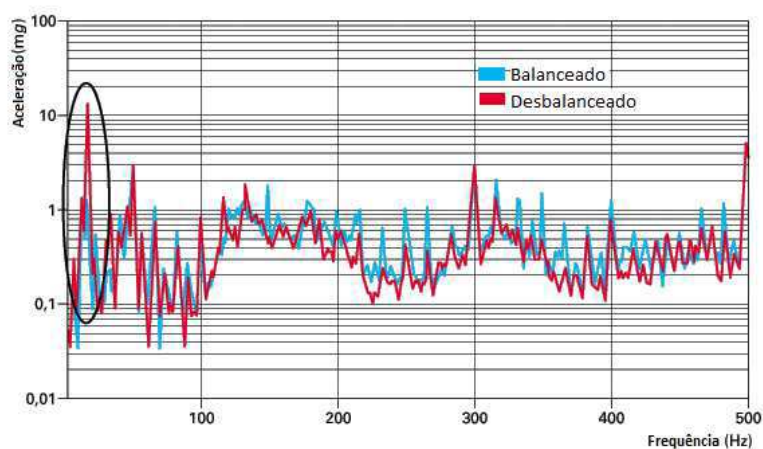


Figura 3.3 – Ilustração das Respostas em Frequência de máquina com e sem desbalanceamento. Fonte: Adaptado de [17]

Este tipo de defeito normalmente ocorre por razões mecânicas, com o desgaste do eixo, atritos ou colisões, ou por razões térmicas, quando há dilatação não uniformes no eixo do rotor[18].

3.3.2 Desalinhamento

O desalinhamento, por sua vez, ocorre quando, em qualquer situação em que há dois eixos conectados, o momento de inércia pelo qual eles estão girando não é o mesmo.

Existem dois tipos fundamentais de desalinhamento sendo angulares quando existe um ângulo diferente de 0 entre o momento de inércia e paralelo quando existe uma distância radial. Ambos são exemplificados na Figura 3.4.

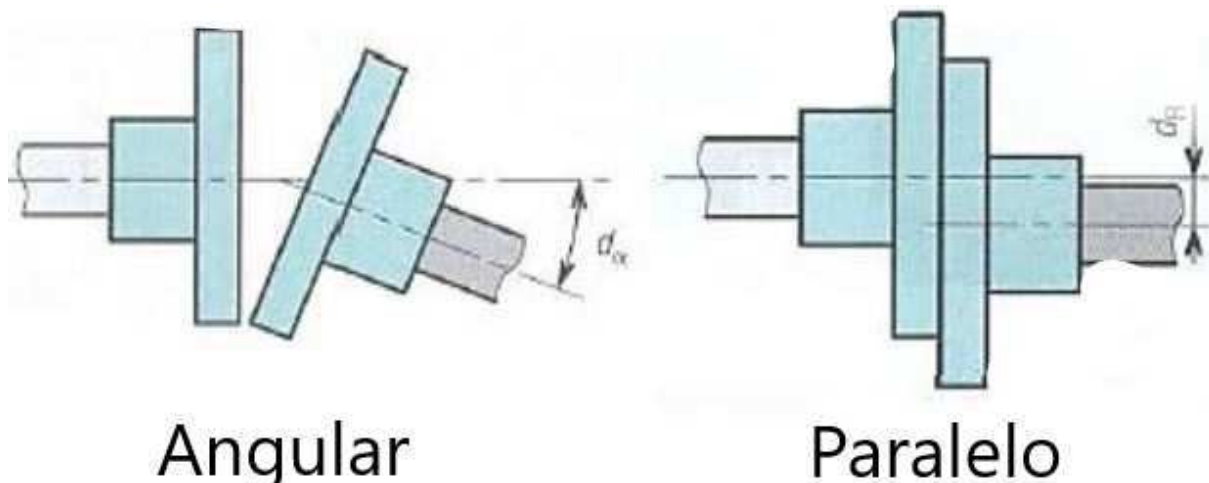


Figura 3.4 – Representação de Diferentes Desalinhamentos. Fonte: Adaptado de [8]

Os efeitos desse tipo de falha também são em harmônicas de baixa ordem, mas neste caso o efeito é mais forte na Segunda. Como pode ser visto nos gráficos comparativos da Figura 3.5

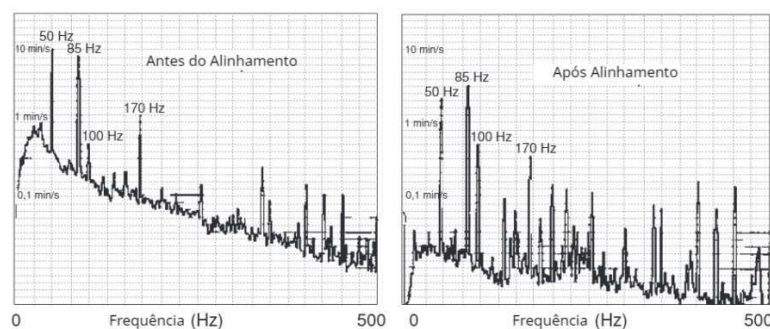


Figura 3.5 – Ilustração das Respostas em Frequência de máquina com e sem desalinhamento. Fonte: Adaptado de [18]

Nesses casos as causas também podem ser mecânicas ou térmicas. Mecânicas no sentido de poder ter sido instalado a estrutura com alguma folga ou o vibrar ao longo do tempo tenha gerado tal folga, de forma a desalinhar algumas das estruturas fixas, como os rolamentos ou diferentes máquinas. A causa térmica é mais uma vez devido à dilatação, mas dessa vez afetando áreas que deveriam ser fixas[18].

3.3.3 Rolamentos

Por fim um dos problemas mais recorrentes em motores elétricos são as faltas devido ao desgaste dos rolamentos.

Uma estrutura típica de rolamento é como a que pode ser vista na Figura 3.6

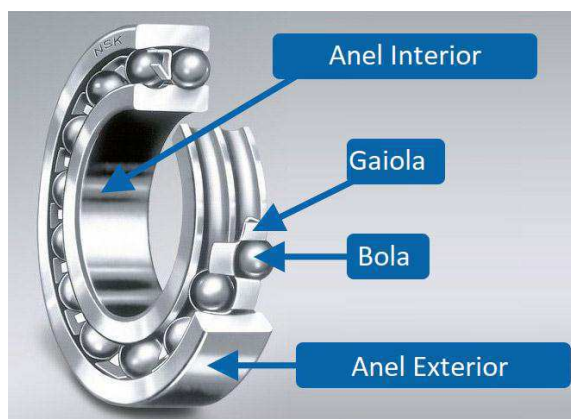


Figura 3.6 – Representação do Rolamento e suas Partes. Fonte: Adaptado de [41]

Algumas das causas de defeitos nos rolamentos podem incluir a entrada de partículas externas entre os anéis externo e interno, como poeira, água ou algum ácido, o óleo lubrificante pode estar velho, podem haver corrente passando pelo rolamento e até mal alinhamento pode levar a problemas com os rolamentos, além do próprio desgaste por uso contínuo.

Os defeitos em rolamentos comumente são analisado em sinais no domínio do tempo, por meio da análise de sua envoltória, como exemplificado na Figura 3.7.

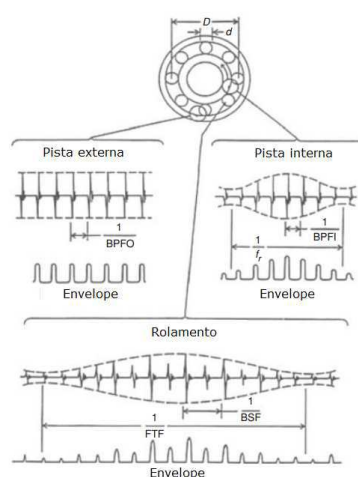


Figura 3.7 – Exemplos de possível gráfico de vibração com rolamento defeituoso. Fonte: Adaptado de [18]

4 Materiais e Métodos

Os dados utilizados neste trabalho foram adquiridos do modelo educacional mostrado na figura 4.1, o qual conta com um motor monofásico de 60 Hz cuja potência nominal é de 0,25 cv associada a um inversor CFW-10 WEG, no entanto, as frequências reais usadas para os experimentos foram de 30 e 60 Hz .

O mesmo foi montado em diferentes configurações, cada uma adequada para uma falha específica, que são melhor explicitadas nas Seções 4.1, 4.2 e 4.3.



Figura 4.1 – Motor na configuração de análise de desbalanceamento. Fonte: Autoria Própria

4.1 Desbalanceamento

A configuração mostrada na Figura 4.1, composta pelo motor, o eixo, dois mancais e o disco branco com parafusos pretos é aquela na qual é possível analisar falhas de desequilíbrio.

O disco branco possui orifícios com ângulos de 20 graus entre eles e o raio do disco é de aproximadamente 6,5 cm, no qual os parafusos são inseridos para simular um eixo cada vez mais desequilibrado, com um comprimento de cerca de 60 cm.

foram realizados testes com 1, 2, 3, 4 e 5 parafusos inseridos para representar diferentes os diferentes níveis de desbalanceamento.

4.2 Desalinhamento

Para a análise do desalinhamento foram feitas algumas alterações em relação à configuração anterior, como pode ser visto na Figura 4.2, onde o eixo teve que ser substituído por dois menores, com 26cm e 31cm cada, conectados por um junta de acrílico, para que fosse possível simular a falha próximo ao meio do eixo.

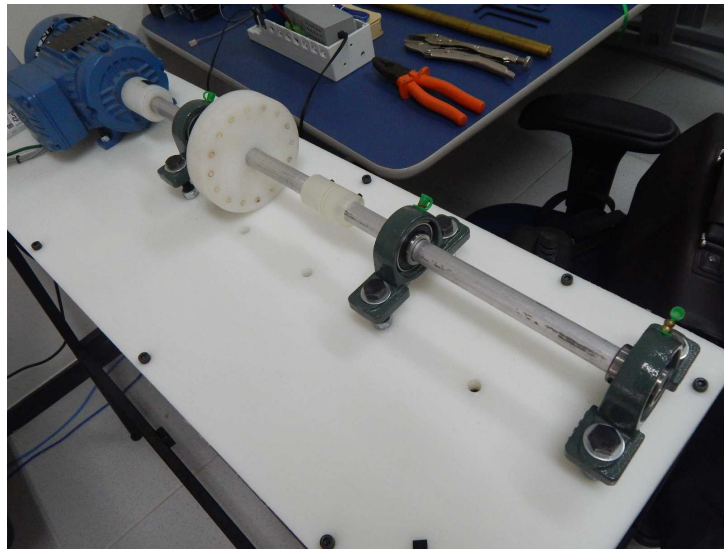


Figura 4.2 – Motor na configuração de análise de desalinhamento. Fonte: Autoria Própria

O tipo de desalinhamento analisado foi o paralelo, onde o primeiro segmento, entre a junta acrílica e o motor, estaria bem alinhado e o segundo, da articulação até o final, desalinhado em diferentes níveis, para cima e para baixo.

Para determinar a gravidade da falha, a diferença no alinhamento, as medidas foram tomadas usando um relógio comparador com precisão de 0,01 mm.

O desalinhamento paralelo induzido em diferentes níveis estava de acordo com os parâmetros estabelecidos pelos manuais da Easy-laser [42], que é fabricante e fornecedor de sistemas de medição e alinhamento baseados em laser.

4.3 Defeitos em Rolamentos

Quanto à análise do rolamento defeituoso, primeiro havia que haver um rolamento nessa situação para possibilitar a extração dos dados necessários.

Existem muitas técnicas que podem ser empregadas para desgastar rolamentos, a exemplo da descrita por [17]. No entanto, o método escolhido foi usar o motor na Figura 4.3 dispostos de forma a ser capaz de alcançar velocidades angulares muito superiores àquela nominal do motor original e sem lubrificação, pois lembra o desgaste real de um rolamento em campo, com a lubrificação ficando menos eficiente.



Figura 4.3 – Motor menor para danificar rolamentos. Fonte: Autoria Própria

Para os testes, foi utilizada a configuração de desbalanceamento mostrada na Figura 4.1 sem parafusos presos ao disco e substituindo o rolamento mais próximo do motor pelo danificado.

4.4 Seleção de Parâmetros

Foram então determinados os parâmetros potencialmente úteis, com base em [18], e os escolhidos foram divididos em três tipos. Frequência, tempo e estatística.

Os componentes de frequência foram os cinco primeiros múltiplos da frequência fundamental pela estimativa de Welch [43]. Os dados de tempo são compostos da razão entre o valor máximo e o valor RMS [17]. As características estatísticas foram as primeiras, que são a média, o segundo, o quarto e o sexto momentos estatísticos, a variância e a curtose [44].

Esta etapa é fundamental porque se destina a ser aplicável em sistemas embarcados, nos quais há uma quantidade limitada de memória e, portanto, é muito interessante saber quais dados são realmente necessários para que possamos economizar o máximo de espaço de armazenamento possível[45].

4.5 Metodologia de Análise

Foram realizados experimentos utilizando acelerômetro que duraram dez minutos com 500 leituras por segundo e depois foram separados em experimentos com 4000 observações cada.

Na posse de dados adequados para análise, foram necessários os instrumentos e procedimentos de análise adequados para atingir o objetivo da pesquisa, sendo estes executados da

seguinte forma.

Os dados adquiridos seriam enviados diretamente à plataforma Microsoft Azure por meio de uma conta de armazenamento blob associada à mesma. Em seguida os dados são pré-processados em python de forma a obter os parâmetros de interesse previamente discriminados.

Os algoritmos de aprendizado de máquina foram então treinados em se utilizando um diagrama na plataforma Azure Machine Learning Studio que pode ser visto na Figura A.1 em anexo. Pode-se observar neste mesmo diagramas as etapas essenciais par ao treino de um modelo preditivo em aprendizado de máquina.

Os primeiros blocos selecionando as colunas, ou seções dos dados, que serão de interesse que neste caso está servindo para padronizar as entradas e para manter algumas informações do teste que são fornecidos pelo próprio dispositivo de captura, tal qual a etiqueta.

Em seguida há o pré-processamento em python, a organização de alguns dados por meio do bloco edit metadata e a normalização. Os dados são entrão divididos entre dados de treino e de testes e por fim é realizado o treinamento, avaliado e salvo o modelo obtido.

Foram utilizados os métodos de regressão logística, redes neurais, floresta de decisão, floresta de decisão, máquina de vetor de suporte e máquina de vetor de suporte localmente profundo, perceptron médio, máquina de ponto Bayes e árvore de decisão aprimorada.

O método com melhores resultados foi então transferido para o diagrama na Figura A.2 em anexo, responsável pelo diagnóstico de futuras medições e envio dos dados adequados para o interfaceamento com os técnicos e engenheiros que venham a utilizar o produto final.

Este segundo diagrama é um pouco mais simples, sendo apenas necessárias algumas alterações nos dados de entrada para que fiquem no mesmo formato de que aqueles utilizados no treino e em seguida tais dados alimentam o modelo que deve ser capaz de gerar um diagnóstico confiável.

4.6 Relatório Final e PowerBI

Na aplicação em mãos existem diversas informações que seriam muito úteis para se ter acesso em tempo real e todas podem ser dispostas por meio de relatórios gerados pela aplicação PowerBI.

O PowerBI é um serviço de análise de dados baseado em nuvem da Microsoft que permite aos usuários gerar facilmente relatórios elegantes e completos, possibilitando, se desejado, visualizar dados importantes em tempo real.

5 Resultados

Conforme declarado antes, os dados adquiridos foram enviados com êxito ao servidor do Azure e, a partir daí, foi possível fazer a análise desejada.

Primeiramente, os resultados provenientes do teste dos diferentes métodos de aprendizado de máquina para problemas de classificação foram os apresentados na Tabela 2. As métricas utilizadas foram a Exatidão, que divide o número de previsões corretas do número total de observações e o F1-Score, que leva em consideração a precisão e a sensibilidade.

Os resultados foram feitos para todo o conjunto de dados, fornecendo a Precisão geral e a pontuação F1 micro-média, e separados por classe e, em seguida, pela média, a partir da qual derivou a precisão média e a pontuação F1 macro-média. As opções calculadas com separação por classe são mais adequadas para os casos em que há um grande desequilíbrio entre a quantidade de dados de cada classe.

Method	Perceptron Médio	SVM - localmente profunda	Máquina de Bayes Point	Árvore de Decisão Boosted	Selva de Decisão	Floresta de Decisão	Rede Neural	Regressão Logística	SVM
Exatidão Geral (%)	79	75	72	96	93	94	73	64	68
Exatidão Média(%)	96	95	94	99	99	99	95	93	94
Micro-Média do F1-Score(%)	79	75	72	96	93	94	73	64	68
Macro-Média do F1-Score(%)	82	81	79	97	93	94.5	78	NaN	76

Tabela 2 – Relação entre os métodos de aprendizado de máquina suas respectivas precisões e F1-Score. Fonte: Autoria Própria

Embora a exatidão seja a mais intuitiva, pois fornece a porcentagem de classificações corretas, o F1-Score é mais adequado para conjuntos de dados desequilibrados, pois leva em consideração a precisão, que é o número de vezes que o preditor classificado corretamente como sendo de um classe específica sobre o total classificado como daquela mesma classe e recordação, que é o número de vezes que o preditor classificado corretamente como sendo de uma classe específica sobre o número total de itens que realmente eram dessa classe.

Observa-se também que há a diferença entre macro-média, ou apenas média, e micro-média, ou global, onde o primeiro leva em consideração todos os dados juntos e o segundo avalia classe por classe e faz a média no final. Apesar dos valores de macro média serem melhores no caso em estudo os de micro-média F1 são mais adequados em geral, por levarem em consideração diferenças nas quantidades de dados para validação.

A partir da tabela, pode-se ver que a maioria dos métodos teve um bom desempenho em relação à métrica de exatidão média, no entanto, quando se trata de exatidão geral ou pontuações F1, isso deixa de ser válido. É algo esperado, já que a maioria deles é conhecida por

outras características. Perceptron médio, Bayes Point Machine e SVMs, proporcionando melhor linearidade e regressão logística, com um tempo de treinamento mais rápido.

Redes neurais são uma exceção. É uma técnica incrivelmente versátil e pode variar drasticamente os resultados com base em muitas características; no entanto, devido a essa versatilidade excessiva, acaba sendo significativamente mais difícil obter um modelo confiável. O apresentado, como é o caso de todos os métodos, na Tabela 2 foi o melhor resultado obtido ao testar diversas configurações.

Por fim, com o modelo treinado para aquele que nos deu o melhor resultado, a floresta de decisão, foi gerado, usando o PowerBI Desktop, o relatório que pode ser visto na Figura 5.1.

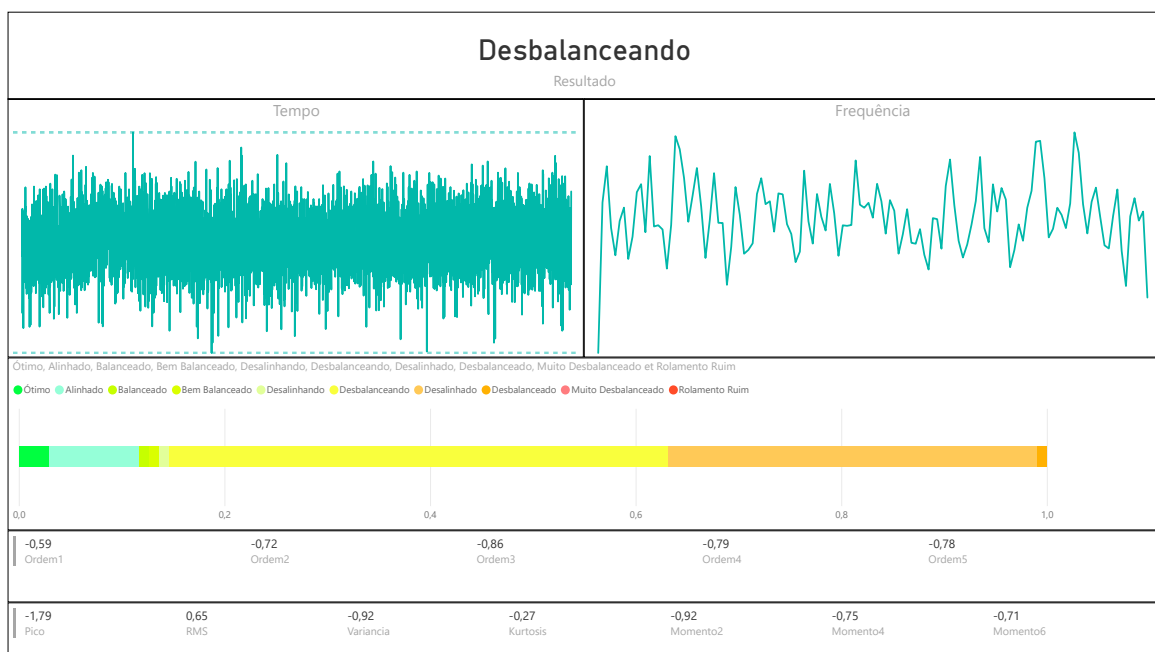


Figura 5.1 – Relatório gerado pelo PowerBI para o diagnóstico. Fonte: Autoria Própria

A interface do relatório final foi projetada de forma a permitir ao operador verificar os resultados apresentando os diagnósticos, bem como os gráficos de tempo e frequência, um gráfico com as probabilidades que o preditor obteve e os valores dos principais parâmetros que são importantes para o diagnóstico.

6 Conclusão

Ao escolher um método, parâmetros diferentes podem ser decisivos. Como os motores podem ser caros, é primordial maximizar a precisão, ou a pontuação F1, pois o tempo de treinamento e a linearidade não são tão importantes quanto os diagnósticos precisos. Sendo assim é natural a escolha dos métodos baseados em floresta de decisão.

Também é importante que no relatório final sejam inclusas as informações necessárias para realização de diagnóstico independente de manutenção preditiva tanta para garantir a validação ao sistema, o que torna o aplicativo em si muito mais robusto.

Contudo o sistema apresentado neste trabalho ainda não pode ser considerado finalizado ou comercial pois não há documentação adequada para o mesmo e a interface entre as diferentes etapas é feita manualmente.

Também seria interessante em etapas futuras que fosse gerada nova base de dados de forma mais exata para o danificar dos rolamentos, possivelmente incluindo outros erros como em engrenagem ou no eixo da máquina.

Outras possibilidades de continuidade deste trabalho são a investigação da possibilidade de aplicação de métodos de detecção de anomalias para a realização do diagnóstico e a geração de dados para treino artificialmente, já que a análise de vibração é um campo com modelos já bem estabelecidos.

Referências

- [1] Kevin Ashton. That 'internet of things' thing. *RFID Journal*, 2009.
- [2] Christian Arnold, Daniel Kiel, and Kai-Ingo Voigt. How the industrial internet of things changes business models in different manufacturing industries. *International Journal of Innovation Management*, 20(08):1640015, 2016.
- [3] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431 – 440, 2015.
- [4] Y Han and YH Song. Condition monitoring techniques for electrical equipment-a literature survey. *IEEE Transactions on Power delivery*, 18(1):4–13, 2003.
- [5] Antoine Grall, Laurence Dieulle, Christophe Bérenguer, and Michel Roussignol. Continuous-time predictive-maintenance scheduling for a deteriorating system. *IEEE transactions on reliability*, 51(2):141–150, 2002.
- [6] Jack McGuinn. Predictive motor maintenance (pdm): Use it or lose it (money). *Power Transmission Engineering*, 2012.
- [7] Rafael B. C. Lima and Eduardo V. Falcão et al. Sistema móvel para inspeção de máquinas girantes - análise de vibração. In *Rio Oil & Gas 2016 expo and conference*, 2016.
- [8] M. GRAD Hicham. Apport de la mcsa par rapport à l'analyse vibratoire pour le suivi de l'état de la machine asynchrone. Master's thesis, Université Mohammed V -Soussi, 2012/2013.
- [9] Moshou D. Pantazi XE. et al. Kateris, D. A machine learning approach for the condition monitoring of rotating machinery. *J Mech Sci Technol*, 28: 61, 2014. <https://doi.org/10.1007/s12206-013-1102-y>.
- [10] David A Maluf and L Daneshmend. Application of machine learning for machine monitoring and diagnosis. In *10th International Florida Artificial Intelligence Research Symposium*, pages 232–236, 1997.
- [11] Ameeth Kanawaday and Aditya Sane. Machine learning for predictive maintenance of industrial machines using iot sensor data. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 87–90. IEEE, 2017.
- [12] Brad Cline, Radu Stefan Niculescu, Duane Huffman, and Bob Deckel. Predictive maintenance applications for machine learning. In *2017 Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–7. IEEE, 2017.

- [13] Yinong Chen and Hualiang Hu. Internet of intelligent things and robot as a service. *Simulation Modelling Practice and Theory*, 34:159 – 171, 2013.
- [14] Artur Arsénio, Hugo Serra, Rui Francisco, Fernando Nabais, João Andrade, and Eduardo Serrano. Internet of intelligent things: bringing artificial intelligence into things and communication networks. In *Inter-cooperative Collective Intelligence: Techniques and Applications*, pages 1–37. Springer, 2014.
- [15] Ping Li, Jin Li, Zhengan Huang, Tong Li, Chong-Zhi Gao, Siu-Ming Yiu, and Kai Chen. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74:76–85, 2017.
- [16] M. Yaseen, D. Swathi, and T. A. Kumar. Iot based condition monitoring of generators and predictive maintenance. In *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, pages 725–729, Oct 2017.
- [17] Omar DJEBILI. *Contribution à la maintenance prédictive par analyse vibratoire des composants mécaniques tournants. Application aux butées à billes soumises à la fatigue de contact de roulement*. PhD thesis, UNIVERSITE DE REIMS CHAMPAGNE ARDENNE, 2013.
- [18] R.B. Randall. *Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications*. EBL-Schweitzer. Wiley, 2011.
- [19] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. "O'Reilly Media, Inc.", 2017.
- [20] Microsoft. Machine learning algorithm cheat sheet for azure machine learning studio, jul 2019.
- [21] J. R. Quinlan. Decision trees and decision-making. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):339–346, March 1990.
- [22] Eduardo Vieira Falcão. Diagnóstico de faltas em bombas, compressores e motores de indução a partir de dados de vibração. Seminário apresentado pelo autor ao LIEC, oct 2015.
- [23] Eduardo Vieira Falcão. Sistema de monitoramento de severidade vibracional e detecção de faltas em máquinas girantes, oct 2016. Monografia (Bacharel em Engenharia Elétrica), UFCG (Universidade Federal de Campina Grande), Campina Grande - PB, Brazil.
- [24] Curiosity. Listen to the princess and the fox, an oddball fairy tale written by an ai, aug 2019.
- [25] Curiosity. Robots of the future will lie to us, and that's a good thing, aug 2019.

- [26] Curiosity. Facebook a.i. programmed to negotiate teaches itself to lie, aug 2019.
- [27] Curiosity. Read these (legit terrifying) horror stories by an artificial intelligence, aug 2019.
- [28] Curiosity. Meet ibm watson, the computer that can argue better than anyone, aug 2019.
- [29] P. Simon. *Too Big to Ignore: The Business Case for Big Data*. Wiley and SAS Business Series. Wiley, 2013.
- [30] T.M. Mitchell. *Machine Learning*. McGraw-Hill international editions - computer science series. McGraw-Hill Education, 1997.
- [31] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [32] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2019.
- [33] Ludolouis. Comment fonctionne un réseau de neurones artificiels, Mar 2019.
- [34] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [35] Bogumił Kamiński, Michał Jakubczyk, and Przemysław Szufel. A framework for sensitivity analysis of decision trees. *Central European Journal of Operations Research*, 26(1):135–159, Mar 2018.
- [36] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- [37] Leo Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at . . . , 1997.
- [38] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780, 1963.
- [39] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [40] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [41] M. Grad Hicham. Apport de la mcsa par rapport à l’analyse vibratoire pour le suivi de l’état de la machine asynchrone. Master’s thesis, Université Mohammed V -Soussi, 2013.
- [42] Easy-laser. *SHAFT ALIGNMENT SYSTEM*, nov 2014.

-
- [43] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.
- [44] Samir S Soliman and S-Z Hsue. Signal classification using statistical moments. *IEEE Transactions on Communications*, 40(5):908–916, 1992.
- [45] Shufu Mao and Tilman Wolf. Hardware support for secure processing in embedded systems. *IEEE Transactions on Computers*, 59(6):847–854, 2010.

ANEXO A –

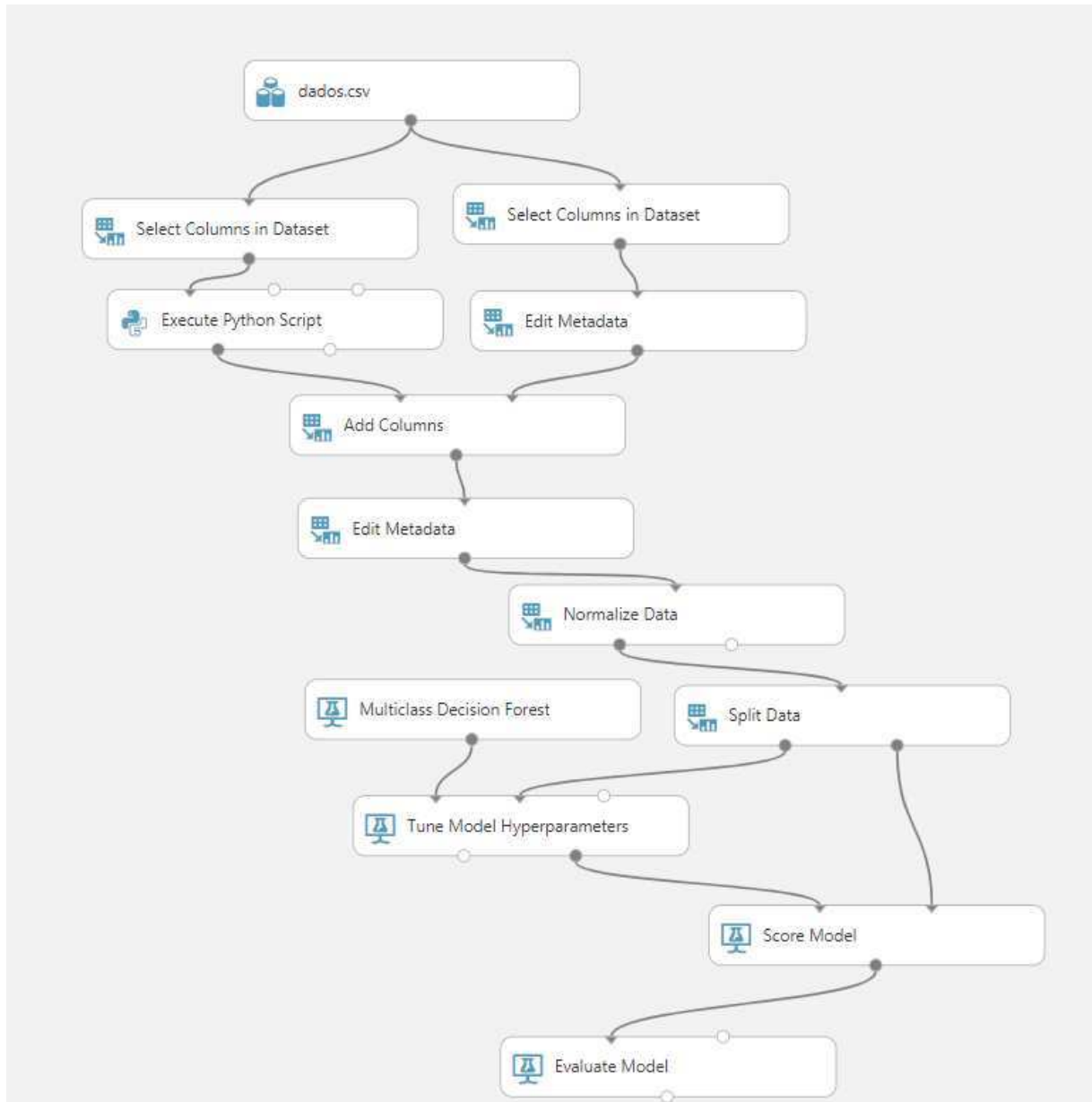


Figura A.1 – Diagrama em AzureMLStudio para treinamento

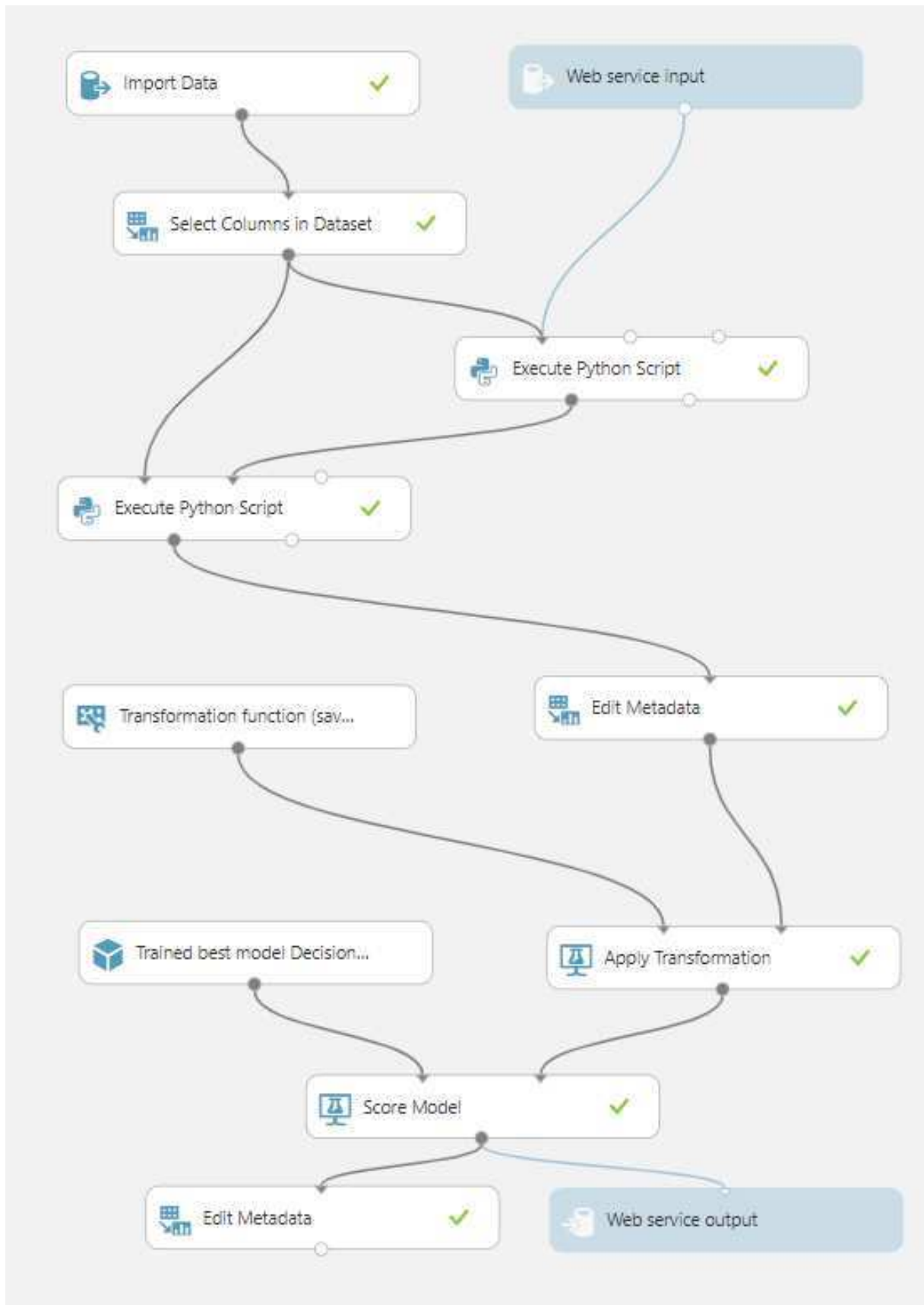


Figura A.2 – Diagrama em AzureMLStudio para diagnóstico