UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENERGIA NUCLEAR
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS ENERGÉTICAS E NUCLEARES

LEONARDO JOSÉ DE PETRIBÚ BRENNAND

DETECÇÃO E ISOLAMENTO DE FALHAS EM AEROGERADORES UTILIZANDO DADOS DO SISTEMA SCADA

Recife
2022

LEONARDO JOSÉ DE PETRIBÚ BRENNAND

DETECÇÃO E ISOLAMENTO DE FALHAS EM AEROGERADORES UTILIZANDO DADOS DO SISTEMA SCADA

Dissertação apresentada ao Programa de Pós-Graduação em Tecnologias Energéticas e Nucleares da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Tecnologias Energéticas e Nucleares.

Área de concentração: Fontes Renováveis de Energia.

Orientadora: Profa. Dra. Olga de Castro Vilela

Coorientador: Prof. Dr. Alexandre Carlos Araújo da Costa

Recife

2022

Catalogação na fonte:
Bibliotecária Rosineide Mesquita Gonçalves da Luz, CRB-4 / 1368

B838d Brennand, Leonardo José de Petribú.
Detecção e isolamento de falhas em aerogeradores utilizando dados do sistema SCADA / Leonardo José de Petribú Brennand. – 2022.
89 f.: il., figs., tabs.

Orientadora: Profa. Dra. Olga de Castro Vilela.
Coorientador: Prof. Dr. Alexandre Carlos Araújo da Costa.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Tecnologias Energéticas e Nucleares, 2022.
Inclui referências.

1. Energia eólica. 2. Acurácia e disponibilidade. 3. Comportamento livre de falhas. 4. Planejamento de manutenção. 5. Energias renováveis. I. Vilela, Olga de Castro (Orientadora). II. Costa, Alexandre Carlos Araújo da (Coorientador). III. Título.

UFPE

621.48 CDD (22. ed.) BCTG/2022-401

LEONARDO JOSÉ DE PETRIBÚ BRENNAND

DETECÇÃO E ISOLAMENTO DE FALHAS EM AEROGERADORES UTILIZANDO DADOS DO SISTEMA SCADA

Dissertação Apresentada ao Programa de Pós Graduação Tecnologias Energéticas e Nucleares da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do título de Mestre em Tecnologias Energéticas e Nucleares. Área de Concentração: Fontes Renováveis de Energia.

Aprovado em: 24/08/2022

BANCA EXAMINADORA

_____
Prof. Dr. Emmanuel Damilano Dutra (Examinador Interno)
Universidade Federal de Pernambuco - UFPE

_____
Prof. Dr. Tsang Ing Ren (Examinador Externo)
Universidade Federal de Pernambuco - UFPE

_____
Prof. Dr. Gustavo de Novaes Pires Leite (Examinador Externo)
Instituto Federal de Pernambuco - IFPE

AGRADECIMENTOS

Agradeço a Deus e a toda minha família, na terra, no céu, de sangue e de coração, que guiaram toda minha trajetória de vida e contribuíram para minha formação como ser humano, essencial para que eu chegasse até aqui;

Agradeço a todos os amigos que cultivei na UFPE, em especial às pessoas brilhantes do Centro de Energias Renováveis – sejam colegas de pesquisa, professores e todos funcionários de apoio;

Agradeço aos meus orientadores, diretos e indiretos, por todos os ensinamentos passados ao longo destes anos, que com certeza irei levar por toda minha vida profissional e acadêmica;

Por fim, agradeço às seguintes entidades pelo apoio financeiro ao longo desta caminhada: a CAPES, pela bolsa no âmbito do programa de pós-graduação em Tecnologias Energéticas e Nucleares; a RAESA/Multiner S.A., pelo auxílio no âmbito do projeto de P&D ANEEL NEO.PROGFALHAS.

RESUMO

O crescimento da energia eólica no mundo levanta a importância cada vez maior da segurança operacional e energética dos aerogeradores. Tais máquinas são acometidas por diversos tipos de falha ao longo de sua vida útil, fazendo com seja necessário detectá-las e localizá-las com antecedência a fim de auxiliar no planejamento do calendário de manutenção dos componentes afetados. Nesse sentido, o objetivo geral deste trabalho é o desenvolvimento de uma metodologia de detecção e isolamento de falhas em aerogeradores a partir da modelagem do comportamento livre de falhas de sinais de temperatura embarcados no sistema SCADA. Para isso, os modelos *Random Forest* e *XGBoost* foram parametrizados com um número elevado de variáveis de entrada, a fim de alcançar uma alta acurácia (tópico muito abordado na literatura). Por outro lado, foram utilizadas redes neurais com um número reduzido de variáveis de entrada, a fim de alcançar uma baixa taxa de ausência de dados na saída do modelo (tópico pouco abordado na literatura). Além disso, uma nova técnica de envelope foi desenvolvida para auxiliar na remoção de *outliers* no conjunto de dados utilizado, sem a necessidade de qualquer intervenção por parte do usuário. Na aplicação da metodologia para dados históricos de um aerogerador em operação, uma falha no rolamento do gerador elétrico e uma falha no transformador foram detectadas e isoladas com 26 e 33 dias de antecedência, respectivamente. Em termos de acurácia, tais resultados são superiores a outros encontrados na literatura, com a contribuição extra de possuir modelos focados em entregar estimativas com elevada disponibilidade (baixa taxa de ausência de dados), o que é muito importante para soluções em tempo real. Futuramente, o desenvolvimento de técnicas de combinação das estimativas de detecção de falhas pode ser realizado a fim de mitigar a influência de falsos positivos na predição final da metodologia.

Palavras-chave: energia eólica; acurácia e disponibilidade; comportamento livre de falhas; planejamento de manutenção; energias renováveis.

ABSTRACT

The growth of wind energy in the world raises the importance of wind turbines' operational safety and energy availability. Such machines are affected by several types of failure throughout their useful life, making it necessary to detect and isolate them in advance to assist in planning the maintenance schedule of the affected components. In this sense, this work's general objective is to develop a methodology for detecting and isolating faults in wind turbines by modelling the fault-free behaviour of temperature signals embedded in the SCADA system. For this, the Random Forest and XGBoost models were parameterized with a high number of input variables to achieve high accuracy (a much-discussed topic in the literature). On the other hand, neural networks with a reduced number of input variables were used to achieve a low rate of missing data in the model's output (a topic rarely addressed in the literature). In addition, a new envelope technique was developed to remove outliers in the dataset without user intervention. When applying the methodology to historical data of an operational wind turbine, a failure in the generator bearing and a failure in the transformer were detected and isolated 26 and 33 days in advance, respectively. In terms of accuracy, these results are better than others found in the literature, with the extra contribution of having models focused on delivering estimates with high availability (low rate of missing data), which is very important for real-time solutions. In the future, the development of techniques to combine fault detection estimates can be carried out to mitigate the influence of false positives in the final prediction of the methodology.

Keywords: wind energy; accuracy and availability; fault-free behavior; maintenance planning; renewable energy.

SUMÁRIO

1 INTRODUÇÃO

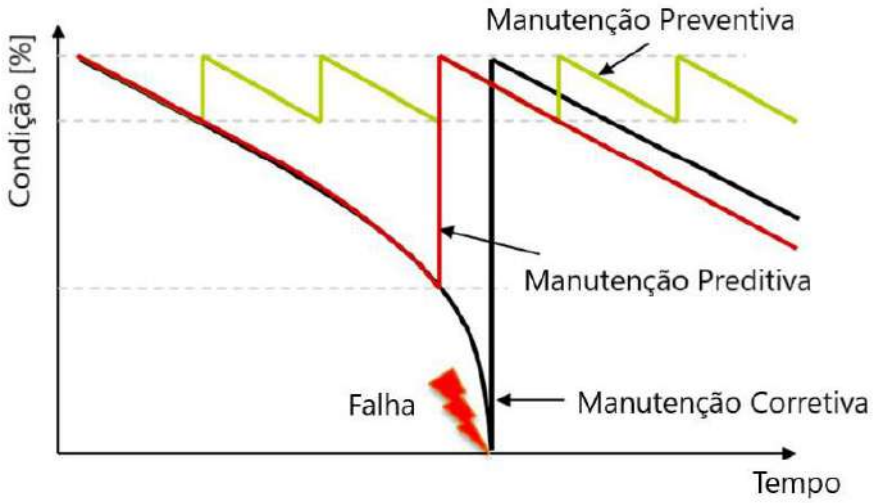
Nos últimos anos, a demanda por energia elétrica vem crescendo vertiginosamente no Brasil e no mundo. No panorama nacional, dados do Plano Decenal de Expansão de Energia mostram que o consumo final de energia elétrica chegou a 563 TWh em 2021, com previsão de demanda de 792 TWh para o ano de 2031, projetando quase 30% de crescimento (EPE, 2022). Considerando essa projeção, e diante da preocupação cada vez maior com a redução no uso de combustíveis fosseis para geração de energia elétrica, a energia eólica desponta como uma das principais soluções para essa problemática. Diante desse cenário, é inerente a preocupação acerca da segurança operacional das centrais eólicas conectadas ao Sistema Interligado Nacional (SIN). O correto funcionamento do processo produtivo de energia elétrica das centrais se torna, portanto, essencial para cumprir tal requisito. No entanto, a quantidade de equipamentos elétricos e mecânicos envolvidos no processo traz um enorme desafio para esta tarefa. Diferentes fatores fazem com que os equipamentos estejam sujeitos a altas probabilidades de falhas, o que afeta a disponibilidade energética da central, diminui sua rentabilidade e põe em risco – a depender da gravidade da falha – aqueles que se encontram próximos do local.

Atualmente, a forma usual de mitigar o impacto e a ocorrência das falhas é por meio de atividades de manutenção, as quais são divididas em três tipos: manutenção corretiva; manutenção preventiva e manutenção preditiva. A Figura 1 ilustra a degradação de um sinal – que nesse caso indica a condição[1] de um componente qualquer – ao longo do tempo, de acordo com o tipo de manutenção empregada. Nota-se que a manutenção corretiva é executada apenas após a ocorrência da falha, o que faz com que exista a possibilidade de quebra do componente, fazendo com que haja a necessidade da sua substituição total (o que pode ser custoso financeiramente). A manutenção preventiva possui a vantagem de ser programável, o que traz maior segurança operacional e reduz significativamente a probabilidade de ocorrência de falhas. No entanto, o não aproveitamento de toda vida útil do componente faz com que muitas vezes tais manutenções sejam desnecessárias, o que torna o procedimento custoso do ponto de vista financeiro. A manutenção preditiva se baseia

[1] Entende-se por condição: o nível de saúde operacional de um certo componente.

na evolução da condição dos sinais monitorados no componente para decidir o melhor período de tempo a ser realizada. Tal estratégia se apoia no fato de que, na prática, 99% das falhas em equipamentos são precedidas por certos sinais, condições ou indicações de que a falha irá ocorrer (AHMAD; KAMARUDDIN, 2012).

Figura 1 - Degradação de um sinal ao longo do tempo de acordo com a manutenção empregada.



Fonte: adaptado de Fischer e Coronado (2015).

Quando os sinais monitorados desviam do seu comportamento esperado acima de certo limiar, constata-se um forte indício de evolução de uma falha – ou seja, parte-se do pressuposto que o sinal de um componente “X” com falha apresenta suficiente dissimilaridade com o sinal do mesmo componente “X” sem falha. O procedimento que estabelece a dissimilaridade entre os sinais “com falha” e “sem falha” é chamado de detecção de falhas, sendo ela a primeira etapa necessária para a realização de uma futura manutenção preditiva. Uma vez detectada a falha, mais informações devem ser coletadas a fim de auxiliar a equipe de O&M (Operação e Manutenção) na tomada de decisão. Nesse sentido, a localização da falha (e.g., subcomponente “x” do componente “X”) é de grande valia para o planejamento da manutenção. O procedimento responsável por entregar estas informações é chamado de isolamento de falhas, sendo ele realizado de forma automática e imediata após a detecção da falha.

Em aplicações reais em aerogeradores, os procedimentos supracitados são embarcados em ferramentas computacionais de operação em tempo real. Assim sendo, existem uma série de desafios inerentes a esse tipo de solução que

ultrapassam as dificuldades puramente metodológicas envolvidas na construção dos modelos. Por exemplo, a usabilidade da ferramenta pelo usuário final é de extrema importância, devendo ser o menos complexa possível a fim de não ter sua acurácia atrelada ao nível de conhecimento técnico de quem está a utilizando. Outro ponto fundamental é a capacidade da ferramenta de fornecer informações ao longo do tempo, o que pode ser caracterizado como sua disponibilidade[2]. Uma ferramenta com baixa disponibilidade possui uma alta taxa de ausência de informações entregues ao longo do tempo, o que pode fazer com que esta deixe de informar possíveis falhas que estejam por ocorrer, causando prejuízo financeiro e operacional à empresa detentora do parque eólico.

Sendo assim, o objetivo principal deste trabalho é o desenvolvimento de uma metodologia para detecção e isolamento de falhas em aerogeradores, utilizando dados do sistema SCADA (ver seção 2.2), que atenda dois pontos principais: a acurácia dos seus modelos (foco principal da literatura) e a disponibilidade de suas estimativas ao longo do tempo (pouco abordada na literatura).

Dentre os objetivos específicos, destacam-se: i) utilização de *targets* específicos para cada componente de interesse do aerogerador ao invés de apenas um *target* representativo para a saúde da máquina como um todo; ii) utilização de técnicas de seleção de atributos para a construção de modelos com poucas variáveis de entrada a fim de mitigar o problema da indisponibilidade de estimativas causada pela ausência de dados; iii) desenvolvimento de uma nova técnica automática, adaptável e não paramétrica de envelope para detecção de *outliers* em sinais bem correlacionados com o *target*.

[2] Aqui, importante não confundir a disponibilidade da ferramenta com a disponibilidade da central eólica. A primeira diz respeito à quantidade de estimativas que a ferramenta é capaz de entregar em um determinado período de tempo (ou seja, é o oposto da taxa de ausência de estimativas). A segunda, por sua vez, diz respeito à disponibilidade energética da central, ou seja, ao percentual de tempo em que os aerogeradores estão aptos a gerar energia.

2 CONCEITOS PRELIMINARES

Aqui, busca-se trazer um pouco dos conceitos preliminares envolvidos nesse trabalho (modelos, técnicas, definições etc.), com o intuito de auxiliar o leitor que não esteja familiarizado com o contexto em que se encaixam os problemas aqui expostos. No entanto, caso o leitor esteja familiarizado com alguns dos conceitos apresentados a seguir, ele poderá optar por não realizar a leitura das seções relativas a tais conceitos, sem que haja qualquer prejuízo no entendimento das demais seções deste documento.

2.1 O AEROGERADOR

Aerogeradores são máquinas rotativas que convertem a energia cinética contida no vento em energia elétrica. Dentre as várias arquiteturas que surgiram ao longo do tempo, a atualmente estabelecida comercialmente é de uma máquina rotativa com três pás acopladas a um eixo horizontal, que se conecta a uma caixa de engrenagem, cuja saída é conectada a um gerador elétrico. A Figura 2 ilustra em maiores detalhes os componentes que compõem um aerogerador comercial de grande porte.

Ao conjunto que consiste nas pás e no *hub* (cubo), dá-se o nome de rotor aerodinâmico. Tal sistema é a excitatriz mecânica do aerogerador, responsável por transferir a energia cinética do vento através do torque para o eixo principal.

O rotor aerodinâmico e o eixo principal constituem a chamada turbina eólica. A estrutura que comporta todos os componentes anexados ao rotor aerodinâmico é chamada de nacele. Tal estrutura possui a função principal de abrigar e proteger os componentes eletromecânicos das condições ambientais externas.

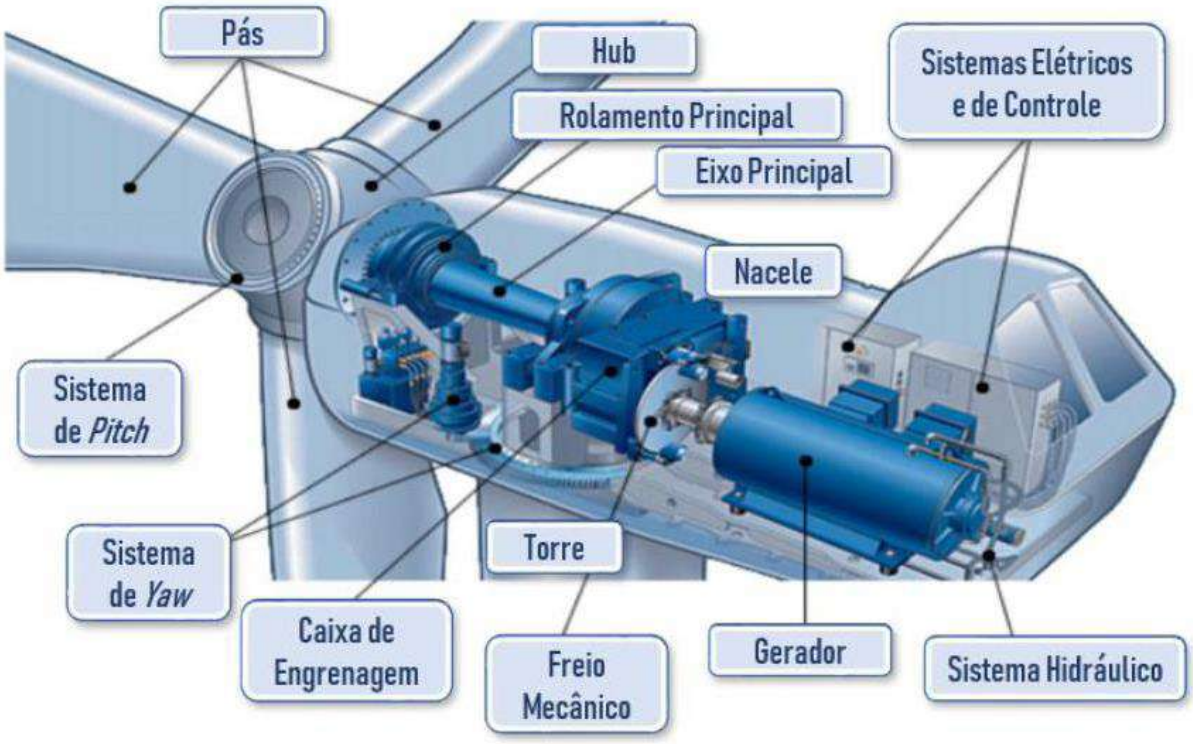
O sistema de *yaw* é aquele responsável pela rotação da nacele em torno do eixo da torre. O principal intuito desse mecanismo é manter o rotor perpendicular à direção predominante do vento. Tal tarefa é atingida mediante a atuação do sistema de controle localizado no interior da nacele, que recebe informações das medições de direção do vento do sensor de direção localizado no topo da nacele.

O sistema de *pitch* controla o ângulo de ataque de cada pá individualmente, visando uma maior captura de energia em condições normais de operação ou, em alguns casos, serve como um mecanismo de frenagem do rotor. O eixo principal e a

caixa de engrenagem constituem o sistema de transmissão do aerogerador. O eixo principal transmite o torque gerado a partir da rotação das pás em uma frequência rotacional baixa. A caixa de engrenagem é responsável por elevar esta frequência de rotação até a entrada do gerador elétrico.

Uma outra arquitetura que vem ganhando espaço atualmente é a de aerogeradores com sistemas de transmissão direta (*direct drive*). Neles, não há caixa de engrenagens, havendo a necessidade de um conversor eletrônico de potência para a conexão com a rede elétrica. O sistema elétrico compreende toda parte de conversão, transmissão de energia e controle. O principal componente deste sistema é o gerador elétrico, que pode ser síncrono ou assíncrono (TONG, 2010; PEDROSA, 2016; BEZERRA, 2019).

Figura 2 - Componentes de um aerogerador comercial de grande porte.

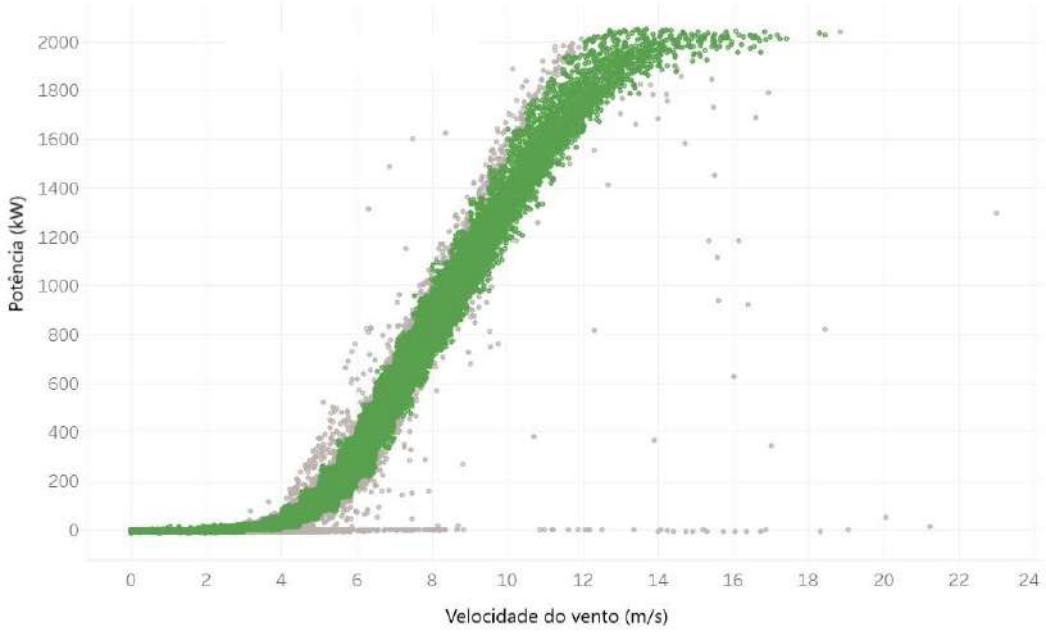


Fonte: Bezerra (2019).

Graficamente, uma das formas de se analisar o desempenho do aerogerador é através da chamada curva de potência da máquina. A Figura 3 ilustra a curva de potência típica de um aerogerador. Nela, cabe destacar 3 regiões importantes (valores usuais em máquinas de grande porte): a região de ausência de potência, para velocidades abaixo de um valor tipicamente igual a 4 ou 5 m/s (valor referente à

chamada velocidade de *cut-in* da máquina); a região de comportamento linear entre a potência e a velocidade, para valores de velocidade tipicamente entre 5 e 12 m/s (i.e., entre a velocidade de *cut-in* e a chamada velocidade nominal da máquina); e a região de potência nominal, na qual a potência máxima é atingida para valores de velocidade tipicamente entre 12 e 25 m/s (i.e., entre a velocidade nominal e a chamada velocidade de *cut-out* da máquina). Como é possível notar na Figura 3, os pontos da curva de potência possuem uma dispersão natural associada à velocidade do vento e à potência, com o seu tamanho variando de forma inversamente proporcional ao *time-step*[3] associado aos dados de potência e velocidade (i.e., quanto menor for o *time-step*, maior o tamanho da dispersão). Na figura, os pontos destacados em verde podem ser considerados como aqueles dentro da região de variabilidade esperada dos dados da curva, enquanto que os pontos em cinza (*outliers*) podem ser considerados dados de ocorrência pouco provável ou fruto de comportamentos indesejados (e.g., medições errôneas).

Figura 3 - Curva de potência de um aerogerador comercial de grande porte. Em verde, estão os pontos considerados dentro da região de variabilidade esperada da curva. Em cinza, estão os pontos considerados fora da região de variabilidade esperada da curva (*outliers*).



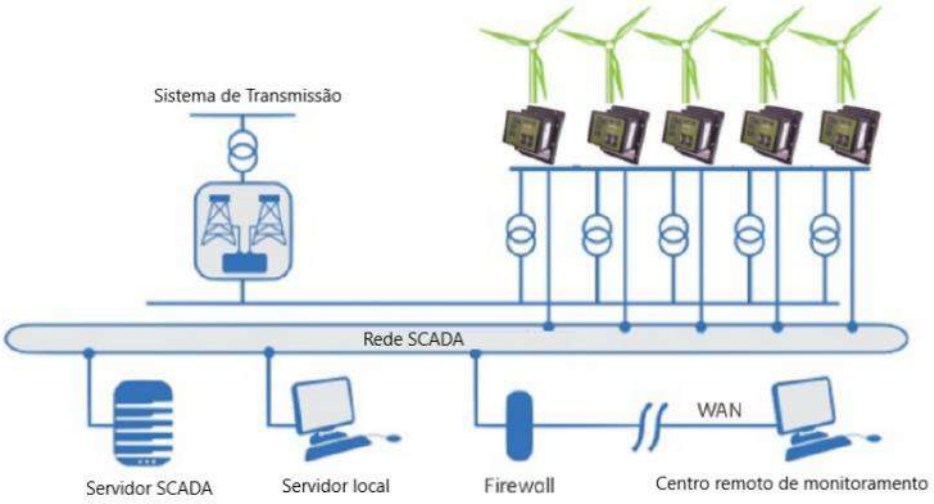
Fonte: Olaoye (2022).

[3] Entende-se por *time-step*: a diferença temporal entre dois registros consecutivos na base de dados observacionais utilizada.

2.2 O SISTEMA SCADA

Sistemas SCADA (*Supervisory Control and Data Acquisition*) são um caso particular de sistemas industriais de controle com o foco em ativos geograficamente dispersos, porém com aquisição e armazenamento centralizado de dados (STOUFFER et al., 2006). Eles permitem que operadores sejam capazes de controlar, monitorar e armazenar dados de aerogeradores, subestações elétricas e torres anemométricas a partir de um centro remoto, assim como mostrado numa possível arquitetura na Figura 4. Tal figura ilustra o modelo de um sistema SCADA de uma central eólica, na qual cada aerogerador é monitorado individualmente. Dentre os dados armazenados pelo sistema SCADA, são encontradas: variáveis ambientais, como velocidade, direção do vento e temperatura ambiente; variáveis elétricas de produção, como a tensão, corrente e potência; variáveis mecânicas, como a temperatura dos componentes e pressão dos óleos; configurações e diversos indicadores de estado (PEDROSA, 2016).

Figura 4 - Arquitetura de um Sistema SCADA para uma central eólica.



Fonte: adaptado de Pedrosa (2016).

2.3 APRENDIZADO SUPERVISIONADO

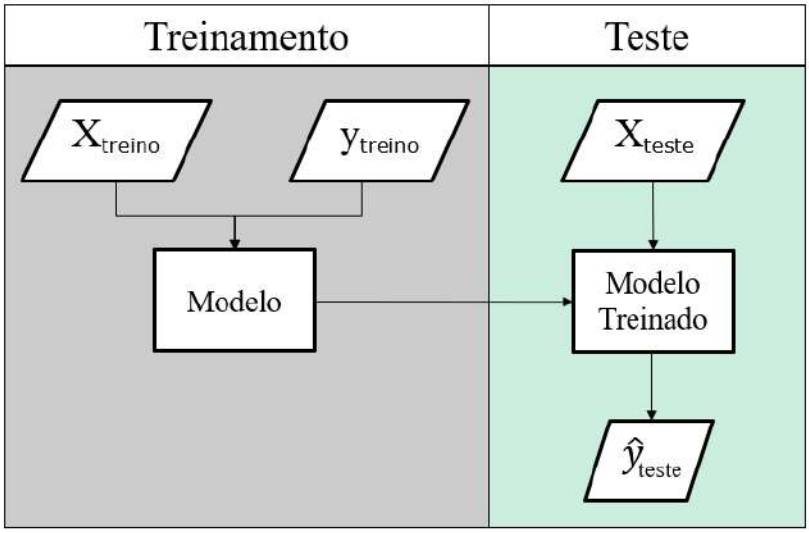
Em problemas de aprendizado de máquina (*machine learning*), o aprendizado supervisionado é o processo pelo qual um modelo (de regressão ou classificação)

estima uma determinada variável mediante a aprendizagem prévia do comportamento de um subconjunto de seus dados (GÉRON, 2019).

A Figura 5 ilustra o processo supracitado. Nela, “y” corresponde à variável que se deseja estimar. Tal variável é denominada de variável alvo ou *target*. Por outro lado, “X” consiste em uma matriz chamada de matriz de atributos – em problemas de regressão, também é chamada de matriz de variáveis regressoras. As colunas dessa matriz correspondem às variáveis utilizadas para estimar o comportamento da variável alvo a partir do modelo. O problema então divide “X” e “y” em dois conjuntos distintos e independentes chamados de conjunto de treinamento e conjunto de teste, com ambos possuindo um subconjunto de “X” e um subconjunto de “y” – ou seja, $\{X_{treino}, y_{treino}\}$ é o conjunto de treinamento e $\{X_{teste}, y_{teste}\}$ é o conjunto de teste.

A fase de treinamento é a responsável pela aprendizagem prévia do modelo acerca do comportamento de “y_{treino}”, utilizando para isso os dados de “X_{treino}”. Após o treinamento, o modelo parametrizado realiza a estimativa ($\hat{y}_{teste}$) da variável alvo (y_{teste}) a partir dos dados de “X_{teste}”. Caso o modelo seja bem treinado, é esperado que suas estimativas realizadas no conjunto de teste ($\hat{y}_{teste}$) sejam as mais próximo possíveis dos valores reais da variável alvo no mesmo conjunto (y_{teste}) – se isso ocorrer, o modelo treinado possui a característica de ter elevada acurácia (i.e., um alto nível de exatidão nas suas estimativas).

Figura 5 - Exemplo de um problema de aprendizado supervisionado. “X” diz respeito à matriz de atributos, “y” se refere ao *target*, “$\hat{y}$” diz respeito à estimativa do *target*, o sufixo “treino” se refere aos dados do conjunto de treinamento e o sufixo “teste” se refere aos dados no conjunto de teste.

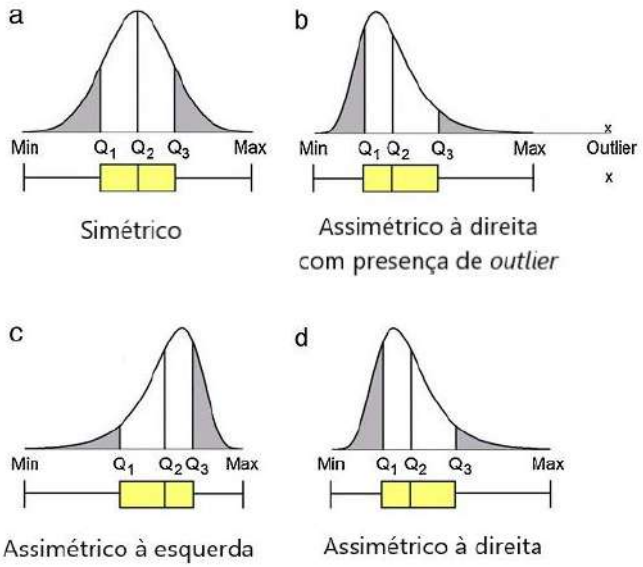


Fonte: o autor (2022).

2.4 TÉCNICA DE BOXPLOT

O diagrama de caixa, *box whiskers* ou *boxplot* é uma técnica estatística utilizada para a determinação de limites superior e inferior para detecção de *outliers* em dados numéricos. A abordagem é denominada “não paramétrica” por não depender dos parâmetros da distribuição estatística da variável em questão, fazendo com seja uma técnica com alto potencial de aplicação. A técnica utiliza-se do conceito de quartis e percentis para determinar a concentração de dados e estimar os limites inferior e superior de detecção de *outliers*, tal como mostra a Figura 6.

Figura 6 – Exemplo genérico da aplicação do *boxplot* a distribuições estatísticas com diferentes assimetrias. No caso “a”, uma distribuição simétrica. No caso “b”, uma distribuição com assimetria à direita e presença de um *outlier*. No caso “c”, uma distribuição com assimetria à esquerda. No caso “d”, uma distribuição com assimetria à direita e sem presença de *outliers*.



Fonte: adaptado de Ferreira et al. (2016).

A Equação 1 e Equação 2 representam, respectivamente, o cálculo dos limites inferior e superior de detecção de *outliers*. Nela, IQR = Q_1 - Q_3, Q_1 representa o primeiro quartil, Q_3 diz respeito ao terceiro quartil e “k” é um parâmetro multiplicador. Usualmente, atribui-se o valor de k=1,5. No entanto, outros valores podem ser utilizados a depender da aplicação e da distribuição dos dados em questão.

$$LI = Q_1 - k \cdot IQR \quad (1)$$

$$LS = Q_3 + \ k \cdot IQR \tag{2}$$

Sendo:
IQR = Q_1 - Q_3;
Q_1 – primeiro quartil;
Q_3 – terceiro quartil;
k – multiplicador;
LS – limite superior;
LI – limite inferior.

Em distribuições estatísticas de curtose muito elevada, as equações supracitadas tendem a rechaçar uma grande quantidade de dados, já que o valor de Q_3 se torna muito próximo ao valor de Q_1. Visando contornar esse problema, Hubert e Vandervieren (2008) propuseram uma modificação na forma de calcular os limites a partir do uso da função estatística *Medcouple* (BRYS et al., 2004). A Equação 3 define a função *Medcouple*. Nela, Q_2 define a mediana da amostra e $h(x_i,x_j)$ é uma função kernel dada pela Equação 4. Sendo assim, os limites inferior e superior serão dados pelos intervalos definidos nas Equações 5 e 6.

$$MC = med_{x_i \leq Q_2 \leq x_j} h(x_i, x_j) \in [-1, 1] \tag{3}$$

$$h(x_i, x_j) = \frac{(x_j - \ Q_2) - \ (Q_2 - \ x_i)}{x_j - x_i} \tag{4}$$

$$[Q_1 - k \cdot e^{-4MC} \cdot IQR; \ Q_3 + k \cdot e^{3MC} \cdot IQR], para \ MC \ \geq 0 \tag{5}$$

$$[Q_1 - k \cdot e^{-3MC} \cdot IQR; \ Q_3 + k \cdot e^{4MC} \cdot IQR], para \ MC < \ 0 \tag{6}$$

Sendo:
IQR = Q_1 - Q_3;
Q_1 – primeiro quartil;
Q_3 – terceiro quartil;

k – multiplicador;

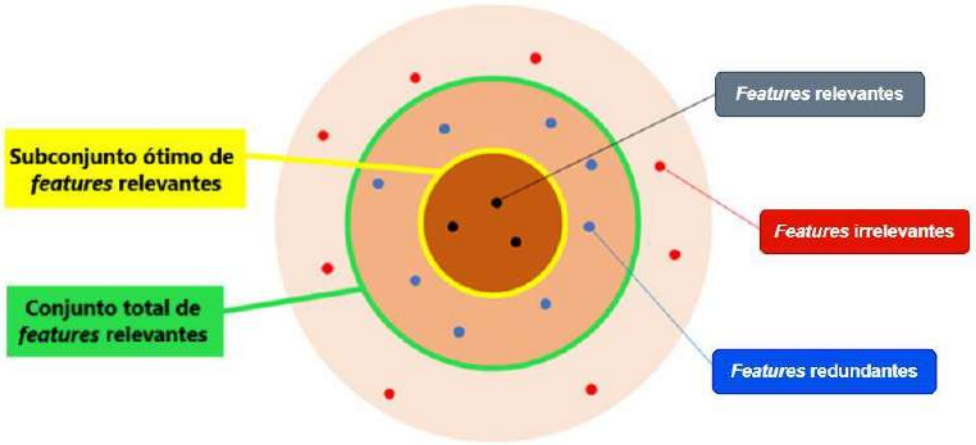
MC – *medcouple*;

$h(x_i,x_j)$ – função kernel.

2.5 ALGORITMO DE MÍNIMA REDUNDÂNCIA MÁXIMA RELEVÂNCIA

O algoritmo de mínima Redundância e Máxima Relevância (mRMR) é voltado essencialmente para a seleção de atributos (*features*) baseando-se em dois principais conceitos: a redundância e a relevância. A Figura 7 ilustra um esquema genérico do grau de importância dos atributos de um modelo qualquer. Nela, observa-se que dentre todas as variáveis (pontos do gráfico), algumas podem ser imediatamente descartadas (pontos vermelhos) por não possuírem um bom grau de relevância com o *target*. Sendo assim, dentre todas as variáveis relevantes (pontos pretos e azuis), existe um subconjunto ótimo que consegue carregar a maior parte das informações desejadas sem que haja redundância entre elas. Ou seja, do conjunto de variáveis relevantes, é possível eliminar aquelas que não acrescentam informações adicionais para o treinamento do modelo de regressão (i.e., são redundantes com relação às outras variáveis relevantes).

Figura 7 - Esquema genérico do grau de importância dos atributos (*features*) de um modelo qualquer. Os pontos em vermelho representam *features* irrelevantes para o modelo. Os pontos dentro do círculo verde representam todas as *features* com algum grau de relevância para o modelo, sendo as representadas por pontos pretos mais relevantes que as representadas por pontos azuis (*features* redundantes).



Fonte: adaptado de Mazzanti (2021).

A técnica de mRMR visa selecionar o subconjunto das “N” variáveis com maior relevância com target e menor redundância entre elas. A métrica para quantificar a relevância pode ser, por exemplo, a informação mútua entre o *target* e cada variável do conjunto original (LATHAM; ROUDI, 2009). Por sua vez, a redundância é composta pela matriz de correlação dos atributos preditivos. O processo então ocorre de maneira iterativa (com “N” iterações), na qual na primeira iteração é escolhida a variável com maior relevância com o *target*. Nas próximas iterações, uma razão “R” é calculada para a continuidade da seleção. Supondo que na i-ésima iteração existam “P” variáveis pré-selecionadas e “Q” variáveis selecionáveis, a razão será dada por: no numerador, a informação mútua entre o *target* e a j-ésima variável do conjunto “Q”; no denominador, a média das correlações entre a j-ésima variável do conjunto “Q” e as variáveis do conjunto “P”. Para cada iteração, será selecionada a variável com maior valor de “R”, até que se atinja a última iteração e se tenha as “N” variáveis selecionadas. Na construção da técnica, o valor de “N” atua como um hiperparâmetro e deve ser previamente estabelecido pelo usuário.

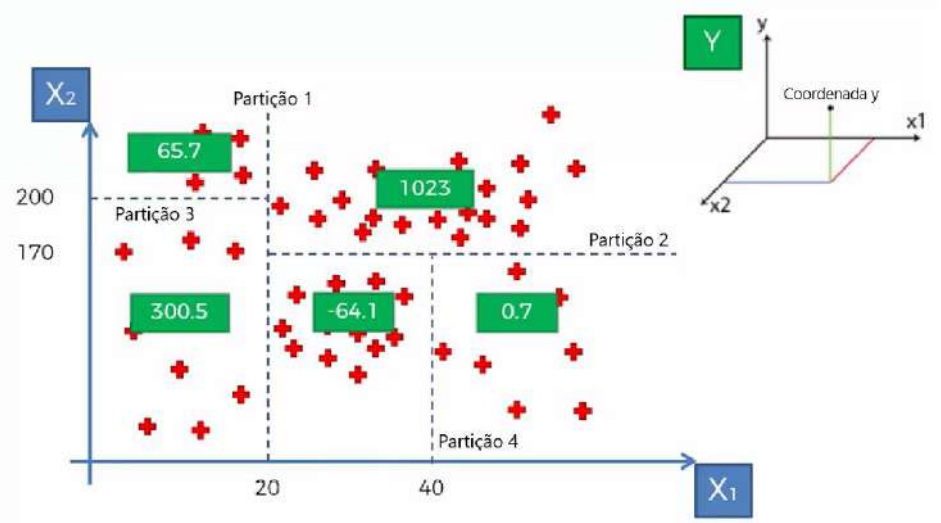
2.6 MODELOS BASEADOS EM ÁRVORES DE DECISÃO

Árvores de decisão consistem em um método não paramétrico de aprendizado supervisionado utilizado para tarefas de regressão e classificação. O objetivo deste tipo de método é estimar o valor de uma variável alvo (*target*) através do aprendizado de regras de decisão que são inferidas a partir de outras variáveis utilizadas como atributos do problema (HASTIE et al., 2009).

A Figura 8 ilustra a formulação de um problema genérico de aprendizado supervisionado com uma variável alvo “y” e dois atributos “X_1” e “X_2”. Nela, o espaço de atributos foi particionado em diferentes regiões de acordo com as regras estabelecidas na Figura 9. Em problemas de regressão, tais partições são realizadas de maneira a minimizar uma função de custo como, por exemplo, o erro quadrático médio (MSE – *Mean Squared Error*), com o número de partições sendo um hiperparâmetro do modelo (SANTANA, 2020). Dessa forma, uma vez realizadas as partições, os valores contidos nas caixas verdes da Figura 8 e da Figura 9 serão referentes à média dos valores de “y” associados a cada ponto contido em cada zona de partição – estes valores, por sua vez, representam a estimativa do *target* para cada zona de partição. O raciocínio detalhado para esse problema é válido para qualquer

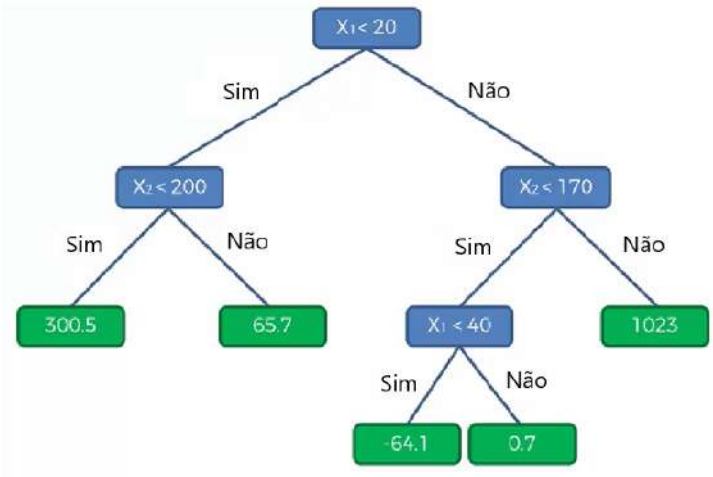
outro problema de aprendizado supervisionado com um *target* “y” e uma matriz “X_M” de “M” atributos.

Figura 8 - Espaço de atributos X_1 e X_2 de um problema genérico de aprendizado supervisionado com target y. Os valores associados a cada zona de partição (expostos em caixas verdes) são referentes à média dos valores de y associados a cada ponto da zona.



Fonte: adaptado de Soni (2022).

Figura 9 - Esquema de árvore de decisão para o problema de aprendizado supervisionado exposto na Figura 8. As regras de partição nas caixas em azul são equivalentes às retas tracejadas da Figura 8. Os valores das caixas verdes equivalem ao resultado das estimativas do *target* para cada zona de partição.



Fonte: adaptado de Soni (2022).

O método de árvores de decisão, por si só, possui vantagens e desvantagens. Dentre as vantagens, cabem citar: i) é um método de fácil compreensão e interpretação (filosofia *white box*); ii) não demanda técnicas de normalização das variáveis de entrada; iii) adaptável a variáveis numéricas e categóricas; iv) possui

habilidade de quantificar a importância marginal de cada atributo na construção de suas regras, e consequentemente, na estimativa do *target* desejado, fazendo com que o algoritmo também possa ser utilizado para seleção univariada de atributos. Por outro lado, algumas desvantagens do modelo que valem ser citadas são: i) árvores muito complexas (e.g., com muitas partições) podem acarretar em super-ajuste (*overfitting*) do modelo; ii) o algoritmo adota uma estratégia de otimização gananciosa (*greedy search*), que busca a melhor solução de forma local (a cada partição) e pode, por consequência, não encontrar a melhor solução global (CHICKERING, 2002); iii) o algoritmo pode criar árvores enviesadas a depender da dominância de classes/comportamentos no conjunto de dados utilizado.

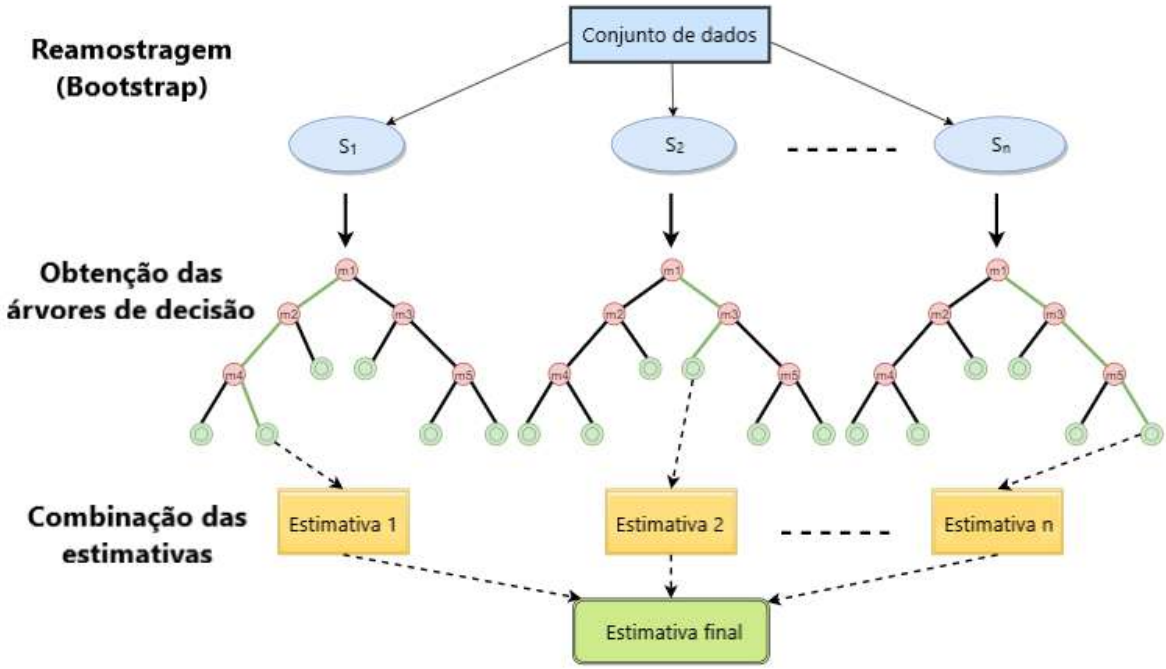
As limitações e desvantagens das árvores de decisão motivaram a criação de novos métodos baseados neste algoritmo com o intuito de vencer ou mitigar os problemas enfrentados pelo método original. Dentre esses novos métodos, dois deles ganharam particular destaque: o modelo *Random Forest* e o modelo *XGBoost*, que são explicados em mais detalhes a seguir.

2.6.1 Modelo Random Forest

Proposta originalmente por Breiman (2001), a Floresta Aleatória ou *Random Forest* é um modelo de *ensemble learning* (SAGI; ROKACH, 2018) que combina a saída de diferentes árvores de decisão a fim de gerar uma estimativa final mais acurada. A Figura 10 ilustra a forma genérica de construção do modelo. Inicialmente, o conjunto de dados original sofre “n” processos de reamostragem através da técnica de *bootstrap* (reamostragem com reposição) (HESTERBERG, 2011), formando “n” novos subconjuntos (S_1 à S_n). Esta estratégia de reamostragem prévia busca solucionar o problema de se obter árvores enviesadas devido à dominância de classes/comportamentos nos dados utilizados. Em seguida, “n” árvores de decisão são treinadas para cada subconjunto obtido, com cada árvore se utilizando de um número “m” de atributos para seu treinamento (sendo “m” menor ou igual ao total de atributos do modelo). Tais estratégias são úteis para evitar a chance de super-ajuste do modelo e para alcançar a melhor solução global para o problema. Além disso, a utilização aleatória dos atributos faz com este modelo seja robusto a situações de alta dimensionalidade, o que o torna ideal para problemas com muitos atributos envolvidos. Por fim, as estimativas individuais são combinadas para gerar uma

estimativa final mais acurada (a partir, por exemplo, da média aritmética dos valores individuais). O processo de reamostragem, ajuste e combinação do modelo é denominado de *bagging* (*bootstrap aggregating*) (SUTTON, 2005).

Figura 10 - Esquema genérico do modelo de *Random Forest*.



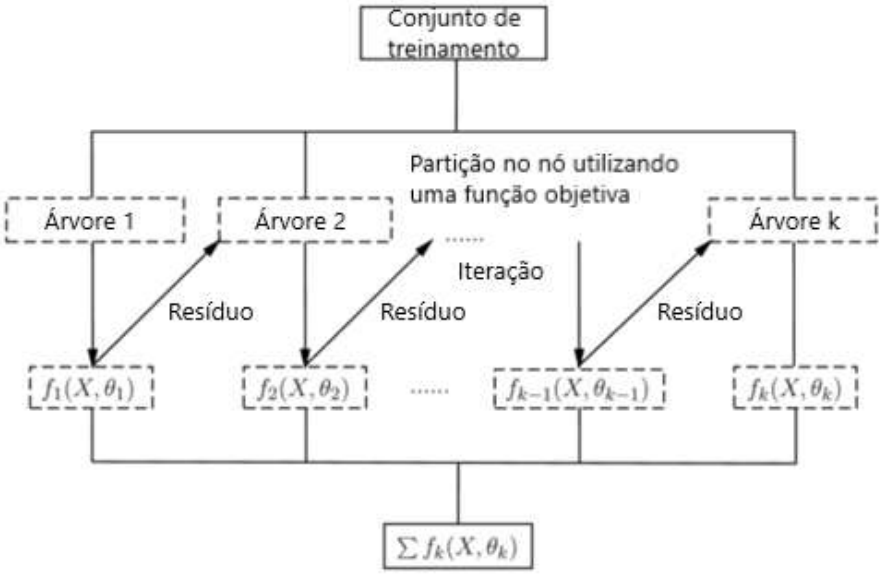
Fonte: adaptado de Dmitrievsky (2022).

2.6.2 Modelo *XGBoost*

O modelo *XGBoost* (*Extreme Gradient Boost*) é um algoritmo de aprendizado de máquina do tipo *ensemble* desenvolvido originalmente por Chen e Guestrin (2016) e voltado para tarefas de classificação e regressão. De forma similar ao *Random Forest*, o *XGBoost* também é um algoritmo baseado em árvores de decisão, entretanto, com a importante diferença de se utilizar da técnica de *boosting* para *ensemble*, diferentemente do *bagging* usado no *Random* Forest. A Figura 11 ilustra o fluxograma geral do modelo *XGBoost*, na qual se observa a característica fundamental por trás do modelo: a cada iteração, o resíduo proveniente da estimativa de uma árvore de decisão é usado como entrada para a próxima árvore. Sendo assim, os erros provenientes do modelo vão sendo atualizados e minimizados a cada iteração através da otimização de uma função de custo (por exemplo, o MSE). Para tornar o modelo mais robusto a *overfitting*, procedimentos de regularização do tipo L1 ou L2

também são adicionados ao modelo. A saída final do modelo é composta por um *ensemble* das saídas individuais, utilizando-se, por exemplo, de uma média aritmética das saídas individuais.

Figura 11 - Esquema genérico do modelo *XGBoost*.

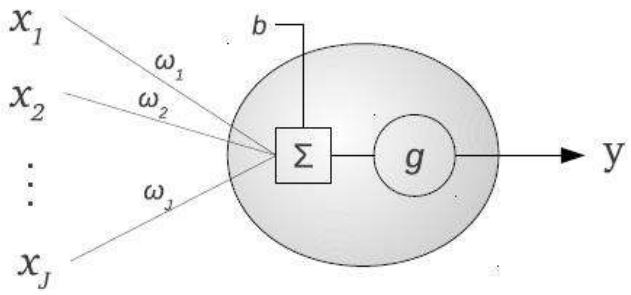


Fonte: adaptado de Zhang et al. (2018).

2.7 MODELOS BASEADOS EM REDES NEURAIS

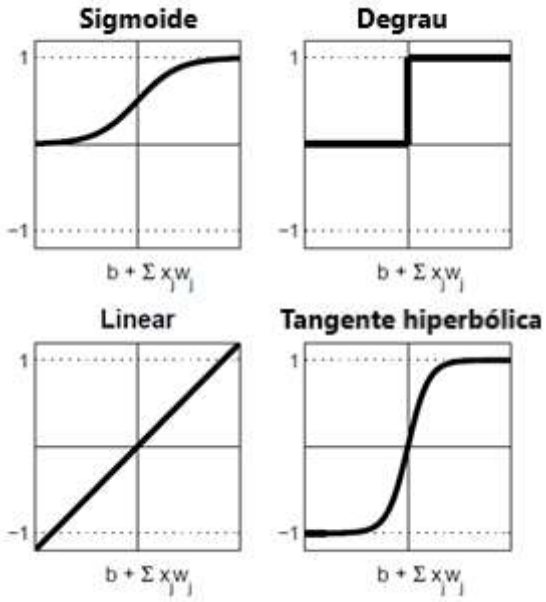
Redes neurais artificiais são modelos que se baseiam na reprodução do comportamento do cérebro humano tendo como referência a interação entre seus neurônios. No âmbito da inteligência artificial, o neurônio é representado como uma unidade chamada perceptron, cujo esquema é representado na Figura 12. Tal unidade consiste em um modelo que realiza a combinação linear dos seus dados de entrada ($\omega_1X_1 + \omega_2X_2 + ... + \omega_JX_J + b$) e transforma o resultado a partir de uma função “g” chamada de função de ativação (por exemplo, aquelas representadas na Figura 13).

Figura 12 - Esquematização do perceptron. X_j se refere à j-ésima variável de entrada; ω_j se refere ao j-ésimo coeficiente (peso sináptico); b se refere ao BIAS; g se refere à função de ativação e y se refere à saída final.



Fonte: Gallego Castillo (2013).

Figura 13 - Exemplos de funções de ativação de um perceptron. X_j se refere à j-ésima variável de entrada; ω_j se refere ao j-ésimo coeficiente (peso sináptico); b se refere ao BIAS.

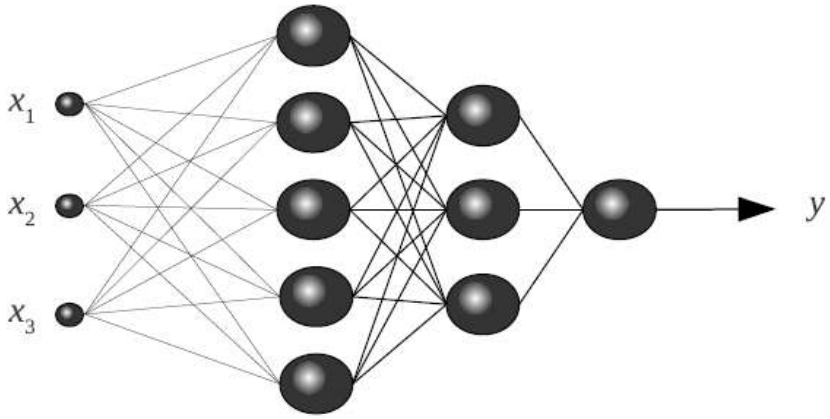


Fonte: adaptado de Gallego Castillo (2013).

Buscando explorar características importantes como a capacidade de aprendizado, generalização e atuação em paralelo dos neurônios, a interconexão entre os perceptrons passou a ser utilizada na literatura, com este tipo de configuração denominada de Perceptron de Múltiplas Camadas (*Multilayer Perceptron – MLP*) (RAMCHOUN et al., 2016), tal como mostra a Figura 14. Ao longo do tempo, diversas outras arquiteturas de redes neurais com diferentes tipos de camadas foram ganhando espaço na literatura. Para a modelagem de séries temporais, dois tipos de camadas se enquadram no atual estado da arte, sendo elas: camadas convolucionais

de 1 dimensão (1D CNN – *Convolutional Neural Network*) (ASSAF et al., 2019; TANG et al., 2020) e camadas do tipo LSTM (*Long Short-Term Memory*) (KARIM et al., 2019; LI et al., 2019). A seguir, explica-se sobre cada uma delas em mais detalhes.

Figura 14 - Arquitetura de uma rede neural MLP.



Fonte: adaptado de Gallego Castillo (2013).

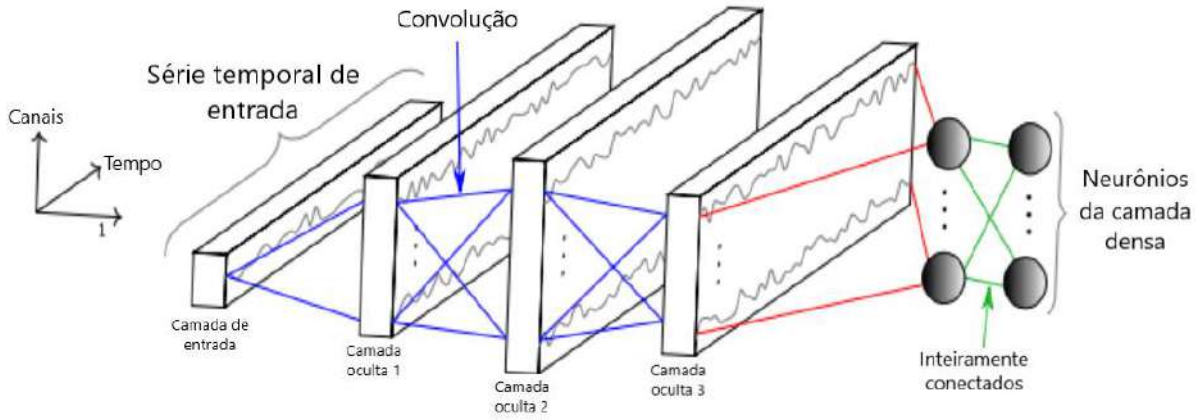
2.7.1 Camadas Convolucionais

Redes neurais convolucionais (CNN – *Convolutional Neural Networks*) são arquiteturas que emergiram inicialmente com forte potencial de aplicação em problemas de processamento de imagens e voz (ABDEL-HAMID et al., 2014; LECUN et al., 1989).

Em camadas convolucionais, o *kernel* executa operações de convolução nos sinais de entrada da camada, além de usar uma função de ativação para gerar os valores de saída. Em aplicações em uma dimensão (usuais para séries temporais), o *kernel* pode ser considerado como uma janela de determinado tamanho que desliza ao longo da série temporal e aplica operações de convolução a cada deslizamento, o que gera um vetor de dados transformados. O processo de convolução é útil para extrair características intrínsecas à série temporal, o que gera novos atributos a serem processados pela rede. Ao aplicar diferentes *kernels* de convolução, a quantidade de novos atributos gerados aumenta, o que pode ser benéfico para aumentar o desempenho do modelo. A Figura 15 ilustra uma aplicação genérica de camadas convolucionais em uma série temporal. As camadas convolucionais são responsáveis por extrair diferentes atributos da camada anterior, aumentando parcialmente a

dimensionalidade do problema. A cada final mostrada na figura é do tipo densa, que nada mais é que uma camada padrão de perceptrons interconectados (tal como nas redes do tipo MLP).

Figura 15 - Esquema de uma rede neural com camadas convolucionais aplicada a uma série temporal. Cada camada convolucional extrai diferentes atributos da camada anterior. A última camada representada é uma cada densa (típicas de redes MLP).



Fonte: adaptado de Ismail Fawaz et al. (2019).

A Equação 7 (WANG et al., 2021) mostra de forma geral a operação aplicada em uma camada convolucional. Nela, $\boldsymbol{K_i^l}$ indica os pesos do i-ésimo *kernel* na camada "l"; $\boldsymbol{b_i^l}$ indica o BIAS do i-ésimo *kernel* na camada "l"; $\boldsymbol{x^l(j)}$ indica a j-ésima região local na camada "l"; $\boldsymbol{y_i^{l+1}(j)}$ indica o *input* do j-ésimo neurônio da camada "l + 1"; a notação "*" é indicativo de um produto interno.

$$y_i^{l+1}(j) = K_i^l * x^l(j) + b_i^l \tag{7}$$

Sendo:

l – contador da camada convolucional;

i – contador do *kernel*;

j – contador da região local;

* – produto interno;

$\boldsymbol{K_i^l}$ – pesos do i-ésimo *kernel* na camada "l";

$\boldsymbol{b_i^l}$ – BIAS do i-ésimo *kernel* na camada "l";

$\boldsymbol{x^l(j)}$ – j-ésima região local na camada "l";

$\boldsymbol{y_i^{l+1}(j)}$ – *input* do j-ésimo neurônio da camada "l + 1".

2.7.2 Camadas LSTM

Redes neurais LSTM (*Long Short-Term Memory*) são um tipo especial de redes neurais recorrentes capazes de aprender dependências de curto e longo prazo. Uma camada LSTM geralmente é composta por algumas células, com cada uma contendo três *gates*: *forget gate, input gate* e *output gate*. A *forget gate* indica quais informações devem ser esquecidas, a *input gate* determina quais *inputs* devem ser lembrados e a *output gate* decide quais informações devem ser entregues na saída. A Figura 16 ilustra a configuração geral de uma camada LSTM, enquanto que as Equações 8 a 13 descrevem o cálculo dos parâmetros envolvidos.

$$f_t = \sigma\big(W_f[h_{t-1}, x_t] + b_f\big) \tag{8}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{9}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{10}$$

$$\widetilde{C_t} = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{11}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{12}$$

$$h_t = o_t * tanh(C_t) \tag{13}$$

Sendo:

f_t – *forget gate*;

i_t – *input gate*;

o_t – *output gate*;

σ – função sigmoide;

W_x – pesos;

b_x – BIAS;

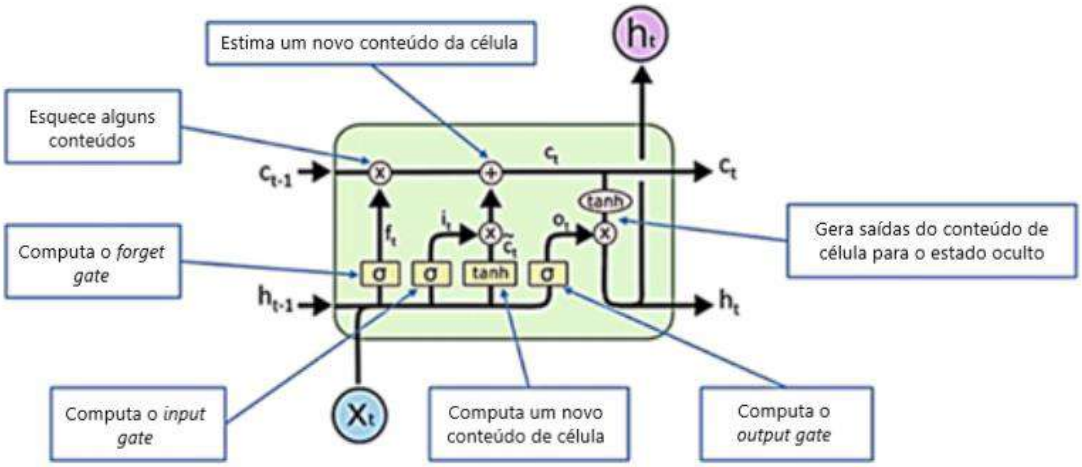
h_{t-1} – saída do nó no *timestamp* t-1;

x_t – valor de entrada no *timestamp* atual;

C_t – estado da célula no *timestamp* atual.

$\widetilde{C_t}$ – valores candidatos ao estado da célula no *timestamp* atual;

Figura 16 - Configuração de uma camada LSTM. f_t é o *forget gate*, i_t é o *input gate*, o_t é o *output gate*, σ é a função sigmoide, h_t é a saída do nó no timestamp atual, e x_t, C_t e $\widetilde{C_t}$ são, respectivamente, o valor de entrada, o estado da célula e os valores candidatos ao estado da célula no *timestamp* atual.



Fonte: Yan et al. (2021).

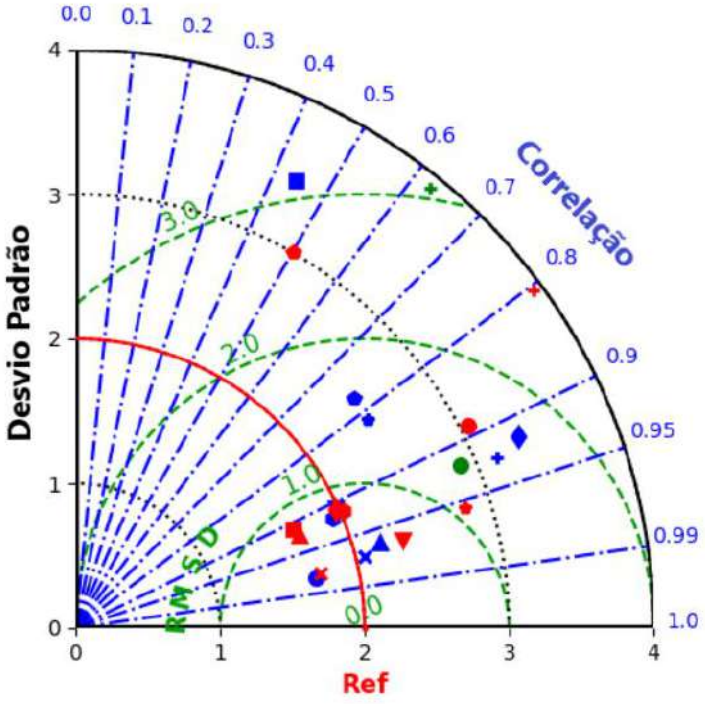
2.8 DIAGRAMA DE TAYLOR

Proposto por Taylor (2001), o diagrama de Taylor é uma forma gráfica de representação combinada de diferentes estatísticos para intercomparação do desempenho de modelos. A Figura 17 ilustra um exemplo genérico do diagrama. Cada ponto do gráfico se refere a um modelo específico. A observação (valor real) é representada pelo ponto “Ref” sobre o eixo horizontal. As coordenadas radiais centradas na origem dizem respeito ao desvio padrão dos modelos (Equação 14). Por outro lado, as coordenadas centradas na observação se referem ao RMSD (*Root Mean Square Deviation*) de cada modelo com a observação (Equação 15). Por fim, a coordenada azimutal centrada na origem se refere ao coeficiente de correlação de Pearson (Equação 16).

No diagrama, quanto menor a distância euclidiana entre o ponto de um modelo e o ponto da observação, melhor é o desempenho deste modelo. Taylor (2001) mostrou que uma forma de quantificar o desempenho geral do modelo fazendo uso de diferentes estatísticos é através do SS4 de Taylor (Equação 17, em que “ρ” é o

coeficiente de correlação entre o modelo e a observação, “s” é o desvio padrão do modelo e “s_{ref}” o desvio padrão da observação). Sendo assim, quanto maior for o SS4 de um modelo, melhor será o seu desempenho.

Figura 17 - Exemplo genérico de um diagrama de Taylor. Cada ponto do gráfico se refere a um modelo específico. A observação (valor real) é representada pelo ponto “Ref” sobre o eixo horizontal.



Fonte: adaptado de Rochford (2022).

$$s = \sqrt{\frac{\sum(x - \bar{x})^2}{N - 1}} \tag{14}$$

$$RMSD = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left[(x - \bar{x}) - \left(x_{ref} - \bar{x}_{ref}\right)\right]^2} \tag{15}$$

$$\rho = \frac{\sum\left[(x - \bar{x}) \cdot \left(x_{ref} - \bar{x}_{ref}\right)\right]}{\sqrt{\sum\left[(x - \bar{x})^2\left(x_{ref} - \bar{x}_{ref}\right)^2\right]}} \tag{16}$$

$$SS4 = \frac{(1 + \rho)^4}{4\left(\frac{s}{s_{ref}} + \frac{s_{ref}}{s}\right)^2} \quad (17)$$

Sendo:

x – dado analisado;

$\bar{x}$ – média aritmética dos “N” valores de “x”;

x_{ref} – dados da observação;

$\bar{x}_{ref}$ – média aritmética dos “N” valores de “x_{ref}”;

s – desvio padrão;

RMSD – *Root Mean Square Deviation*;

ρ – coeficiente de correlação entre o modelo e a observação;

s_{ref} – desvio padrão da observação.

3 REVISÃO DE LITERATURA

Aqui, a revisão de literatura foi estruturada de forma a fornecer a fundamentação teórica e o estado da arte por trás deste trabalho. Sendo assim, três subseções são apresentadas com esse objetivo. A primeira delas traz o embasamento conceitual para as falhas em sistemas mecânicos. A segunda apresenta uma descrição da literatura de uma série de falhas em componentes de aerogeradores. Por último, a terceira fornece o estado da arte por trás da detecção e isolamento de falhas em aerogeradores. A seguir, comenta-se em mais detalhes sobre cada uma delas.

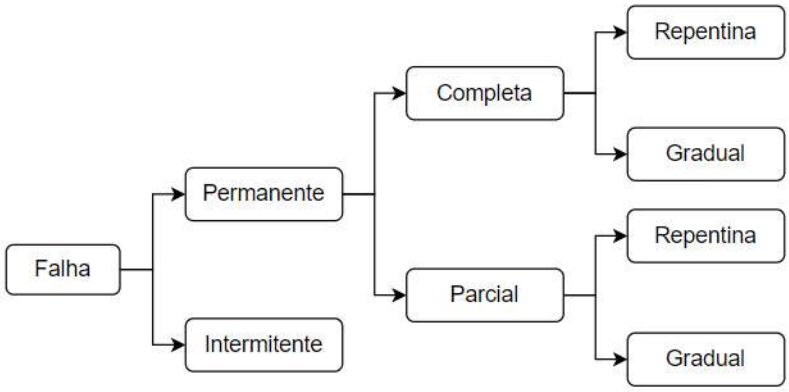
3.1 FALHAS EM SISTEMAS MECÂNICOS

Na literatura, os trabalhos escritos na língua inglesa fazem uso de duas palavras para descrever a falha em sistemas mecânicos: *fault* e *failure*. Isermann (2005) descreve *fault* como “um desvio não permitido do aceitável, usual ou condição padrão de pelo menos uma propriedade característica do sistema”. Por outro lado, o autor descreve *failure* como “uma interrupção permanente da habilidade de um sistema em executar uma função desejada sob condições operacionais específicas”. Para o autor, a evolução do primeiro fenômeno descrito (*fault*) pode levar à ocorrência do segundo (*failure*).

Collacott (2012) descreve *fault* e *failure* como termos equivalentes e propõe uma classificação do fenômeno tal como mostra a Figura 18. Para o autor, uma falha intermitente é aquela que causa a perda de alguma função do componente por um breve período de tempo, com essa função sendo restaurada por si só após a falha. Uma falha permanente pode ocasionar na inabilidade total ou parcial de um componente em executar sua função (falha completa e falha parcial respectivamente), com esta habilidade sendo recuperada apenas mediante reparo ou manutenção. Toda falha permanente – seja ela completa ou parcial – pode ocorrer de forma repentina ou gradual. As falhas de ocorrência gradual são aquelas que evoluem gradativamente ao longo do tempo e podem ser monitoradas e ter seu comportamento estimado (e.g, falhas que afetam a temperatura de componentes do aerogerador). Já as falhas repentinas ocorrem pontualmente no tempo, são difíceis de se monitorar e possuem

comportamento difícil de se estimar (e.g., falhas na regulação de *pitch* do aerogerador).

Figura 18 – Estrutura de classificação de falhas em sistemas mecânicos de acordo com Collacott (2012).

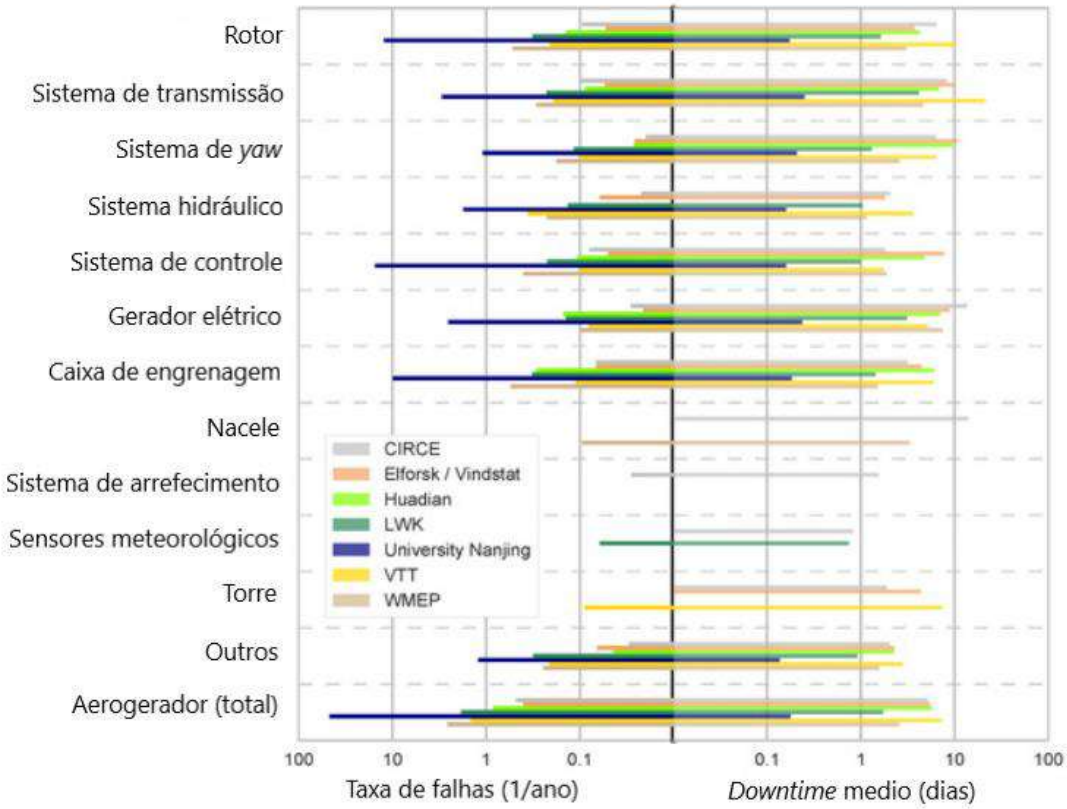


Fonte: adaptado de Collacott (2012).

3.2 FALHAS EM AEROGERADORES DESCRITAS NA LITERATURA

Vários estudos foram realizados nos últimos anos no intuito de identificar taxas de falha e os tempos de inatividade (*downtime*) associados em componentes de aerogeradores (CARTER et al., 2016; JUNG et al., 2015; TAVNER, 2011). Pfaffel et al. (2017) realizaram um compilado de 7 desses estudos e compararam os resultados apresentados por cada um deles (Figura 19). Na figura, é possível observar que os resultados do estudo realizado pela Universidade de Nanjing (barras em azul no gráfico) possuem valor muito superior aos demais no que se refere à taxa anual de falhas de cada componente. Os autores comentam que essa diferença se deve a particularidades específicas do trabalho, mas ressaltam que o *ranking* de taxa anual de falhas por componente se mantém similar para todos os estudos realizados. Nesse sentido, cabe-se destacar o impacto de componentes importantes como a caixa de engrenagem, o gerador elétrico e os demais componentes do sistema de transmissão – estes possuem valores significativos, seja pela taxa anual de falhas ou pelo tempo médio de inatividade (*downtime* médio) quando se está à espera da manutenção.

Figura 19 - Frequência de falhas em subsistemas de aerogeradores e respectivo tempo médio de inatividade (*downtime* médio) causado pelas falhas nesses componentes.



Fonte: adaptado de Pfaffel et al. (2017).

No que diz respeito à particularidade das falhas que acometem os componentes principais do aerogerador, Qiao e Lu (2015a) realizaram uma extensa revisão que está brevemente resumida nas subseções a seguir.

3.2.1 Pás e cubo do aerogerador

Dentre as possíveis falhas causadas no rotor aerodinâmico, destacam-se as assimetrias no rotor e as falhas que acometem as pás, tal como a fadiga, rachaduras, redução de rigidez e deformação plástica (CASELITZ; GIEBHARDT, 2005; GONG; QIAO, 2012). A assimetria no rotor é usualmente causada por erros no ângulo de *pitch* (assimetria aerodinâmica) ou pelo desbalanceamento de massa nas pás (GARDELS; QIAO, 2010; ZENG et al., 2013). A fadiga nas pás é causada pelo envelhecimento do material e pelas cargas variáveis recebidas no impacto com o vento. A longo prazo, a fadiga pode resultar em delaminação na estrutura da pá, o que acarreta em redução na sua rigidez. Da mesma forma, a fadiga a longo prazo também pode causar

rachaduras na superfície e na estrutura interna das pás. A deformação plástica na pá é geralmente causada por cargas desbalanceadas de longo prazo e redução na sua rigidez.

3.2.2 Caixa de engrenagens

Caixas de engrenagens são consideradas como um dos subsistemas mais suscetíveis a falha num aerogerador, contribuindo com cerca de 20% do tempo de parada da máquina (devido a falhas) ao longo de sua vida útil (RIBRANT; BERTLING, 2007). Vários fatores contribuem para as falhas em caixas de engrenagens, tais como: erros de dimensionamento, erros de fabricação, erros de instalação, desalinhamento, sobrecargas de torque, degradação de superfície e fadiga. Outras falhas severas em engrenagens incluem rachaduras, quebra e fraturas nos dentes da engrenagem.

3.2.3 Rolamentos

Rolamentos são elementos utilizados em vários subsistemas do aerogerador, como o rotor, eixo principal, gerador elétrico, caixa de engrenagens, sistema de *pitch* e sistema de *yaw*. Tais elementos estão sujeitos a falhas na sua pista interna, pista externa, gaiola e elementos rolantes. Tais falhas, em geral, aparecem inicialmente na forma de degradação de superfície em certas partes, desenvolvendo-se em seguida para falhas mais severas, como fadiga, quebra e rachaduras. Os detritos produzidos por uma falha no rolamento causarão a abrasão de outros componentes do subsistema, como as engrenagens, por exemplo. Problemas relacionados à lubrificação são a principal causa de falhas nesse componente (LEITE, 2018).

3.2.4 Sistema hidráulico

O sistema hidráulico é amplamente utilizado em aerogeradores. Nos sistemas de *pitch* e *yaw*, ele é responsável por entregar a potência hidráulica para os motores que realizam o controle angular. Tal sistema também é responsável por controlar o freio mecânico do aerogerador. O sistema hidráulico pode sofrer diferentes falhas, como o vazamento de óleo e o bloqueio da válvula deslizante. Tais falhas podem ser

diagnosticadas utilizando sinais adquiridos por meio de sensores – por exemplo, os de nível e de pressão (WU et al., 2011; CHEN et al., 2013).

3.2.5 Gerador e motores

As falhas nos motores e geradores elétricos podem ser classificadas como falhas elétricas (e.g., danos no isolamento do estator ou do rotor) e falhas mecânicas (e.g., falha no rolamento do gerador). Falhas nos enrolamentos, como o curto circuito de bobinas, são as mais comuns de ocorrerem em máquinas de indução utilizadas nos aerogeradores (POPA et al., 2003). A assimetria no campo magnético geralmente é notada durante uma falha nos enrolamentos (YANG et al., 2009). Tais falhas também causam um aumento na temperatura nestes componentes (ZAHER et al., 2009). Falhas de circuito aberto em estatores irão modificar o espectro das correntes de linha dos estatores, bem como sua potência instantânea. Desbalanceamento elétrico é outra falha comum em máquinas elétricas. No rotor, por exemplo, o desbalanceamento elétrico causa vibração no eixo. Da mesma forma, o desbalanceamento elétrico no estator causa mudança na corrente e na potência de saída do gerador (POPA et al., 2003).

3.2.6 Sensores

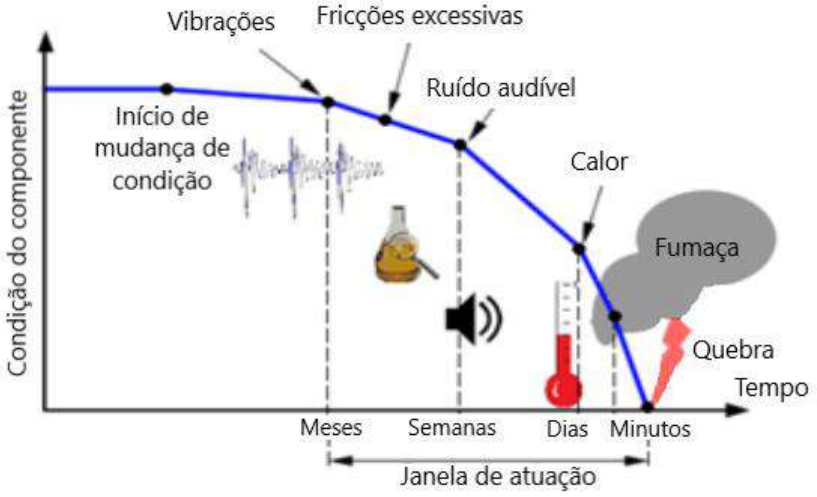
Uma variedade de sensores, como anemômetros, birutas, termômetros e sensores de nível de óleo estão instalados em aerogeradores para monitoramento e controle da máquina. Tais sensores estão sujeitos a várias falhas, como mal funcionamento do *hardware*, falhas físicas no elemento sensor ou falhas no processamento e aquisição de dados. Estima-se que as falhas em sensores constituem mais de 14% das falhas em aerogeradores (RIBRANT; BERTLING, 2007). Uma falha num sensor pode causar degradação na performance do aerogerador, falha nos subsistemas elétrico, mecânico e de controle ou até mesmo o desligamento forçado do aerogerador (PEDROSA, 2016).

3.3 ESTADO DA ARTE EM DETECÇÃO E ISOLAMENTO DE FALHAS EM AEROGERADORES

Com relação à modelagem e estimativa do comportamento da falha em um sistema mecânico, Gao, Cecati e Ding (2015) descrevem dois processos importantes de se mencionar: sua detecção e o seu isolamento. De acordo com os autores, a detecção de falhas é o processo responsável por indicar se existe um mau funcionamento ou uma falha no sistema e informar o tempo em que o fenômeno foi detectado. Por outro lado, o isolamento da falha é o processo pelo qual se informa a sua localização no sistema de interesse (e.g., qual componente do aerogerador está sendo afetado). Essa informação pode dar maiores indícios da causa raiz do fenômeno, o que auxilia a equipe de O&M no planejamento do reparo ou da manutenção do componente acometido pela falha.

A Figura 20 mostra o tempo de manifestação de uma falha em vários tipos de sinais extraídos de um sistema mecânico. Nota-se que os sinais de vibração sentem a influência da falha com antecedência de meses. Embora não tão comum, alguns sistemas SCADA possuem sinais de vibração embarcados em seu monitoramento. Em todo caso, a grande maioria possui medições de temperatura, o que possibilita a detecção de falhas com dias de antecedência.

Figura 20 - Sinais que detectam o desenvolvimento de uma falha mecânica.



Fonte: adaptado de Tchakoua et al. (2014).

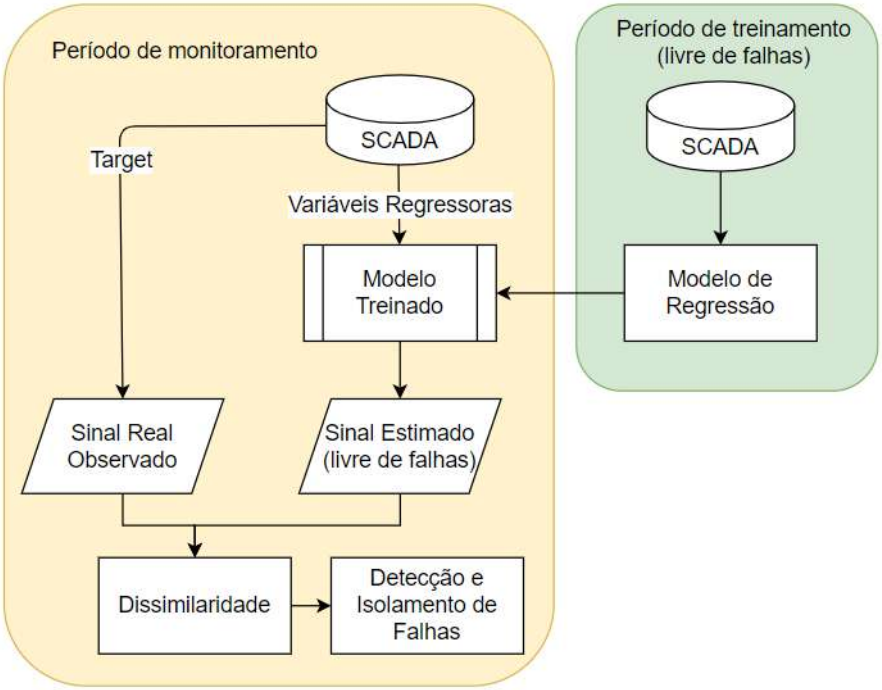
Tautz-Weinert e Watson (2016) trazem vários grupos de técnicas de detecção de falhas em aerogeradores, dentre as quais, destacam-se as técnicas de modelagem do comportamento normal (NBM – *Normal Behaviour Modelling*). Nesse trabalho, entende-se por comportamento normal (ou comportamento esperado) como aquele associado à operação da máquina em condições saudáveis e livre de qualquer tipo de falhas. Entretanto, para evitar confusões do leitor, optou-se por renomear este termo como “Comportamento Livre de Falhas”, a fim de evitar associações errôneas entre este comportamento (o livre de falhas) e o comportamento referente a uma distribuição gaussiana (também chamada de distribuição normal). Note que não necessariamente o comportamento livre de falhas possui uma distribuição gaussiana – o que vai definir a distribuição estatística do comportamento livre de falhas é a variável alvo que se deseja avaliar. Dito isso, por ser o foco deste trabalho, algumas contribuições da literatura referentes à modelagem do comportamento livre de falhas em aerogeradores são expostas a seguir.

A modelagem do comportamento livre de falhas visa estimar o comportamento esperado do sinal de interesse e compará-lo com seu valor observado durante o monitoramento em tempo real. Para isso, um modelo de regressão é treinado com dados do sistema SCADA em períodos livres de qualquer tipo de falha ou anomalia. Uma vez treinado e com a parametrização final definida, este mesmo modelo é utilizado durante o monitoramento em tempo real para estimar o comportamento livre de falhas do sinal de interesse e compará-lo com o seu valor real no mesmo instante de tempo. A comparação entre os sinais é feita mediante alguma métrica de dissimilaridade – por exemplo, o resíduo entre os sinais –, com uma falha sendo detectada e isolada sempre quando o valor da dissimilaridade superar um limiar pré-estabelecido (limiar de detecção de falha). A Figura 21 ilustra o processo comentado acima.

A Figura 22, por sua vez, mostra a aplicação da metodologia supracitada em um estudo de caso envolvendo uma falha no rolamento principal de um aerogerador (WANG et al., 2019). Na figura à esquerda, é possível observar o descolamento entre os sinais vermelho (valor real observado) e azul (valor estimado), o que indica o aumento da temperatura decorrente da evolução de uma falha no componente. À direita, a distância de Mahalanobis (MCLACHLAN, 1999) foi escolhida como métrica de dissimilaridade, sendo representada pelo sinal azul, enquanto que a reta vermelha tracejada representa o limiar de detecção de falha. Sendo assim, a falha foi detectada

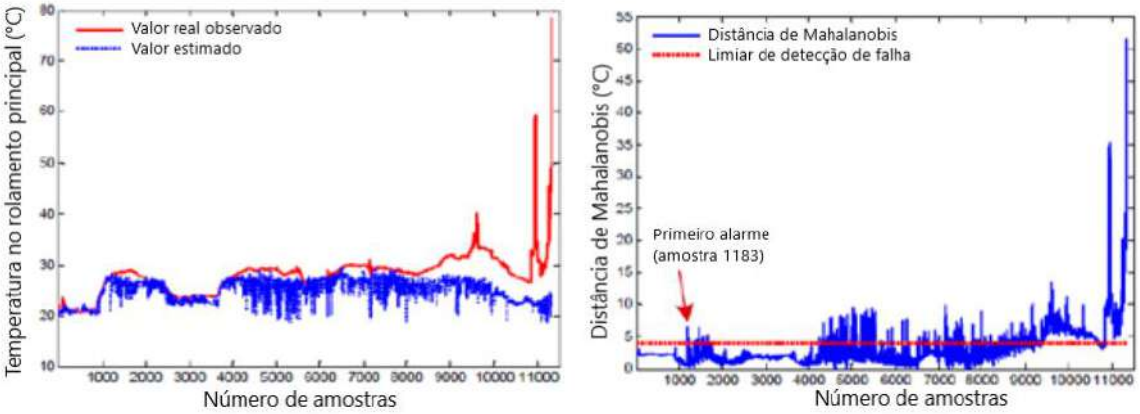
na amostra de número 1183, com o processo isolamento informando que ela estava localizada no rolamento principal do aerogerador.

Figura 21 - Fluxograma geral da modelagem do comportamento livre de falhas em aerogeradores. Um modelo de regressão é treinado com dados do sistema SCADA em períodos livre de falhas. Uma vez parametrizado, este modelo é utilizado para estimar o comportamento livre de falhas durante o período de monitoramento. O sinal estimado é então comparado com o sinal real observado a partir de uma métrica de dissimilaridade.



Fonte: o autor (2022).

Figura 22 - Á esquerda, a comparação entre o sinal real observado (vermelho) e o sinal estimado (azul) da temperatura em um rolamento principal. À direita, uma medida de dissimilaridade entre os dois sinais (azul) e o limiar de detecção de falhas (vermelho).



Fonte: adaptado de Wang et al. (2019).

Inicialmente, alguns modelos estatísticos utilizados na literatura para a modelagem deste tipo de comportamento foram: modelos lineares e polinomiais (GARLICK et al., 2009; WILKINSON et al., 2014); redes neurais artificiais (GARCIA et al., 2006; BRANDÃO et al., 2010) e logica difusa (JANG et al., 1997; SCHLECHTINGEN et al., 2013).

Atualmente, com a consolidação das técnicas de aprendizado de máquina (*machine learning*) e aprendizado profundo (*deep learning*), diferentes técnicas são utilizadas para a modelagem do comportamento livre de falhas dos componentes monitorados. Stetco et al. (2019) realizaram uma extensa revisão acerca dos métodos de aprendizado de máquina utilizados no monitoramento de condição de aerogeradores[4]. Alinhados à filosofia atual das metodologias de aprendizado de máquina, os autores subdividem a revisão em diferentes etapas: i) aquisição de dados e pré-processamento; ii) seleção e extração de atributos; iii) seleção de modelos; iv) métricas de validação.

Dados extraídos de aerogeradores compartilham dos desafios impostos em ambientes "*big data*". Baseado no modelo dos "4 V's" (DONG; SRIVASTAVA, 2013), dados eólicos atuais possuem as seguintes características: Volume, Velocidade, Variedade e Veracidade. Por esse motivo, vários autores desenvolvem seus trabalhos utilizando ferramentas de mercado que viabilizam a solução destes problemas. Por exemplo, Canizo et al. (2017) desenvolveram seus trabalhos de manutenção preditiva utilizando tecnologias como: i) HDFS – sistema primário de armazenamento de dados utilizado por aplicações Hadoop (KARUN; CHITHARANJAN, 2013); ii) Spark – sistema de processamento distribuído utilizado em ambientes *big data* (SPARK, 2018); iii) Apache Mesos – gerenciador de *clusters* em ambiente distribuído (FRAMPTON, 2018); iv) Zookeeper – servidor de código aberto para coordenação distribuída de aplicativos em nuvem (JUNQUEIRA; REED, 2013).

Com relação às técnicas de pré-processamento, a remoção de *outliers* dos dados avaliados desempenha um papel importante na modelagem. Marti-Puig et al. (2017) investigaram os efeitos da remoção de outliers na detecção de falhas em aerogeradores. Utilizando técnicas de filtro como *studentized deviate test* (RYU et al., 2021) e identificador Hampel (YAO et al., 2019), os autores mostraram que a remoção de *outliers* reduz o erro na fase de treinamento dos modelos. Por outro lado, a

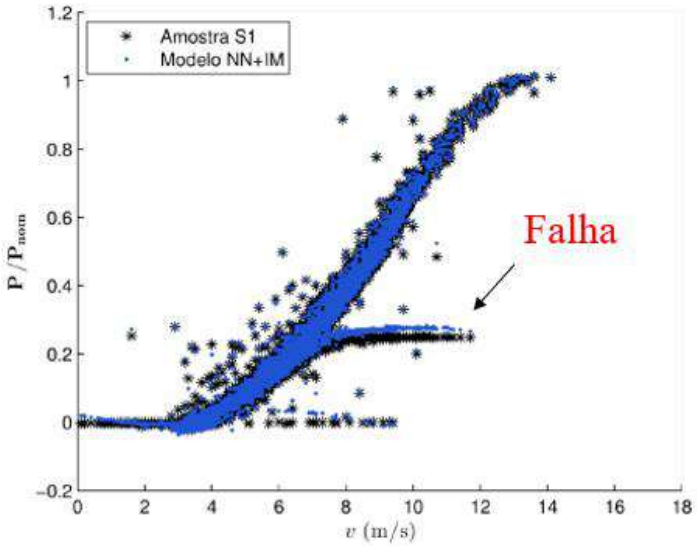
[4] O monitoramento de condição é o processo pelo qual se monitoram os sinais de interesse do aerogerador com o intuito de realizar detecção de falhas e posterior manutenção preditiva.

remoção no conjunto de teste pode fazer com que parte do processo evolutivo da falha seja desconsiderado, tendo em vista que estes *outliers* podem ocorrer devido ao comportamento com falha. Leahy et al. (2019) apresentaram uma revisão acerca dos problemas relativos à qualidade dos dados no monitoramento de condição de aerogeradores. Os autores mostraram que diversos trabalhos na literatura relatam problemas na qualidade dos dados utilizados, com muitos recorrendo a metadados (*logs*, alarmes, registros etc.) como solução para eliminação de dados errôneos (MARVUGLIA; MESSINEO, 2012; PARK et al., 2014). Outros autores se utilizam de técnicas estatísticas para remoção de *outliers*, principalmente na curva de potência de aerogeradores (ZHAO et al., 2017a).

Os métodos de seleção estatística de atributos podem ser divididos em três tipos: métodos *wrapper*, métodos *embedded* e métodos *filter*. Marti-Puig et al. (2019) compararam diferentes técnicas de seleção de atributos utilizando dados do sistema SCADA de aerogeradores, sendo essas: Informação Mútua, Informação Mútua Condicional, Informação Mútua Conjunta, Mínima Redundância Máxima Relevância, *Double Input Symmetrical Relevance*, *Conditional Mutual Info Maximisation* e *Interaction Capping*. Dentre as técnicas utilizadas, o melhor resultado obtido foi utilizando Informação Mútua Condicional, enquanto que o pior foi utilizando Informação Mútua.

A seleção física de atributos é o procedimento pelo qual se retiram do conjunto de dados aqueles atributos que, além do *target*, podem ter seu comportamento livre de falhas modificado pela falha estudada. Felgueira et al. (2019) mostraram que a seleção física de atributos desempenha um papel importante na acurácia de modelos de detecção de falhas. Os autores apontaram que modelos que fazem uso de variáveis com relação de simultaneidade com o *target* tendem a reproduzir o comportamento da falha durante a operação. Tal fato impede que se observe o aumento da dissimilaridade entre os sinais, o que consequentemente dificulta a detecção da falha. Por simultaneidade, entende-se o processo pelo qual o comportamento de uma variável “x” repercute diretamente no comportamento de outra variável “y”, e vice-versa. Pedrosa (2016) enfrentou esse tipo de problema ao modelar a curva de potência de um aerogerador sem realizar a seleção física de atributos. A Figura 23 mostra que a utilização de variáveis que possuem relação física com a potência (e.g., a corrente elétrica) fez com que a falha fosse reproduzida pelo modelo (os pontos em azul reproduzem a regulação de potência).

Figura 23 - Modelo de curva de potência (pontos em azul) reproduzindo o comportamento dos dados observados (pontos em preto), inclusive durante o período de regulação.



Fonte: adaptado de Pedrosa (2016).

Com relação aos modelos regressivos utilizados, Stetco et al. (2019) mostram que a grande parte é voltada para a modelagem do comportamento livre de falhas dos componentes monitorados. Orozco et al. (2018) utilizaram modelos regressivos para modelar o comportamento livre de falhas de dados de temperatura de componentes de um aerogerador. Além dos modelos clássicos, como regressão linear e polinomial, os autores utilizaram técnicas de aprendizado de máquina como a floresta aleatória (*random forest*), além de técnicas de aprendizado profundo como as redes neurais. Guo et al. (2017), por sua vez, utilizaram redes neurais recorrentes para construir um indicador de saúde operacional para manutenção preditiva em rolamentos de aerogeradores. Ulmer et al. (2020) se utilizaram de redes neurais convolucionais para detectar falhas em rolamentos de geradores de aerogeradores, com o método retornando melhores resultados em comparação com redes neurais MLP (*Multi-Layer Perceptron*). Trizoglou et al. (2021) utilizaram o método de *XGBoost* (*Extreme Gradient Boosting*) para detectar falhas a partir do sinal de temperatura do gerador elétrico de um aerogerador *offshore*. Os autores mostraram que tal método foi capaz de superar a acurácia de redes neurais LSTM *(Long Short Term Memory*), além de demandar um menor tempo para treinamento do modelo. Zhang et al. (2018) também utilizaram do método *XGBoost* para detectar diferentes falhas em aerogeradores. O trabalho mostrou que uma seleção prévia de atributos utilizando o método de *Random*

Forest foi capaz de criar um estimador final mais robusto e menos propenso a *overfitting*.

Stetco et al. (2019) mostram que estratégias usuais de validação cruzada como a *k-fold* ainda são negligenciadas por alguns autores na literatura de detecção de falhas. No que tange a avaliação de desempenho dos modelos, métricas usuais voltadas para modelos de regressão e classificação são utilizadas. Dentre as métricas voltadas para regressão, destacam-se: MAE, MAPE, sMAPE, RMSE e R^2 (MARVUGLIA; MESSINEO, 2012). As métricas voltadas para modelos de classificação são utilizadas ao final do procedimento de detecção de falhas, buscando entender a capacidade da metodologia em classificar instantes com falha e sem falha. Sendo assim, destacam-se: Acurácia, Precisão, *Recall*, Especificidade e F1 *score* (LI et al., 2021).

Com relação às técnicas de isolamento de falhas, alguns trabalhos forneceram importantes contribuições ao estado da arte no âmbito da energia eólica. Por exemplo, Li et al. (2018) utilizaram o modelo *Random Forest* para o isolamento de falhas em aerogeradores simuladas computacionalmente. O modelo foi responsável por classificar qual dos diferentes resíduos gerados possuía maior relação com a falha detectada, com cada resíduo sendo referente à diferença entre o valor observado de um sensor do aerogerador e seu respectivo valor estimado em regime de CLF.

Pedrosa (2016) realizou o isolamento de falhas que repercutem na *performance* do aerogerador a partir do cálculo da informação mútua entre o resíduo do sinal de potência e as demais variáveis do SCADA, com a localização da falha sendo referente à localização do sensor com maior grau de informação mútua obtido. Utilizando de uma filosofia similar, alguns autores aplicaram técnicas de seleção estatística de variáveis a fim de observar quais sinais contribuíam diretamente na reprodução do resíduo em períodos de falha. Dentre as técnicas utilizadas, cabem citar: *foward selection* (WANG; JIANG, 2009), regressão LASSO (ZOU; QIU, 2009) e técnicas baseadas em reconstrução de falhas (LIU; DU; YE, 2021).

A grande maioria dos trabalhos da literatura (por exemplo, os aqui citados anteriormente) foca em apresentar soluções para a melhoria da acurácia dos modelos de detecção e isolamento de falhas. No entanto, pouco se fala acerca dos desafios enfrentados ao se colocar esses modelos em operação no âmbito de ferramentas operacionais. Por exemplo, a disponibilidade das estimativas que a ferramenta realiza ao longo do tempo é crucial para o sucesso no auxílio às tomadas de decisão da

equipe de O&M. Nesse sentido, uma ferramenta com alto nível de acurácia e baixo nível de disponibilidade tem pouca serventia de utilização.

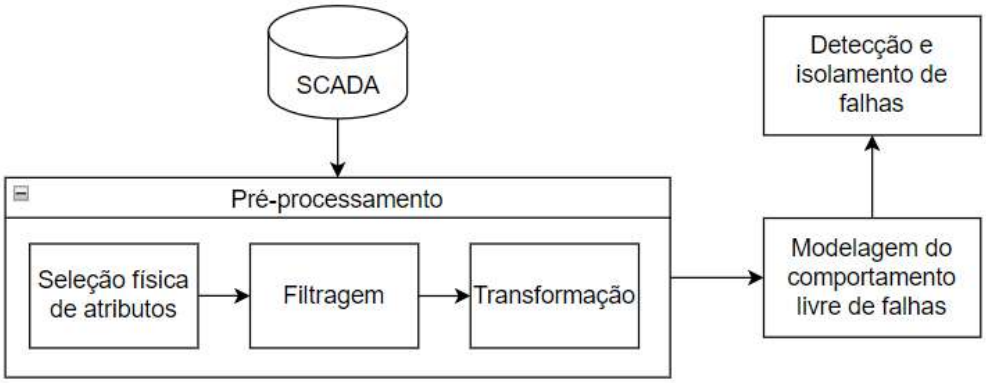
Um dos principais fatores que ocasionam a indisponibilidade de estimativas de uma ferramenta é a ausência de dados em uma ou mais variável de entrada do modelo utilizado. Nesse sentido, a forma mais usual na literatura de se mitigar este problema é mediante a estimativa dos dados ausentes a partir de técnicas de imputação de dados (MARTINEZ-LUENGO; SHAFIEE; KOLIOS, 2019; MORSHEDIZADEH et al., 2017; QU et al., 2020).

No escopo da imputação de dados, uma das formas de se realizar a estimativa do valor “x” ausente de uma variável “X” é mediante uma função do tipo $\hat{x} = f(S)$, sendo “S” um subconjunto de “n” variáveis distintas e “f” um modelo qualquer de regressão. No entanto, esse tipo de estimativa também pode sofrer com o problema da indisponibilidade, bastando apenas que haja a ausência de pelo menos uma variável do subconjunto “S”. Para mitigar este obstáculo, existem técnicas de imputação que fazem uso apenas da própria variável “X” para recuperar seus dados ausentes – por exemplo, substituindo-os pela média aritmética dos dados presentes. Contudo, esse tipo de solução pode levar a problemas na acurácia no modelo que se alimenta de tais dados, tendo em vista que essas técnicas tendem a gerar estimativas enviesadas (SOLEY-BORI, 2013). Dessa forma, fica evidente que outras soluções podem contribuir para o estado da arte no que diz respeito à disponibilidade de estimativas de ferramentas de detecção e isolamento de falhas em aerogeradores.

4 METODOLOGIA

A Figura 24 mostra o diagrama geral da metodologia aqui proposta. Nele, 4 fases principais são apresentadas: i) Fonte de dados do problema; ii) Pré-processamento dos dados (que por sua vez é subdividido em 3 subfases: seleção física de atributos, filtragem e transformação); iii) Modelagem do comportamento livre de falhas; iv) Detecção e isolamento de falhas. A seguir, cada fase da metodologia será apresentada em maiores detalhes.

Figura 24 - Fluxograma geral da metodologia.



Fonte: o autor (2022).

4.1 FONTE DE DADOS

Os dados da central eólica foram fornecidos em regime de livre acesso pela EDP (Energias de Portugal), para o concurso de detecção de falhas exposto em seu *website* (EDP, 2019). Foram fornecidos dados de cinco aerogeradores localizados numa central offshore na costa da África, tal como mostra a Figura 25. Os aerogeradores numerados na figura à direita se referem àqueles que possuem dados disponíveis em regime de livre acesso. Os números de identificação de cada máquina são equivalentes aos fornecidos pela EDP. Os dados são relativos ao sistema SCADA da central, com cerca de 81 variáveis continuamente monitoradas durante 2 anos (2016 e 2017), obtidos a cada 10 minutos, o que totaliza 105.120 registros para cada variável. Além disso, alarmes relativos à operação da central e das máquinas também são fornecidos. Estes auxiliarão no pré-processamento e filtragem dos dados.

Figura 25 - Localização da central eólica e dos respectivos aerogeradores. Visão global, à esquerda, com zoom local, à direita.

Fonte: EDP (2019).

Os aerogeradores possuem as mesmas características técnicas. A potência nominal é de 2 MW, com velocidade de arranque de 4 m/s, velocidade nominal de 12 m/s e velocidade de parada de 25 m/s. O sistema de transmissão consiste em uma caixa de engrenagens do tipo planetária. O gerador elétrico é do tipo assíncrono. Dentre os componentes do aerogerador, 5 são considerados de maior interesse pela empresa de serem monitorados, sendo eles: gerador elétrico, rolamento do gerador elétrico, caixa de engrenagens, transformador e grupo hidráulico.

Com respeito às falhas que acometeram a central ao longo dos dois anos, 28 registros entre os cinco aerogeradores foram observados pela equipe de operação. Dentre estas falhas, a metodologia aqui proposta visa abordar algumas daquelas cujo a temperatura nos componentes analisados sofre algum tipo de alteração previamente ao evento de falha. Em geral, falhas de natureza mecânica podem gerar um aumento gradual na temperatura, que é possível de ser identificado pelos modelos de detecção de falhas. Falhas de natureza elétrica também podem gerar aumentos de temperatura nos componentes analisados, apesar da dinâmica (velocidade) da falha ser mais rápida quando comparadas às falhas mecânicas.

Uma vez caracterizado o escopo das falhas que se deseja analisar, a escolha da variável alvo (*target*) deve ser realizada. Esta variável é aquela que deverá ser reproduzida pelo modelo de comportamento livre de falhas, seguindo uma estratégia

de aprendizado supervisionado. A escolha do *target* deve ser feita em linha com a falha que se deseja detectar. Por exemplo, caso o interesse seja na análise de uma falha no transformador, a variável escolhida como *target* pode ser a temperatura em uma das fases do transformador. Em um caso operacional, é benéfico que se faça a escolha de múltiplos *targets* para cada componente de interesse, a fim de ampliar o escopo de sinais monitorados que possam sofrer influência da falha. Sendo assim, todo o *pipeline* da metodologia deve ser rodado para cada um dos *targets*, de forma paralela, gerando múltiplas saídas de detecção e isolamento de falhas.

Esta estratégia também é interessante no sentido de mitigar o problema da indisponibilidade nas estimativas de cada modelo. Quanto mais modelos entregando estimativas ao longo do tempo, menor a chance da ferramenta não entregar nenhuma informação em cada instante de tempo (ou seja, caso um dos modelos sofra com ausência de dados, é possível que outros não tenham esse problema).

Sendo assim, a Tabela 1 ilustra os possíveis *targets* a serem escolhidos para cada componente de interesse do aerogerador. Tais variáveis foram elencadas pensando nas falhas que repercutem na temperatura dos componentes. Em uma ferramenta operacional, é desejável que se tenham modelos para todos os *targets*, tal como comentado anteriormente.

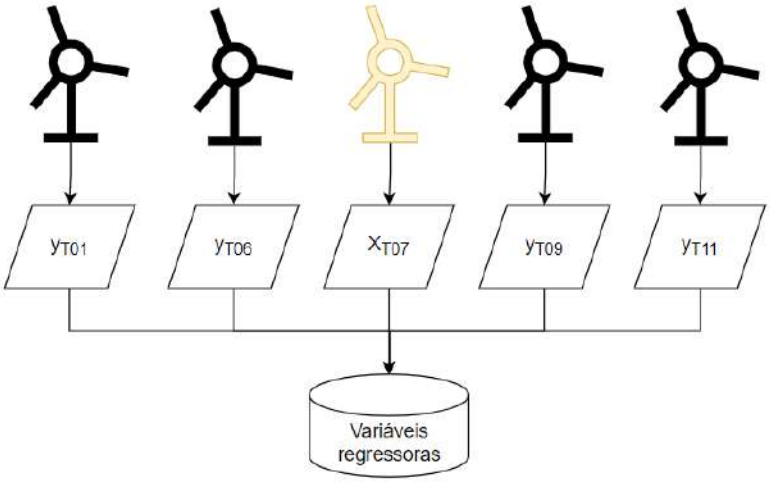
Tabela 1 - Possíveis targets para cada componente de interesse do aerogerador. As variáveis foram elencadas por representar a temperatura em locais específicos de cada componente.

Componente	**Targets**	**Localização do sensor de temperatura**
Gerador Elétrico	*gen_phase1_temp_avg*	Fase 1 do gerador
	gen_phase2_temp_avg	Fase 2 do gerador
	gen_phase3_temp_avg	Fase 3 do gerador
	gen_slipring_temp_avg	*Slipring* do gerador
Rolamento do Gerador	*gen_bear_temp_avg*	Rolamento do gerador (*non-drive end)*
	gen_bear2_temp_avg	Rolamento do gerador (*drive end)*
Gearbox	*gear_bear_temp_avg*	Rolamento da *gearbox*
	gear_oil_temp_avg	Óleo da *gearbox*
Transformador	*hvtrafo_phase1_temp_avg*	Fase 1 do transformador
	hvtrafo_phase2_temp_avg	Fase 2 do transformador
	hvtrafo_phase3_temp_avg	Fase 3 do transformador
Grupo Hidráulico	*hyd_oil_temp_avg*	Óleo do grupo hidráulico

Fonte: o autor (2022).

A Figura 26 mostra a montagem do conjunto de atributos, ou em outras palavras, do conjunto de variáveis regressoras (i.e., variáveis de entrada em um modelo de regressão) para cada modelo criado. A numeração de cada máquina segue o padrão fornecido pela EDP. Nesse caso, o aerogerador em análise é o T07, que está pintado em amarelo na figura. "X_{T07}" significa o conjunto de todas as variáveis do sistema SCADA do aerogerador T07, exceto a variável "y", que consiste no target do problema. Ou seja, caso o *target* seja a temperatura no rolamento do gerador do aerogerador T07, as variáveis regressoras do problema serão todas as variáveis do sistema SCADA do aerogerador T07 (exceto o *target*), além da temperatura no rolamento do gerador de todas as demais máquinas (i.e., T01, T06, T09 e T11). É importante salientar que é formado um conjunto diferente desse tipo para cada *target* utilizado.

Figura 26 - Esquema de montagem do conjunto de variáveis regressoras de cada modelo criado. Aqui, o aerogerador de análise é o T07 (as numerações são de acordo com as fornecidas pela EDP). "X_{T07}" significa o conjunto de todas as variáveis do sistema SCADA do aerogerador T07, exceto a variável "y", que consiste no *target* do problema.



Fonte: o autor (2022).

4.2 PRÉ-PROCESSAMENTO

Uma vez adquiridos os dados observacionais, é importante que seja realizado um tratamento prévio na base de dados que irá ser submetida ao treinamento dos modelos de comportamento livre de falhas e detecção/isolamento de falhas. O intuito principal é a remoção de possíveis *outliers* e comportamentos que não são de

interesse para o processo de aprendizado. Com isso, a fase de pré-processamento é subdividida em 3 subfases, sendo elas: seleção física de atributos, filtragem e transformação. A seguir, comenta-se em maiores detalhes a respeito de cada uma delas.

4.2.1 Seleção física de atributos

De forma geral, considerando que a variável alvo represente um comportamento de algum subsistema da máquina, deve-se retirar da análise todas aquelas variáveis que apresentem relação de simultaneidade com esta variável, uma vez que a modelagem de comportamento livre de falhas não pode ser afetada pela falha no processo de aprendizagem.

4.2.2 Filtragem

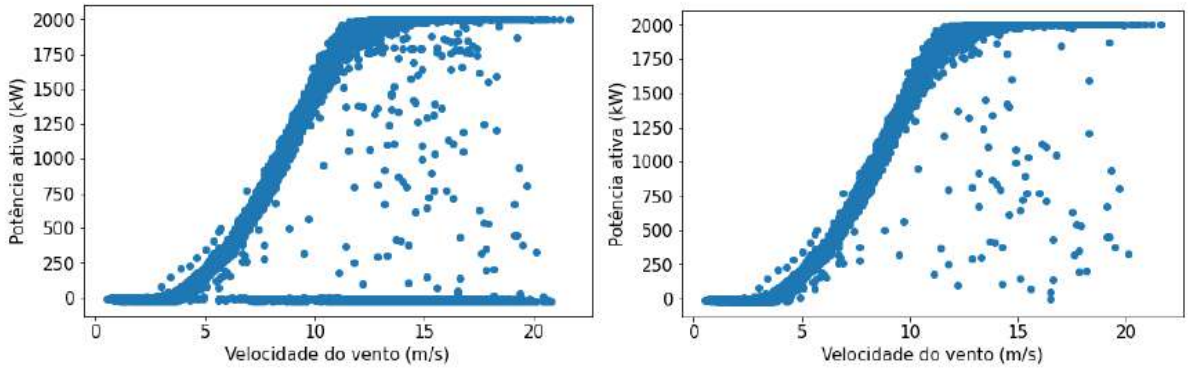
Aqui, deseja-se remover aqueles dados que sejam associados a comportamentos anômalos e problemáticos, como períodos de falha, mal funcionamento de sensores e períodos de alarmes do sistema SCADA. Além disso, dados que não necessariamente sejam errôneos, mas que distem do seu valor esperado e das demais observações do conjunto de dados analisado devem ser tratados como *outliers* e eliminados sempre que possível. Sendo assim, o processo de filtragem aplicado na metodologia foi dividido em 3 diferentes testes: teste de atuações de *pitch*, teste de envelope e teste de ocorrência de alarmes e falhas. A seguir, comenta-se em mais detalhes a respeito de cada um deles.

4.2.2.1 Teste de atuações de *pitch*

Uma forma de eliminar comportamentos indesejados é removendo aqueles períodos em que a máquina está parada, sem gerar energia. É importante realizar a remoção de tais períodos uma vez que os sensores de medição de várias variáveis operacionais continuam operando normalmente mesmo com a máquina parada, o que seria prejudicial para os modelos de aprendizagem, que iriam considerar a associação de tais variáveis com a potência nula em vários períodos. Sendo assim, tais períodos serão eliminados mediante a remoção dos dados nos instantes em que o ângulo de

pitch da máquina supere 80°, o que caracteriza a posição de freio aerodinâmico (ação de parada da máquina). A Figura 27 mostra o processo de filtragem da curva de potência do aerogerador T07 do parque aqui analisado. Nota-se que a curva antes da filtragem apresentava vários valores de potência nula para velocidades entre 5 e 20 m/s, o que também foi observado na curva de potência das demais turbinas do conjunto de dados. Tais valores foram removidos após a filtragem.

Figura 27 - Curva de potência da turbina T07 antes (à esquerda) e depois (à direita) da filtragem pelo ângulo de *pitch*.



Fonte: o autor (2022).

4.2.2.2 Teste de envelope

Aqui, propõe-se o desenvolvimento de uma nova técnica de envelope com o intuito de filtrar os *outliers* remanescentes no conjunto de dados após a remoção das atuações de *pitch*. Por exemplo, na Figura 27 à direita, observa-se que a porção abaixo da curva de potência ainda permanece com vários *outliers*, mesmo após a retirada dos períodos de atuação de *pitch* para frenagem do rotor.

Na literatura, vários autores costumam utilizar técnicas de *clustering* para remoção de *outliers* em distribuições bivariadas (BANGALORE et al., 2017; GONZALES et al., 2019; TRIZOGLOU et al., 2021). Tais técnicas, apesar de retornarem resultados satisfatórios, normalmente precisam da definição prévia de alguns hiperparâmetros[5] para ajuste do modelo, tais como o número de *clusters* ou percentual de contaminação. Em métodos semi-objetivos, tal fato demanda a necessidade de análises exploratórias e certo conhecimento empírico do usuário, o

[5] Entende-se por hiperparâmetros: parâmetros definidos pelo usuário previamente ao treinamento do modelo.

que pode se tornar bastante custoso à medida que se trabalhe com um grande número de variáveis.

Pensando nessas problemáticas, uma nova metodologia automática de detecção de *outliers* em distribuições bivariadas é proposta neste trabalho. Tal metodologia consiste em um teste de envelope, que é realizado para encontrar *outliers* em dispersões (x, y) entre o *target* (y) e variáveis regressoras (x) com correlação acima de 0,7 com o *target.* A seguir, apresenta-se o passo a passo para a aplicação do teste em cada uma das dispersões (x, y).

i. adquira 6 meses de dados (ou o máximo que conseguir) até o mês de interesse para o pré-processamento (e.g., mês de treinamento dos modelos);

ii. aplique as Equações 1 e 2 individualmente para “x” e para “y” utilizando k = 1,5 e elimine os *outliers* encontrados;

iii. normalize as variáveis “x” e “y” de forma que ambas variem de 0 a 1 – por exemplo, a partir da técnica de *MinMaxScaler* (ZHANG et al., 2019);

iv. divida “x” em 10 intervalos (*bins*) equidistantes entre si e calcule a mediana dos valores de “y” em cada intervalo, tendo assim 10 pontos posicionados sobre a dispersão (com os valores de “x” correspondentes ao ponto médio de cada intervalo);

v. realize uma interpolação cúbica (HUYNH, 1993) entre os 10 pontos, fazendo com que a distância no eixo “x” entre dois pontos consecutivos da curva ajustada seja equivalente à menor distância no eixo “x” entre dois pontos consecutivos da dispersão;

vi. com a curva de ajuste obtida, recupere os dados originais da dispersão, incluindo inclusive aqueles *outliers* que foram eliminados no passo “ii”;

vii. obtenha a menor distância euclidiana entre a curva ajustada e cada ponto da dispersão e armazene estas distâncias em um vetor (i.e., cada

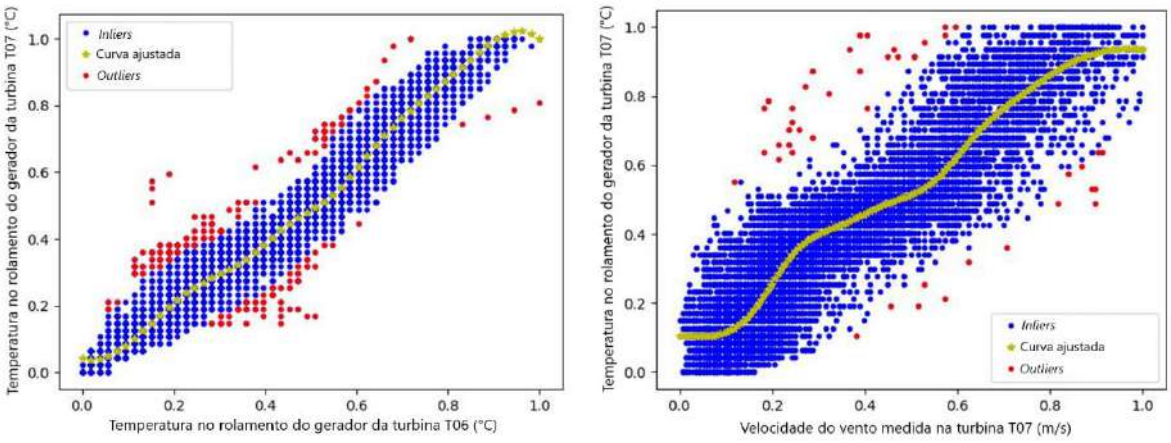
posição do vetor armazena a menor distância equivalente a cada ponto da dispersão);

viii. aplique a Equação 2 ao vetor de distâncias e obtenha o limite superior para k = 3;

ix. os pontos da dispersão cuja distância esteja acima do limite superior serão os *outliers* finais encontrados pela metodologia;

A título de exemplo, a Figura 28 mostra a aplicação do teste de envelope para duas dispersões envolvendo a temperatura no rolamento da turbina T07. Nota-se que o método foi capaz de fornecer um bom ajuste da curva aos dados originais da distribuição, além de identificar boa parte dos dados visualmente díspares da distribuição. É importante salientar que esta metodologia e as parametrizações que a envolvem são de certa forma generalizáveis, desde que se trabalhe com dados do sistema SCADA a cada 10 minutos. Para outras configurações (e.g., dados horários), é preciso reavaliar os valores utilizados na parametrização que mais se ajustam ao caso específico.

Figura 28 - Exemplos da aplicação do teste de envelope. À esquerda, a dispersão entre a temperatura no rolamento do gerador da turbina T07 e a mesma variável na turbina T06. À direita, a dispersão entre a temperatura no rolamento do gerador da turbina T07 e a velocidade média do vento medida na mesma turbina. Em azul, os dados classificados como *inliers* (i.e., dados que não são *outliers*). Em amarelo, a curva de ajuste obtida. Em vermelho, os *outliers* identificados.



Fonte: o autor (2022).

4.2.2.3 Teste de ocorrência de alarmes e falhas

O teste de ocorrência de alarmes visa eliminar do conjunto de dados os períodos relativos à ocorrência de alarmes de temperatura do sistema SCADA da turbina monitorada. Tais períodos são indesejáveis para a modelagem do comportamento livre de falhas, uma vez que em geral estão associados a comportamentos de temperaturas mais elevadas que o normal. Além disso, também são retirados do conjunto de dados os períodos relativos às falhas registradas na turbina de interesse. Em um caso operacional, também seriam retirados da análise os períodos de falha previamente detectados pela ferramenta.

4.2.3 Transformação

Após a filtragem dos dados, optou-se também pela realização de uma mudança no intervalo de integração (*time-step*) do conjunto de dados escolhido para modelagem. Com isso, os dados originais com *time-step* de 10 minutos foram transformados para um *time-step* horário. O procedimento se deu pelo cálculo da média dos dados originais de 6 em 6 instâncias (quantidade de dados de 10 minutos presentes em 1 hora). Tal procedimento foi realizado com dois intuitos: minimizar a influência de possíveis *outliers* remanescentes nos dados originais (que contribuem para o aumento do ruído na entrada dos modelos) e diminuir o esforço computacional empregado no processamento de um grande volume de dados. A partir de algumas análises preliminares, observou-se que o *time-step* horário foi capaz de reduzir o tamanho do conjunto de dados a um nível adequado em termos de custo computacional, ao mesmo tempo que manteve boa parte do comportamento de alta frequência da série temporal original. Ainda assim, uma análise de sensibilidade mais extensa a respeito desta escolha pode ser feita futuramente.

4.3 MODELAGEM DO COMPORTAMENTO LIVRE DE FALHAS

Como discutido anteriormente, a modelagem do comportamento livre de falhas é responsável pela criação de um ou mais modelos que estimem a cada instante de tempo o valor do *target* em condições saudáveis (i.e., livre de falhas) do componente monitorado. Sendo assim, em períodos de operação livre de falhas do aerogerador, é

desejável que se tenha o menor erro possível entre o valor estimado pelo modelo e o valor efetivamente observado durante a operação.

Neste trabalho, a modelagem do comportamento de falhas é constituída por modelos de regressão precedidos de técnicas de seleção de atributos (*feature selection*). Nessa configuração, 3 diferentes tipos de modelos foram utilizados, sendo eles: *Random Forest,* para seleção de atributos e modelo de regressão final; *XGBoost*, para seleção de atributos e modelo de regressão final; e rede neural com camadas convolucionais e LSTM (modelo CNN-LSTM), que atua como modelo regressivo final e é precedido pela técnica de mínima Redundância Máxima Relevância (mRMR) como técnica de seleção de atributos.

A estratégia supracitada foi adotada a fim de direcionar a metodologia para o objetivo geral proposto. Ou seja, as técnicas de seleção de atributos servem para otimizar as estimativas dos modelos em termos de acurácia (no caso dos modelos *Random Forest* e *XGBoost*, que possuem bom desempenho utilizando um grande número de atributos) e disponibilidade (no caso do modelo CNN-LSTM, que consegue desempenhos satisfatórios com uma quantidade reduzida de atributos). Os modelos supracitados foram utilizados tendo em vista o bom desempenho alcançado por eles em trabalhos descritos na literatura (trabalho esses que foram citados na seção de revisão bibliográfica). A seguir, comenta-se em maiores detalhes a respeito de cada uma das configurações adotas.

4.3.1 Modelo Random Forest

Aqui, o modelo será utilizado primeiramente para selecionar os atributos mais relevantes, para em seguida ser retreinado apenas com estes atributos. A definição dos melhores hiperparâmetros a serem utilizados será feita a partir da busca aleatória de uma rede extensa de combinações previamente estabelecida em um esquema de validação cruzada (*randomized search cross validation*) (BERGSTRA; BENGIO, 2012). A Tabela 2 mostra o conjunto de hiperparâmetros utilizado para a busca da melhor parametrização do modelo. Nela, o hiperparâmetro *threshold* é relativo à etapa de seleção de atributos e representa o limiar pelo qual os atributos serão selecionados através da sua importância marginal para a predição final do modelo. Por exemplo, um *threshold* = 1,5 * *median* seleciona apenas os atributos cuja importância marginal esteja acima de um fator de 1,5 da mediana das demais importâncias marginais. Os

demais hiperparâmetros da Tabela 2 são relativos ao próprio modelo *Random Forest*. Na validação cruzada, serão escolhidos aleatoriamente 50 combinações dentre todas as possíveis da Tabela 2. A quantidade de combinações foi escolhida subjetivamente considerando-se o equilíbrio entre custo computacional[6] e quantidade de combinações minimamente suficiente para alcançar uma boa acurácia.

Tabela 2 - Rede de hiperparâmetros para o modelo *Random Forest*.

Parâmetro	Valores
max_depth	[80; None]
max_features	[10; auto]
min_samples_leaf	[1; 3]
min_samples_split	[2; 8]
n_estimators	[100; 500; 1000]
threshold	[“median”; “1.5*median”]

Fonte: o autor (2022).

4.3.2 Modelo *XGBoost*

Da mesma forma que no *Random Forest*, o modelo será utilizado primeiramente para selecionar os atributos mais relevantes, para em seguida ser retreinado apenas com estes atributos. A definição dos melhores hiperparâmetros a serem utilizados também será feita a partir da técnica de *randomized search cross validation*. A Tabela 3 mostra o conjunto de hiperparâmetros utilizado para a busca da melhor parametrização do modelo. Nela, a definição do hiperparâmetro threshold se dá de maneira equivalente ao caso anterior. Os demais hiperparâmetros da Tabela 3 são relativos ao próprio modelo XGBoost. Na validação cruzada, serão escolhidos aleatoriamente 50 combinações dentre todas as possíveis da Tabela 3. A justificativa pela escolha é equivalente àquela apresentada no caso do modelo *Random Forest*.

[6] A máquina utilizada para treinar os modelos possui 16 GB RAM, processador Intel core i5, CPU com 4 núcleos e GPU NVIDIA GEFORCE GTX.

Tabela 3 - Rede de hiperparâmetros para o modelo *XGBoost*.

Parâmetro	Valores
booster	[“gbtree”; “gblinear”]
learning_rate	[0,01; 0,05; 0,1]
max_depth	[5; 10; 20]
min_child_weight	[1; 5; 10]
col_sample_bytree	[0,5; 0,75; 1]
n_estimators	[100; 500; 1000]
reg_alpha	[0; 0,25; 0.5]
reg_lambda	[1; 2,5; 5]
gamma	[0; 1; 2]
threshold	[“median”; “1.5*median”]

Fonte: o autor (2022).

4.3.3 Modelo CNN-LSTM

A modelagem do comportamento livre de falhas a partir das redes neurais convolucionais e LSTM (modelo CNN-LSTM) está essencialmente dividida em duas etapas. A primeira diz respeito à seleção de atributos do conjunto de dados pré-processados para formar o conjunto de treinamento. Para essa tarefa, o modelo de mínima Redundância e Máxima Relevância foi utilizado. A segunda é relativa à estimativa final do *target* em regime de comportamento livre de falhas, na qual a rede neural foi utilizada. A seguir, comenta-se em mais detalhes a respeito de cada uma delas.

4.3.3.1 Mínima Redundância Máxima Relevância (mRMR)

A técnica de mRMR seleciona um número “N” de atributos de entrada para o modelo desejado, com a quantidade “N” sendo definida pelo usuário. Sendo assim, enxerga-se um claro benefício em ter a liberdade de selecionar um número pré-

definido de atributos para a construção de um modelo. A escolha de “n” valores distintos para “N” permite criar “n” diferentes modelos CNN-LSTM para comportamento livre de falhas. Do ponto de vista operacional, a criação de múltiplos modelos com diferentes números de atributos de entrada traz maior probabilidade de solução a um difícil problema: a disponibilidade da ferramenta em cada instante de tempo.

Dentre os fatores possíveis que causam a indisponibilidade de uma ferramenta, um dos mais prováveis é a ausência de uma ou mais variáveis de entrada do modelo ao longo do tempo. Ou seja, uma ferramenta cujo o modelo é definido por $y = f(x_1, x_2, x_3)$ necessita que os valores x_1, x_2 e x_3 estejam todos disponíveis ao mesmo tempo para que se possa estimar o valor de “y”. Sendo assim, modelos estarão sujeitos a alta indisponibilidade à medida que se apresentem várias ausências em suas variáveis de entrada, cenário este que se torna mais provável quanto maior for a quantidade de variáveis utilizadas.

Nesse sentido, é importante que se tenha uma contrapartida em relação aos modelos *Random Forest* e *XGBoost*, que costumam fazer uso de um elevado número de atributos e podem sofrer com indisponibilidade em um cenário operacional da ferramenta. Por esse motivo, optou-se por utilizar o modelo CNN-LSTM em um cenário de menor número de variáveis de entrada, a fim de mitigar a chance de indisponibilidade operacional. O *trade-off* referente a esta escolha é a possível perda de acurácia nestes modelos com relação a soluções que fazem uso de um maior número de atributos – por isso que também é interessante que se disponha de modelos dedicados ao aumento de acurácia sem a preocupação do número de variáveis de entrada, como o *Random Forest* e o *XGBoost*.

Com isso, a solução final adotada neste trabalho para o CNN-LSTM foi de utilizar duas variantes do modelo: uma com 5 atributos selecionados pelo mRMR (N = 5) e outra com 10 atributos selecionados pela mesma técnica (N = 10).

4.3.3.2 Arquitetura e hiperparâmetros do modelo CNN-LSTM

As duas variantes do modelo CNN-LSTM (N = 5 e N = 10) se utilizarão da mesma arquitetura base e da mesma rede de hiperparâmetros durante a validação cruzada. A arquitetura base utilizada é do tipo sequencial, com as camadas definidas na seguinte sequência e com as seguintes parametrizações:

i. camada de entrada com *shape* (*batch_size*, *n_time_steps*, *n_features*);

ii. camada convolucional 1D com número de filtros igual a *n_units*, tamanho da janela igual *n_time_steps* e função de ativação ReLU (*Rectified Linear Unit*);

iii. camada de *dropout* a uma taxa de 20%;

iv. camada LSTM com número de células igual a *n_units*, tamanho da janela igual a *n_time_steps* e função de ativação tangente hiperbólica;

v. camada de *dropout* a uma taxa de 20%;

vi. camada LSTM com número de células igual a *n_units*, tamanho da janela igual a *n_time_steps* e função de ativação tangente hiperbólica;

vii. camada densa com 4 neurônios e função de ativação ReLU;

viii. camada densa de saída com 1 neurônio e função de ativação linear.

A Tabela 4 mostra a rede de hiperparâmetros utilizadas. Tais hiperparâmetros serão testados em um esquema de *randomized cross validation*, na qual serão selecionadas aleatoriamente 5 combinações para testar o desempenho do modelo e escolher a parametrização final. O número reduzido de combinações se deve às limitações computacionais enfrentadas para o treinamento do modelo (muito mais custoso computacionalmente que os demais métodos utilizados nesse trabalho). Para cada combinação, a rede será treinada 3 vezes a fim de buscar diferentes inicializações dos parâmetros da rede neural. A melhor inicialização em termos de acurácia será aquela considerada na análise de cada rodada de validação, sendo a acurácia quantificada através do maior SS4 de Taylor, exposto na Equação 17.

O modelo será treinado se utilizando de 250 épocas e de um *batch size* de 16, tendo a função de otimização “adam” como escolhida (ZHANG, 2018). Ao final da validação cruzada, o modelo com a parametrização que retornou melhores resultados

será retreinado 6 vezes se utilizando de todo conjunto de treinamento, com a melhor inicialização (em termos de SS4) sendo armazenada como modelo final.

Tabela 4 - Rede de hiperparâmetros para o modelo CNN-LSTM.

Parâmetro	Valores
n_time_steps	[1, 3, 6]
n_units	[8, 16, 32, 64]

Fonte: própria.

4.4 DETECÇÃO E ISOLAMENTO DE FALHAS

Uma vez parametrizado o modelo de comportamento livre de falhas, a predição pode ser realizada para o conjunto de teste (monitoramento). Sendo assim, uma medida de dissimilaridade entre o valor predito ($\hat{y}_{teste}$) e o valor real (y_{teste}) deve ser calculada. A falha será detectada quando esta dissimilaridade superar determinado limiar. Aqui, a métrica de dissimilaridade a ser utilizada é o valor absoluto do resíduo entre os dois sinais, tal como mostra a Equação 18. Além disso, o limiar de detecção adotado neste trabalho será calculado através do *boxplot,* com limite superior obtido através da Equação 5 ou Equação 6 (a depender do valor do parâmetro MC). Por fim, uma falha será detectada sempre quando o limiar de detecção for superado por no mínimo dois *timestamps* consecutivos (i.e., uma dissimilaridade acima do normal por no mínimo 2 horas consecutivas).

$$r(t) = |\hat{y}(t) - y(t)| \tag{18}$$

Sendo:
ŷ(t) – estimativa do modelo no instante de tempo “t”;
y(t) – valor real observado no instante de tempo “t”;
r(t) – resíduo no instante de tempo “t”.

Uma vez detectada a falha, a etapa de isolamento é automaticamente acionada para informar sua localização a fim de auxiliar sobre a identificação da possível causa raiz pela equipe de O&M. Nesse trabalho, a forma de escolha do *target* já permite

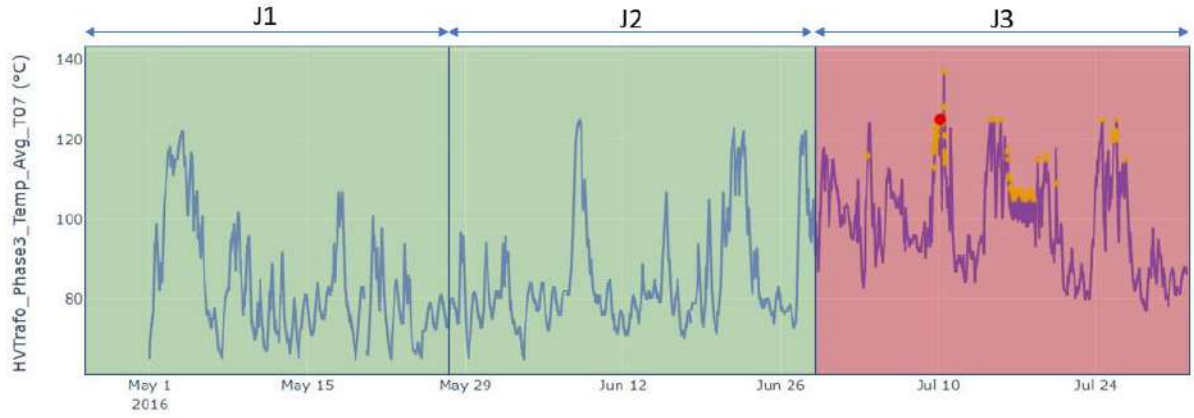
informar com certa clareza a respeito da localização da falha. Por exemplo, o *target HVTrafo_Phase3_Temp_Avg* se refere à temperatura na fase 3 do transformador do aerogerador. Uma falha detectada utilizando este sinal já dá um claro indício sobre a sua localização (i.e., algum problema na fase 3 do transformador). Sendo assim, o isolamento de falhas neste trabalho é um módulo da ferramenta que aguarda por um indicativo de falha vindo do módulo de detecção, e automaticamente informa a sua localização como sendo aquela relativa ao sinal do sistema SCADA que está sendo utilizado como *target*.

A EDP, empresa detentora da central, especifica que as falhas da base de dados aqui utilizada devem ser detectadas dentro de um intervalo de 2 a 60 dias com relação ao instante de registro que consta nos metadados fornecidos (i.e., a tabela que fornece os instantes de tempo referentes aos momentos em que a equipe de O&M percebeu a ocorrência de cada falha após as inspeções realizadas no parque). Para a empresa, falhas detectadas com antecedência menor que 2 dias do registro não permitiriam tempo hábil de ação por parte da equipe de O&M no reparo do componente. Por outro lado, a empresa considera que as falhas relatadas nesta base de dados não carregam padrões que possam ser evidenciados nos sinais do SCADA em um período superior a 60 dias dos seus respectivos registros. Isso faz com que detecções com antecedência maior que essa sejam consideradas com falso positivo.

4.5 *PIPELINES* DE TREINAMENTO E OPERAÇÃO

Pensando em um caso operacional, a metodologia apresentada neste trabalho deve se apoiar em fluxogramas específicos (*pipelines*) para o treinamento dos modelos e para a utilização destes durante a operação a fim de detectar e diagnosticar falhas no componente monitorado. A Figura 29 busca esquematizar a atuação da ferramenta ao longo de um mês específico de operação (janela J3 pintada em vermelho). Por fins didáticos neste mês específico houve uma falha no componente monitorado, cujo registro pela equipe de O&M ocorreu no instante assinalado pelo círculo vermelho. Os asteriscos amarelos indicam períodos de ocorrência de alarmes de temperatura no sistema SCADA. Sendo assim, para que a ferramenta possa entregar saídas ao longo do mês de operação, é necessário que se obtenha 2 meses anteriores ao mês de operação (janelas J1 e J2, em verde) a fim de treinar os modelos de comportamento livre de falhas e gerar os respectivos limiares de detecção.

Figura 29 - Esquematização da atuação da ferramenta ao longo de um mês de específico de operação (J3) e tendo 2 meses anteriores (J1 e J2) para treinamento dos modelos de comportamento livre de falhas e estimativa dos limiares de detecção. O sinal representado é a temperatura na fase 3 do transformador da turbina T07. Os asteriscos amarelos representam alarmes do sistema SCADA, enquanto que o círculo vermelho representa o instante de registro da falha pela equipe de O&M.



Fonte: o autor (2022).

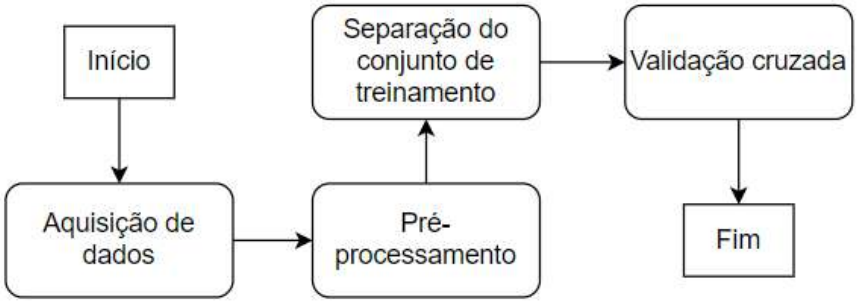
4.5.1 *Pipeline* de treinamento

A Figura 30 mostra o *pipeline* utilizado para treinamento dos modelos de comportamento livre de falhas empregados nesse trabalho. Todo esse *pipeline* é referente ao treinamento de 1 mês específico de dados (e.g., J1 ou J2). O tamanho da janela foi escolhido a partir do seguinte critério: ter dados suficientes para garantir o processo de aprendizagem do modelo durante o treinamento, ao mesmo tempo que se tenha um custo computacional viável. No entanto, cabe investigar futuramente o ganho em acurácia proporcionado pela utilização de uma janela maior para o treinamento do modelo (principalmente aqueles baseados em redes neurais).

A montagem do conjunto de dados é feita de acordo com o que foi exposto na seção 4, adquirindo inicialmente até seis meses de dados, sendo estes relativos ao mês de treinamento e aos cinco meses anteriores a ele. O pré-processamento é feito tal como explica a seção 4.2. Com os dados pré-processados, o conjunto de treinamento é feito selecionando os dados relativos ao mês de treinamento. Caso este conjunto possua um percentual de ausência de dados superior a 10%, mais dados vão sendo adquiridos até que se cumpra esse percentual (com o limite máximo de até 1 mês a mais de dados). Com isso, o procedimento de validação cruzada pode ser aplicado a fim de se obter as parametrizações finais dos modelos de CLF.

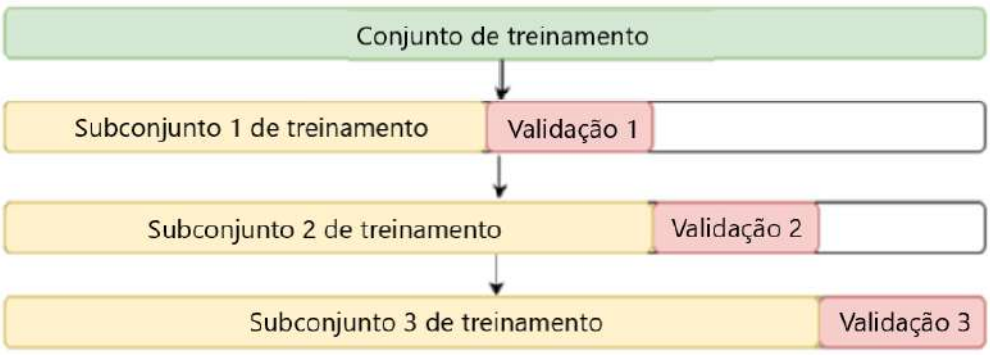
O procedimento de validação cruzada escolhido foi o *k-fold* sequencial, tal como mostra a Figura 31 (SHRIVASTAVA, 2022). Aqui, pela extensão temporal dos dados, foi escolhido um número de k = 3 *folds*. Sendo assim, o procedimento consiste em 3 etapas independentes, na qual em cada uma delas o conjunto de treinamento (em verde) é dividido em um subconjunto de treinamento (em amarelo) e um subconjunto de validação (em vermelho) – com o subconjunto de treinamento tendo tamanho variável a cada etapa e o subconjunto de teste um tamanho fixo. Em cada etapa, o modelo treinado no subconjunto de treinamento estima os dados do subconjunto de validação, com métricas de desempenho sendo calculadas após isso. Sendo assim, a parametrização final escolhida será aquela que retornar as melhores métricas de desempenho ao final do processo.

Figura 30 - *Pipeline* de treinamento da ferramenta.



Fonte: o autor (2022)

Figura 31 - Validação cruzada do tipo *k-fold* sequencial com k = 3. Aqui o conjunto de treinamento (em verde) é subdividido 3 vezes em um subconjunto de treinamento (em amarelo) de tamanho variável e um subconjunto de validação (em vermelho) de tamanho fixo. Nesse trabalho, os modelos utilizados neste procedimento são aqueles referentes à modelagem de CLF.



Fonte: adaptado de Shrivastava (2022).

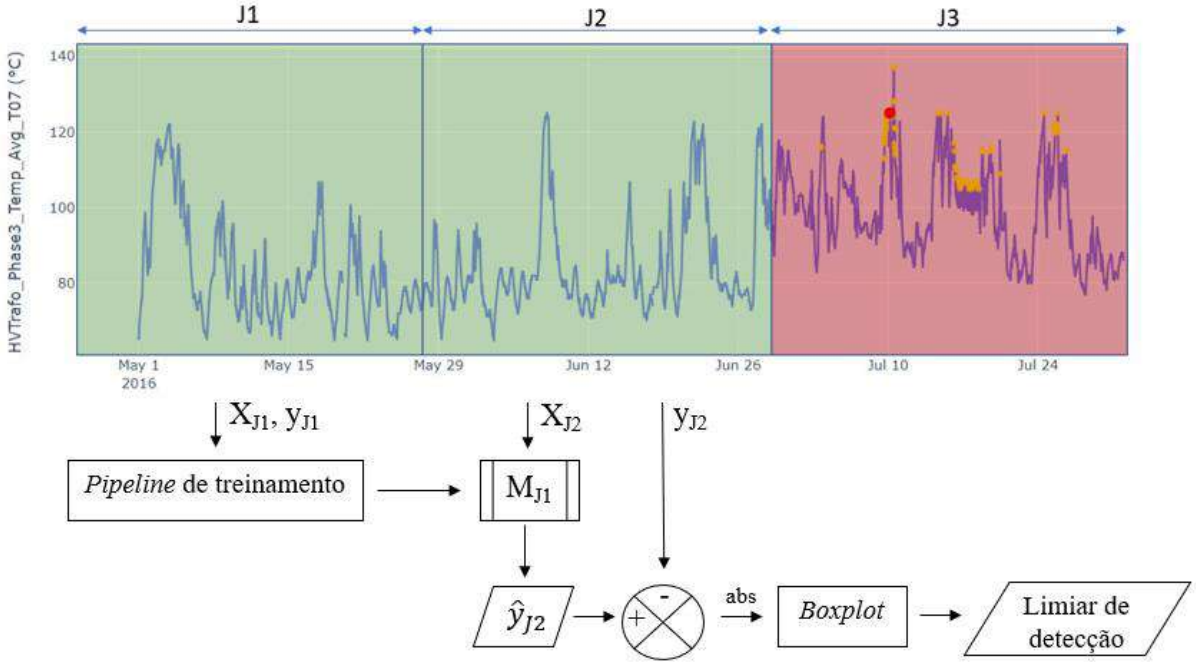
4.5.2 *Pipeline* de operação

O *pipeline* de operação consiste no passo a passo necessário para realizar as estimativas da ferramenta ao longo do tempo, sendo dividido em duas etapas principais: i) obtenção do limiar de detecção; ii) detecção e isolamento de falhas. A seguir, comenta-se em mais detalhes a respeito de cada uma delas. Após isso, detalha-se a respeito da saída final da metodologia – que consistiria na saída final da ferramenta caso a metodologia fosse aplicada em tempo real.

4.5.2.1 Obtenção do limiar de detecção

A Figura 32 ilustra o processo por trás da obtenção do limiar de detecção de falhas no *pipeline* de operação. Nesta etapa, todo o *pipeline* de treinamento (Figura 30) é executado utilizando dados da janela mensal J1 (ou seja, a matriz de atributos X_{J1} e o *target* y_{J1}). Uma vez finalizado o *pipeline* de treinamento, o modelo decorrente deste treinamento (M_{J1}) é utilizado para estimar o comportamento livre de falhas dos dados da janela mensal J2 ($\widehat{y_{J2}}$), usando para isso a matriz de atributos dos dados dessa janela (X_{J2}). A estimativa $\widehat{y_{J2}}$ realizada é então comparada com o sinal real observado y_{J2}, com o valor absoluto do resíduo entre eles sendo calculado tal como a Equação 18. Com isso, o limiar de detecção de falhas é estimado a partir da utilização da técnica de *boxplot* (Equações 5 e 6) sobre o resíduo previamente obtido.

Figura 32 - Etapa de obtenção do limiar de detecção no *pipeline* de operação. "X_{Ji}" significa a matriz de atributos na janela mensal "Ji" e "y_{Ji}" é o *target* na janela mensal "Ji", com "i" variando de 1 a 3. M_{J1} é o modelo treinado a partir do *pipeline* de treinamento e utilizando dados da janela J1. "$\widehat{y_{J2}}$" é a estimativa do *target* na janela "J2" e "abs" significa o valor absoluto.

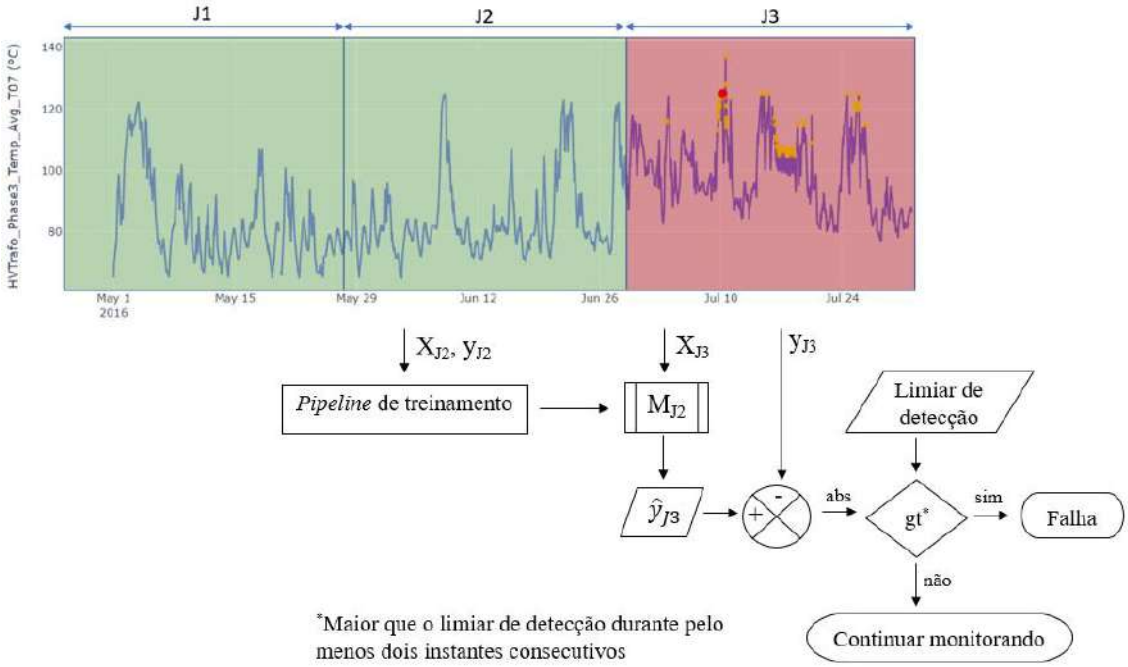


Fonte: o autor (2022).

4.5.2.2 Detecção e isolamento de falhas

A Figura 33 mostra o processo por trás da etapa de detecção e isolamento de falhas do *pipeline* operacional. De forma análoga à etapa anterior, todo o *pipeline* de treinamento (Figura 30) é executado, mas dessa vez utilizando dados da janela mensal J2 (ou seja, a matriz de atributos X_{J2} e o *target* y_{J2}). Uma vez finalizado o *pipeline* de treinamento, o modelo decorrente deste treinamento (M_{J2}) é utilizado para estimar o comportamento livre de falhas dos dados da janela de operação J3 ($\widehat{y_{J3}}$), usando para isso a matriz de atributos dos dados de operação (X_{J3}). A estimativa $\widehat{y_{J3}}$ realizada é então comparada com o sinal real observado y_{J3}, com o valor absoluto do resíduo entre eles sendo calculado tal como a Equação 18. A partir daí, caso o valor do resíduo supere o limiar de detecção (obtido na etapa anterior) por 2 instantes de tempo consecutivos, uma falha é então detectada no aerogerador. De forma automática, o processo de isolamento informa a localização dessa falha, que será referente à localização do sensor responsável por medir a variável alvo utilizada no problema.

Figura 33 - Etapa de detecção e isolamento de falhas no *pipeline* de operação. "X_{Ji}" significa a matriz de atributos na janela mensal "Ji" e "y_{Ji}" é o *target* na janela mensal "Ji", com "i" variando de 1 a 3. M_{J2} é o modelo treinado a partir do *pipeline* de treinamento e utilizando dados da janela J2. "$\widehat{y_{J3}}$" é a estimativa do target na janela "J3", "abs" significa o valor absoluto e "gt*" significa ser maior que o limiar de detecção durante pelo menos dois instantes consecutivos.

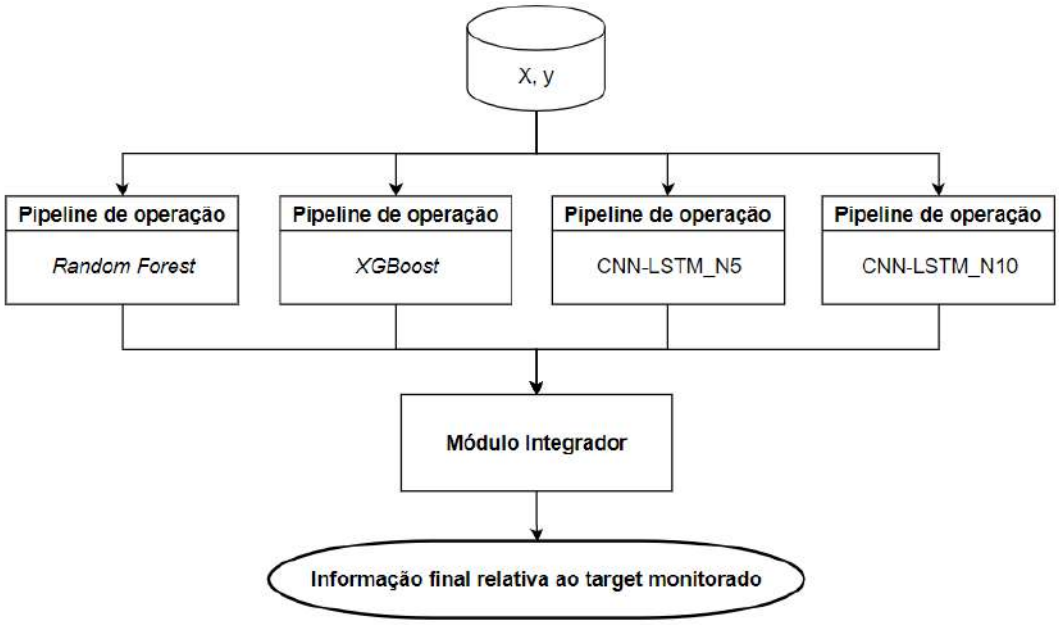


Fonte: o autor (2022).

4.5.3 Saída final da metodologia

Uma vez caracterizado o *pipeline* de operação, deve-se compreender como a saída da metodologia deve ser apresentada para cada *target* monitorado, que é o que mostra a Figura 34. Na figura, nota-se que o *pipeline* de operação deve ser executado em paralelo para cada modelo de CLF previamente definido. Assim sendo, as saídas individuais do *pipeline* de cada modelo são posteriormente fornecidas a um módulo integrador, que informará a saída da metodologia para o *target* monitorado. Com isso, a cada instante de tempo, uma falha será informada com respeito à sua ocorrência e localização sempre quando ao menos uma das saídas individuais tenha entregado alguma informação nesse sentido.

Figura 34 - Saída integrada da metodologia com respeito ao *target* monitorado. O fluxograma abaixo deve ser rodado em paralelo cada *target* “y” do problema, sendo “X” sua matriz de atributos. O módulo integrador observa em cada instante de tempo a saída de cada *pipeline* dos modelos, informando uma falha e sua localização sempre quando ao menos uma das saídas individuais tenha entregado essa informação.



Fonte: o autor (2022).

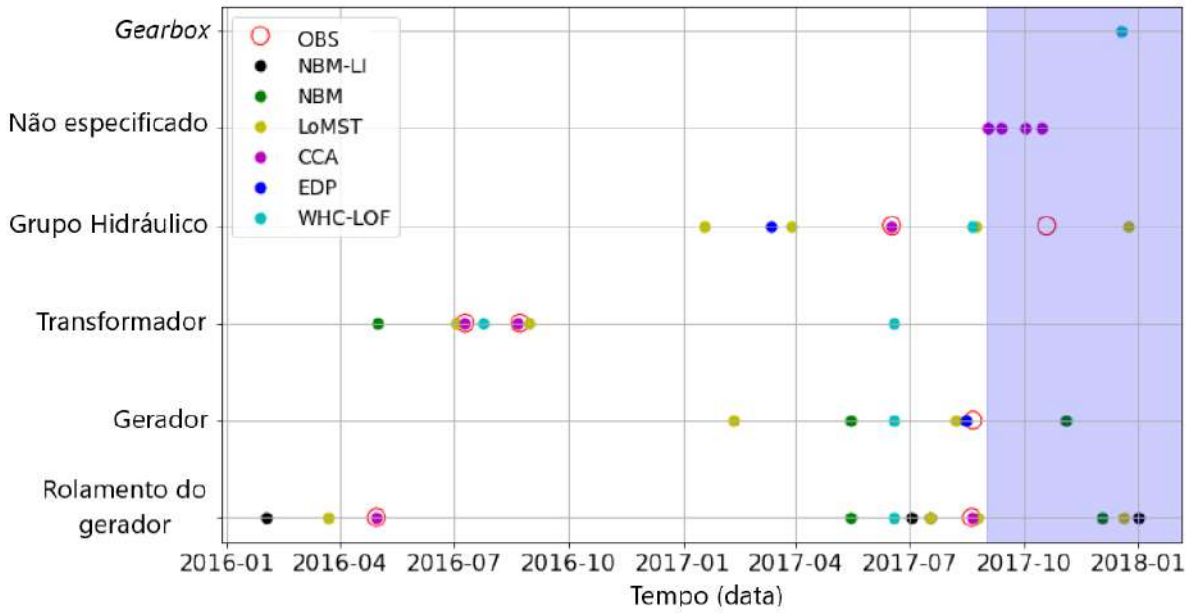
Em vista do que foi discutido, caso essa metodologia seja aplicada a uma situação real com aerogeradores em operação, a equipe de O&M receberá informações ao longo do tempo para cada *target* monitorado, a partir da filosofia apresentada na Figura 34. É importante ressaltar que a utilização da metodologia em um caso operacional traria a necessidade da aplicação para todos os *targets* expostos na Tabela 1, bem como para todos os meses (i.e., as janelas da Figura 29 se moveriam a cada mês com passos de 1 mês). Isso não foi feito neste trabalho em virtude do grande custo computacional que demandaria. Por exemplo, no computador utilizado[7] para obtenção dos resultados aqui apresentados, os modelos CNN-LSTM demoravam cerca de 3 horas cada para conclusão do treinamento de cada mês, mesmo utilizando processamento em paralelo.

[7] A máquina utilizada para treinar os modelos possui 16 GB RAM, processador Intel core i5, CPU com 4 núcleos e GPU NVIDIA GEFORCE GTX.

5 RESULTADOS

A Figura 35 traz resultados da literatura (BARBER et al., 2022) para a mesma base de dados utilizada neste trabalho. Tais resultados são fruto da submissão de diferentes soluções para um concurso realizado pela EDP (Energias De Portugal) em parceria com a WeDoWind, que visava detectar as falhas nos aerogeradores da EDP com maior antecedência possível no intervalo de 2 a 60 dias anteriores à observação da falha. Na Figura 35, os pontos coloridos informam a data de detecção da falha de um modelo específico para um componente específico da turbina T07. O círculo vazado (OBS, na legenda) representa o instante em que a falha foi observada pela equipe de O&M. No total, 6 modelos foram apresentados, sendo eles: i) NBM – *Normal Behaviour Models*; ii) NBM-LI – *Normal Behaviour Models with Lagged Inputs*; iii) LoMST-CUSUM – *Combined Local Minimum Spanning Tree and Cumulative Sum of Multivariate Time Series Data*; iv) CCA – *Canonical Correlation Analysis*; v) EDP – modelo não informado utilizado pela própria empresa; vi) WHC-LOF – *Combined Ward Hierarchical Clustering and Novelty Detection with Local Outlier Factor*.

Figura 35 – Resultados da literatura que trazem instantes de detecção de falha de 6 diferentes modelos aplicados a componentes específicos da turbina T07. O instante de observação da falha pela equipe de O&M é representado pelo círculo vazado (OBS, na legenda). Sendo: NBM – *Normal Behaviour Models*; NBM-LI – *Normal Behaviour Models with Lagged Inputs*; LoMST – *Local Minimum Spanning Tree*; CCA - *Canonical Correlation Analysis*; EDP – modelo não informado utilizado pela própria empresa; WHC-LOF – *Combined Ward Hierarchical Clustering and Novelty Detection with Local Outlier Factor*.



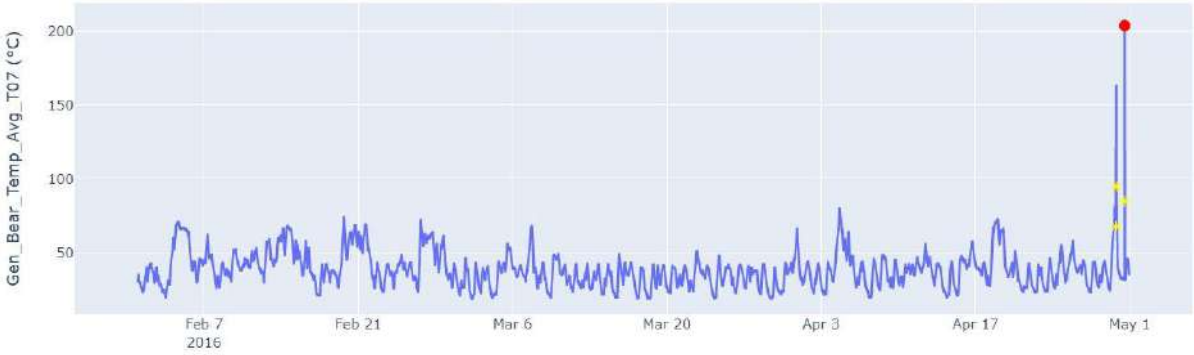
Fonte: adaptado de Barber et al. (2022).

Para as análises a serem realizadas neste trabalho, duas falhas foram selecionadas da base de dados para servirem de estudo de caso para aplicação da metodologia apresentada. A primeira falha acometeu o sensor de temperatura do rolamento do gerador elétrico da turbina T07. A segunda falha, também na turbina T07, indicou problemas de altas temperaturas no transformador. Tais falhas foram convenientemente escolhidas por terem repercutido em sinais de temperatura do sistema SCADA, estando assim dentro do escopo do trabalho. Além disso, os meses das janelas de operação foram escolhidos de forma que o mês da falha coincidisse com o mês de operação. A seguir, apresenta-se em mais detalhes os resultados de cada estudo de caso.

5.1 ESTUDO DE CASO 1: ROLAMENTO DO GERADOR ELÉTRICO

O primeiro estudo de caso a ser analisado diz respeito a uma falha que acometeu o sensor de temperatura acoplado ao rolamento do gerador elétrico da turbina T07. A data de registro da falha pela equipe de O&M foi no dia 30/04/2016. Para essa falha, o *target* escolhido para detecção e isolamento foi a temperatura no rolamento do gerador (*Gen_Bear_Temp_Avg*). A Figura 36 ilustra a série temporal desta variável. Nela, os asteriscos amarelos representam alarmes de temperatura do sistema SCADA, enquanto que o círculo vermelho representa o instante de registro da falha pela equipe de O&M. Como é possível notar, o registro da falha se deu no final do mês, com algumas ocorrências de alarmes sendo observadas próximas a este instante. Visualmente, é difícil perceber alterações significativas nos padrões do sinal de temperatura, a não ser pelos instantes próximos ao registro de falha.

Figura 36 - Série temporal da temperatura no rolamento do gerador elétrico da turbina T07. O ponto vermelho indica o instante de tempo em que ocorreu o registro da falha por parte da equipe de O&M. Os asteriscos amarelos indicam alarmes de temperatura do sistema SCADA.



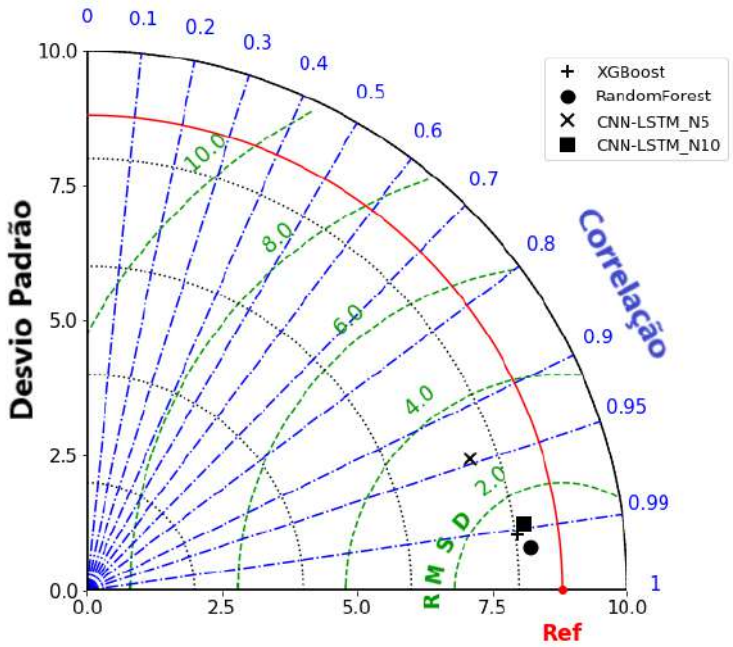
Fonte: o autor (2022).

De forma similar ao que foi apresentado na Figura 29, aqui foram obtidos 3 conjuntos de dados distintos para a aplicação da metodologia. O primeiro, mês J1, é referente ao mês de fevereiro de 2016. O segundo, mês J2, é referente ao mês de março de 2016. O terceiro conjunto é referente ao mês de abril de 2016 e foi utilizado como mês de operação. Aqui, partiu-se do pressuposto de que a falha pode ser detectada com antecedência e ainda dentro do período referente ao mês de operação, já que o registro se deu no final do mês (i.e., uma detecção útil entre 2 e 29 dias antes do evento que aconteceu no dia 30/04). Uma antecedência ainda maior poderia ser obtida (e.g., detecção prévia com mais de 30 dias). Se fosse o caso, o aumento do resíduo (nesse recorte de estudo de caso) seria notado durante a modelagem do CLF no mês J2. Entretanto, em uma situação operacional, tal preocupação não existiria, já que a operação seria feita mês a mês ao invés de um recorte para um mês específico.

A primeira etapa executada do *pipeline* de treinamento (Figura 30) foi a modelagem do comportamento livre de falhas do *target* do mês J2 ($\hat{y}_{J2}$) a partir do treinamento dos modelos de CLF no período J1 (M_{J1}). A Figura 37 mostra o diagrama de Taylor que resume as métricas de desempenho obtidas para a comparação de $\hat{y}_{J2}$ com y_{J2}. Nela, observa-se que os modelos *Random Forest*, *XGBoost* e CNN-LSTM_N10 (CNN-LSTM com 10 atributos selecionados) foram os que apresentaram melhores resultados, com ligeira vantagem para o primeiro. O modelo CNN-LSTM_N5 (CNN-LSTM com 5 atributos selecionados) destoou um pouco dos demais, ainda que os resultados também tenham sido razoavelmente satisfatórios. Isso pode ser devido

ao fato deste estudo de caso necessitar de uma quantidade maior de atributos para representar o comportamento do *target*.

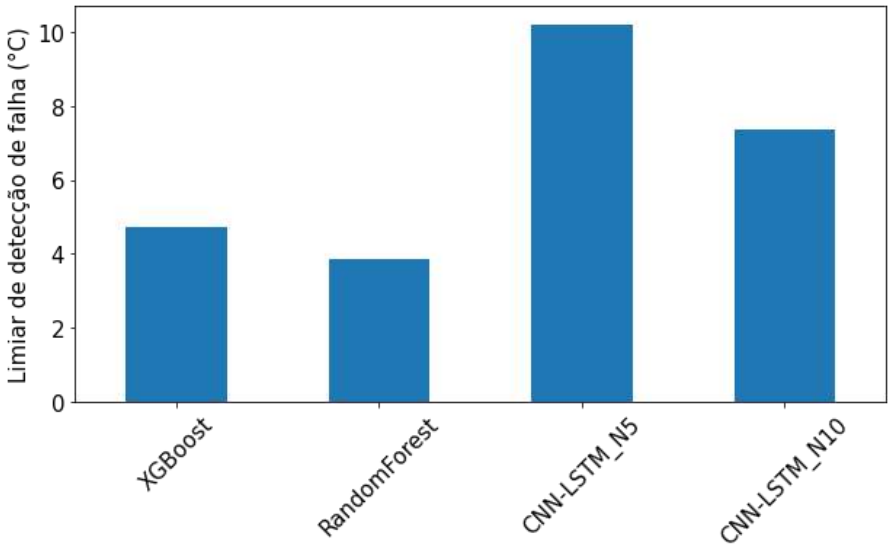
Figura 37 - Diagrama de Taylor do primeiro estudo de caso para as métricas de desempenho de $\hat{y}_{J2}$ obtidas para os 4 modelos apresentados na metodologia.



Fonte: o autor (2022).

Uma vez estimado o $\hat{y}_{J2}$, seu resíduo com y_{J2} pode ser calculado. Tal resíduo, para modelos bem acurados, deve ser representativo apenas das flutuações naturais do modelo, se aproximando de um ruído branco no melhor dos casos (excelente modelo e observação completamente livre de falhas). Sendo assim, o limiar de detecção de falhas pode ser calculado através da aplicação do *boxplot* no resíduo obtido. A Figura 38 mostra os limiares de detecção estimados para cada um dos modelos apresentados na metodologia. Como era de se esperar, a magnitude do limiar possui certa relação com a acurácia do modelo em questão. Isso pode ser observado pelo fato de o pior modelo retornar o limiar mais elevado (CNN-LSTM_N5) e do melhor modelo retornar o limiar menos elevado (*Random Forest*).

Figura 38 - Limiares de detecção de falhas do primeiro estudo de caso para todos os modelos expostos na metodologia.

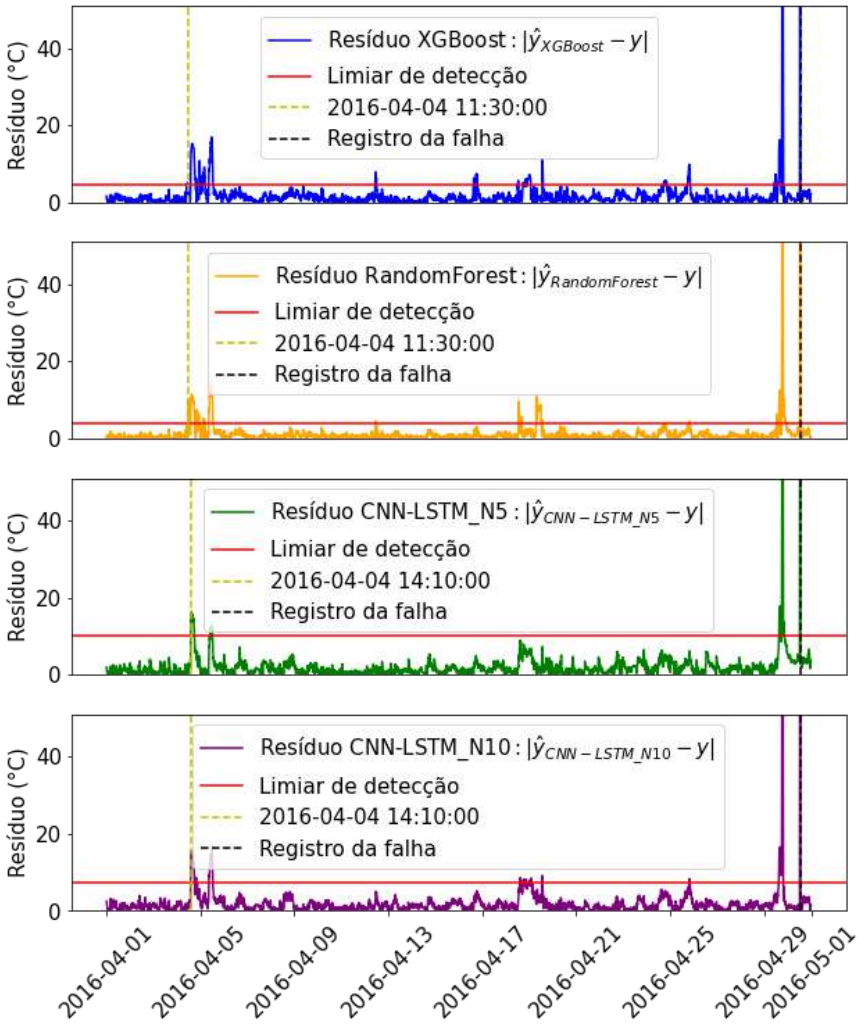


Fonte: o autor (2022).

Uma vez determinados os limiares de detecção de falhas, a etapa de detecção e isolamento de falhas do *pipeline* de operação pode ser executada. O comportamento livre de falhas do mês de operação ($\hat{y}_{J3}$) é estimado a partir do modelo de CLF treinado com os dados de J2 (M_{J2}). O resíduo de $\hat{y}_{J3}$ com y_{J3} é então gerado e uma falha será detectada caso este resíduo supere seu respectivo limiar por no mínimo dois *timestamps* consecutivos.

A Figura 39 mostra o sinal do resíduo e seu respectivo limiar para cada um dos modelos da metodologia. Observa-se que todos os modelos foram capazes de detectar a falha com 26 dias de antecedência, com os modelos *Random Forest* e *XGBoost* detectando-a 2h40min antes que os modelos CNN-LSTM_N5 e CNN-LSTM_N10. Note-se que mesmo com um desempenho ligeiramente inferior, os modelos CNN-LSTM se fazem extremamente necessários para garantir uma maior taxa de disponibilidade da ferramenta.

Figura 39 - Detecção de falhas do primeiro estudo de caso para os 4 modelos da metodologia. O instante de detecção de cada modelo está indicado pela linha vertical tracejada em amarelo, cujo timestamp associado está exposto na legenda de cada gráfico.

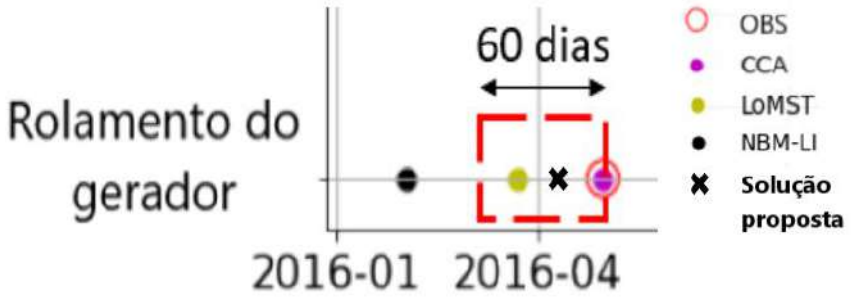


Fonte: o autor (2022).

Uma vez detectada a falha, sua localização é informada imediatamente à equipe de O&M através do processo de isolamento. Para esse estudo de caso, a localização é referente ao rolamento do gerador elétrico do aerogerador T07, baseando-se nos critérios estabelecidos na seção 4.4. Este quase 1 mês de antecedência na detecção e isolamento da falha permitiria um melhor planejamento da equipe de manutenção a respeito do melhor momento para parada da máquina, além de fornecer tempo hábil para questões logísticas como a compra de equipamentos e, caso necessário, contratação de serviços para realização da atividade.

A Figura 40 mostra um *zoom* aplicado à Figura 35, para a região específica da falha aqui analisada. Nela, um retângulo de largura equivalente a 60 dias (em escala aproximada à escala da figura) foi colocado para identificar a região de “verdadeiro positivo” para as falhas detectadas (i.e., 2 a 60 dias, de acordo com a EDP). Observa-se que os modelos CCA e NBM-LI posicionaram-se fora da região delimitada, com o primeiro detectando a falha em um período inferior a 2 dias de antecedência da observação e o segundo em um período superior a 60 dias de antecedência da observação. O modelo LoMST foi capaz de detectar a falha com antecedência de 30 a 60 dias da observação. O “x” colocado na figura representa o resultado retornado para a solução proposta neste trabalho (i.e., 26 dias de antecedência).

Figura 40 – *zoom* aplicado à Figura 35 para a região específica da falha do primeiro estudo de caso, comparando a antecedência de detecção da solução proposta (“x” em preto) com os demais modelos da literatura apresentados para o mesmo estudo de caso. Sendo: OBS – instante de observação da falha pela equipe de O&M; NBM-LI – *Normal Behaviour Models with Lagged Inputs*; LoMST – *Local Minimum Spanning Tree*; CCA - *Canonical Correlation Analysis*.



Fonte: adaptado de Barber et al. (2022).

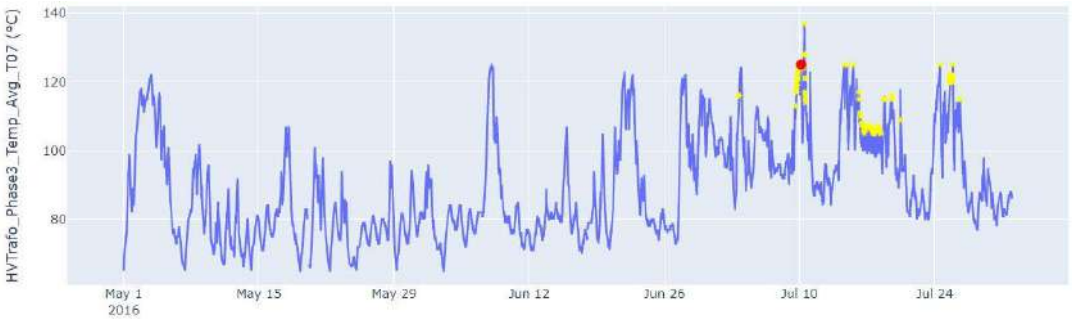
Ao comparar com a literatura, é possível notar que a solução aqui proposta trouxe um resultado satisfatório para esse estudo de caso, apresentando um resultado superior a 5 dos 6 modelos expostos em Barber et al. (2022) – vale salientar que cada um destes modelos foi desenvolvido por candidatos diferentes do concurso. Além disso, a solução deste trabalho traz a contribuição extra de possuir modelos focados no aumento na disponibilidade de estimativas, o que não se observa nas soluções trazidas da literatura.

5.2 ESTUDO DE CASO 2: TRANSFORMADOR

O segundo estudo de caso a ser analisado diz respeito a uma falha que acometeu o transformador da turbina T07. A data de registro da falha pela equipe de O&M foi no dia 10/07/2016. Para essa falha, o *target* escolhido para detecção e isolamento foi a temperatura na fase 3 do transformador (*HVTrafo_Phase3_Temp_Avg*).

A Figura 41 ilustra a série temporal do *target* escolhido. Nela, os asteriscos amarelos representam alarmes de temperatura do sistema SCADA, enquanto que o círculo vermelho representa o instante de registro da falha pela equipe de O&M. Como é possível notar, o registro da falha se deu no início do mês de julho, com algumas ocorrências de alarmes sendo observadas próximas a este instante, tanto antes quanto depois. Visualmente, percebe-se uma certa tendência de aumento do sinal a partir de meados do mês de junho, fato que motivou com que este fosse o mês escolhido para operação, ao invés do mês de julho (mês de registro da falha). Aqui, acredita-se que a falha pode ser detectada já no mês de junho, o que não impede que ela possa ser detectada inclusive antes – a discussão a respeito dessa possibilidade e como essa metodologia lida com esse problema se dá de forma similar ao que foi discutido no estudo de caso anterior. Sendo assim, o mês J1 será referente ao mês de abril, o mês J2 será referente ao mês de maio e o mês de junho será o mês de operação (mês J3).

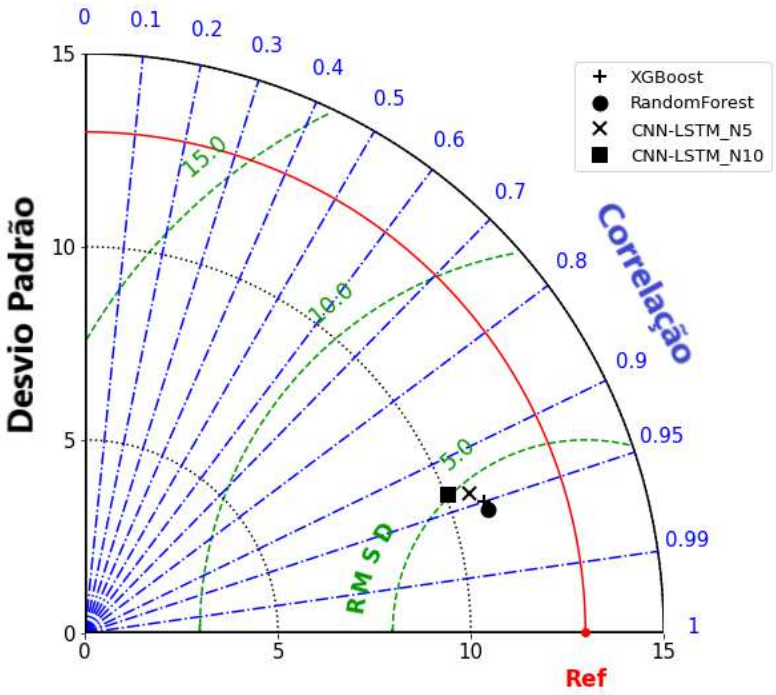
Figura 41 - Série temporal da temperatura na fase 3 do transformador da turbina T07. O ponto vermelho indica o instante de tempo em que ocorreu o registro da falha por parte da equipe de O&M. Os asteriscos amarelos indicam alarmes de temperatura do sistema



Fonte: o autor (2022).

A primeira etapa executada do *pipeline* de treinamento (Figura 30) foi a modelagem do comportamento livre de falhas do target do mês J2 (ŷJ2) a partir do treinamento dos modelos de CLF no período J1 (MJ1). A Figura 42 mostra o diagrama de Taylor que resume as métricas de desempenho obtidas para a comparação de $\hat{y}_{J2}$ com y_{J2}. Observa-se que apesar dos modelos retornarem resultados satisfatórios, estes estão próximos (em termos de métricas) do pior modelo do primeiro estudo de caso, o que evidencia uma maior dificuldade em se modelar o comportamento livre de falhas para este estudo de caso (o estudo de caso 2).

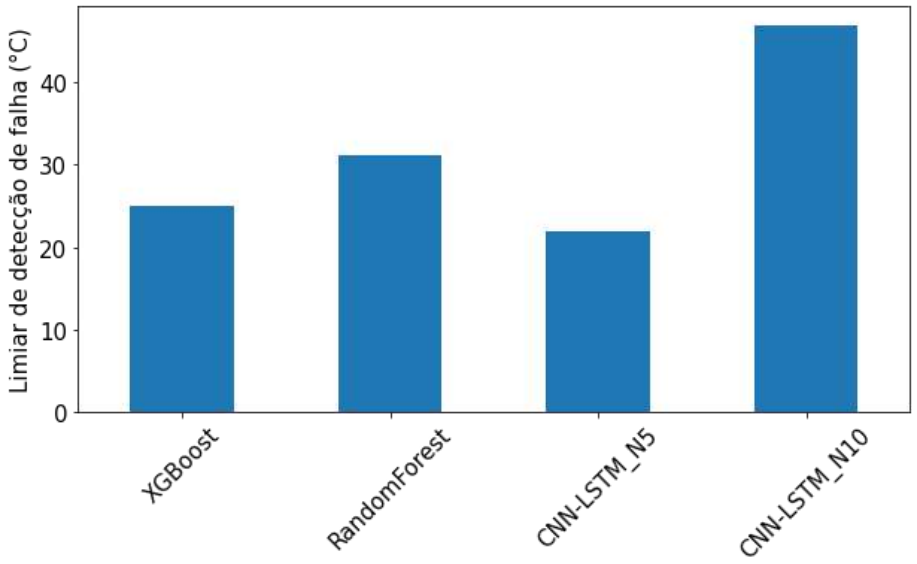
Figura 42 - Diagrama de Taylor do segundo estudo de caso para as métricas de desempenho de $\hat{y}_{J2}$ obtidas para os 4 modelos apresentados na metodologia.



Fonte: o autor (2022).

Uma vez estimado o $\hat{y}_{J2}$, seu resíduo com y_{J2} pode ser calculado de forma análoga àquela exposta no estudo de caso anterior. Após o cálculo dos resíduos, o limiar de detecção de falhas deve ser calculado para cada um dos resíduos estimados, também de forma análoga ao que foi feito anteriormente. A Figura 43 ilustra os limiares de detecção obtidos para cada um dos modelos da metodologia. Observa-se que tais limiares possuem magnitudes muito superiores àqueles estimados no estudo de caso anterior, o que reforça a dificuldade dos modelos em modelar o comportamento livre de falhas para esse *target*.

Figura 43 - Limiares de detecção de falhas do segundo estudo de caso para todos os modelos expostos na metodologia.



Fonte: o autor (2022).

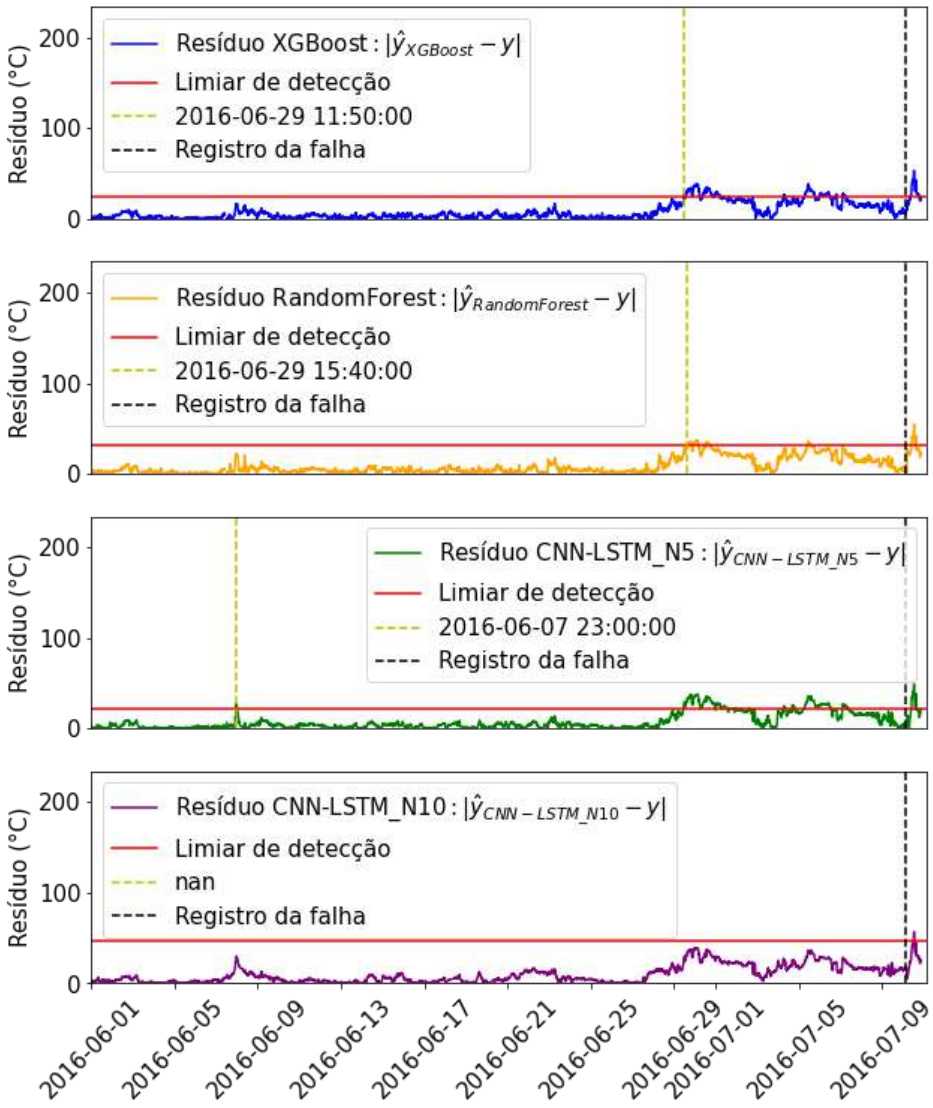
Uma vez determinados os limiares de detecção de falhas, a etapa de detecção e isolamento do *pipeline* de operação pode ser executada de forma análoga àquela exposta no estudo de caso anterior. A Figura 44 mostra o sinal do resíduo e seu respectivo limiar para cada um dos modelos da metodologia. Diferentemente do primeiro estudo de caso, aqui nota-se uma maior heterogeneidade na detecção de falhas por parte dos modelos. Os modelos *XGBoost* e *Random Forest* detectaram a falha com 11 dias de antecedência, o que pode ser considerado um verdadeiro positivo pelos critérios da EDP (falha detectada entre 2 e 60 dias antes do registro). O modelo ConvLSTM_N5 retornou um indicativo de falha no dia 07/06, com 33 dias de antecedência ao registro, o que também é tido como um verdadeiro positivo pelos critérios da EDP. Já o modelo ConvLSTM_N10 não foi capaz de detectar a falha com antecedência, o que pode ser devido ao fato dele possuir menor acurácia com relação ao demais, tal como mostra a Figura 42.

A construção metodológica desse trabalho faz com que uma possível tomada de decisão pela equipe de O&M seja guiada pelos resultados individuais de cada modelo. Isto é, se um modelo detecta uma falha, a equipe de O&M deve prontamente checar se a informação se confirma na prática. Uma boa estratégia de melhoria em perspectivas futuras é a criação de modelos de combinação das saídas individuais de detecção de falhas. Tal estratégia poderia, por exemplo, ponderar as saídas de cada

modelo de acordo com a acurácia obtida na fase da modelagem de CLF, o que poderia diminuir a ocorrência de falsos positivos da ferramenta.

De forma análoga ao estudo de caso anterior, o isolamento de falhas para esse estudo de caso se daria de forma automática e logo após a detecção, com o próprio *target* indicando a localização da falha e fornecendo indícios à equipe de O&M a respeito da possível causa raiz. Aqui, a localização seria a fase 3 do transformador da turbina T07.

Figura 44 - Detecção de falhas do segundo estudo de caso para os 4 modelos da metodologia. O instante de detecção de cada modelo está indicado pela linha vertical tracejada em amarelo, cujo timestamp associado está exposto na legenda de cada gráfico.

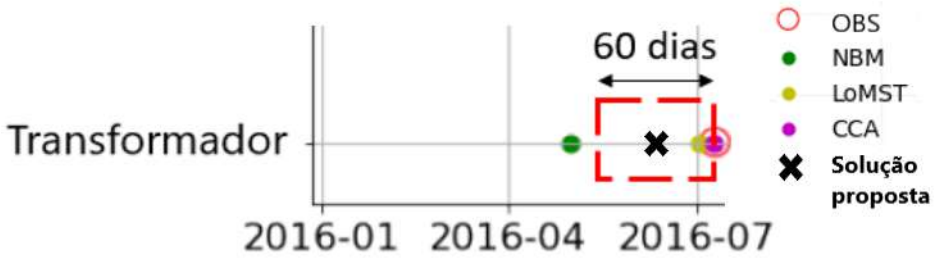


Fonte: o autor (2022).

Analogamente ao estudo de caso anterior, a Figura 45 mostra a comparação da solução aqui proposta com os resultados da literatura para a falha em análise. Para esse caso, percebe-se que a solução proposta neste trabalho foi capaz de superar todos os modelos da literatura, com a falha sendo detectada com 33 dias de antecedência, contra 10 dias de antecedência do melhor modelo da literatura para esse estudo de caso – note que o modelo NBM detectou a falha com antecedência superior a 60 dias, o que caracteriza essa estimativa como falso positivo de acordo com os critérios da EDP.

Além disso, reitera-se a capacidade desta solução em gerar modelos focados em maior disponibilidade de estimativas, o que não se observa nos trabalhos da literatura aqui apresentados.

Figura 45 – *zoom* aplicado à Figura 35 para a região específica da falha do segundo estudo de caso, comparando a antecedência de detecção da solução proposta (“x” em preto) com os demais modelos da literatura apresentados para o mesmo estudo de caso. Sendo: OBS – instante de observação da falha pela equipe de O&M; NBM – *Normal Behaviour Models*; LoMST – *Local Minimum Spanning Tree*; CCA - *Canonical Correlation Analysis*.



Fonte: adaptado de Barber et al. (2022).

6 CONCLUSÕES E PERSPECTIVAS FUTURAS

Este trabalho apresentou uma metodologia conjunta para detecção e isolamento de falhas em aerogeradores utilizando dados do sistema SCADA. A detecção e isolamento prévios da ocorrência de falhas são procedimentos de fundamental importância para o auxílio à tomada de decisão da equipe de O&M do parque eólico, que se beneficia dos indicativos de evolução de uma falha e consegue realizar um melhor planejamento operacional para realização da manutenção – favorecendo, por exemplo, questões logísticas como a compra de materiais e contratação de serviços, além de garantir uma maior disponibilidade de energia para a central.

A estratégia de detecção de falhas adotada foi a análise de resíduos entre a observação do sinal que se deseja monitorar e a estimativa deste por meio de modelos de comportamento livre de falha (CLF), com os limiares de detecção de falhas sendo calculados através da aplicação do *boxplot* nos resíduos. Aqui, optou-se pela utilização de diferentes modelos de CLF, sendo dois deles baseados em árvores de decisão (*Random Forest* e *XGBoost*) e um deles baseado em redes neurais com camadas convolucionais e LSTM.

A construção metodológica do trabalho foi inteiramente guiada para servir de utilização imediata em ferramentas operacionais, apresentando soluções a problemas que são inerentes a aplicações em tempo real e por vezes não são muito discutidos na literatura. Por exemplo, muito se preocupa com respeito à acurácia dos modelos de CLF, mas pouco se fala a respeito da disponibilidade destes modelos em gerar saídas ao longo do tempo. Uma ferramenta baseada em um modelo muito acurado mas com baixa disponibilidade é extremamente problemática para o auxílio à tomada de decisão da equipe de O&M em tempo real. Uma das soluções propostas nesse trabalho foi fazer uso de vários modelos de CLF a fim de mitigar o problema da disponibilidade sem abrir mão da acurácia. Para isso, tais modelos foram parametrizados com diferentes variáveis de entrada, convenientemente selecionados através de técnicas de seleção de atributos como: mínima Redundância Máxima Relevância, utilizada anteriormente ao modelo baseado em redes neurais; e a importância marginal de cada atributo em modelos baseados em árvores de decisão. A heterogeneidade nas variáveis de entrada dos modelos (quando comparados entre si) é benéfica para a disponibilidade da ferramenta, uma vez que a ausência de dados

em determinados sinais é uma das principais causas da indisponibilidade ao longo do tempo.

Outro ponto de contribuição deste trabalho foi com relação ao pré-processamento dos sinais realizado anteriormente ao treinamento dos modelos de CLF. Além das técnicas usuais da literatura, como a retirada de períodos de freio aerodinâmico do rotor por ângulo de *pitch* e a retirada de períodos de alarmes de temperatura do sistema SCADA, uma nova técnica de remoção de *outliers* em distribuições bivariadas foi proposta, sendo ela denominada de envelope. Tal técnica também tem como motivação a facilitação ao usuário e a maior autonomia de ferramentas operacionais. Abordagens usuais de remoção de *outliers* na literatura geralmente necessitam de alguns hiperparâmetros a serem informados pelo usuário para sua utilização, como por exemplo, algumas técnicas de *clustering*. Por vezes, são necessárias análises exploratórias prévias para essa definição, bem como a utilização de conhecimento empírico, o que traz uma desvantagem para aplicações em tempo real. A técnica aqui proposta mostrou boa adequação a diferentes sinais do sistema SCADA com a parametrização sugerida, o que deixa o processo mais automatizado e menos dependente do usuário.

Anteriormente à seleção estatística dos atributos (*statistical-based feature selection*), uma seleção física de atributos também foi empregada. Tal seleção foi feita para eliminar possíveis relações de causalidade entre os atributos de entrada e o *target*, para que assim a falha não seja reproduzida pelo modelo de CLF durante a detecção.

O isolamento de falhas foi desenvolvido para ser realizado de forma automática e após a detecção, informando a localização da falha de acordo com a posição específica do sensor referente ao *target*. Por exemplo, caso seja detectada uma falha e o *target* seja a temperatura na fase 3 do transformador, o isolamento retorna que a falha acometeu exatamente essa região (fase 3 do transformador).

Dois estudos de caso foram selecionados para aplicação da metodologia. O primeiro foi relativo a uma falha no rolamento do gerador elétrico da turbina T07. O segundo, uma falha no transformador da mesma turbina. Os resultados finais foram satisfatórios para ambos estudos de caso. Para o primeiro, a falha foi detectada com 26 dias de antecedência, retornando saídas similares para todos os modelos. Além disso, a solução aqui proposta teve desempenho superior a 5 de 6 modelos da literatura aplicados ao mesmo estudo de caso. Para o segundo estudo de caso,

informações heterogêneas foram retornadas pelos modelos, com um deles detectando a falha com 33 dias de antecedência, outros dois com 11 dias de antecedência e outro não conseguindo detectar a falha (já que foi o que obteve menor acurácia durante o treinamento). Os modelos capazes de detectar a falha obtiveram resultados superiores a todos os modelos da literatura aplicados ao mesmo estudo de caso, o que demonstra o bom potencial da metodologia.

A divergência entre resultados para um mesmo estudo de caso evidencia a importância de se ter diferentes modelos atuando em paralelo, para que se possa aumentar as chances de ter estimativas mais acuradas e com uma maior disponibilidade.

Como perspectivas futuras, planeja-se estender os resultados deste trabalho para uma simulação mais próxima do caso operacional, aplicando os *pipelines* de treinamento e operação para todos os meses deste conjunto de dados, bem como para todos os *targets*. Além disso, planeja-se a inclusão de mais modelos (e variantes de modelos) de comportamento livre de falhas, a fim de aumentar o potencial da ferramenta. Por fim, uma boa contribuição ao trabalho seria o desenvolvimento de uma técnica de combinação das saídas individuais de detecção de falhas. Tal estratégia poderia ser baseada em uma ponderação das saídas com relação à acurácia de cada modelo de CLF durante a fase de estimativa dos limiares de detecção (em que se usa o mês J2 para testar o modelo parametrizado em J1). A contribuição desta escolha poderia ser, por exemplo, a redução da quantidade de falsos positivos retornados pela ferramenta.

REFERÊNCIAS

ABDEL-HAMID, O. et al. Convolutional neural networks for speech recognition. **IEEE/ACM Transactions on audio, speech, and language processing**, v. 22, n. 10, p. 1533-1545, 2014.

AHMAD, R.; KAMARUDDIN, S. An overview of time-based and condition-based maintenance in industrial application. **Computers & Industrial Engineering**, v. 63, n. 1, p. 135-149, 2012.

ARAZA, A. Integrating time series forest loss into streamflow prediction by random forest in key watersheds of the philippines. 2018.

ASSAF, R. et al. Mtex-cnn: Multivariate time series explanations for predictions with convolutional neural networks. In: **2019 IEEE International Conference on Data Mining (ICDM)**. IEEE, 2019. p. 952-957.

BADIHI, H. et al. A Comprehensive Review on Signal-Based and Model-Based Condition Monitoring of Wind Turbines: Fault Diagnosis and Lifetime Prognosis. **Proceedings of the IEEE**, 2022.

BANGALORE, P. et al. An artificial neural network-based condition monitoring method for wind turbines, with application to the monitoring of the gearbox. **Wind Energy**, v. 20, n. 8, p. 1421-1438, 2017.

BARBER, S. et al. Enabling Co-Innovation for a Successful Digital Transformation in Wind Energy Using a New Digital Ecosystem and a Fault Detection Case Study. **Energies**, v. 15, n. 15, p. 5638, 2022.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of machine learning research**, v. 13, n. 2, 2012.

BEZERRA, C. C. A. **Detecção de Falhas em Rolamentos de Turbinas Eólicas Utilizando Modelos de Aprendizagem de Máquina**. 2019. Trabalho de Conclusão de Curso. Universidade Federal de Pernambuco.

BRANDÃO, R. F. M. et al. Neural Networks for Condition Monitoring of Wind Turbines. **Int. Symp. Mod. Electr. Power Syst**. Wroclaw, Pol., 2010.

BREIMAN, L. Random forests. **Machine learning**, v. 45, n. 1, p. 5-32, 2001.

BRYS, G.; HUBERT, M.; STRUYF, A. A robust measure of skewness. **Journal of Computational and Graphical Statistics**, v. 13, n. 4, p. 996-1017, 2004.

CANIZO, M. et al. Real-time predictive maintenance for wind turbines using Big Data frameworks, in: 2017 IEEE Int. Conf. Progn. Heal. Manag., 2017, pp. 70e77.

CARTER, C. M. et al. **Continuous Reliability Enhancement for Wind (CREW). Program Update**. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2016.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. 2016. p. 785-794.

CHICKERING, D. M. **Optimal structure identification with greedy search.** Journal of machine learning research, v. 3, n. Nov, p. 507-554, 2002.

CLEMENTE, J. O. **Previsão de rampas em séries temporais de potência de saída de centrais eólicas**. 2018. Dissertação de Mestrado. Universidade Federal de Pernambuco.

COLLACOTT, R. Mechanical fault diagnosis and condition monitoring. **Springer Science & Business Media**, 2012.

DMITRIEVSKY, M. **Random Decision Forest In Reinforcement Learning**. Disponível em: < https://www.mql5.com/en/articles/3856>. Acesso em: 22 de jul. 2022.

DONG, X. L.; SRIVASTAVA, D. Big data integration. Proceedings of the VLDB Endowment, 6(11):1188–1189, 2013.

EDP. **EDP open data**. Disponível em: <https://opendata.edp.com/pages/homepage/>. Acesso em: 23 de abr. 2019.

EPE. **Plano Decenal de Expansão de Energia 2031**. Disponível em: <https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/Documents/PDE%202031_RevisaoPosCP_rvFinal.pdf>. Acesso em: 22 de jul. 2022.

FELGUEIRA, T. et al. The Impact of Feature Causality on Normal Behaviour Models for SCADA-based Wind Turbine Fault Detection. **arXiv preprint arXiv:1906.12329**, 2019.

FERNANDES, L. C. **Previsão de potência eólica de curtíssimo prazo baseada na análise espectral e decomposição da série temporal**. 2018. Dissertação de Mestrado. Universidade Federal de Pernambuco.

FERREIRA, J. E. V. et al. Graphical representation of chemical periodicity of main elements through boxplot. **Educación química**, v. 27, n. 3, p. 209-216, 2016.

FISCHER, K.; CORONADO, D. A. Condition monitoring of wind turbines: state of the art, user experience and recommendations. **Fraunhofer-IWES, Bremerhaven**, 2015.

FRAMPTON, M. Apache mesos. In: **Complete Guide to Open Source Big Data Stack**. Apress, Berkeley, CA, 2018. p. 97-137.

GALLEGO CASTILLO, C. J. **Statistical models for short-term wind power ramp forecasting**. 2013. Tese de Doutorado. Aeronauticos.

GAO, Z.; CECATI, C.; DING, S. X. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. **IEEE transactions on industrial electronics**, v. 62, n. 6, p. 3757-3767, 2015.

GARCIA, M. C. et al. SIMAP: Intelligent System for Predictive Maintenance: Application to the health condition monitoring of a windturbine gearbox. **Computers in Industry**, v. 57, n. 6, p. 552-568, 2006.

GARLICK, W. G. et al. A model-based approach to wind turbine condition monitoring using SCADA data. **ICSE**. 2009.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems**. " O'Reilly Media, Inc.", 2019.

GONZALEZ, E. et al. Using high-frequency SCADA data for wind turbine performance monitoring: A sensitivity study. **Renewable energy**, v. 131, p. 841-853, 2019.

GUO, L. et al. A recurrent neural network based health indicator for remaining useful life prediction of bearings, Neurocomputing 240 (May 2017) 98e109.

HASTIE, T. et al. **The elements of statistical learning: data mining, inference, and prediction.** New York: springer, 2009.

HESTERBERG, T. Bootstrap. **Wiley Interdisciplinary Reviews: Computational Statistics**, v. 3, n. 6, p. 497-526, 2011.

HUBERT, M.; VANDERVIEREN, E. An adjusted boxplot for skewed distributions. **Computational statistics & data analysis**, v. 52, n. 12, p. 5186-5201, 2008.

HUYNH, H. T. Accurate monotone cubic interpolation. **SIAM Journal on Numerical Analysis**, v. 30, n. 1, p. 57-100, 1993.

ISERMANN, R. Fault-diagnosis systems: an introduction from fault detection to fault tolerance. **Springer Science & Business Media**, 2005.

ISMAIL FAWAZ, H. et al. Deep learning for time series classification: a review. **Data mining and knowledge discovery**, v. 33, n. 4, p. 917-963, 2019.

JANG, J. et al. Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [Book Review]. **IEEE Transactions on automatic control**, v. 42, n. 10, p. 1482-1484, 1997.

JUNG, H. et al. Abschlussbericht: Erhöhung der Verfügbarkeit von Windenergieanlagen EVW-Phase 2. **FGW eV Wind Energy and Other Decentralized Energy Organizations: Berlin, Germany**, 2015.

JUNQUEIRA, F.; REED, B. **ZooKeeper: distributed process coordination**. " O'Reilly Media, Inc.", 2013.

KARIM, F. et al. Multivariate LSTM-FCNs for time series classification. **Neural Networks**, v. 116, p. 237-245, 2019.

KARUN, A. K.; CHITHARANJAN, K. A review on hadoop—HDFS infrastructure extensions. In: **2013 IEEE conference on information & communication technologies**. IEEE, 2013. p. 132-137.

LATHAM, P. E.; ROUDI, Y. Mutual information. **Scholarpedia**, v. 4, n. 1, p. 1658, 2009.

LEAHY, K. et al. Issues with data quality for wind turbine condition monitoring and reliability analyses. **Energies**, v. 12, n. 2, p. 201, 2019.

LECUN, Y. et al. Handwritten digit recognition with a back-propagation network. **Advances in neural information processing systems**, v. 2, 1989.

LEITE, G. N. P. **Diagnóstico de falhas em componentes de turbinas eólicas através da aplicação de quantificadores da teoria da informação**. 2018. Tese de doutorado. Universidade Federal de Pernambuco.

LI, M. et al. A data-driven residual-based method for fault diagnosis and isolation in wind turbines. **IEEE Transactions on Sustainable Energy**, v. 10, n. 2, p. 895-904, 2018.

LI, Y. et al. Wind turbine fault diagnosis based on transfer learning and convolutional autoencoder with small-scale data. **Renewable Energy**, v. 171, p. 103-115, 2021.

LI, Y. et al. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. **Knowledge-Based Systems**, v. 181, p. 104785, 2019.

LIU, X.; DU, Juan; YE, Zhi-Sheng. A condition monitoring and fault isolation system for wind turbine based on SCADA data. **IEEE Transactions on Industrial Informatics**, v. 18, n. 2, p. 986-995, 2021.

LIU, Z.; ZHANG, L. A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings. **Measurement**, v. 149, p. 107002, 2020.

MARTINEZ-LUENGO, M.; SHAFIEE, M.; KOLIOS, Athanasios. Data management for structural integrity assessment of offshore wind turbine support structures: data cleansing and missing data imputation. **Ocean Engineering**, v. 173, p. 867-883, 2019.

MARTI-PUIG, P. et al. Effects of the pre-processing algorithms in fault diagnosis of wind turbines, Environ. Model. Software (2018) 0e1, no. February 2017.

MARTI-PUIG, P. et al. Feature selection algorithms for wind turbine failure prediction. **Energies**, v. 12, n. 3, p. 453, 2019.

MARVUGLIA, A. MESSINEO, A. “Monitoring of wind farms' power curves using machine learning techniques, Appl. Energy 98 (2012) 574e583.

MAZZANTI, S. (2021). “MRMR” Explained Exactly How You Wished Someone Explained to You. **Towards Data Science**. Disponível em: < https://towardsdatascience.com/mrmr-explained-exactly-how-you-wished-someone-explained-to-you-9cf4ed27458b>. Acesso em: 10 de mar. 2022.

MCLACHLAN, G. J. Mahalanobis distance. **Resonance**, v. 4, n. 6, p. 20-26, 1999.

MORSHEDIZADEH, M. et al. Application of imputation techniques and adaptive neuro-fuzzy inference system to predict wind turbine power production. **Energy**, v. 138, p. 394-404, 2017.

OLAH, C. Understanding lstm networks. 2015.

OLAOYE, A. **Wind energy analytics toolbox: Iterative power curve filter**. Disponível em: <https://towardsdatascience.com/wind-energy-analytics-toolbox-iterative-power-curve-filter-fec258fdb997>. Acesso em: 18 de jul. 2022.

OROZCO, R. et al. Diagnostic models for wind turbine gearbox components using SCADA time series data preprint, in: 2018 IEEE Int. Conf. Progn. Heal. Manag., 2018, pp. 1e9, no. July.

PANG Y, et al. Spatio-temporal fusion neural network for multi-class fault diagnosis of wind turbines based on SCADA data. **Renewable Energy**. 2020.

PARK, J. et al. Development of a novel power curve monitoring method for wind turbines and its field tests. **IEEE Transactions on Energy Conversion**, v. 29, n. 1, p. 119-128, 2014.

PEDROSA, G. T. M. C. **Detecção e diagnóstico de falhas na performance de aerogeradores**. 2016. Dissertação de Mestrado. Universidade Federal de Pernambuco.

PFAFFEL, S. et al. Performance and reliability of wind turbines: a review, Energies 10 (11) (2017).

QU, F. et al. A novel wind turbine data imputation method with multiple optimizations based on GANs. **Mechanical Systems and Signal Processing**, v. 139, p. 106610, 2020.

RAMCHOUN, H. et al. Multilayer perceptron: Architecture optimization and training. 2016.

ROCHFORD, P. **SkillMetrics: Taylor Diagram Example 10**. Disponível em: <https://github.com/PeterRochford/SkillMetrics/wiki/Taylor-Diagram-Example-10>. Acesso em: 24 de jul. 2022.

RYU, M. et al. Online sequential extreme studentized deviate tests for anomaly detection in streaming data with varying patterns. **Cluster Computing**, v. 24, n. 3, p. 1975-1987, 2021.

SAGI, O.; ROKACH, Lior. Ensemble learning: A survey. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 8, n. 4, p. e1249, 2018.

SANTANA, F. B. et al. Floresta aleatória para desenvolvimento de modelos multivariados de classificação e regressão em química analítica. 2020.

SCHLECHTINGEN, M., et al. Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description. **Applied Soft Computing**, v. 13, n. 1, p. 259-270, 2013.

SCIKIT-LEARN. **Random Forest Regressor**. Disponível em: < https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html/>. Acesso em: 03 de fev. 2022.

SHRIVASTAVA, S. **Cross Validation in Time Series**. Disponível em: < https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4/>. Acesso em: 28 de jan. 2022.

SOHONI, V.; GUPTA, S. C.; NEMA, R. K. A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems. Journal of Energy, v. 2016, 2016. 50

SOLEY-BORI, M. Dealing with missing data: Key assumptions and methods for applied analysis. **Boston University**, v. 4, n. 1, p. 19, 2013.

SONI, M. **What is Decision Tree Regression?**. Disponível em: <https://maniksonituts.medium.com/what-is-decision-tree-regression-dcd0ea40a323>. Acesso em: 22 de jul. 2022.

SPARK, A. Apache spark. **Retrieved January**, v. 17, n. 1, p. 2018, 2018.

STETCO, A. et al. Machine learning methods for wind turbine condition monitoring: A review. **Renewable energy**, v. 133, p. 620-635, 2019.

STOUFFER, K. et al. **Guide to Supervisory Control and Data Acquisition (SCADA) and industrial Control Systems Security**. In: SPIN. [s.n.], 2006.

SUTTON, C. D. Classification and regression trees, bagging, and boosting. **Handbook of statistics**, v. 24, p. 303-329, 2005.

TANG, W. et al. Rethinking 1d-cnn for time series classification: A stronger baseline. **arXiv preprint arXiv:2002.10061**, 2020.

TAUTZ-WEINERT, J.; WATSON, S. J. Using SCADA data for wind turbine condition monitoring–a review. **IET Renewable Power Generation**, v. 11, n. 4, p. 382-394, 2016.

TAVNER, P. How are we going to make offshore wind farms more reliable?. **Supergen Wind**, 2011.

TAYLOR, K. E. Summarizing multiple aspects of model performance in a single diagram. Program for Climate Model Diagnosis and Intercomparison, Lawrence Livermore National Laboratory, University of California, 2001.

TCHAKOUA, P. et al. Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges. **Energies**, v. 7, n. 4, p. 2595-2630, 2014.

TONG, W. **Wind power generation and wind turbine design**. WIT press, 2010.

TRIZOGLOU, P; LIU, X.; LIN, Z. Fault detection by an ensemble framework of Extreme Gradient Boosting (XGBoost) in the operation of offshore wind turbines. **Renewable Energy**, v. 179, p. 945-962, 2021.

ULMER, M. et al. Early fault detection based on wind turbine scada data using convolutional neural networks. In: **5th European Conference of the Prognostics and Health Management Society, Virtual Conference, 27-31 July 2020**. PHM Society, 2020.

WANG, H. et al. Early fault detection of wind turbines based on operational condition clustering and optimized deep belief network modeling. **Energies**, v. 12, n. 6, p. 984, 2019.

WANG, K.; JIANG, W. High-dimensional process monitoring and fault isolation via variable selection. **Journal of Quality Technology**, v. 41, n. 3, p. 247-258, 2009.

WANG, X.; MAO, D.; LI, X. Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network. **Measurement**, v. 173, p. 108518, 2021.

WILKINSON, M. et al. Comparison of methods for wind turbine condition monitoring with SCADA data. **IET Renewable Power Generation**, v. 8, n. 4, p. 390-397, 2014.

YAN, X.; WEIHAN, W.; CHANG, M. Research on financial assets transaction prediction model based on LSTM neural network. **Neural Computing and Applications**, v. 33, n. 1, p. 257-270, 2021.

YAO, Z. et al. Using hampel identifier to eliminate profile-isolated outliers in laser vision measurement. **Journal of Sensors**, v. 2019, 2019.

ZHANG, D. et al. A data-driven design for fault detection of wind turbines using random forests and XGboost. **Ieee Access**, v. 6, p. 21020-21031, 2018.

ZHANG, T. et al. Research on gas concentration prediction models based on LSTM multidimensional time series. **Energies**, v. 12, n. 1, p. 161, 2019.

ZHANG, Z. Improved adam optimizer for deep neural networks. In: **2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)**. Ieee, 2018. p. 1-2.

ZHAO, Y. et al. Fault prediction and diagnosis of wind turbine generators using SCADA data. **Energies**, v. 10, n. 8, p. 1210, 2017b.

ZHAO, Y. et al. Data-driven correction approach to refine power curve of wind farm under wind curtailment. **IEEE Transactions on Sustainable Energy**, v. 9, n. 1, p. 95-105, 2017a.

ZOU, C.; QIU, P. Multivariate statistical process control using LASSO. **Journal of the American Statistical Association**, v. 104, n. 488, p. 1586-1596, 2009.