



Humberto Seghetto dos Santos

**Aprendizado de Máquina Aplicado à Detecção
de Falhas em Sistemas Mecânicos**

Trabalho de Conclusão de Curso

Orientador : Prof. Helon Vicente Hultmann Ayala
Coorientador: Pedro Henrique Domingues

Rio de Janeiro
Novembro de 2023

Resumo

Seghetto dos Santos, Humberto; Hultmann Ayala, Helon Vicente; Domingues, Pedro Henrique. **Aprendizado de Máquina Aplicado à Detecção de Falhas em Sistemas Mecânicos**. Rio de Janeiro, 2023. 34p. Trabalho de Conclusão de Curso – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Nas diversas áreas da engenharia é fundamental garantir a saúde das estruturas utilizadas. O *Structural Health Monitoring* (SHM) é a disciplina específica que tem como objetivo avaliar e manter a integridade de estruturas, possibilitando a detecção precoce de falhas e prolongando sua vida útil. Neste trabalho iremos apresentar uma aplicação de *Machine Learning* (ML) como uma ferramenta auxiliar no processo de identificação de falhas essencial ao SHM. Em particular, analisaremos um estudo de caso desenvolvido com ênfase nas pás de turbinas eólicas e utilizaremos os dados deste caso como uma base para demonstrar a aplicabilidade dos conceitos teóricos. Dentre as técnicas utilizadas damos destaque a metodologia focada na redução dimensional, tanto através de métodos tradicionais como o *Principal Component Analysis* quanto mais modernos como o *Feature Engineering*, aliados aos modelos altamente adaptativos de ML. Essas ferramentas são fundamentais para ser possível tirar máximo proveito da imensa quantidade de dados que são produzidos pelas tecnologias de sensores e de monitoramento moderno. A integração de ML ao SHM não apenas aprimora a precisão na identificação de falhas, mas também exemplifica a adaptabilidade e eficácia de tecnologias inovadoras na proteção da saúde estrutural.

Palavras-chave

Monitoramento da Saúde Estrutural; Aprendizado de Máquina; Feature Engineering; Análise de Componentes Principais; Classificação; Wind Turbine Blades.

Abstract

Seghetto dos Santos, Humberto; Hultmann Ayala, Helon Vicente (Advisor); Domingues, Pedro Henrique (Co-Advisor). **Machine Learning Applied to Fault Diagnosis in Mechanical Systems**. Rio de Janeiro, 2023. 34p. Trabalho de Conclusão de Curso – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

In various engineering fields, ensuring the health of utilized structures is crucial. Structural Health Monitoring (SHM) is the specific discipline aimed at assessing and maintaining the integrity of structures, enabling early fault detection, and extending their lifespan. In this work, we will present an application of Machine Learning (ML) as an auxiliary tool in the essential fault identification process within SHM. Specifically, we will analyze a case study with an emphasis on wind turbine blades, utilizing the data from this case as a basis to demonstrate the applicability of theoretical concepts. Among the techniques employed, we highlight the methodology focused on dimensional reduction, utilizing both traditional methods such as Principal Component Analysis and more modern approaches like feature engineering, coupled with highly adaptive ML models. These tools are crucial to harness the maximum potential of the vast amount of data generated by modern sensor and monitoring technologies. The integration of ML into SHM not only enhances precision in fault identification but also exemplifies the adaptability and effectiveness of innovative technologies in safeguarding structural health.

Keywords

Structural Health Monitoring; Machine Learning; Feature Engineering; Principal Component Analysis; Classification; Wind Turbine Blades.

Sumário

1	Introdução	7
2	Metodologia	9
2.1	Estudo de Caso	9
2.2	Feature Engineering	14
2.3	Features Estatísticas	14
2.3.1	Média	15
2.3.2	Variância	15
2.3.3	Assimetria	16
2.3.4	Curtose	16
2.4	PCA e Detecção de Outliers	16
2.5	Modelos Utilizados	19
2.5.1	Árvore de Decisão	19
2.5.2	Análise de Discriminante Linear	20
2.5.3	Máquinas de Vetores de Suporte	20
2.6	Treino e Avaliação dos Modelos	22
2.6.1	Busca de hiperparâmetros	22
2.6.2	Cross-Validation	24
2.7	Métricas	24
2.7.1	Sensibilidade	26
2.7.2	Precisão	26
2.7.3	Acurácia Balanceada	26
2.7.4	F1-Score	27
3	Resultados	28
4	Conclusão	31
5	Bibliografia	33

Lista de figuras

Figura 2.1	Modelo Windspot 3.5kW WT	9
Figura 2.2	(a) Overview da montagem do experimento original na câmara climática; (b) Controlador Climático; (c) Shaker com isolamento de isopor; (d) strain gauges s1 e s2 montados no lado de baixa pressão; (e) strain gauges s23 e s24 montados no lado de alta pressão; (f) Sensor Climático; (g) Trincas 1 e 2; e (h) Trinca 3	10
Figura 2.3	Excitação senoidal, Casos R (subfiguras A e B) e D (subfiguras C e D), Medidas de deformação no sensor 6.2.	12
Figura 2.4	Excitação com ruído branco, Casos R (subfiguras A e B) e D (subfiguras C e D), Medidas de deformação no sensor 6.1.	12
Figura 2.5	Excitação senoidal, Caso R, Medidas de deformação no sensor de deformação 1 (subfiguras A e B) e 23 (subfiguras C e D).	13
Figura 2.6	Excitação senoidal, Casos R (subfiguras A e B) e D (subfiguras C e D), Medidas de aceleração no sensor 7.	13
Figura 2.7	Diagrama esquemático representando o pré-processamento de dados	15
Figura 2.8	Exemplo de um PCA em dados bidimensionais.	17
Figura 2.9	Gráfico com número de componentes e variância explicada para os casos de PCA antes do filtro e PCA depois do filtro, respectivamente. Ambos utilizaram o menor número de componentes necessários para atingir 99% de variância explicada.	18
Figura 2.10	Correlação dos 3 primeiros componentes principais para os casos PCA sem filtro e com filtro, respectivamente.	19
Figura 2.11	SVC em um conjunto de dados linearmente separável.	21
Figura 2.12	SVC em um conjunto de dados não linearmente separável. Observe que antes da aplicação do kernel não existe um hiperplano adequado para a separação dos dados	22
Figura 2.13	Diagrama representando o pipeline de treinamento e avaliação dos modelos.	23
Figura 2.14	Matriz de confusão para classificação binária.	25

Lista de tabelas

Tabela 2.1	Lista de casos de teste e descrições	11
Tabela 3.1	Resultados para o nível de referência	28
Tabela 3.2	Resultados para a busca de hiperparâmetros	29
Tabela 3.3	Comparação entre a busca de hiperparâmetros e a referência	29

1

Introdução

A implementação da detecção de falhas emergiu como uma ferramenta extraordinariamente eficaz no âmbito da manutenção e reparo de sistemas mecânicos, desempenhando um papel crucial na área de *Structural Health Monitoring* (SHM). Graças ao emprego de tecnologias inovadoras, como as técnicas de *Machine Learning* (ML), agora é viável realizar análises automatizadas utilizando dados coletados em tempo real por uma rede de sensores diversificados (Lei, 2020). Esse avanço tecnológico proporciona um processo substancialmente mais ágil, eliminando a necessidade de inspeções manuais conduzidas por técnicos altamente especializados, característica intrínseca aos métodos convencionais baseados no tempo de uso. Estes últimos, por sua vez, estão gradativamente cedendo lugar a abordagens mais contemporâneas, fundamentadas na avaliação da condição estrutural (Ou, 2020).

Essa transição para métodos baseados no monitoramento contínuo da condição estrutural com uso de sensores não apenas representa uma evolução significativa no campo da manutenção de sistemas mecânicos, mas também destaca a relevância crescente das tecnologias de detecção de falhas na busca por eficiência e confiabilidade operacional.

No vasto espectro de aplicações de SHM, destaca-se de maneira proeminente o seu emprego em turbinas eólicas (WTs, do inglês *Wind Turbines*), uma área que tem conquistado crescente notoriedade devido às suas potenciais contribuições para a geração de energia de maneira mais eficiente (Ou, 2020).

A relevância dessa aplicação também encontra respaldo no expressivo aumento do interesse global por fontes de energia limpa e renovável, notadamente evidenciado nos últimos anos. Entre essas fontes, a energia eólica assume um papel de destaque, em virtude de sua alta densidade energética, associada a benefícios significativos, como baixo impacto ambiental e custo reduzido por kWh gerado (Santelo, 2022).

No cenário atual, a necessidade premente de soluções energéticas sustentáveis têm impulsionado a pesquisa e o desenvolvimento de tecnologias inovadoras para aprimorar a eficiência dos parques eólicos. A aplicação do SHM em turbinas eólicas tornou-se, assim, um componente essencial nesse esforço coletivo, permitindo uma abordagem proativa na detecção e correção de falhas,

otimizando a operação e prolongando a vida útil desses equipamentos cruciais para a matriz energética sustentável.

Em Gharehbaghi, 2021, os autores apresentam um compilado de definições, métodos e perspectivas na área de SHM. O objetivo do SHM é monitorar a resposta de estruturas e detectar danos, mantendo a performance elevada e barateando a manutenção ao evitar falhas dos componentes. O artigo apresenta comparações entre os métodos tradicionais utilizados na área, com suas vantagens e desvantagens, citando também trabalhos anteriores de múltiplos autores.

De acordo com Santelo, 2022, a imprevisibilidade de falhas em WTs reduzem sua confiabilidade, gerando interrupções, perdas de produção e custos de manutenção elevados. SHM é uma das ferramentas para a melhoria da confiabilidade de WTs, e que deve aumentar em importância no futuro dada as tendências do mercado.

Em Sun, 2022, é apresentado um modelo de aprendizado auto supervisionado que treina utilizando apenas leituras de estruturas saudáveis, que são mais abundantes e mais fáceis de obter que de estruturas danificadas. O método utiliza data augmentation para gerar amostras negativas com pseudo-labels. Resultados obtidos mostram uma performance melhor que o estado-da-arte e uma excelente capacidade de generalização de representações para múltiplas WTs diferentes.

Diante da importância e aplicabilidades dos sistemas de monitoramento para falhas em sistemas mecânicos, o presente trabalho busca realizar a detecção de trincas em pás de WTs através de diferentes modelos de ML com o apoio de técnicas de redução dimensional como o Principal Component Analysis (PCA).

2 Metodologia

Para este trabalho foi utilizado um conjunto de dados de acesso público desenvolvido por uma equipe de pesquisadores da ETH-Zurich no artigo “Vibration-based monitoring of a small-scale wind turbine blade under varying climate conditions. Part I: An experimental benchmark” (Ou et al., 2020). Neste capítulo iremos detalhar as características do conjunto de dados utilizado, conforme o artigo original, bem como as técnicas e ferramentas utilizadas neste trabalho para o desenvolvimento dos modelos de previsão.

2.1 Estudo de Caso

Descrevemos aqui o estudo de caso realizado no artigo (Ou,2020), primeira parte de um estudo realizado em duas partes, com o objetivo de criar um benchmark para o estudo de técnicas de SHM aplicadas em WTs, com foco na integridade estrutural das pás das turbinas. Para tal, utilizaram as pás de um modelo Windspot 3.5kW WT, turbina de pequeno porte que foi fornecida pela Sonkyo Energy. Observa-se que as pás são um dos componentes mais importantes e os expostos às condições mais adversas, e suas falhas acarretam consideráveis prejuízos.

Para realizar os testes, a pá da turbina foi instalada em uma estrutura de aço fixada ao solo através de três colunas de concreto, na qual a pá foi firmemente fixada em uma de suas extremidades por quatro parafusos, criando assim uma estrutura de apoio fixo-livre, análoga às condições reais de uso da



Figura 2.1: Modelo Windspot 3.5kW WT

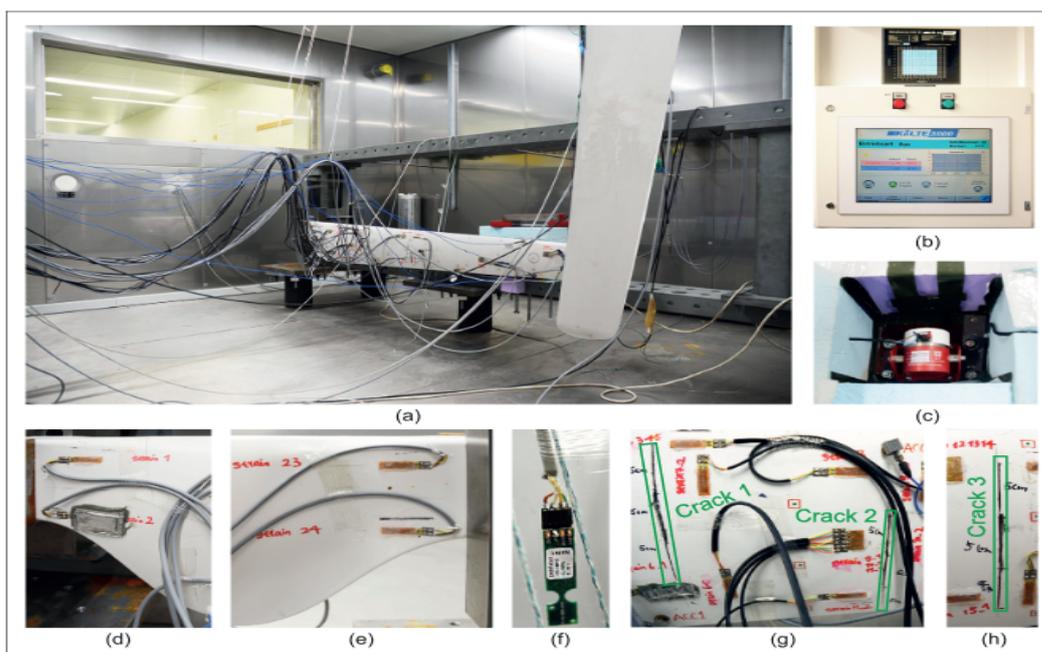


Figura 2.2: (a) Overview da montagem do experimento original na câmara climática; (b) Controlador Climático; (c) Shaker com isolamento de isopor; (d) strain gauges s1 e s2 montados no lado de baixa pressão; (e) strain gauges s23 e s24 montados no lado de alta pressão; (f) Sensor Climático; (g) Trincas 1 e 2; e (h) Trinca 3

turbina. Para minimizar o potencial ruído proveniente das operações laboratoriais adjacentes durante o processo de teste, foram colocados absorvedores de borracha entre a amostra e a estrutura.

A estrutura foi testada em estados saudáveis e danificados sob diferentes condições, incluindo 13 casos diferentes, dos quais: 1 caso base (*healthy*), 3 com massas adicionais (simulando o acúmulo de gelo) e 9 com trincas em múltiplas posições (sendo essas as três posições mais comuns para o surgimento das falhas) e de tamanhos variáveis (simulando problemas de diferentes severidades).

Para cada caso foram feitas leituras em diferentes temperaturas, de -15°C até $+40^{\circ}\text{C}$ com passos de 5°C (totalizando 12 temperaturas) e umidade relativa constante de 60%, ambas mantidas nos níveis desejados pelo uso de uma câmara climática onde foram realizados os experimentos. Dessa forma, é possível observar os impactos de variações de temperatura sobre a estrutura, mantendo sob controle possíveis variações na densidade do ar que poderiam ser causadas pela umidade. Também foram instalados sensores de temperatura e umidade na superfície do corpo de prova para garantir que as condições desejadas foram atingidas.

Na Tabela 2.1, detalhamos as condições em que foram realizados cada

Tabela 2.1: Lista de casos de teste e descrições

Caso	Descrição	Número de Testes
R	Caso saudável / Não dano	12 x 21
A	Massa adicional: 1x44g	12 x 6
B	Massa adicional: 2x44g	12 x 6
C	Massa adicional: 2x44g	12 x 6
D	Trinca 1: 5 cm;	12 x 6
E	Trinca 1: 5 cm; Trinca 2: 5 cm;	12 x 6
F	Trinca 1: 5 cm; Trinca 2: 5 cm; Trinca 3: 5 cm;	12 x 6
G	Trinca 1: 10 cm; Trinca 2: 5 cm; Trinca 3: 5 cm;	12 x 6
H	Trinca 1: 10 cm; Trinca 2: 10 cm; Trinca 3: 5 cm;	12 x 6
I	Trinca 1: 10 cm; Trinca 2: 10 cm; Trinca 3: 10 cm;	12 x 6
J	Trinca 1: 15 cm; Trinca 2: 10 cm; Trinca 3: 10 cm;	12 x 6
K	Trinca 1: 15 cm; Trinca 2: 15 cm; Trinca 3: 10 cm;	12 x 6
L	Trinca 1: 15 cm; Trinca 2: 15 cm; Trinca 3: 15 cm;	12 x 6

um dos testes, bem como a quantidade de testes para cada um dos casos de dano e não dano.

A excitação da pá da turbina foi realizada utilizando um shaker capaz de gerar vibrações mecânicas, que então foram transmitidas até a pá e medidas com um sensor de forças, determinando assim o sinal realmente observado pelo corpo. Foram feitos dois tipos de testes, ambos de 120 segundos cada, mas com diferentes tipos de excitação. No primeiro, foi gerado um sinal de ruído branco, com frequências variando de 0 até 400 Hz. Destacamos que esse teste foi repetido 20 vezes no caso R e 5 vezes nos demais. No segundo, foi feito um sine sweep com frequências de 1 até 300 Hz. Estas frequências foram escolhidas de forma a possibilitar a observação dos seis primeiros modos de ressonância da estrutura, conforme determinado com uma análise de elementos finitos.

Os sinais de saída foram obtidos utilizando uma rede de 8 acelerômetros e 18 *strain gauges*, ambos instrumentos comuns de uso geral em SHM. Em especial, para os *strain gauges* foram utilizadas duas configurações distintas, sendo na primeira estes alinhados com o eixo Z e na segunda alinhados com o eixo Y, permitindo assim o monitoramento de deformações nos dois eixos. Além disso, foi utilizado um segundo corpo de prova, não excitado e com *strain gauges* instalados de forma que fosse possível corrigir efeitos da temperatura nas leituras. Nas figuras abaixo, temos uma comparação visual de algumas diferentes leituras que foram obtidas e disponibilizadas em Ou,2020.

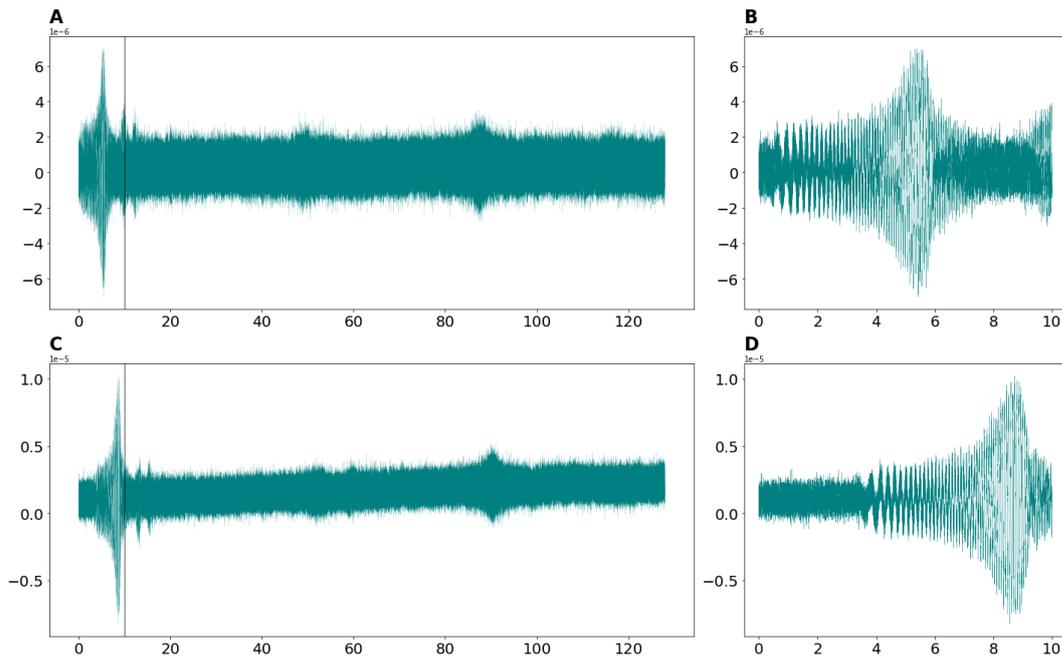


Figura 2.3: Excitação senoidal, Casos R (subfiguras A e B) e D (subfiguras C e D), Medidas de deformação no sensor 6.2.

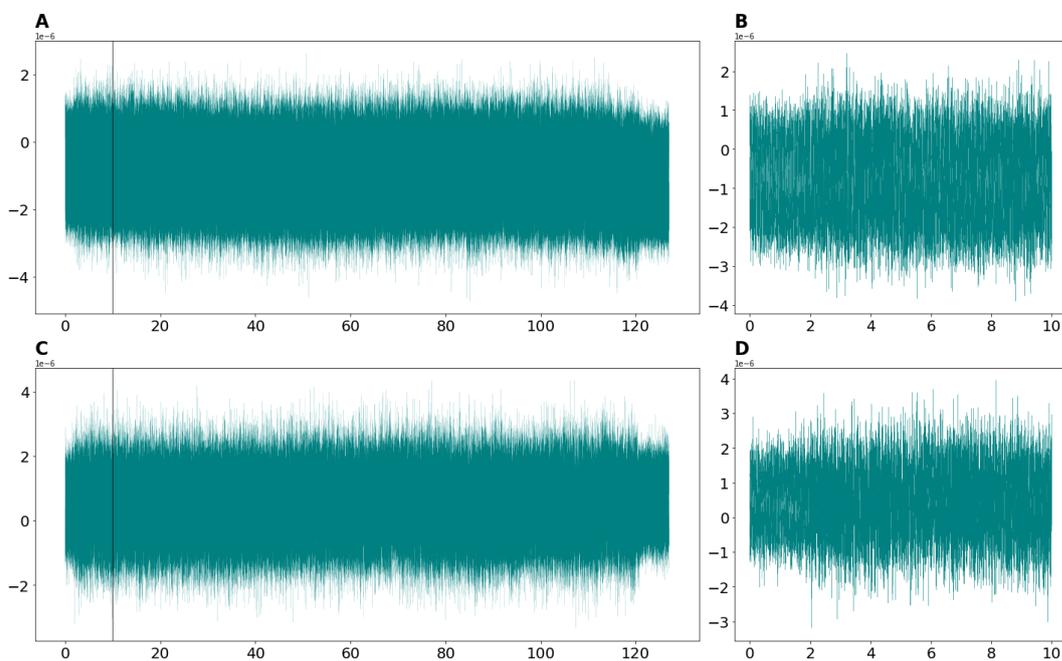


Figura 2.4: Excitação com ruído branco, Casos R (subfiguras A e B) e D (subfiguras C e D), Medidas de deformação no sensor 6.1.

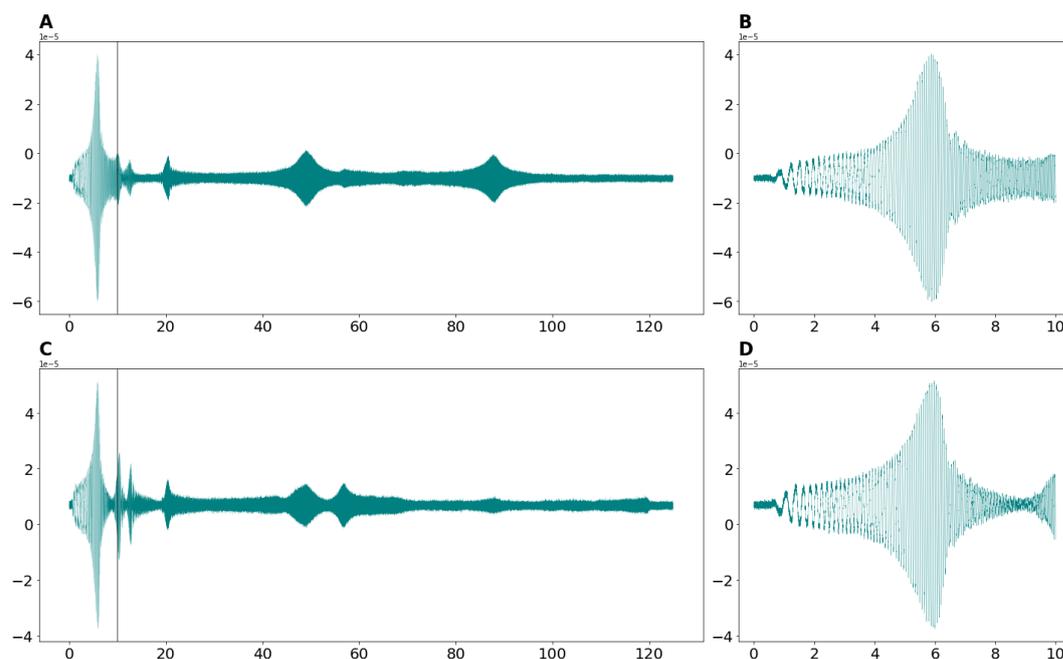


Figura 2.5: Excitação senoidal, Caso R, Medidas de deformação no sensor de deformação 1 (subfiguras A e B) e 23 (subfiguras C e D).

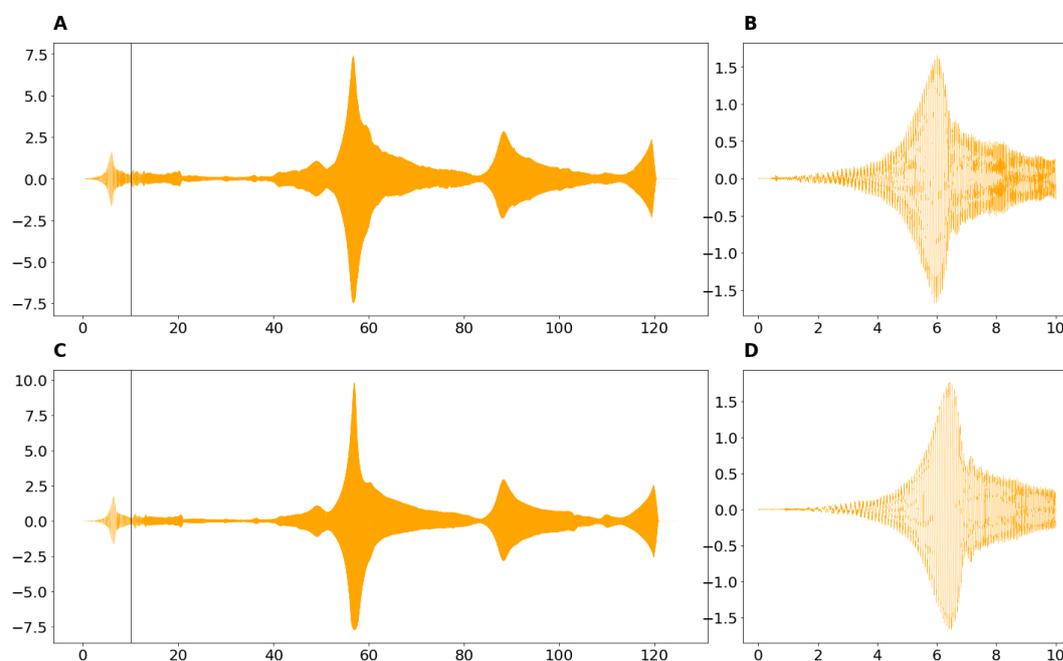


Figura 2.6: Excitação senoidal, Casos R (subfiguras A e B) e D (subfiguras C e D), Medidas de aceleração no sensor 7.

2.2

Feature Engineering

Utilizando a linguagem de programação *Python* e os arquivos disponibilizados pela pesquisa de Ou, 2020, foi desenvolvido um programa para realizar todas as etapas de tratamento dos dados bem como todo o processo de criação, treinamento e avaliação de cada modelo.

Devido às dificuldades de trabalhar com o peso excessivamente grande dos arquivos originais, o primeiro passo realizado foi uma extração de *features* manual utilizando de técnicas de *feature engineering*. O *dataset* original foi determinado como possuindo um total de 2402 casos individuais provenientes das múltiplas configurações de testes de ruído branco e *sinesweep* aplicados aos diferentes estados de dano em diferentes temperaturas. Cada um desses casos individuais possui até 19 canais de leitura para os *strain gauges*, além de mais 8 canais associados aos acelerômetros e as leituras de temperatura e umidade relativa. Cada um desses canais possui até mais de 200.000 leituras.

De acordo com Brunton, 2019, *Machine Learning* se baseia em técnicas de otimização do uso de dados, com o objetivo de tanto encontrar um subespaço de baixa dimensionalidade para representar os dados como de encontrar métodos de regressão e classificação adequados aos diferentes dados utilizados. Dessa forma, temos um conjunto de métodos matemáticos para a extração de *features* dos dados. Parte fundamental desse processo é a escolha correta de quais dados ou *features* devem ser utilizados e de que forma, a esse processo de extração de *features* dos dados brutos damos o nome de *Feature Engineering*.

Neste processo buscamos, portanto, criar representações mais significativas e informativas dos dados, pois a qualidade e relevância dessas *features* desempenham um papel fundamental no desempenho final do modelo, contribuindo para a melhoria da precisão e capacidade de generalização. Para tal, podemos tanto aplicar transformações e outras medidas de pré-processamento de dados como também criar novos *features* com técnicas como *data augmentation*. Neste trabalho, optamos por criar essas *features* utilizando os dados estatísticos das *features* originais.

2.3

Features Estatísticas

Para este estudo, foi decidido utilizar apenas as leituras de aceleração de cada caso, e a fim de capturar informações pertinentes para as leituras foram retiradas as medidas estatísticas de mínimo, média, máximo, variância, assimetria e curtose de cada canal individual. Sendo assim, o *dataset* resultante mantém todos os 2402 casos de testes, mas resume as leituras de cada um deses

em 48 *features* estatísticos. O diagrama da figura 2.7 representa de maneira esquemática o processo realizado.

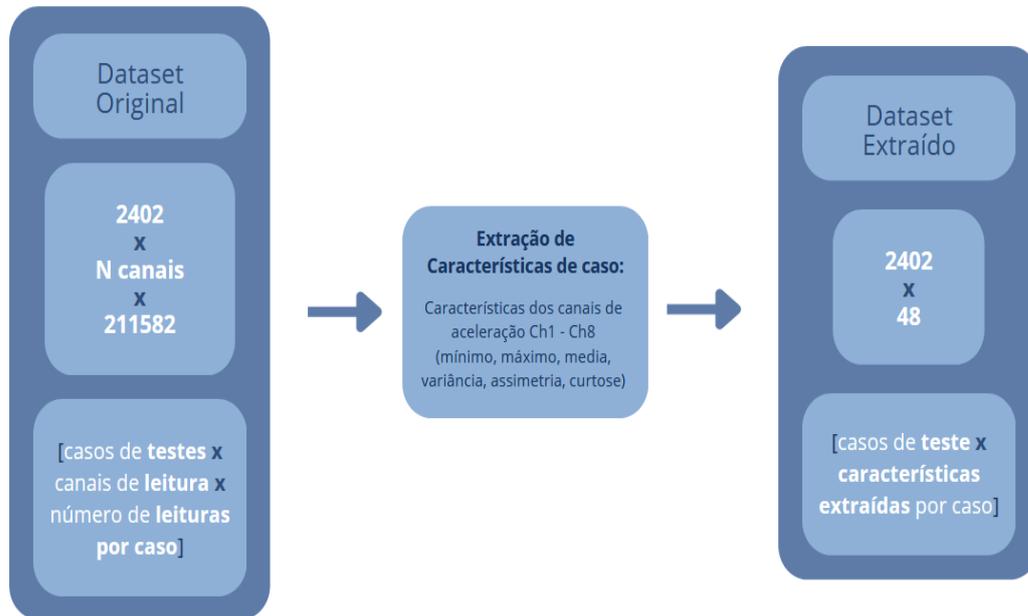


Figura 2.7: Diagrama esquemático representando o pré-processamento de dados

Dentre as estatísticas utilizadas, destacamos as seguintes definições:

2.3.1

Média

A média é uma medida estatística que representa o valor central de um conjunto de dados. É calculada somando todos os valores e dividindo pelo número total de observações. A média é sensível a valores extremos (*outliers*), pois é afetada por todas as observações no conjunto de dados.

$$\text{Média} = \mathbb{E}(X) = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x} \quad (2-1)$$

2.3.2

Variância

A variância é uma medida que expressa a dispersão dos dados em relação à média. Ela é equivalente ao segundo momento central, assim sendo calculado como a média da soma dos quadrados da diferença entre cada leitura e o valor médio ou valor esperado. Uma variância maior indica uma maior dispersão dos dados em torno da média, enquanto uma variância menor indica uma distribuição mais concentrada.

$$\text{Variância} = \mathbb{E}[(X - \bar{x})^2] = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 = \sigma^2 \quad (2-2)$$

2.3.3

Assimetria

A assimetria é uma medida que descreve a falta de simetria em uma distribuição de dados e é equivalente ao terceiro momento central da distribuição. Uma distribuição simétrica tem uma assimetria zero e equivale ao terceiro momento central. Se a cauda da distribuição se estende mais para a direita em relação a média, a distribuição é positivamente assimétrica (ou inclinada para a direita). Se a cauda se estende mais para a esquerda, a distribuição é negativamente assimétrica (ou inclinada para a esquerda).

$$\text{Assimetria} = \mathbb{E}\left[\left(\frac{X - \bar{x}}{\sigma}\right)^3\right] = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^{3/2}} \quad (2-3)$$

2.3.4

Curtose

A curtose pode ser interpretada como o grau de "achatamento" ou "pico" de uma distribuição em relação a uma distribuição normal. Uma curtose alta indica que os dados têm caudas mais pesadas ou são mais "pontigudos" em relação a uma distribuição normal. Uma curtose baixa indica que os dados têm caudas mais leves ou são mais "achatados" em comparação com uma distribuição normal. A curtose é frequentemente comparada à curtose de uma distribuição normal, que é 3.

$$\text{Curtose} = \mathbb{E}\left[\left(\frac{X - \bar{x}}{\sigma}\right)^4\right] = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^2} \quad (2-4)$$

2.4

PCA e Detecção de Outliers

Com o *dataset* reduzido em mãos, foram realizados múltiplos testes para os diferentes modelos de classificação previamente selecionados. No primeiro caso de teste, foi utilizado como input dos modelos os *features* extraídos no passo anterior sem qualquer outra medida de tratamento, enquanto nos casos subsequentes foi utilizado um *Principal Component Analysis* (PCA) a fim de

verificar o impacto e a viabilidade da aplicação de redução dimensional no dataset extraído.

De acordo com Géron, 2019, PCA é uma técnica de redução de dimensionalidade que transforma um conjunto de dados de alta dimensionalidade em um conjunto de dados de baixa dimensionalidade, preservando a maior parte da informação original. Isso é feito encontrando um conjunto de vetores ortogonais, chamados de componentes principais, que capturam a maior parte possível da variância nos dados originais para a dada dimensão reduzida. Esse procedimento nos permite que ao trabalhar com dados de alta dimensionalidade e algum grau de correlação, seja possível representar esses dados ignorando as últimas dimensões de forma que as informações perdidas sejam mínimas.

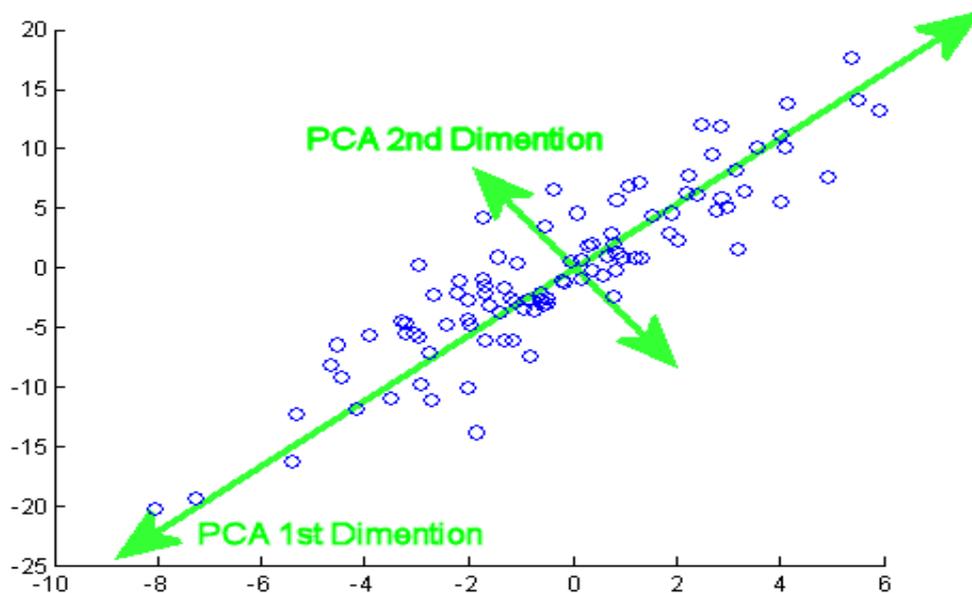


Figura 2.8: Exemplo de um PCA em dados bidimensionais.

A quantidade de componentes principais a serem utilizados é um parâmetro importante do PCA. Quanto mais componentes forem utilizados, mais informações serão preservadas, mas também mais dimensionalidade será preservada, existe assim um *trade-off* entre a redução dimensional e a precisão que pode ser obtida.

Essa quantidade de componentes pode ser definida escolhendo um número fixo arbitrário antes da execução do PCA, ou após a execução do PCA com base em um critério como a variância explicada total. A variância explicada total é a proporção da variância total dos dados que é explicada pelos componentes principais.

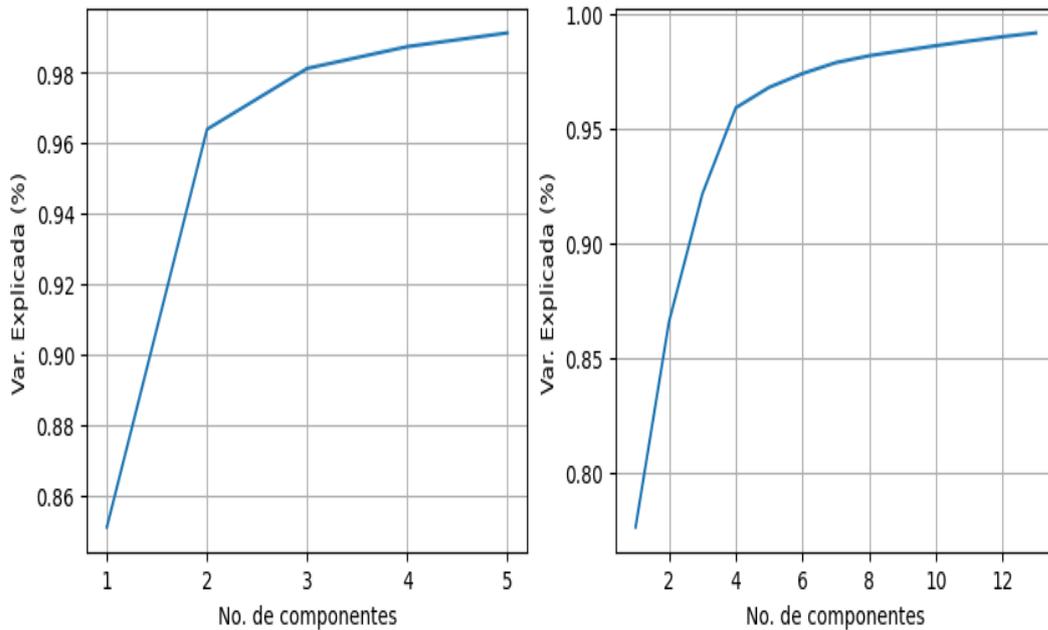


Figura 2.9: Gráfico com número de componentes e variância explicada para os casos de PCA antes do filtro e PCA depois do filtro, respectivamente. Ambos utilizaram o menor número de componentes necessários para atingir 99% de variância explicada.

De acordo com Brunton, 2019, PCA é extremamente sensíveis a *outliers* devido ao escalonamento quadrático das distâncias entre os pontos do conjunto e os componentes principais a serem determinados. A fim de combater possíveis problemas advindos deste fato, optamos por realizar uma filtragem de *outliers*. Notamos que para realizar o tratamento de *outliers*, é possível tanto descartá-los, caso suas ausências não sejam prejudiciais à abrangência do conjunto de dados, ou realizar intervenções diretas com pré-processamento de dados. O método de filtragem escolhido para determinar quais pontos são *outliers* foi simples: tomamos a mediana dos valores e rejeitamos aqueles que desviam em mais de 40% dela. Para avaliar os impactos dessa filtragem durante os testes utilizando PCA, foram adotados dois procedimentos de filtragem diferentes:

No primeiro, aplicamos o PCA no conjunto de dados e em seguida aplicamos o filtro de *outliers* no resultado do PCA, descartando todos os pontos filtrados.

No segundo, aplicamos o filtro antes do PCA, permitindo realizar o ajuste da transformação utilizando apenas os dados considerados não-*outliers*. Com a matriz de transformação em mãos, aplicamos a mesma ao conjunto de dados completo, de forma que os *outliers* ainda poderão ser utilizados como input para os modelos.

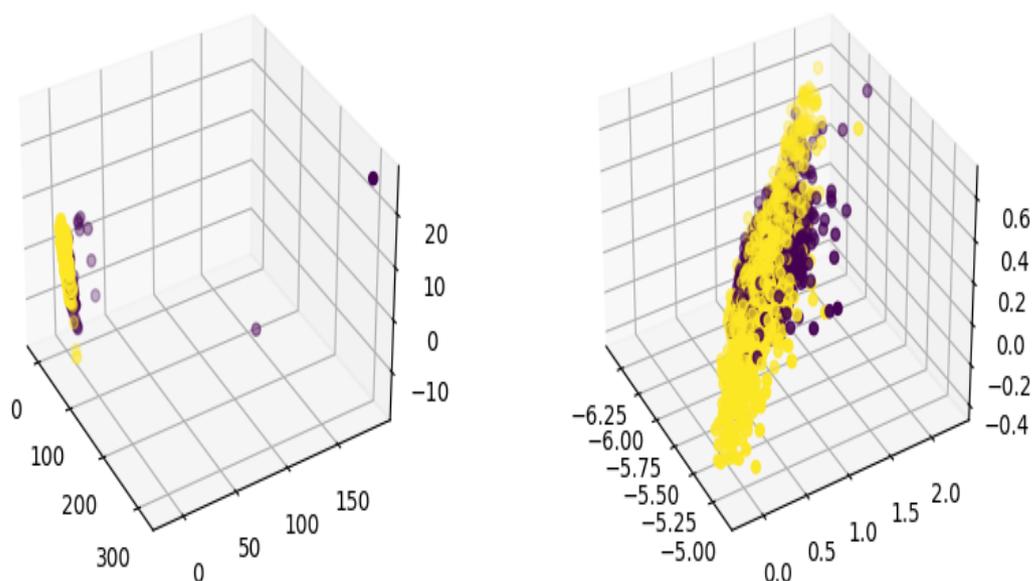


Figura 2.10: Correlação dos 3 primeiros componentes principais para os casos PCA sem filtro e com filtro, respectivamente.

2.5 Modelos Utilizados

Quanto aos modelos que foram selecionados para os testes, estes foram os chamados *Decision Trees* (DTs), *Linear Discriminant Analysis* (LDAs) e *Support Vector Machines* (SVMs), que iremos descrever abaixo:

2.5.1 Árvore de Decisão

Uma árvore de decisão, ou *Decision Tree*, é uma representação de um modelo de decisão que utiliza uma estrutura em forma de árvore, composta por nodos e ramos, para mapear decisões e suas possíveis consequências. Cada nodo interno da árvore representa uma escolha com base em uma característica ou atributo do conjunto de dados, enquanto os ramos conectam os nodos, indicando os diferentes resultados possíveis dessas decisões.

Os nodos folha representam as saídas ou resultados finais do processo de tomada de decisão. As árvores de decisão são amplamente utilizadas em aprendizado de máquina e mineração de dados para classificação e regressão. O processo de construção da árvore envolve a escolha iterativa dos atributos mais relevantes para dividir os dados em subconjuntos homogêneos, buscando maximizar a pureza das classes nos nodos folha. Essa abordagem torna as árvores de decisão altamente interpretáveis e eficazes para a tomada de decisões

em uma variedade de domínios, além de serem suscetíveis a técnicas como poda para evitar *overfitting* e melhorar a generalização do modelo.

2.5.2

Análise de Discriminante Linear

A Análise Discriminante Linear, ou *Linear Discriminant Analysis*, é uma técnica estatística utilizada para encontrar combinações lineares de variáveis que melhor separam duas ou mais classes em um conjunto de dados. O principal objetivo desse método é maximizar a variabilidade entre as classes enquanto minimiza a variabilidade dentro de cada classe. Isso proporciona uma técnica poderosa para a redução de dimensionalidade e classificação de padrões. A LDA é frequentemente aplicada em problemas de reconhecimento de padrões e aprendizado de máquina, sendo particularmente útil quando se deseja identificar as características mais discriminantes que distinguem diferentes grupos em um conjunto de dados. Ao transformar os dados em um novo espaço de características, a LDA busca otimizar a separação entre as classes, melhorando assim o desempenho de algoritmos de classificação.

O *Ridge Classifier* pode ser considerado um subgrupo da LDA com uma abordagem específica para lidar com multicolinearidade e regularização. Enquanto a LDA tradicional busca encontrar as combinações lineares de variáveis que melhor separam diferentes classes em um conjunto de dados, o *Ridge Classifier* incorpora um termo de regularização L2 (também conhecido como penalidade de *Ridge*) para lidar com a multicolinearidade, ou seja, a presença de alta correlação entre variáveis independentes. Esse termo de regularização é especialmente útil quando as variáveis estão altamente correlacionadas, evitando que o modelo atribua pesos extremos a variáveis semelhantes.

Assim, o *Ridge Classifier*, ao integrar a regularização L2 à estrutura da LDA, contribui para a estabilidade do modelo e melhora sua capacidade de generalização em conjuntos de dados complexos. Essa combinação de técnicas torna o *Ridge Classifier* uma escolha eficaz em situações em que a multicolinearidade pode impactar negativamente a performance do modelo de Análise Discriminante Linear convencional.

2.5.3

Máquinas de Vetores de Suporte

As Máquinas de Vetores de Suporte, ou *Support Vector Machine*, são uma gama versátil de técnicas de Aprendizado de Máquina que encontram aplicação tanto em problemas de classificação quanto em problemas de regressão. Neste

contexto, onde o desafio é a classificação, concentraremos nossa atenção nos *Support Vector Classifiers* (SVC), que se especializam em categorizar dados.

Em termos de classificação, o princípio fundamental dos SVCs é separar um conjunto de dados em suas classes usando hiperplanos. Em um cenário bidimensional de classificação binária, você pode imaginar isso como o uso de uma linha reta que divide o espaço em duas regiões distintas, cada uma representando uma classe.

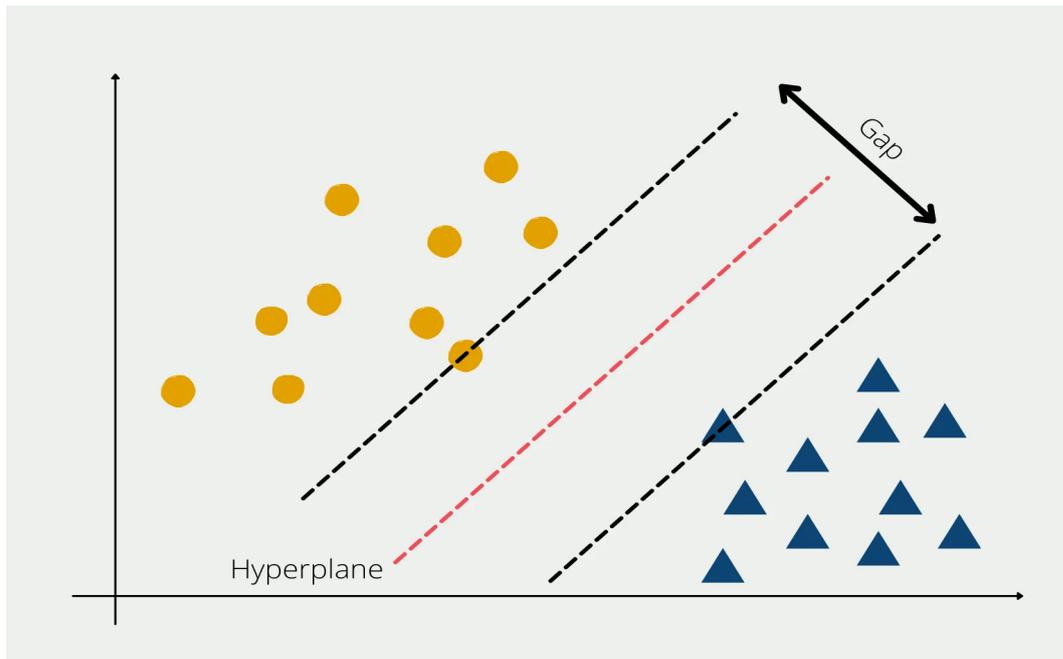


Figura 2.11: SVC em um conjunto de dados linearmente separável.

Além disso, ao realizar esta divisão temos também como objetivo maximizar o tamanho da margem criada entre o hiperplano e os conjuntos de dados ao mesmo tempo que minimizamos a quantidade de pontos classificados erroneamente.

No entanto, na prática, essa abordagem tem limitações, pois exige que os dados sejam linearmente separáveis. Na maioria dos casos reais, os dados não se encaixam nesse critério, o que nos leva à necessidade de utilizar “kernels”. Os *kernels* são funções que aumentam a dimensionalidade dos dados, tornando mais fácil a separação das classes.

Esses *kernels* permitem que as SVM lidem com dados que não são linearmente separáveis, tornando-as uma ferramenta poderosa e flexível para a classificação em uma ampla variedade de situações do mundo real. Eles podem ser aplicados a problemas complexos onde a separação das classes não é simples, tornando as SVM uma técnica valiosa no campo de ML.

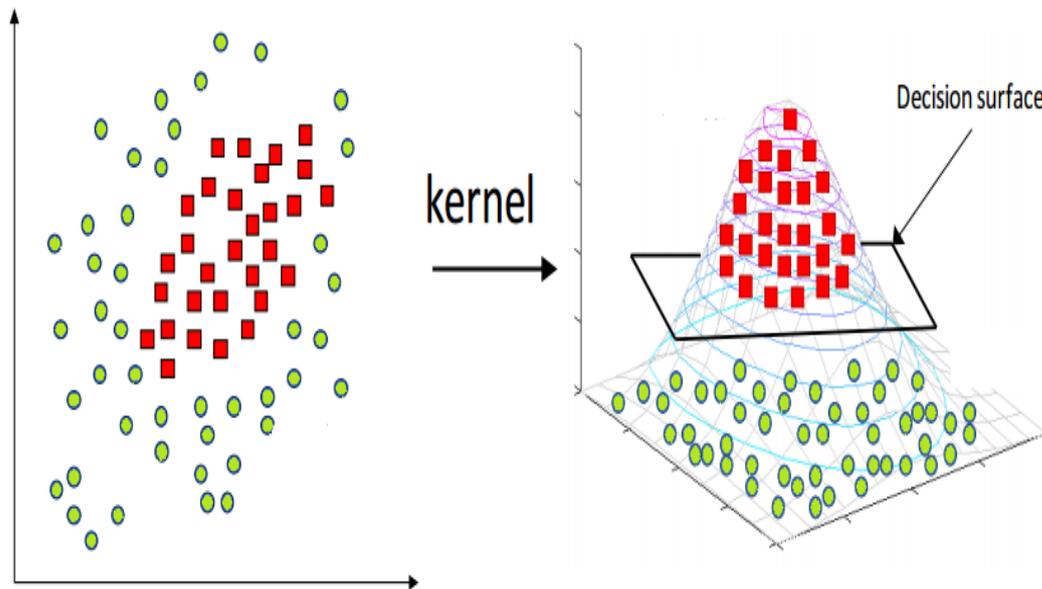


Figura 2.12: SVC em um conjunto de dados não linearmente separável. Observe que antes da aplicação do kernel não existe um hiperplano adequado para a separação dos dados

2.6

Treino e Avaliação dos Modelos

Tendo descritos todos os procedimentos de pré-processamento de dados utilizados bem como os modelos que foram treinados, resta apenas detalhar o procedimento de treinamento em si e as métricas pelos quais estes serão avaliados. A figura 2.13 representa, em uma abstração de alto nível, os passos descritos nos itens anteriores, complementando assim o digrama da figura 2.7 de forma a representar o pipeline completo utilizado.

Entre os vários modelos utilizados, dentro do mesmo tipo de modelo, pode haver uma ampla variação de desempenho. Em primeiro lugar, os modelos de aprendizado de máquina possuem parâmetros que devem ser definidos antes de sua criação e não podem ser ajustados durante o treinamento. Esses parâmetros são conhecidos como hiperparâmetros.

2.6.1

Busca de hiperparâmetros

Para garantir que a seleção de hiperparâmetros seja apropriada para o objetivo pretendido, é necessário realizar uma busca no espaço de possibilidades dos hiperparâmetros em um processo denominado otimização de hiperparâmetros. Existem diversas técnicas para conduzir esse processo, mas aqui

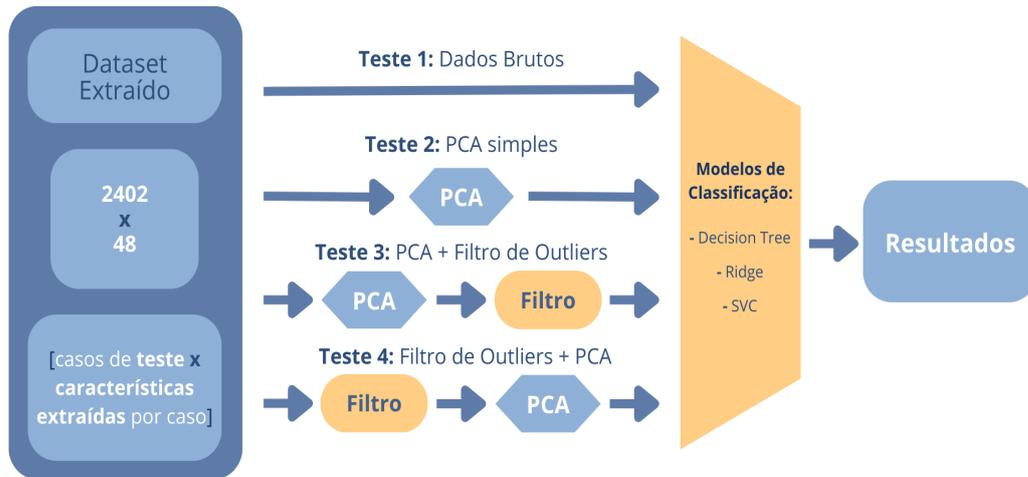


Figura 2.13: Diagrama representando o pipeline de treinamento e avaliação dos modelos.

abordaremos duas delas: *Grid Search* e *Random Search*.

O *Grid Search* representa o método mais tradicional e consiste em uma abordagem de busca exaustiva, explorando sistematicamente todas as combinações de parâmetros a partir de um conjunto finito de valores previamente definidos. Como o método de *Grid Search* busca todas as combinações possíveis de maneira exaustiva, seu custo computacional é mais elevado que o *Random Search*, e cresce exponencialmente junto do número de hiperparâmetros e seus valores a serem testados.

Por outro lado, o *Random Search* adota uma abordagem mais flexível. Em vez de explorar todas as combinações de parâmetros, ele opta por uma seleção aleatória a partir dos conjuntos de valores possíveis para cada hiperparâmetro. Se tratando do *random search*, esses conjuntos de valores possíveis podem ser finitos ou infinitos.

Em ambos os casos, o procedimento é semelhante. Para cada conjunto de parâmetros selecionado, um modelo correspondente é criado, treinado e avaliado usando o conjunto de dados e a função objetivo desejada. Esse processo de avaliação repetitivo é fundamental para determinar quais combinações de hiperparâmetros resultam no melhor desempenho do modelo, permitindo uma escolha informada da configuração ideal.

2.6.2 Cross-Validation

Em geral, o processo de treinamento de um modelo de aprendizado de máquina ocorre de maneira estocástica, o que significa que podem ocorrer variações no desempenho de dois modelos, mesmo que sejam criados e treinados seguindo os mesmos procedimentos. Para obter uma avaliação mais robusta do desempenho de cada modelo, uma avaliação que seja menos suscetível às flutuações inerentes aos métodos, fazemos uso de uma técnica denominada *cross-validation*.

De maneira geral, *Cross-validation* se baseia em subdividir o conjunto de dados disponíveis para treinamento em K partes de igual tamanho, chamadas *folds*, das quais designamos de maneira aleatória uma ou mais partes como um conjunto de teste e o restante como um conjunto de treino. Os parâmetros do modelo são ajustados utilizando apenas o conjunto de treino, mas o modelo é avaliado nos dois conjuntos. Esse procedimento é então repetido até que todas as partes do conjunto de dados tenham sido utilizadas na validação e no treino das múltiplas iterações do modelo. Utilizamos assim uma média da função de perda em cada iteração para melhor avaliar o desempenho do modelo.

Como medida adicional, também é comum utilizar repetições adicionais nos processos de *cross-validation* criando assim mais uma camada de redundância.

2.7 Métricas

Quando estamos treinando um modelo de machine learning, seja ele qual for, precisamos de meios de avaliar seus desempenhos a fim de escolher aquele que melhor se adapta ao conjunto de dados e tipo de tarefa a ser realizado. Neste trabalho foram consideradas as métricas de *Balanced Accuracy*, *Recall* e *F1-Score*. Antes de detalhar essas métricas, tomemos a matriz de confusão, pois entendendo esta é mais fácil compreender as outras métricas.

A matriz de confusão é uma tabela $N \times N$, onde N é a quantidade de classes presente no *dataset*. Ela é uma ferramenta útil para avaliar a performance de um modelo de classificação, permitindo visualizar quantos exemplos foram classificados corretamente e quantos foram classificados de maneira incorreta pelo modelo. Sua construção é feita a partir de linhas e colunas representando os valores reais e valores previstos, onde cada elemento (i,j) da matriz representa a quantidade de exemplos pertencentes a classe i que foram classificados como parte da classe j .

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figura 2.14: Matriz de confusão para classificação binária.

Para o caso da classificação binária, a matriz de confusão assume um tamanho 2 x 2 e damos nomes específicos a cada um de seus elementos:

- Verdadeiros Positivos (TP): São os casos em que o modelo classificou corretamente exemplos como pertencentes à classe positiva.
- Verdadeiros Negativos (TN): São os casos em que o modelo classificou corretamente exemplos como pertencentes à classe negativa.
- Falsos Positivos (FP): São os casos em que o modelo classificou incorretamente exemplos da classe negativa como pertencentes à classe positiva. Em outras palavras, o modelo errou ao indicar que algo é positivo quando, na verdade, é negativo.
- Falsos Negativos (FN): São os casos em que o modelo classificou incorretamente exemplos da classe positiva como pertencentes à classe negativa. Nesse caso, o modelo errou ao não identificar corretamente algo como positivo.

As métricas que foram utilizadas para avaliação dos modelos podem ser obtidas facilmente a partir desses 4 elementos. Abaixo, discutimos as métricas utilizadas.

2.7.1 Sensibilidade

O *recall*, também conhecido como sensibilidade, é uma métrica que avalia a capacidade de um modelo em identificar corretamente todos os itens de uma classe específica em relação ao total de itens que realmente pertencem a essa classe. Ele é particularmente útil quando se deseja minimizar a ocorrência de falsos negativos, ou seja, quando a não identificação de itens da classe positiva é indesejável. No entanto, o *recall* não leva em consideração a métrica dos falsos positivos, o que significa que um modelo treinado com foco exclusivo no *recall* pode incorrer em um grande número de falsos negativos, classificando incorretamente muitos elementos como não pertencentes à classe em questão. Em um cenário extremo, esse modelo pode rotular a maioria dos dados como pertencentes à classe negativa, resultando em um *recall* alto, mas com uma capacidade limitada de distinguir efetivamente entre as classes.

$$\text{Recall} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}} \quad (2-5)$$

2.7.2 Precisão

A precisão é uma métrica que avalia a porcentagem de itens corretamente classificados em uma classe específica em relação ao total de itens previstos como pertencentes a essa classe. Ela é particularmente útil quando se busca minimizar a ocorrência de falsos negativos, ou seja, quando a classificação incorreta de itens que realmente pertencem à classe em questão é indesejável. No entanto, a precisão não considera a métrica de falsos positivos, o que significa que um modelo treinado com ênfase exclusiva na precisão pode incorrer em um grande número de falsos positivos, classificando erroneamente muitos elementos como pertencentes à classe em questão. Em um cenário extremo, esse modelo pode simplesmente rotular todos os dados como pertencentes à classe positiva, resultando em uma alta precisão, mas sem uma capacidade efetiva de distinguir entre as classes.

$$\text{Precision} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}} \quad (2-6)$$

2.7.3 Acurácia Balanceada

A *Balanced Accuracy*, ou acurácia balanceada, é uma métrica que leva em consideração o desequilíbrio de classes em um conjunto de dados. Em situações onde as classes estão desequilibradas (ou seja, há muito mais instâncias de uma

classe do que de outra), a acurácia padrão pode ser enganosa. A acurácia balanceada considera a taxa de verdadeiros positivos (TP) e verdadeiros negativos (TN) de cada classe e calcula a média dessas taxas. Isso ajuda a avaliar o desempenho do modelo de maneira mais justa quando as classes não estão equilibradas.

$$\text{Balanced Accuracy} = \frac{\text{Recall} + \frac{\text{Verdadeiros Negativos}}{\text{Verdadeiros Negativos} + \text{Falsos Positivos}}}{2} \quad (2-7)$$

2.7.4

F1-Score

O *F1-Score* é equivalente a média harmônica entre o recall e a precisão, resultando em uma maior penalização caso qualquer um dos dois tenha um valor baixo. Devido as propriedades da média harmônica, essa métrica sempre resultara em uma penalização maior associada a parte de valor mais baixo, tendo um resultado menor que a media aritmética simples.

$$\text{Recall} = \frac{2 \times \text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (2-8)$$

3 Resultados

Neste capítulo iremos discutir os resultados obtidos para os modelos de previsão de dano desenvolvidos utilizando as técnicas previamente descritas. Ressaltamos novamente os 4 casos de teste considerados, observe que os casos 3 e 4 diferem na ordem em que as operações foram realizadas:

- Caso 1: *Features* Estatísticas Brutas
- Caso 2: *Features* Estatísticas com aplicação de PCA
- Caso 3: *Features* Estatísticas com PCA e filtragem de *outliers*
- Caso 4: *Features* Estatísticas com filtragem de *outliers* e PCA

Para cada um dos casos de teste foram treinados os 3 tipos de modelos para os quais foram realizadas as buscas de hiperparâmetros, as quais foram repetidas 50 vezes para que fosse possível determinar o desempenho médio do processo. A fim de obter um nível basal de referência para que também fosse possível avaliar o impacto e eficiência da busca, também criado e treinado um modelo utilizando os hiperparâmetros *default* da biblioteca.

Observamos pela tabela 3.1 que, em média, o melhor caso de teste foi o caso 1, com um *recall* superior a 95%, enquanto os piores foram os casos 2 e 3. Estes resultados indicam que as *features* extraídas sozinhas são mais adequadas

Tabela 3.1: Resultados para o nível de referência

		Balanced Accuracy		F1-Score		Recall	
		Média	Std dev	Média	Std dev	Média	Std dev
Caso 1	DTC	0,950	0,010	0,962	0,008	0,962	0,009
	RDG	0,911	0,010	0,939	0,007	0,980	0,007
	SVC	0,884	0,014	0,920	0,009	0,964	0,009
Caso 2	DTC	0,785	0,015	0,828	0,012	0,828	0,016
	RDG	0,565	0,014	0,755	0,012	0,934	0,010
	SVC	0,787	0,017	0,842	0,014	0,874	0,023
Caso 3	DTC	0,758	0,014	0,799	0,015	0,799	0,022
	RDG	0,806	0,012	0,845	0,010	0,862	0,016
	SVC	0,831	0,012	0,862	0,011	0,870	0,015
Caso 4	DTC	0,845	0,011	0,876	0,010	0,877	0,017
	RDG	0,811	0,012	0,875	0,010	0,948	0,013
	SVC	0,871	0,012	0,912	0,008	0,962	0,010

Tabela 3.2: Resultados para a busca de hiperparâmetros

		Balanced Accuracy		F1-Score		Recall	
		Média	Std Dev	Média	Std Dev	Média	Std Dev
Caso 1	DTC	0,960	0,009	0,969	0,007	0,969	0,009
	RDG	0,925	0,011	0,942	0,008	0,951	0,012
	SVC	0,958	0,007	0,966	0,006	0,967	0,010
Caso 2	DTC	0,802	0,015	0,841	0,011	0,842	0,024
	RDG	0,565	0,014	0,755	0,012	0,934	0,010
	SVC	0,793	0,015	0,847	0,012	0,878	0,020
Caso 3	DTC	0,780	0,015	0,816	0,016	0,816	0,033
	RDG	0,807	0,011	0,846	0,010	0,861	0,018
	SVC	0,831	0,012	0,862	0,010	0,870	0,015
Caso 4	DTC	0,852	0,012	0,882	0,010	0,886	0,024
	RDG	0,814	0,013	0,874	0,010	0,936	0,028
	SVC	0,873	0,011	0,913	0,008	0,960	0,010

Tabela 3.3: Comparação entre a busca de hiperparâmetros e a referência

		Balanced Accuracy		F1-Score		Recall	
		Abs.	%	Abs.	%	Abs.	%
Caso 1	DTC	0,00826	0,87	0,00656	0,68	0,00650	0,67
	RDG	0,01420	1,56	0,00271	0,29	-0,02888	-2,95
	SVC	0,07386	8,35	0,04606	5	0,00243	0,25
Caso 2	DTC	0,01611	2,05	0,01320	1,59	0,01387	1,67
	RDG	-0,00009	-0,02	-0,00006	-0,01	-0,00007	-0,01
	SVC	0,00540	0,69	0,00426	0,51	0,00455	0,52
Caso 3	DTC	0,02163	2,85	0,01741	2,18	0,01669	2,09
	RDG	0,00101	0,13	0,00032	0,04	-0,00087	-0,1
	SVC	0,00006	0,01	0,00001	0	-0,00012	-0,01
Caso4	DTC	0,00639	0,76	0,00592	0,68	0,00860	0,98
	RDG	0,00299	0,37	-0,00123	-0,14	-0,01137	-1,2
	SVC	0,00150	0,17	0,00051	0,06	-0,00152	-0,16

para a determinação da presença de dano, dispensando a necessidade do uso de outras técnicas de tratamento de maneira conjunta.

Simultaneamente, o desempenho comparativamente ruim dos casos 2 e 3 indica que a aplicação do PCA sem o tratamento prévio dos *outliers* foi notavelmente prejudicial ao modelo, fato que é facilmente verificável uma vez que o caso 4 obteve um melhor resultado geral que os dois.

Na tabela 3.2 podemos observar que o comportamento foi similar ao caso da tabela 3.1 . De fato, a tabela 3.3 confirma que para maior parte das combinações de caso teste e tipo de modelo não houve uma melhora significativa no desempenho, com o único ponto em que houve um ganho superior a 3% sendo para o modelo SVC no caso 1. Também chama atenção o modelo ridge no caso 1 que apesar de um canho de 1.5% de *balanced accuracy*

perde 2.95% de seu *recall*.

De maneira geral é possível afirmar que os três tipos de modelos considerados obtiveram resultados comparáveis, não sendo observável nenhuma dominância clara de um deles. Ocorre, na verdade, um *trade-off* entre as métricas de avaliação. Um ponto de destaque interessante é o modelo ridge no caso 2, que obteve 93% de *recall* mas apenas 56% de *balanced accuracy*, sendo possível inferir que sua precisão foi de aproximadamente 20%. Isso provavelmente se deve a este ser um caso com presença de *outliers* que podem ter sido exacerbados por um PCA sem a presença de nenhum filtro de *outliers*, fato ao que o modelo ridge estaria especialmente vulnerável quando comparado aos outros modelos por ser um modelo linear.

4 Conclusão

A aplicação de técnicas de aprendizado de máquina ao monitoramento estrutural (SHM) tem se mostrado uma abordagem promissora para a detecção e identificação de danos em estruturas. Ao integrar o poder analítico das máquinas no monitoramento contínuo da integridade estrutural, evidenciamos não apenas melhorias na detecção de falhas, mas também uma resposta adaptativa e proativa a potenciais problemas.

Do desenvolvimento do trabalho podemos concluir que apesar de alguns métodos demonstrarem resultados aquém do esperado, em especial o uso do PCA para redução dimensional tendo se mostrado redundante e a busca por hiperparâmetros ineficiente, ainda foi possível desenvolver modelos capazes de identificar a presença de dano nas estruturas com grande grau de confiabilidade.

Ressaltamos que com os custos elevados associados as rotinas de manutenção é interessante minimizar os casos de falsos positivos que poderiam gerar uma desnecessária paralisação dos geradores apenas para que fosse verificado que na verdade a manutenção ainda não era necessária. Também é importante que dado a natureza incremental do agravamento do tipo de falha aqui estudado se torna um trade-off favorável a não identificação de falhas pequenas, e que são mais difíceis de se identificar, de forma que não comprometam a segurança estrutural se assim for possível garantir que os casos saudáveis também não serão identificados incorretamente.

Sob esta ótica, os dados destacam a importância da sinergia entre SHM e Machine Learning para a manutenção preditiva e a extensão da vida útil de estruturas. Embora existam variações nos desempenhos dos métodos, a tendência geral é a de benefícios substantivos na integridade estrutural e na redução de custos.

Em suma, este estudo destaca a eficácia do Machine Learning no SHM, oferecendo uma perspectiva promissora para a manutenção preditiva e a vida útil de estruturas críticas. Os avanços notáveis reforçam a necessidade de pesquisa e inovação para superar desafios específicos e melhorar a robustez do SHM com Machine Learning. À medida que avançamos para o futuro, a integração dessas tecnologias promete revolucionar a eficiência operacional e

fortalecer a resiliência estrutural, contribuindo para ambientes mais seguros e sustentáveis.

5 Bibliografía

BRUNTON, Steven L; KUTZ, Jose N. Data-driven science and engineering: machine learning, dynamical systems, and control. Cambridge University Press, 2019.

FLAH, Majdi; et al. Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review. *Archives of Computational Methods in Engineering*, v. 27, n. 4, p. 2621-2643, 2020. DOI: 10.1007/s11831-020-09471-9.

GÉRON, Aurélien; Hands-on machine learning with Scikit-Learn, Keras TensorFlow. O'Reilly Media, Inc., 2 ed., Sebastopol, United States: O'Reilly Media, Inc., 2019.

GHAREHBAGHI, Vahid Reza; et al. A critical review on structural health monitoring: definitions, methods, and perspectives. *Archives of Computational Methods in Engineering*, v. 29, jun. 2022. DOI: <https://doi.org/10.1007/s11831-021-09665-9>. Disponível em: <https://link.springer.com/article/10.1007/s11831-021-09665-9>. Acesso em: 24 jun. 2022.

LEI, Yaguo; et al. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, v. 138, abr. 2020. DOI: <https://doi.org/10.1016/j.ymssp.2019.106587>. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0888327019308088?via%3Dihub>. Acesso em: 24 jun. 2022.

OU, Yaowen; et al. Vibration-based monitoring of a small-scale wind turbine blade under varying climate conditions. Part I: An experimental benchmark. *Structural Control and Health Monitoring*, v. 28, n. 6, jun. 2021. DOI: <https://doi.org/10.1002/stc.2660>. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1002/stc.2660>. Acesso em: 24 jun. 2022.

SANTELO, Thiago Naufal; et al. Wind Turbine Failures Review and Trends. *Journal of Control, Automation and Electrical Systems*, v. 33, abr. 2022. DOI: <https://doi.org/10.1016/j.apenergy.2022.118882>. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0306261922003105?via%3Dihub>. Acesso em: 24 jun. 2022.

SUN, Shilin; et al. Condition monitoring of wind turbine blades based on self-supervised health representation learning: A conducive technique to effective and reliable utilization of wind energy. *Applied Energy*, v.

313, maio 2022. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0306261922003105?via%3Dihub>. Acesso em: 24 jun. 2022.

ZHENG, Alice; CASARI, Amanda. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media, Inc., 1 ed., Sebastopol, United States: O'Reilly Media, Inc., 2018.