CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

IDENTIFICAÇÃO DE FALHAS EM TURBINAS EÓLICAS: UMA ABORDAGEM DE APRENDIZADO DE MÁQUINA CENTRADA EM DADOS

Danielle Rodrigues Pinna

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:
Diego Nunes Brandão
Rodrigo Franco Toso

Rio de Janeiro,

Novembro de 2024

Identificação de Falhas em Turbinas Eólicas: Uma Abordagem de Aprendizado de Máquina Centrada em Dados

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

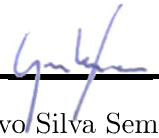
Danielle Rodrigues Pinna

Aprovada por:

Presidente, Prof. Diego Nunes Brandão, D.Sc. (orientador)

Rodrigo Franco Toso, PhD. (coorientador) - Microsoft

Rafaelli de Carvalho Coutinho, D.Sc. - CEFET/RJ



Gustavo Silva Semaan, D.Sc. - UFF

Ângela Ferreira, PhD. - IPB/Portugal

Rio de Janeiro,

Novembro de 2024

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

P656 Pinna, Danielle Rodrigues
Identificação de falhas em turbinas eólicas: uma abordagem de aprendizado de máquina centrada em dados / Danielle Rodrigues Pinna. — 2024.
61f. + apêndice : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2024.
Bibliografia : f. 57-61
Orientador: Diego Nunes Brandão
Coorientador: Rodrigo Franco Toso

1. Turbinas. 2. Localização de falhas (Engenharia). 3. Aprendizado do computador. I. Brandão, Diego Nunes. (Orient.). II. Toso, Rodrigo Franco. III. Título.

CDD 621.24

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

DEDICATÓRIA

Life is not easy for any of us. But what of that?
We must have perseverance and above all confidence in ourselves. We must believe that we are gifted for something, and that this thing, at whatever cost, must be attained". (Marie Curie)

AGRADECIMENTOS

Inicialmente, gostaria de expressar minha profunda gratidão à minha família, que sempre me apoiou e incentivou a continuar estudando, sem desistir.

Agradeço ao meu orientador Diego Brandão e co-orientador Rodrigo Toso pelo tempo de dedicação e conhecimento compartilhado, que foram essenciais para a realização deste trabalho

Por fim, agradeço à instituição CEFET/RJ e a todos os professores pelo aprendizado, suporte e recursos oferecidos, sem os quais a realização deste trabalho não seria possível.

RESUMO

Identificação de Falhas em Turbinas Eólicas: Uma Abordagem de Aprendizado de Máquina Centrada em Dados

Danielle Rodrigues Pinna

Orientadores:
Diego Nunes Brandão
Rodrigo Franco Toso

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

Os últimos anos têm sido marcados pela transição da matriz energética mundial, predominantemente com as fontes eólica e solar, que são consideradas energias limpas. As turbinas eólicas, responsáveis pelo processo de conversão energética, constituem-se por equipamentos complexos e de alto custo, suscetíveis a diversas falhas devido a múltiplos fatores operacionais e ambientais. O monitoramento contínuo dos componentes das turbinas é essencial para a detecção precoce de falhas, o que pode reduzir significativamente os custos de manutenção e aumentar a eficiência operacional. Este trabalho foca na aplicação e comparação de técnicas de aprendizado de máquina para a detecção de falhas em turbinas eólicas, utilizando uma abordagem centrada em dados. A pesquisa enfatiza a importância do pré-processamento dos dados, destacando técnicas de balanceamento de classes, particionamento de dados e seleção de atributos. Além disso, são comparados diferentes algoritmos de aprendizado de máquina, com foco na otimização de hiperparâmetros. Os resultados demonstram que um pré-processamento adequado dos dados é crucial para o desempenho dos modelos de aprendizado de máquina. Também é evidenciada a importância do tempo computacional na otimização dos hiperparâmetros e na seleção do algoritmo mais apropriado para o contexto específico do problema.

Palavras-chave:
Turbinas Eólicas, Aprendizado de Máquina, Classificação de Falhas

Rio de Janeiro,

Novembro de 2024

ABSTRACT

Identificação de Falhas em Turbinas Eólicas: Uma Abordagem de Aprendizado de Máquina Centrada em Dados

Danielle Rodrigues Pinna

Advisors:
Diego Nunes Brandão
Rodrigo Franco Toso

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

The last few years have been marked by the transition of the world energy matrix, predominantly with wind and solar sources, which are considered clean energies. Wind turbines, responsible for the energy conversion process, are complex and expensive equipment, susceptible to various failures due to multiple operational and environmental factors. Continuous monitoring of turbine components is essential for early fault detection, which can significantly reduce maintenance costs and increase operational efficiency. This work uses a data-centric approach to apply and compare machine learning techniques for fault detection in wind turbines. The research emphasizes the importance of data preprocessing, highlighting techniques such as class balancing, data partitioning, and attribute selection. Additionally, different machine learning algorithms are compared, focusing on hyperparameter optimization. The results demonstrate that adequate data preprocessing is crucial for the performance of machine learning models. The importance of computational time in optimizing hyperparameters and selecting the most appropriate algorithm for the specific problem context is also highlighted.

Key-words:
Wind Turbine, Machine Learning, Fault Classification

Rio de Janeiro,

Novembro de 2024

Lista de Figuras

I.1 Matriz Elétrica Brasileira. [ABEEólica, 2021] 16

I.2 *Ranking* Capacidade Total Instalada *Onshore* [ABEEólica, 2021] 17

III.1 Estrutura hierárquica que mostra a distinção entre regressão, classificação e agrupamento com base na divisão de domínio de dados. Adaptado de [Suthaharan, 2016]. . 24

III.2 Visualização de algumas divisões de domínio dos dados utilizando a classificação de algoritmos do aprendizado supervisionado. Adaptado de [Suthaharan, 2016]. 25

III.3 Ilustração dos processos de classificação de dados. Adaptado de [Suthaharan, 2016]. . 26

III.4 Ilustração *holdout*. Adaptado de [Raschka, 2015]. 28

III.5 Ilustração *k-fold*. Adaptado de [Raschka, 2015]. 29

III.6 Gráfico da função logística. Adaptado de [Raschka, 2015]. 31

III.7 Ilustração KNN. Adaptado de [Raschka, 2015]. 32

III.8 Ilustração de uma Árvore de Decisão. 33

III.9 Ilustração da Máquina de Vetores de Suporte (SVM). 34

III.10 Ilustração *k-fold* com ajuste de hiperparâmetros. Adaptado de [Raschka, 2015]. . . . 36

III.11 Representação do *Halving Random Search* sucessivo [Soper, 2023]. 37

III.12 Problema de ajuste de hiperparâmetros em um espaço de busca 2D. 38

III.13 Matriz de confusão para um problema com duas classes. Fonte: Elaborada pelo autor 38

IV.1 Investimentos do setor de energia nos últimos 5 anos. Adaptado de [GWEC, 2024] . 41

IV.2 Tendência do tamanho das turbinas *onshore* e *offshore*, 1980-2030. Adaptado de [GWEC, 2024] 42

IV.3 Componentes de uma turbina eólica. 43

IV.4 Frequência de Falhas nos Componentes da Turbina Eólica. 46

IV.5 Frequência de Falhas por Turbina Eólica. 46

IV.6 Falha 47

IV.7 Percentual de falhas em cada componente da turbina eólica 48

IV.8 Curva de Potência 49

IV.9 Falha Rolamento Gerador T07 49

IV.10 Falha Rolamento Gerador T07 50

IV.11Falha Gerador T06 50
IV.12Falha Transformador T07 51
IV.13Falha Transformador T07 51
IV.14Análise temporal das variáveis meteorológicas 52
IV.15Rank das Correlações Cruzada do *Metmast* 53

V.1 *Pipeline* da metodologia de aprendizado de máquina adotada. 54
V.2 *Scatter Plot* das 2 primeiras Componentes Principais. 56
V.3 *Pipeline* da metodologia adotada no Fluxo de Trabalho 2. 58
V.4 *Pipeline* da metodologia adotada no Fluxo de Trabalho 3. 61
V.5 *Pipeline* da metodologia adotada no Fluxo de Trabalho 4. 64
V.6 *Pipeline* da metodologia adotada no Fluxo de Trabalho 5. 65

Lista de Tabelas

II.1 Quantidade de artigos por ano de publicação da pesquisa sistemática na base *Scopus*. 19

IV.1 Descrição dos conjuntos de dados. 45
IV.2 Amostra das primeiras observações do dataset *Metmast*. 45
IV.3 Falhas das Turbinas por Componente 47

V.1 Autovalores e Variância das 10 primeiras componentes. 55
V.2 Autovetores das 6 primeiras componentes selecionadas. 57
V.3 Métricas dos modelos das componentes da Turbina na base de teste. 59
V.4 Comparação dos resultados com o *benchmark*. 60
V.5 Resultado da Árvore de Decisão em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica. 62
V.6 Resultado do KNN em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica. 62
V.7 Resultado da Regressão Logística em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica. 62
V.8 Resultado do *Naive Bayes* em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica. 62
V.9 Resultado da Floresta Aleatória em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica. 63
V.10 Resultado do SVM em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica. 63
V.11 Resultado para cada otimizador e seleção de atributos na base de teste. 64
V.12 Comparação da métrica F_1-*Score* dos resultados para o *benchmark*. 65
V.13 Resultado para cada técnica de balanceamento de classes na base de teste. 66
V.14 Métricas do OCSVM sem otimização de hiperparâmetros. 67
V.15 Métricas do OCSVM com otimização de hiperparâmetros. 67
V.16 Comparação classificador binário e OCSVM 68

VII.1 Descrição das variáveis do conjunto de dados *Metmast* 77

VII.2Descrição das variáveis do conjunto de dados *Signals* 78

Lista de Acrônimos

AD	Árvore de Decisão.
AM	Aprendizado de Máquina.
AUC	Área sob a curva.
CNN	Redes Neurais Convolucionais.
EDP	Energias de Portugal.
FA	Floresta Aleatória.
GWEC	Conselho Global de Energia Eólica.
HRS	*Halving Random Search*.
IA	Inteligência Artificial.
KNN	K-Vizinhos mais Próximos.
LSTM	Algoritmo de Memória de Curto Prazo.
MCC	Coeficiente de Correlação de Matthews.
MI	Informação Mútua.
NB	Naive Bayes.
OCSVM	Máquina de Vetor de Suporte de Classe Única.
PCA	Análise de Componentes Principais.
RL	Regressão Logística.
RNA	Redes Neurais Artificiais.
RS	*Random Search*.
RUL	Vida Útil Remanescente.

SCADA	Sistema de Controle e Aquisição de Dados.
SMOTE	Método Sintético de Sobreamostragem Minoritária.
SVM	Máquinas de Vetores de Suporte.

Sumário

I Introdução **16**

II Trabalhos Relacionados **19**

III Referencial Teórico **22**

III.1 Aprendizado de Máquina 22

III.2 Pré-processamento dos dados 26

III.2.1 Dados Desbalanceados 27

III.2.2 Partição dos dados 27

III.2.3 Seleção e Extração de Atributos 29

III.3 Métodos de Classificação 30

III.3.1 Regressão Logística 30

III.3.2 *Naive Bayes* 31

III.3.3 K-Vizinhos mais Próximos 31

III.3.4 Árvores de Decisão 32

III.3.5 Floresta Aleatória 33

III.3.6 Máquina de Vetores de Suporte 34

III.4 Otimização de Hiperparâmetros 35

III.4.1 *Grid Search* 36

III.4.2 *Random Search* 36

III.4.3 *Halving Random Search* 37

III.5 Avaliação dos Modelos Preditivos 38

IV Conjunto de Dados **41**

IV.1 Energia Eólica 41

IV.2 Turbinas Eólicas 42

IV.3 Base de Dados 44

V Resultados **54**

V.1 Fluxo de Trabalho 1 - Análise das Variáveis do Mastro Meteorológico 54

V.2 Fluxo de Trabalho 2 - Comparação de Algoritmos de Aprendizado de Máquina 58
V.3 Fluxo de Trabalho 3 - Comparação dos Métodos de Otimização de Hiperparâmetros 60
V.4 Fluxo de Trabalho 4 - Comparação das Técnicas de Seleção e Extração de Atributos 63
V.5 Fluxo de Trabalho 5 - Tratamento de Dados Desbalanceados 65
V.6 Fluxo de Trabalho 6 - Classificador de uma única classe OCSVM 66
V.7 Sumário 67

VI Considerações Finais 69
VI.1 Artigos Publicados 71

Referências 72

VII Apêndice A 77

Capítulo I Introdução

A energia eólica é um importante recurso de energia limpa e renovável disponível na natureza que possui benefícios de interesse global e econômico. Uma das metas que o Brasil se comprometeu no Acordo de Paris, realizado em 2015, foi no aumento do uso de fontes alternativas de energia para redução das emissões de gases de efeito estufa.

De acordo com os dados do Boletim Anual da Associação Brasileira de Energia Eólica [ABEEólica, 2021], a indústria de energia eólica no Brasil fechou o ano de 2022 com $25,6$GW de capacidade instalada, representando um crescimento de $18,85\%$ em relação ao ano anterior. Os 904 Parques Eólicos existentes no país estão distribuídos por 12 estados, com maior concentração na região Nordeste. A nova capacidade eólica instalada em 2022 fez a fonte eólica atingir uma participação de 13,4% da matriz elétrica brasileira, ficando apenas atrás da energia hidrelétrica com 54,1% de participação, conforme ilustrado na Figura I.1.

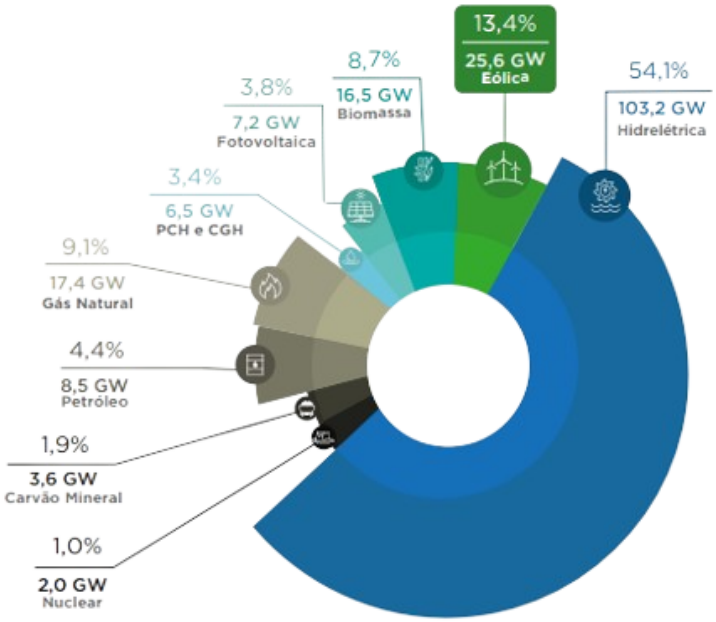


Figura I.1: Matriz Elétrica Brasileira. [ABEEólica, 2021]

Há ainda os benefícios ambientais, de acordo com ABEEólica [2021], em 2022 a fonte eólica evitou o total de 26,88 milhões de toneladas de emissões de dióxido de carbono (CO_2), o equivalente à emissão anual de cerca de 22 milhões de automóveis de passeio. Para efeito de comparação, a

cidade de São Paulo possui uma frota de cerca de 10 milhões de automóveis. Além disso, de 2016 a 2024 o setor eólico brasileiro terá evitado emissões de gases de efeito estufa valoradas entre R$ 60 e 70 bilhões.

Segundo o Conselho Global de Energia Eólica (GWEC), o Brasil vem sendo classificado como um potência eólica e já ocupa o sexto lugar no *Ranking* global da capacidade eólica total instalada *onshore* (terrestre) em 2022, conforme ilustrado na Figura I.2. Além disso, o país ocupa a terceira posição de nova capacidade eólica instalada. Um fator que explica o eficiente desenvolvimento da energia eólica é o seu grande potencial. Estima-se que o Brasil tenha, em terra, um potencial de mais de 700GW, podendo alcançar a liderança global de produção.

POSIÇÃO	PAÍS	Capacidade total instalada onshore (GW)
1	China	334,0
2	EUA	144,2
3	Alemanha	59,0
4	Índia	41,9
5	Espanha	29,8
6	**Brasil**	**25,6**
7	França	20,7
8	Canadá	15,3
9	Reino Unido	14,6
10	Suécia	14,4

Figura I.2: *Ranking* Capacidade Total Instalada *Onshore* [ABEEólica, 2021]

Além do benefício ambiental, a criação de parques eólicos no Brasil contribui positivamente em benefícios sociais e econômicos, como a geração de empregos diretos e indiretos, geração de renda com os arrendamentos de terras de pequenos proprietários, e também no impacto de aumento de arrecadação de impostos que, com adequado gerenciamento público, podem significar melhorias para a população e o município [ABEEólica, 2021].

Um importante decreto do Governo Brasileiro publicado no final de Janeiro de 2022[1] dispõe sobre o aproveitamento dos recursos naturais no mar para a geração de energia elétrica a partir de parques eólicos *offshore* (alto mar). Este é um avanço crucial na transição da matriz energética do Brasil para fontes cada vez mais limpas e sustentáveis.

Com o crescente desenvolvimento deste setor surgem também os desafios relacionados à redução dos custos de operação e manutenção (O&M) das turbinas eólicas, que são sistemas sofisticados, complexos e caros. De acordo com Blanco-M et al. [2017], a O&M das turbinas são responsáveis por cerca de 25% a 35% dos custos de geração.

[1] https://www.in.gov.br/en/web/dou/-/decreto-n-10.946-de-25-de-janeiro-de-2022-376016988

Os problemas relacionados à manutenção da turbina eólica normalmente são as falhas de componentes do sistema elétrico e as provenientes das condições climáticas extremas. Algumas falhas dos componentes não ocorrem com tanta frequência, mas uma única perturbação pode ocasionar horas de perda da produtividade ou até mesmo o desligamento da turbina. Por essa razão, a maneira mais eficaz de reduzir os custos de manutenção é monitorar o *status* dos geradores e prever o seu mau funcionamento antes que o sistema falhe [Qin et al., 2017]. Assim, o diagnóstico precoce das falhas é um fator chave para reduzir significativamente os custos de manutenção.

Atualmente, os aerogeradores modernos possuem um sistema de coleta e armazenamento de dados, conhecido como Sistema de Controle e Aquisição de Dados (SCADA) [Marti-Puig et al., 2018]. Como o nome sugere, este sistema de monitoramento é alimentado por vários sensores que fornecem medições em intervalos de 10 minutos de variáveis relacionadas às turbinas eólicas.

Este sistema registra diversas variáveis, como por exemplo, dados do sistema hidráulico, do rotor, bem como variáveis meteorológicas, como, temperatura, pressão atmosférica e velocidade do vento, resultando em uma quantidade significativa de dados. Contudo, extrair respostas que permitam a prevenção de falhas de uma grande quantidade de dados para identificação precoce de falhas não é uma tarefa simples e exige a aplicação de métodos cada vez mais sofisticados [Corley et al., 2021].

Segundo Pandit and Wang [2024], o monitoramento de condições baseado em dados SCADA possui um potencial significativo para melhorar as operações e manutenção de turbinas eólicas. No entanto, é necessário o pré-processamento dos dados SCADA para a garantia da integridade dos dados. Os modelos de Aprendizado de Máquina (AM) baseados em dados estão entre os métodos existentes para detecção precoce de falhas para turbinas eólicas.

A maioria dos trabalhos sobre detecção de falhas em turbinas eólicas são fundamentados em conjuntos de dados operacionais e de eventos, como os fornecidos pelo SCADA [Stetco et al., 2019]. Neste trabalho, o processamento da detecção de falhas divide-se em duas etapas: no aprendizado de máquina centrado em dados e na abordagem centrada no modelo. A primeira se concentra na otimização do pré-processamento de dados SCADA para melhorar a qualidade dos dados, incluindo a técnica de seleção de atributos, e a segunda, na otimização das arquiteturas do modelo e seus parâmetros.

O objetivo deste trabalho é a aplicação de *pipelines* de aprendizado de máquina, considerando de forma iterativa a etapa de pré-processamento para avaliação de cada estratégia na detecção de falhas que afetam os componentes das turbinas.

O presente trabalho está organizado em mais cinco capítulos. No Capítulo 2 é apresentado os trabalhos relacionados. O Capítulo 3 apresenta o referencial teórico, o Capítulo 4 apresenta o conjunto de dados, os resultados são discutidos no Capítulo 5 e, por fim, o Capítulo 6 apresenta as considerações finais.

Capítulo II Trabalhos Relacionados

Este capítulo apresenta os trabalhos encontrados na literatura a partir de um mapeamento sistemático [Grant and Booth, 2009] sobre detecção de falhas em turbinas eólicas, com o objetivo de identificar e avaliar as pesquisas disponíveis para esse tema.

A pesquisa foi realizada em fevereiro de 2024 na base *Scopus*, utilizando a *string* de busca: TITLE-ABS-KEY ("*wind turbine*" and "scada" and "*fault detection*"), onde foram encontrados 284 artigos. Visando limitar apenas os artigos que continham o objetivo principal de classificação das falhas sob a abordagem de aprendizado de máquina, foi adicionada a *string* "*machine learning*" no termo de busca, resultando em 56 artigos base. Desses 56, mais da metade foram publicados entre os anos de 2021 e 2023, conforme mostrado na Tabela II.1, ressaltando a sua recente importância.

Tabela II.1: Quantidade de artigos por ano de publicação da pesquisa sistemática na base *Scopus*.

Ano da publicação	**Documentos**
2016	1
2017	3
2018	3
2019	4
2020	5
2021	18
2022	8
2023	12
2024	2

Alguns trabalhos exploraram abordagens para criação de rótulo de falhas, como em Hu et al. [2017] que criaram os rótulos de classe para os dados operacionais com falhas usando os dados de status e aviso do sistema SCADA. Também aplicaram uma etapa para converter os dados operacionais em representações de séries temporais contínuas de uma hora usando recursos defasados, eliminando assim a correlação temporal da série.

Helbing and Ritter [2018] fizeram uma revisão bibliográfica desde 2009 com as aplicações não supervisionadas e supervisionadas de Redes Neurais Artificiais (RNAs) para o monitoramento de condições em turbinas eólicas. Constataram que as abordagens não supervisionadas são predominantes na literatura de RNAs, devido a dificuldade da obtenção de conjuntos de dados rotulados para treinamento supervisionado.

Antes de aplicar os algoritmos de prognóstico para turbinas eólicas, Marti-Puig et al. [2018]

ressaltaram a importância de implementar uma etapa de pré-processamento, que muitas das vezes é subestimada por não considerar seu grande impacto nos resultados finais. Nesse estudo, os autores avaliaram o impacto da remoção de valores extremos (*outliers*) e constataram que a remoção de *outliers* não é uma boa prática, pois esses valores são o modo de operação da turbina menos frequente (estados de falha).

Outro artigo que faz uma extensa revisão da literatura sobre os modelos de aprendizado de máquina aplicados ao monitoramento e detecção de falhas de turbinas eólicas é evidenciado em [Stetco et al., 2019]. As etapas analisadas dos modelos de AM nesse artigo foram: fonte de dados, seleção e extração de recursos, seleção e validação do modelo e a tomada de decisão. Os resultados mostraram que os algoritmos de Redes Neurais, Máquinas de Vetores de Suporte (SVM, do inglês *Support Vector Machine*) e Árvores de Decisão são os mais utilizados.

Mammadov et al. [2021] compararam os métodos de aprendizado de máquina SVM, RNAs e AdaBoost para determinar o melhor modelo de classificação binária para prevenção de falhas com base em dados históricos de uma fazenda eólica localizada no Canadá. Os resultados demonstraram a capacidade do método proposto, que abrange desde a coleta e pré-processamento de dados até a modelagem e validação dos modelos, em prever falhas de geradores eólicos com uma precisão de até 83%, com destaque para o algoritmo AdaBoost, que foi o mais rápido para executar e o mais fácil de ajustar.

O trabalho de Yi et al. [2021] foca na classificação de dados desbalanceados, uma questão comum na detecção de falhas em turbinas eólicas. Eles utilizaram os classificadores Bayesiano, SVM e a Árvore de Decisão para comparação dos resultados sem e com sobreamostragem, demonstrando uma melhoria significativa na capacidade de detecção de falhas com dados balanceados.

Velandia-Cardenas et al. [2021] utilizaram técnicas de processamento de dados, sobreamostragem aleatória e divisão de tempo para melhorar o desempenho dos algoritmos de aprendizado de máquina KNN e SVM. A abordagem destaca a importância de lidar adequadamente com o desequilíbrio de dados para evitar falsos positivos. Para tanto, propõem uma metodologia de detecção de falhas utilizando os classificadores de aprendizado de máquina com pesquisa de grade aleatória e validação cruzada para treinamento do modelo com ajuste de hiperparâmetro em dados reais SCADA.

Bilendo et al. [2021] propuseram um método não supervisionado de AM para obter os rótulos dos dados normais e anormais (candidato a falha) de um conjunto de sinal de dados SCADA. Além disso, aplicaram os algoritmos *k-means* e a análise discriminante linear com base na função densidade de probabilidade. Posteriormente, Zhang et al. [2022] apresentaram um método de previsão de pseudo-rótulo que combina a votação majoritária de subdomínio e iterações gerais (SMV-I) para rotular os dados.

O estudo de Waqas Khan and Byun [2022] apresentou um novo método de detecção de falhas

em turbinas eólicas baseado em um classificador de empilhamento, do inglês *Stacking Ensemble*. Utilizaram três modelos de classificação, AdaBoost, K-vizinhos mais próximos e regressão logística. A saída dessas três classificações foi combinada e usada como entrada para o meta-modelo do classificador AdaBoost. Este método combinado se mostrou mais robusto em comparação com a abordagem tradicional dos modelos de aprendizado de máquina, demonstrando a eficácia de combinar múltiplas técnicas para uma detecção de falhas mais precisa em turbinas eólicas.

Trabalhos recentes [Feng et al., 2019; Ayman et al., 2022] empregam o algoritmo de memória de curto prazo (LSTM, do inglês *Long Short Term Memory*) para o monitoramento de turbinas eólicas, uma vez que, em termos de dados de séries temporais, o LSTM pode construir a correlação entre as informações previamente conhecidas e o ambiente atual. Entretanto, Trizoglou et al. [2021] compararam o LSTM com o XGBoost para projetar um modelo de comportamento normal do gerador para fins de detecção de falhas, no qual observaram que o XGBoost superou o LSTM em precisão preditiva e eficiência computacional.

Uma outra abordagem, conforme Rahimilarki et al. [2022], consiste na aplicação de Redes Neurais Convolucionais (CNN) para detecção e classificação de falhas em aerogeradores, neste caso os dados SCADA são convertidos para imagens 2-D em escala de cinza. Uma das vantagens das CNNs é sua capacidade de extrair automaticamente características significativas de dados visuais, capturando tendências que métodos tradicionais de análise de falhas podem não conseguir detectar.

Garan et al. [2022] discutiram que a maioria dos artigos da literatura é baseada em modelos e não em pré-processamento de dados, enfatizando que mais esforço deveria ser colocado na qualidade do conjunto de dados a fim de melhorar o desempenho das medidas de classificação. Destacaram que a etapa da preparação dos dados é tão crítica quanto a escolha do algoritmo de aprendizado de máquina, de modo que propõem uma metodologia datacêntrica, no qual as etapas orientada a dados são executadas de forma iterativas na detecção de falhas. O estudo explora várias técnicas de AM, incluindo árvores de decisão, florestas aleatórias e redes neurais, avaliando sua eficácia na identificação de falhas iminentes que afetam cinco componentes diferentes das turbinas eólicas.

Esses trabalhos ilustram o crescente interesse e a aplicação de técnicas de aprendizado de máquina na detecção de falhas em turbinas eólicas. A combinação de análise de dados SCADA e aprendizado de máquina abre espaço para o monitoramento eficaz de turbinas eólicas, visando otimizar a produção de energia eólica e garantir a operação segura e eficiente de parques eólicos.

Capítulo III Referencial Teórico

O objetivo deste capítulo é abordar os conceitos que fundamentam esta pesquisa. Para tanto, ele está dividido em cinco seções. A Seção III.1 discute sobre a técnica do aprendizado de máquina. A Seção III.2 aborda as técnicas de preparação dos dados. A Seção III.3 apresenta os algoritmos de aprendizado de máquina usados neste trabalho. A Seção III.4 detalha os métodos de ajustes de hiperparâmetros empregados. Finalmente, a Seção III.5 descreve as métricas de desempenho utilizadas para avaliar os modelos.

III.1 Aprendizado de Máquina

Nos últimos anos, com o crescente volume de dados gerados e com uma maior complexidade dos problemas a serem tratados computacionalmente, tornou-se necessário o uso de ferramentas computacionais mais sofisticadas e autônomas. Um exemplo são os algoritmos de aprendizado de máquina que são capazes de modelar os dados e resolver problemas complexos. Além disso, é considerada uma das técnicas mais empregadas no processo de mineração de dados dos últimos anos [Faceli et al., 2011].

Mitchell [1997] define o aprendizado de máquina como a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência. Para Marsland [2015] o aprendizado de máquina consiste em fazer com que os computadores modifiquem ou adaptem suas ações para que estas se tornem mais precisas.

De acordo com Sambasivan et al. [2021], como o AM depende em grande parte de seus dados, ter dados de alta qualidade tem um papel decisivo na construção de modelos confiáveis e robustos, em oposição a apenas um bom algoritmo de treinamento. No entanto, os dados costumam ser o aspecto menos incentivado em relação ao trabalho de ajustar novos modelos e algoritmos. Para garantir que o sistema de AM possa representar os dados e prever com precisão o fenômeno que pretende medir, é crucial ter uma boa qualidade dos dados que pode levar a melhorias significativas dos modelos.

Os algoritmos de aprendizado de máquina possuem diversas subdivisões baseadas no tipo de problema que precisa ser resolvido. Existem quatro categorias principais: aprendizado supervisionado, aprendizado não supervisionado, aprendizado semi-supervisionado e aprendizado por reforço

[Russell and Norvig, 2010].

O aprendizado de máquina supervisionado é uma das técnicas de aprendizado comum que exploram dados rotulados, cujo objetivo é aprender um modelo preditivo a partir de um conjunto de dados, composto por uma variável alvo e um conjunto de variáveis explicativas. Esse modelo deve ser capaz de generalizar o conhecimento adquirido para dados desconhecidos a fim de se ter uma boa capacidade preditiva.

Uma aplicação importante e amplamente utilizada são os problemas de regressão e classificação em que o objetivo é fazer previsão. Por outro lado, para algoritmos de aprendizado de máquina não supervisionados, os dados não possuem rótulos predefinidos. O algoritmo aprende a partir de exemplos simples sem resposta associada, deixando para o algoritmo determinar padrões nos dados de forma independente [Bishop and Nasrabadi, 2006]. Entre as tarefas mais comuns de aprendizagem não supervisionada, destacam-se a redução de dimensionalidade, o agrupamento (clusterização) e a detecção de anomalias.

Algoritmos de aprendizado de máquina semi-supervisionados utilizam dados rotulados e não rotulados para treinamento de modelo, geralmente combinando métodos supervisionados e não supervisionados. Outra subdivisão são os algoritmos de aprendizado de máquina por reforço, nos quais os algoritmos aprendem com o ambiente. Se tiverem um bom desempenho, recebem uma recompensa, e o objetivo é maximizar essa recompensa [Bishop and Nasrabadi, 2006].

Uma representação hierárquica das abordagens de aprendizado de máquina, supervisionado e não supervisionado, é mostrada na Figura III.1, com base na divisão: dados $\rightarrow$ domínio $\rightarrow$ divisão. De cima para baixo na hierarquia, o fluxograma mostra primeiro duas categorias de dados: se o conjunto de dados não for rotulado, então é um problema de aprendizado não supervisionado, e se os dados estiverem rotulados, então as características do domínio de dados devem ser compreendidas.

Nesse caso, se o domínio de dados não pode ou não deve ser dividido, então a abordagem a ser usada é de regressão, e se puder ser dividido, então é um problema de classificação e a divisão do domínio deve ser analisada. Se os pontos de dados associados às classes forem separáveis, então o domínio de dados original poderá ser dividido e a classificação poderá ser aplicada. Entretanto, se as classes forem inseparáveis, então o domínio original deve ser transformado, chamado de espaço de características [Suthaharan, 2016].

Figura III.1: Estrutura hierárquica que mostra a distinção entre regressão, classificação e agrupamento com base na divisão de domínio de dados. Adaptado de [Suthaharan, 2016].

Neste estudo, o interesse é no problema de aprendizado de máquina supervisionado, mais especificamente no problema de classificação binária para reconhecer as falhas e as operações sem falha de uma turbina eólica, considerando os dados SCADA. O algoritmo de classificação binária lida somente com duas classes, 0 ou 1. A classe 0 indica uma observação sem falha (saudável) e a classe 1 indica observações com falha (defeituosa). O propósito do algoritmo é desenvolver um modelo capaz de determinar a classe de exemplos que não estão rotulados.

A ilustração apresentada na Figura III.2 considera um exemplo de classificação supervisionada de duas classes, vermelho e azul. Nota-se que a separação dessas classes pode ser feita de quatro formas diferentes, a primeira (a) traçando uma linha reta, na segunda opção (b) é feito alguns cortes verticais e horizontais, na terceira opção (c) tem vários cortes verticais e horizontais, e na última opção (d) é feito um corte suave. Cada uma dessas opções representa um tipo de algoritmo aprendizado de máquina que será discutido neste trabalho.

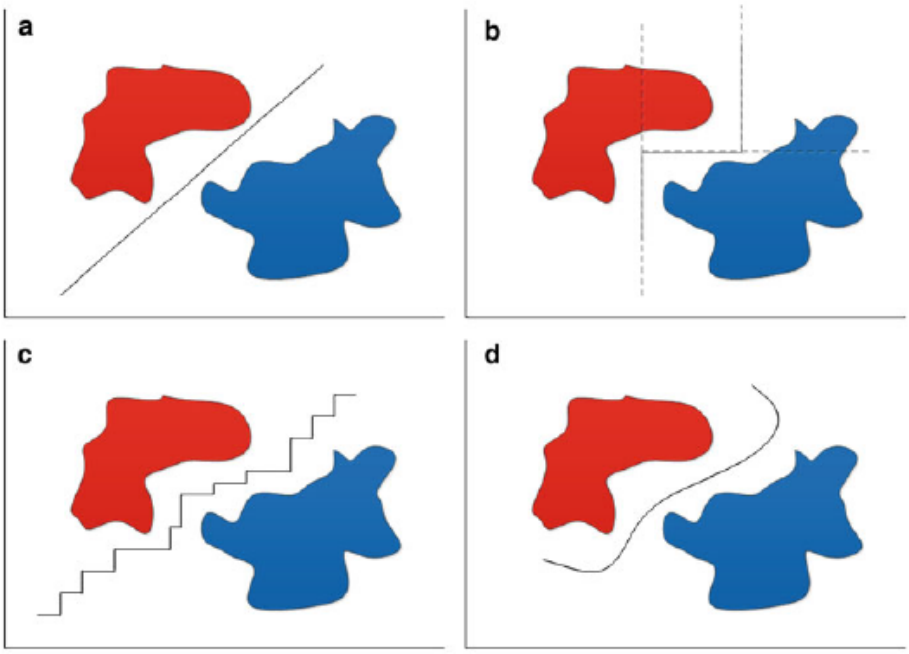


Figura III.2: Visualização de algumas divisões de domínio dos dados utilizando a classificação de algoritmos do aprendizado supervisionado. Adaptado de [Suthaharan, 2016].

Apesar da essência principal do AM consistir na construção de algoritmos supervisionados ou não supervisionados, está não é a única etapa que envolve o processo. É de suma importância entender o problema a ser resolvido e definir os objetivos. Nos últimos anos, o desenvolvimento de metodologias baseadas em Inteligência Artificial (IA) cresceu substancialmente em diversos domínios. Modelos de aprendizado de máquina e aprendizado profundo são frequentemente complexos e carecem de explicações claras sobre seu processo de tomada de decisão, sendo chamados de "caixas pretas". Segundo Hassija et al. [2024], isso dificulta sua adoção em áreas críticas como bancos, saúde e segurança. A opacidade desses modelos, devido à complexidade das redes neurais profundas, compromete a transparência e exige que os algoritmos de IA sejam explicáveis para decisões cruciais.

O fluxograma apresentado na Figura III.3 descreve um processo genérico de classificação de AM, abrangendo desde a coleta dos dados de entrada, a compreensão dos dados, a tecnologia até as técnicas de modelagem dos dados. A etapa inicial de coleta e análise dos dados é fundamental para a construção do modelo, pois é a partir dela que são obtidas as informações essenciais para desenvolver um modelo eficaz e confiável.

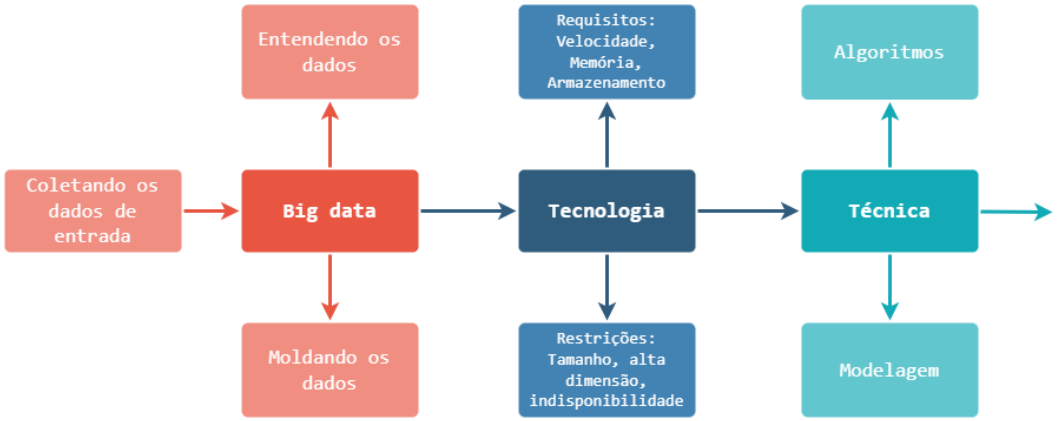


Figura III.3: Ilustração dos processos de classificação de dados. Adaptado de [Suthaharan, 2016].

Desta forma, é recomendado que as etapas de coleta e preparação dos dados, seleção de variáveis, escolha do algoritmo, determinação dos parâmetros, treinamento e avaliação do desempenho do modelo construído, sejam seguidas a fim de encontrar um modelo de melhor desempenho preditivo.

III.2 Pré-processamento dos dados

A qualidade dos dados e a quantidade de informações úteis que eles contêm são fatores chave para determinar o quão bem um algoritmo de aprendizado de máquina pode aprender. A aplicação de técnicas de preparação dos dados é importante para melhorar a qualidade dos dados e para ajudar os algoritmos de aprendizado de máquina a construir modelos mais fiéis à distribuição real dos dados [Faceli et al., 2011].

O pré-processamento dos dados é uma das etapas iniciais mais importantes do *pipeline* de aprendizado de máquina, cujo foco é dividir uma tarefa completa de AM em um fluxo de trabalho composto por várias etapas. Entre as tarefas frequentemente realizadas nessa fase estão a limpeza e imputação de valores ausentes, a amostragem, o balanceamento das classes do conjunto de dados, a transformação dos dados e a seleção de atributos relevantes para a construção do modelo.

Antes de iniciar o pré-processamento, é fundamental conduzir uma análise descritiva dos dados, a fim de compreender suas medidas estatísticas, como médias, variâncias, covariâncias e correlações, bem como identificar possíveis padrões ocultos nos dados. Essas análises ajudam a caracterizar o conjunto de dados com base em seu tamanho, número de padrões distintos, dispersão dos padrões, entre outros aspectos relevantes [Suthaharan, 2016].

Uma medida simples mas muito útil é a contagem. A partir da medida de contagem, algumas perguntas iniciais podem ser respondidas, como: Quantas observações existem? Quantas características existem? E quantas classes existem? Essa medida é a principal para identificar os problemas de dados desequilibrados ou desbalanceados.

III.2.1 Dados Desbalanceados

Dados desbalanceados significam que as classes não são equilibradas, ou seja, não são igualmente informativas. Usando um exemplo de duas classes, quando o número de observações de uma classe é significativamente menor do que na outra classe, esta classe é chamada de uma classe minoritária e a outra classe de classe majoritária.

Segundo Ramentol et al. [2012], as técnicas para lidar com o desequilíbrio de classes dividem-se em duas categorias: abordagens no nível do algoritmo de aprendizado e no nível de dados. As abordagens no nível de dados são mais versáteis, pois podem ser aplicadas independentemente do classificador escolhido. Além disso, o pré-processamento dos conjuntos de dados pode ser feito uma única vez e os dados processados podem ser usados para treinar diferentes classificadores, otimizando o tempo de computação.

As soluções no nível de dados consiste em balancear a distribuição das classes por meio da sobreamostragem (*oversampling*) da classe minoritária, ou da subamostragem (*undersampling*) da classe majoritária, ou aplicando modelos híbridos que combinam as técnicas anteriores.

Os métodos de subamostragem criam um subconjunto do conjunto de dados original eliminando alguns dos exemplos da classe majoritária. Já os métodos de sobreamostragem criam um superconjunto do conjunto de dados original replicando alguns dos exemplos da classe minoritária ou criando novos a partir das instâncias da classe minoritária original. Existe também as abordagens híbridas que combinam subamostragem e sobreamostragem com a finalidade de melhorar a qualidade dos exemplos sintéticos gerados.

O método sintético de sobreamostragem minoritária SMOTE (do inglês, *Synthetic Minority Oversampling Technique*) é o mais considerado em problemas de detecção de falhas [Gad and Hassanien, 2021; Zhang and Li, 2021]. O SMOTE consiste em gerar dados sintéticos para a classe minoritária a partir dos casos já existentes. Ele faz isso interpolando entre exemplos existentes da classe minoritária, evitando o sobreajuste e expandindo as fronteiras de decisão da classe minoritária no espaço da classe majoritária.

III.2.2 Partição dos dados

Em aprendizado de máquina é de suma importância avaliar o desempenho de um modelo de aprendizado de máquina em dados que ele ainda não viu. Um modelo pode sofrer de *underfitting* (alto viés) se o modelo for muito simples, ou de *overfitting* (alta variância) se o modelo for muito complexo para o conjunto de dados de treinamento. Para encontrar um equilíbrio aceitável entre viés e variância, é preciso avaliar o modelo cuidadosamente. As técnicas denominadas de validação cruzada, "*holdout*" e "*k-fold*", podem ajudar a obter estimativas confiáveis do erro de generalização do modelo, ou seja, como o modelo se comporta em dados não vistos.

Segundo Raschka [2015], os principais métodos de amostragem podem ser descritos como:

- *Holdout*: O conjunto de dados é dividido em uma porcentagem p para treinamento do modelo e $(1-p)$ para teste, usado para estimar seu desempenho. A porcentagem p para treinamento ainda pode ser dividida para obtenção do conjunto de validação. O conjunto de validação é interessante para ajustar e comparar diferentes configurações de parâmetros para melhorar ainda mais o desempenho na previsão em dados não vistos.

  - *Treino*: Amostra destinada para treinar o modelo, nesta parte não é possível avaliar o desempenho do modelo dentro da própria amostra para o qual o mesmo foi elaborado, já que modelo não é capaz de generalizar em dados não vistos, podendo ocasionar um *overfitting*.

  - *Validação*: Amostra na qual é selecionado o modelo, ou seja, onde os valores ideais de parâmetros de ajuste (também chamados de hiperparâmetros) são selecionados.

  - *Teste*: Amostra na qual é avaliada o desempenho do modelo construído com a amostra de treino.

  A Figura III.4 ilustra o conceito de validação cruzada do tipo "*holdout*", onde é usado um conjunto de validação para avaliar repetidamente o desempenho do modelo após o treinamento usando diferentes valores de parâmetros. Assim que o ajuste dos valores dos parâmetros são satisfatórios, é estimado o erro de generalização dos modelos no conjunto de teste.

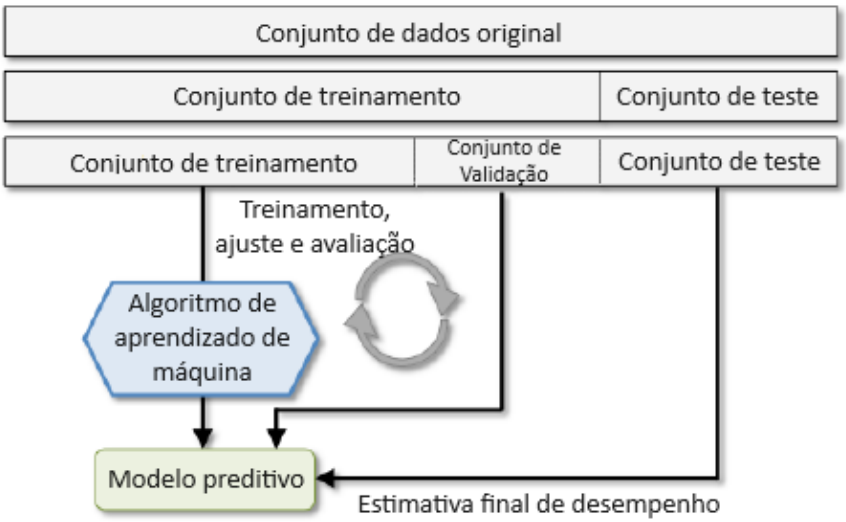


Figura III.4: Ilustração *holdout*. Adaptado de [Raschka, 2015].

- Validação Cruzada *k-fold*: Neste método, o conjunto de dados é particionado aleatoriamente em k subconjuntos sem reposição, onde $k-1$ subconjuntos são utilizados para treinamento e um subconjunto é usado para teste. Este procedimento é repetido k vezes, obtendo assim k modelos e estimativas de desempenho. Normalmente, a validação cruzada *k-fold* é usada para ajuste do modelo, ou seja, para encontrar os valores ideais de hiperparâmetros que gerem um

desempenho de generalização satisfatório.

A Figura III.5 ilustra o conceito de validação cruzada *k-fold* com $k = 10$. O conjunto de dados de treinamento é dividido em 10 dobras (*folds*), e durante as 10 iterações, 9 dobras são usadas para treinamento, e 1 dobra é usada como conjunto de teste para a avaliação do modelo. Além disso, os desempenhos estimados de cada dobra são utilizados para calcular o desempenho médio do modelo final, conforme mostrado na Figura III.5, representado pelo desempenho médio (E).

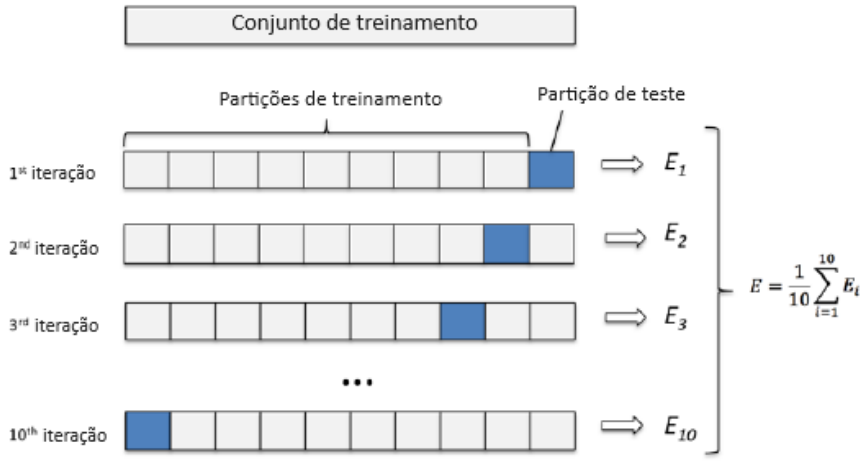


Figura III.5: Ilustração *k-fold*. Adaptado de [Raschka, 2015].

III.2.3 Seleção e Extração de Atributos

Muitas vezes conjuntos de dados reais contêm uma grande quantidade de características, também chamada de atributos ou variáveis, entretanto nem todas são informativas para o processo que devem descrever. Desta maneira, faz-se necessária a aplicação de técnicas de seleção ou extração de atributos para manter os dados que são de fato úteis para o problema em análise. A seleção de características é usada para selecionar um subconjunto das características originais. Já na extração de características, as informações do conjunto de dados são comprimidas para construir um novo subespaço de características.

Os algoritmos de seleção de atributos são usados para reduzir um espaço de atributos d-dimensional inicial para um subespaço de atributos k-dimensional, onde $k < d$. A motivação por trás dos algoritmos de seleção de atributos é selecionar automaticamente um subconjunto de atributos mais relevantes para o problema, melhorando a eficiência computacional ou reduzindo o erro de generalização do modelo ao remover atributos irrelevantes ou ruídos [Raschka, 2015].

A informação mútua (MI, do inglês *Mutual Information*) é considerada um método de seleção de características. Ela é uma medida de independência estatística que possui duas propriedades principais. Primeiro, pode medir qualquer tipo de relacionamento entre variáveis aleatórias, in-

cluindo relacionamentos não lineares. Segundo, a MI é invariante sob transformações no espaço de características que são invertíveis e diferenciáveis, por exemplo, translações, rotações e qualquer transformação que preserve a ordem dos elementos originais dos vetores de características [Vergara and Estévez, 2014].

Semelhante à seleção de atributos, a extração de atributos é usada para reduzir o número de atributos em um conjunto de dados. No entanto, a extração de atributos é usada para transformar ou projetar os dados em um novo espaço de atributos, melhorando a eficiência computacional.

A Análise de Componentes Principais (PCA) é uma técnica de transformação linear não supervisionada amplamente utilizada em diferentes campos, principalmente para redução de dimensionalidade [Raschka, 2015]. O PCA ajuda a identificar padrões nos dados com base na correlação entre as variáveis.

Segundo Mingoti [2007], o PCA tem como objetivo explicar a estrutura de variância e covariância de um vetor aleatório por meio da construção de combinações lineares das variáveis originais. Estas combinações são chamadas de componentes principais e são não correlacionadas entre si. Em geral, o objetivo dessa análise é reduzir a quantidade de dados e facilitar a interpretação das análises realizadas. Assim, a informação contida nas d variáveis originais é substituída pela informação contida em k ($k << d$) componentes principais não correlacionadas.

A informação mútua [Hu et al., 2017] e a análise de componentes principais [Velandia-Cardenas et al., 2021; Correa-jullian et al., 2022] são alguns dos métodos encontrados na literatura para o problema de detecção de falhas em turbinas eólicas.

III.3 Métodos de Classificação

A Regressão Logística, o *Naive Bayes*, o *K-Vizinhos mais Próximos*, as Árvores de Decisão, a Floresta Aleatória e a Máquina de Vetores de Suporte são alguns dos principais algoritmos de classificação de aprendizado de máquina supervisionado. Por outro lado, a Máquina de Vetor de Suporte de Classe Única (OCSVM, do inglês *One-class Support Vector Machine*), é um problema de aprendizagem não supervisionada, mas que pode ser usado para tarefas de classificação binária, especialmente para conjuntos de dados desbalanceados.

III.3.1 Regressão Logística

A Regressão Logística é a transformação da Regressão Linear utilizada para classificação binária, e é também considerada um algoritmo de aprendizagem supervisionado. Tem como objetivo estimar valores discretos (valores binários como 0/1, sim/não, verdadeiro/falso) com base em um determinado conjunto de variáveis explicativas. Normalmente, a Regressão Logística usa uma função "*Sigmoid*" (ou função logística), que possui curva em formato "S", conforme mostrado na Figura

III.6. Esta figura é utilizada para a classificação binária que converte valores para o intervalo [0,1], podendo ser interpretados como a probabilidade de determinada instância pertencer ou não a determinada classe. Por padrão, se a previsão exceder 0.5, uma classe de sucesso é prevista; caso contrário, uma classe de falha é prevista. O limite de 0.5 pode ser ajustado para otimizar a precisão conforme necessário.

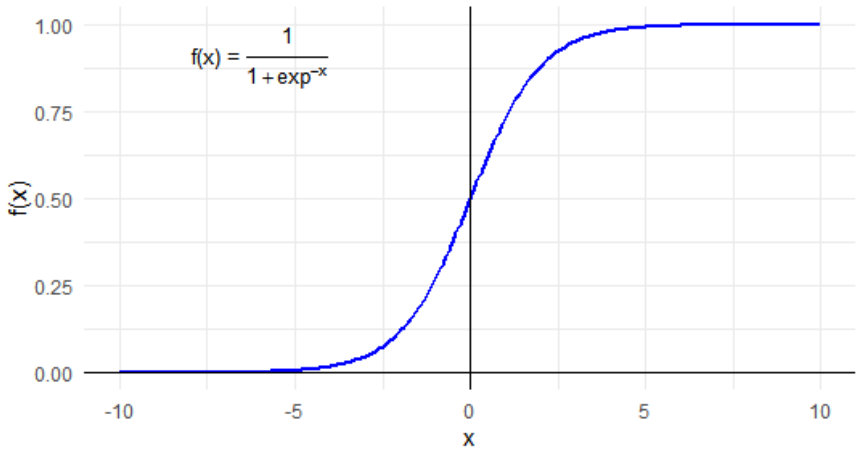


Figura III.6: Gráfico da função logística. Adaptado de [Raschka, 2015].

III.3.2 *Naive Bayes*

O classificador *Naive Bayes* é um algoritmo de classificação baseado em probabilidade. É um dos algoritmos de classificação mais antigos e, mesmo na sua forma mais simples, é surpreendentemente eficaz devido a sua simplicidade. O modelo é fácil de implementar, computacionalmente eficiente e tende a funcionar particularmente bem em conjuntos de dados relativamente pequenos em comparação com outros algoritmos.

O modelo probabilístico desse classificador é baseado no Teorema de Bayes, e o adjetivo "ingênuo" (*naive*) vem da suposição de que as características em um conjunto de dados são mutuamente independentes. Na prática, essa suposição de independência é frequentemente violada, mas os classificadores *Naive Bayes* ainda tendem a ter um bom desempenho mesmo que essa suposição ingênua de Bayes não seja verdadeira [Raschka, 2014].

III.3.3 *K*-Vizinhos mais Próximos

O algoritmo K-Vizinhos mais Próximos, (KNN, do inglês *K-Nearest Neighbours*) é um método baseado no conceito de distância, ou seja, na proximidade entre os dados, que usa informações dos dados de K-vizinhos por um classificador baseado em memória. Por conta de memorizar os dados de treinamento é chamado de aprendizado "preguiçoso".

De acordo com [Faceli et al., 2011], a hipótese base é que dados similares tendem a estar con-

centrados em uma mesma região no espaço de entrada e, da mesma forma, os dados que não são similares estarão distantes entre si. O parâmetro K é definido pelo usuário e em problemas de classificação, é comum utilizar valores ímpares para evitar empates. A proximidade pode ser medida por meio de medidas como a distância euclidiana.

A Figura III.7 mostra como o KNN classifica novos pontos com base nas regiões de decisão formadas pelos dados de treinamento. Um novo ponto de dado recebe o rótulo da classe com base na votação por maioria entre seus K vizinhos mais próximos, ou seja, mais semelhantes.

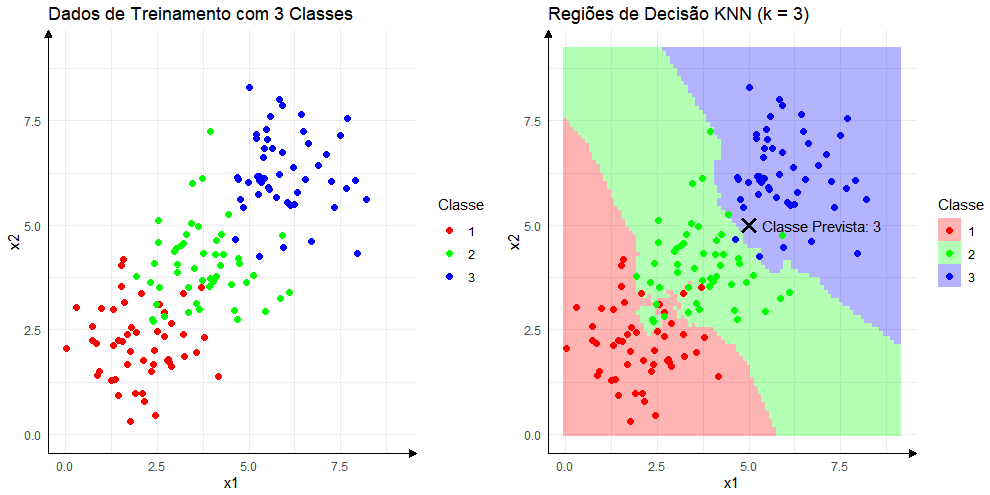


Figura III.7: Ilustração KNN. Adaptado de [Raschka, 2015].

III.3.4 Árvores de Decisão

As Árvores de Decisão são uma importante técnica para implementar a tarefa de classificação pois sua representação é simples, intuitiva e de fácil compreensão. São modelos atraentes por conta da interpretabilidade, e seu nome reflete a forma como eles funcionam: dividindo dados por meio de uma série de perguntas ou decisões.

A ideia geral dos métodos baseados em árvores é particionar o espaço recursivamente em retângulos (sub-regiões), nos quais um modelo simples é aprendido. Estes modelos utilizam a estratégia de dividir para conquistar: um problema complexo é decomposto em sub-problemas mais simples e recursivamente esta técnica é aplicada a cada sub-problema [Raschka, 2015].

O processo do algoritmo de decisão começa na raiz da árvore e divide os dados com base na característica que oferece o maior ganho de informação. A divisão é repetida em cada nó filho até que as folhas estejam puras, ou seja, todas as amostras de um nó pertencem à mesma classe. No entanto, isso pode resultar em árvores muito profundas, com muitos nós, levando ao *overfitting*. Para evitar isso, geralmente é aplicada uma técnica chamada de poda, que estabelece um limite para a profundidade máxima da árvore [Raschka, 2015].

A Figura III.8 ilustra um exemplo do problema de falhas em turbinas eólicas. Com base nas

características do conjunto de treinamento, o modelo de árvore de decisão aprende uma série de perguntas para inferir as classes dos exemplos. No caso específico das turbinas eólicas, essas perguntas podem envolver variáveis como a temperatura do óleo da caixa de engrenagens, a temperatura do rolamento do gerador e a velocidade do vento.

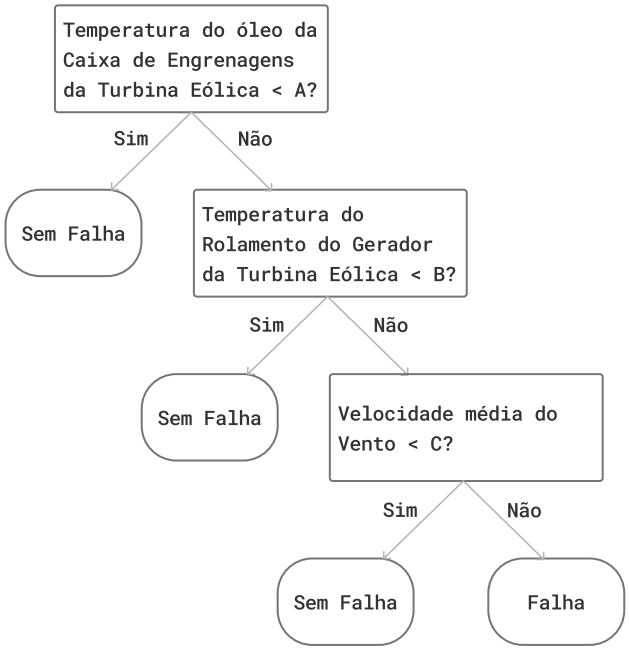


Figura III.8: Ilustração de uma Árvore de Decisão.

III.3.5 Floresta Aleatória

A Floresta Aleatória consiste em um conjunto de árvores de decisão geradas dentro de um mesmo objeto. Cada objeto (conjunto de árvores) passa por um mecanismo de votação (*bagging*) que elege a classificação mais votada. Tal método é uma combinação de preditores de árvores, de modo que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente e com a mesma distribuição para todas as árvores da floresta [Breiman, 2001].

Pode-se interpretar o modelo de Floresta Aleatória como uma paralelização das árvores de decisão, pois várias Árvores de Decisão são construídas simultaneamente para a classificação. Para realizar essa tarefa, o método gera um subconjunto do conjunto de treinamento com reposição (*bootstrap*) para criar um novo conjunto de dados de treinamento e aplica a técnica de árvore de decisão a cada subconjunto para gerar classificadores. Essa estrutura paralela das florestas aleatórias pode ajudar na classificação de grandes volumes de dados [Suthaharan, 2016].

III.3.6 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte (SVM, do inglês *Support Vector Machine*) é um modelo de AM poderoso e versátil, capaz de realizar classificação linear ou não linear, regressão e até detecção de *outliers*.

A ideia fundamental por trás das SVMs é encontrar o hiperplano de margem máxima que melhor separa os pontos das diferentes classes. Os pontos de treinamento equidistantes do hiperplano de margem máxima mais próximo dele são chamados de vetores de suporte, os quais são os principais responsáveis pela autoridade deste hiperplano. Para acomodar limites não lineares entre classes, o espaço dimensional dos dados é aumentado por meio do uso de *kernels*, tornando o algoritmo mais flexível [Witten and James, 2013].

O conceito básico por trás dos métodos de *kernel* para lidar com dados linearmente inseparáveis é criar combinações não lineares das características originais para projetá-las em um espaço de maior dimensionalidade por meio de uma função de mapeamento, tornando-as linearmente separáveis.

Por exemplo, para um problema que não pode ser resolvido por um hiperplano linear, os métodos de *kernel* podem projetar o problema para um espaço tridimensional, onde ele pode ser resolvido por um hiperplano. Esta capacidade de transformar problemas não lineares em lineares com *kernels* é uma das razões pelas quais SVMs são úteis para muitos problemas de classificação [Raschka, 2015].

Na Figura III.9 é ilustrada o algoritmo SVM, onde o objetivo é maximizar a margem. A margem é definida como a distância entre o hiperplano separador (limite de decisão) e as amostras de treinamento que estão mais próximas deste hiperplano, as chamadas vetores de suporte.

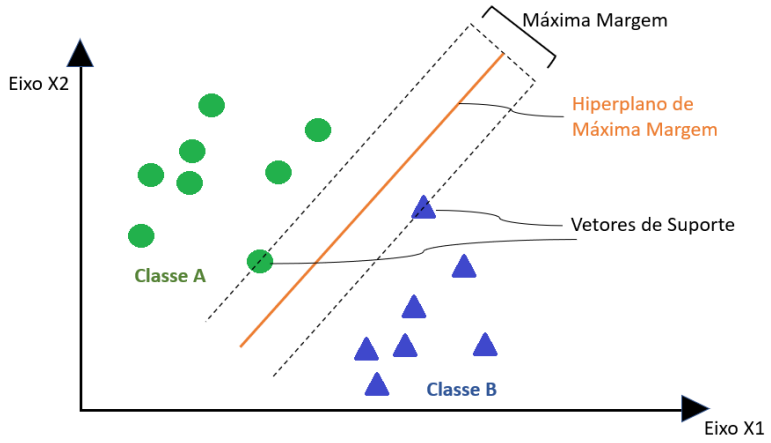


Figura III.9: Ilustração da Máquina de Vetores de Suporte (SVM).

Máquina de Vetor de Suporte de Classe Única

O algoritmo SVM de uma classe, OCSVM do inglês *One-Class* SVM, introduzido por Schölkopf et al. [1999], é uma adaptação do algoritmo SVM para o problema de classificação de uma única

classe. Os classificadores de classe única normalmente são utilizados na detecção de padrões raros. Em geral, utiliza-se apenas a classe comum (estado normal) no treinamento do modelo e na validação há uma mistura de instâncias normais e anormais.

Em um SVM binário, o objetivo é encontrar o hiperplano que separa duas classes com a maior margem possível. No OCSVM, durante o treinamento, existem apenas dados rotulados positivamente. O hiperplano correspondente à classe negativa é definido para ser a origem do sistema de coordenadas. O objetivo do OCSVM se resume a encontrar um hiperplano mais distante da origem, de forma que os dados rotulados positivamente fiquem no semi-espaço positivo do hiperplano [Perera et al., 2021].

Embora não sejam projetados especificamente para problemas de classificação binária, os algoritmos de classificação de uma única classe podem ser eficazes em conjuntos de dados desbalanceados, onde há poucos exemplos da classe minoritária. Nesse contexto, a classe majoritária é considerada "normal", enquanto a classe minoritária é tratada como uma anomalia.

III.4 Otimização de Hiperparâmetros

No aprendizado de máquina, a otimização ou ajuste de hiperparâmetros é o processo de encontrar a combinação certa de valores de hiperparâmetros para obter o máximo desempenho dos dados em um período de tempo razoável. Um pré-requisito para treinar modelos de AM em geral é criar uma combinação específica de valores de hiperparâmetros. Somente após a escolha de um conjunto específico de hiperparâmetros é que o processo de treinamento pode ajustar os parâmetros do modelo [Japa et al., 2023]. Isso pode ser particularmente importante ao comparar o desempenho de diferentes modelos de AM em um conjunto de dados.

Hiperparâmetros são valores de parâmetros usados para controlar o processo de aprendizagem e afetam significativamente o desempenho dos modelos. Segundo Agrawal [2021], os modelos de AM são compostos por dois tipos diferentes de parâmetros: hiperparâmetros, ou seja, parâmetros que o usuário pode definir arbitrariamente antes de iniciar o treinamento, e Parâmetros do modelo, aqueles aprendidos durante o treinamento do modelo.

A maioria dos algoritmos de AM vem com valores padrão para seus hiperparâmetros [Agrawal, 2021]. Mas os valores padrão nem sempre funcionam bem em diferentes tipos de projetos de AM. É por isso que precisa otimizá-los para obter a combinação certa e oferecer o melhor desempenho. Alguns exemplos comuns de hiperparâmetros incluem taxa de aprendizagem, abandono e função de ativação para redes neurais, profundidade máxima da árvore para florestas aleatórias e taxa de regularização para regressão linear regularizada, entre outros.

Os profissionais geralmente ajustam esses hiperparâmetros usando métodos de força bruta padrão, como pesquisar sistematicamente uma grade de hiperparâmetros (*grid search*) ou amostrar

hiperparâmetros aleatoriamente (*random search*). Entre os métodos existentes para acelerar a otimização de hiperparâmetros, o *halving* sucessivo emergiu como um algoritmo de "parada antecipada" popular e de última geração [Li et al., 2020].

A Figura III.10 exemplifica o conceito de validação cruzada combinado com a pesquisa de grade de hiperparâmetros. Essa junção é útil para ajuste fino do desempenho de um modelo de aprendizado de máquina ao variar os valores de seus hiperparâmetros e também para escolher entre os diferentes algoritmos de aprendizado de máquina [Raschka, 2015].

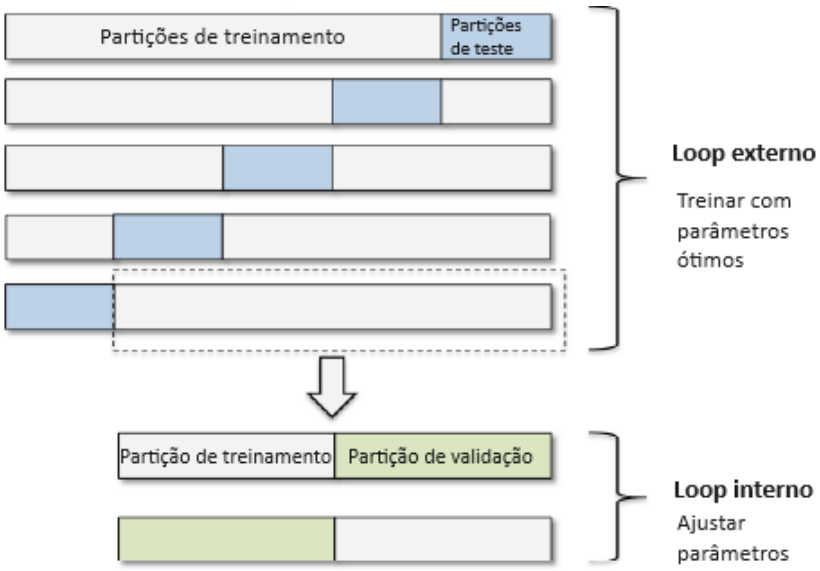


Figura III.10: Ilustração *k-fold* com ajuste de hiperparâmetros. Adaptado de [Raschka, 2015].

III.4.1 *Grid Search*

O *Grid Search* usa uma abordagem de força bruta para testar todas as combinações de uma lista predefinida de valores de hiperparâmetros e encontrar o modelo com o melhor conjunto de parâmetros que fornece precisão máxima. De acordo com Agrawal [2021], este método é a maneira mais segura de encontrar o melhor conjunto de hiperparâmetros, uma vez que todas as combinações são avaliadas, mas também tem suas desvantagens, quando se trata de dimensionalidade, ele sofre quando o número de hiperparâmetros cresce exponencialmente, exigindo mais tempo para ser executado.

III.4.2 *Random Search*

O *Random Search* é um método que consome menos tempo e recursos do que o *Grid Search*. Ele testa hiperparâmetros aleatoriamente a partir de combinações aleatórias de um intervalo de valores, cria um conjunto e treina o modelo nele. Este método pode não encontrar o melhor conjunto de hiperparâmetros, mas pode fornecer um modelo que se aproxime do ideal em termos de desempenho, economizando muito tempo computacional. Uma desvantagem da pesquisa aleatória é que ela não

tenta melhorar com base em combinações de hiperparâmetros previamente testadas [Japa et al., 2023].

III.4.3 *Halving Random Search*

Finalmente, o *Halving Random Search* implementa uma estratégia de torneio de forma sucessiva. Isto quer dizer que ele começa com um pequeno número de casos de treinamento para identificar e selecionar rapidamente modelos candidatos pouco promissores. Os modelos que sobrevivem para a próxima rodada são avaliados usando uma proporção maior dos dados disponíveis. Esse processo se repete até restar apenas alguns modelos candidatos, que são então treinados e avaliados usando todos os dados disponíveis [Soper, 2023].

A Figura III.11 apresenta uma representação gráfica do algoritmo de *Halving Random Search* aplicado de forma sucessiva, no qual o número de modelos candidatos diminui exponencialmente de uma iteração para a próxima, enquanto o número de casos de treinamento aumenta exponencialmente de uma iteração para a próxima.

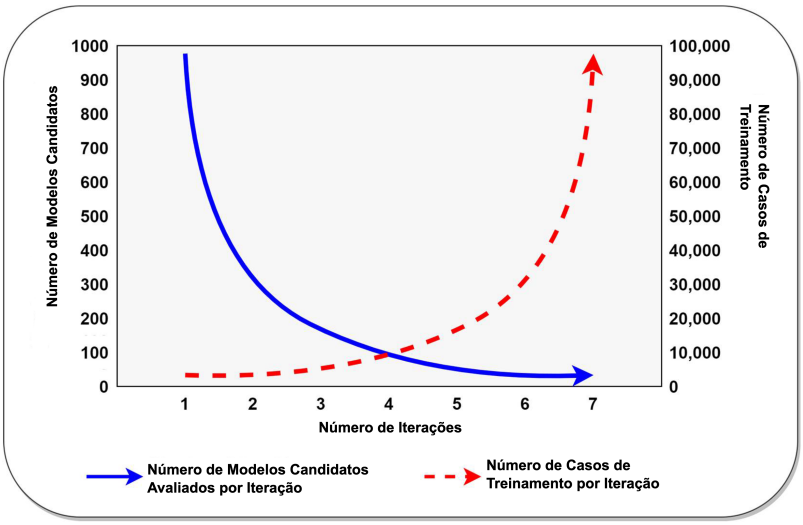


Figura III.11: Representação do *Halving Random Search* sucessivo [Soper, 2023].

A Figura III.12 [1] mostra um problema de ajuste de hiperparâmetros com um espaço de busca 2D, onde cada ponto representa uma configuração específica de hiperparâmetros e cores mais quentes correspondem a um melhor desempenho. Vale ressaltar que métodos de seleção adaptativos para ajuste de hiperparâmetros, como o *halving* sucessivo, procedem sequencialmente e concentram-se em regiões promissoras do espaço de busca [Li et al., 2020].

[1] Adaptado de: *https://blog.ml.cmu.edu/2018/12/12/massively-parallel-hyperparameter-optimization/*

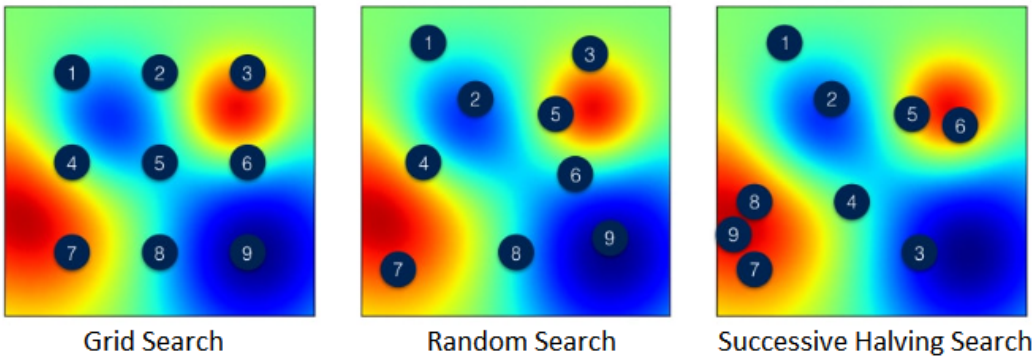


Figura III.12: Problema de ajuste de hiperparâmetros em um espaço de busca 2D.

Os métodos de hiperparâmetros são essenciais para encontrar o modelo mais adequado. No entanto, esses métodos podem aumentar significativamente o tempo computacional do processamento, dependendo da técnica de busca utilizada. Para tal, neste trabalho são comparadas três técnicas de busca de hiperparâmetros, analisando os resultados obtidos por cada uma e o tempo de processamento necessário.

III.5 Avaliação dos Modelos Preditivos

Para medir o desempenho dos algoritmos de classificação e verificar a capacidade do modelo de generalizar para um conjunto de exemplos nunca antes vistos, diversas métricas podem ser utilizadas no contexto de classificação binária, como por exemplo, a matriz de confusão, a acurácia, a precisão, a sensibilidade e o F_1-*Score* [Provost and Kohavi, 1998].

Um método bastante utilizado para analisar os resultados produzidos pelos classificadores é a matriz de confusão e as medidas de desempenho que dela resultam. A matriz de confusão é exibida na Figura III.13, que fornece as quantidades preditas e observadas em cada classe da variável resposta.

		Classe predita	
		0	**1**
Classe verdadeira	**0**	VN	FP
	1	FN	VP

Figura III.13: Matriz de confusão para um problema com duas classes. Fonte: Elaborada pelo autor

Verdadeiros Positivos (VP): Número de exemplos da classe positiva e que foram corretamente classificados, neste caso representa a detecção correta de falhas; Verdadeiros Negativos (VN): Número de exemplos da classe negativa e que foram corretamente classificados, ou seja, os exemplos que não tiveram falhas; Falsos Positivos (FP): Número de exemplos da classe negativa e que foram incorretamente classificados pelo modelo, compreendendo os falsos alarmes; e Falsos Negativos

(FN): Número de exemplos da classe positiva e que foram incorretamente classificados, isto é, as falhas não detectadas.

As medidas de desempenho baseadas na matriz de confusão são descritas a seguir:

- **Acurácia** - A acurácia é a proporção de todos os casos corretamente classificados pelo número total de exemplos no conjunto de teste. É uma métrica intuitiva e fácil de interpretar, sendo especialmente útil quando as classes estão balanceadas, ou seja, quando o número de exemplos em cada classe é aproximadamente igual. No entanto, em cenários onde há um desbalanceamento significativo entre as classes (por exemplo, em problemas de detecção de fraudes, onde a maioria das transações são legítimas e apenas uma pequena fração é fraudulenta), a acurácia pode ser uma métrica enganosa. Isso ocorre porque um modelo que sempre prediz a classe majoritária pode obter uma alta acurácia sem ser realmente eficaz na identificação das classes minoritárias.

$$acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{III.1}$$

- **Precisão** - Corresponde a proporção de resultados positivos classificados corretamente entre todos aqueles preditos como positivos. É especialmente relevante em contextos onde o custo de falsos positivos é alto.

$$precisão = \frac{VP}{VP + FP} \tag{III.2}$$

- **Sensibilidade ou Revocação** - Corresponde à taxa de acerto na classe positiva verdadeira. É importante em situações onde perder um falso negativo é mais crítico.

$$sensibilidade = \frac{VP}{VP + FN} \tag{III.3}$$

- **F1-*Score*** - É a média harmônica entre a precisão e a revocação. É útil quando se deseja um equilíbrio entre precisão e revocação.

$$F_1\text{-}score = 2 * \frac{\text{precisão} * revocação}{\text{precisão} + revocação} \tag{III.4}$$

- **Coeficiente de Correlação de Matthews (*Matthews Correlation Coefficient* - MCC)** - O MCC assume valores entre -1 e 1. Uma pontuação de 1 indica concordância perfeita entre os valores previstos e reais. O MCC leva em conta todos os quatro valores da matriz de confusão: VN, VP, FP e FN. Isso proporciona uma medida mais equilibrada do desempenho do modelo, especialmente útil em contextos de classes desbalanceadas.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{III.5}$$

- **Área sob a curva ROC (*Receiver Operating Characteristic*)** - É uma curva de probabilidade e a Área sob a curva (AUC) representa o grau ou medida de separabilidade. Indica o quão bem o modelo é capaz de distinguir entre classes. Quanto maior a AUC, melhor é o modelo

Capítulo IV Conjunto de Dados

Neste capítulo é abordado o contexto dos dados utilizados para o desenvolvimento deste trabalho. Na Seção IV.1 é apresentada uma breve discussão sobre a importância da energia eólica, destacando tanto o papel estratégico do setor em termos de investimentos quanto seu potencial para inovações tecnológicas. Na Seção IV.2 é detalhado o funcionamento do sistema de uma turbina eólica, com a descrição de seus principais componentes e sua relevância no processo de geração de energia. Por fim, na Seção IV.3, são discutidos os dados empregados neste estudo, incluindo uma análise exploratória que visa compreender melhor as características da base de dados utilizada.

IV.1 Energia Eólica

Os números globais de investimento nos setores de energias renováveis, fósseis, redes e energia nuclear nos últimos cinco anos mostram que a energia renovável cresceu mais de 20% desde 2019, Figura IV.1. Em contraste, o investimento em combustíveis fósseis tem se mantido estável ou até mesmo declinante. Estes investimentos ressaltam cada vez mais a importância das energias renováveis na produção de energia global, como alternativa a energias fósseis de modo a reduzir o impacto no ambiente e no aquecimento global.

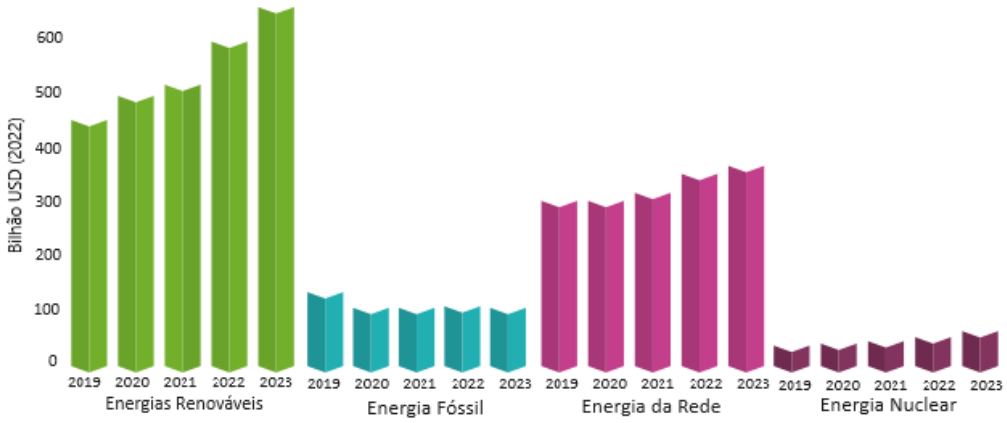


Figura IV.1: Investimentos do setor de energia nos últimos 5 anos. Adaptado de [GWEC, 2024]

De acordo com o GWEC [2024], assim como qualquer outro setor industrial, a energia eólica deve uma grande parte de seu sucesso à sua busca pela inovação. À medida que a energia eólica se expandiu para todas as partes do mundo, a indústria também expandiu sua capacidade de se

adaptar às condições e requisitos locais, identificando soluções que entregam volumes maiores de energia de forma mais confiável e eficiente.

Uma corrida tecnológica para desenvolver novas turbinas é não apenas um investimento caro em pesquisa e desenvolvimento, mas também um risco para uma cadeia de suprimentos sustentável. Turbinas maiores geralmente resultam em maiores capacidade de produção, porém também incluem a adaptação às condições específicas do local, a padronização dos componentes e a otimização de todo o processo. A Figura IV.2 exibe a evolução do tamanho das turbinas ao longo dos anos, incluindo a estimativa de produção e magnitude do equipamento esperada em 2030.

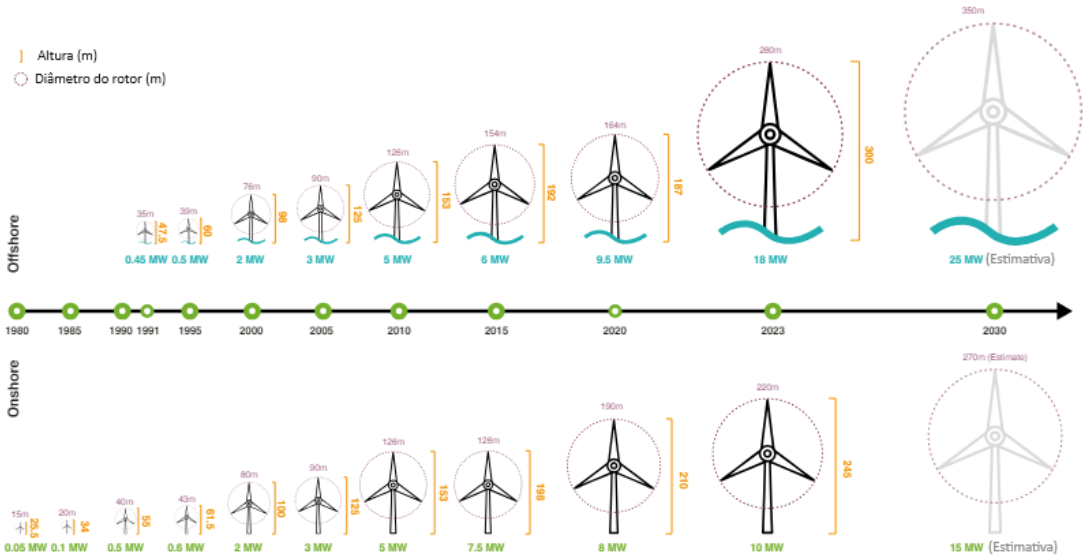


Figura IV.2: Tendência do tamanho das turbinas *onshore* e *offshore*, 1980-2030. Adaptado de [GWEC, 2024]

IV.2 Turbinas Eólicas

De acordo com Letcher [2023], a geração de eletricidade a partir de turbinas eólicas teve suas origens nos Estados Unidos na década de 1970. Esse desenvolvimento foi impulsionado pela necessidade de substituir a energia derivada de combustíveis fósseis por formas renováveis de energia. De todas as formas renováveis de energia (eólica, solar, geotérmica e hidrelétrica), a energia eólica e solar destacaram-se pelo crescimento significativo e positivo.

A turbina eólica é responsável por converter a energia cinética do vento em energia elétrica. Isso ocorre quando o vento movimenta as pás, fazendo girar o rotor, que então transmite essa rotação ao gerador [Letcher, 2023]. Na Figura IV.3[1], é possível visualizar o sistema de uma turbina eólica e seus principais componentes. Os componentes externos incluem a torre, o rotor, o anemômetro, as pás e a nacele, que protege todos os componentes internos. Dentro da nacele, na parte interna, estão a caixa de engrenagens, o gerador e o controlador.

[1] Adaptado de *https://windmillstech.com/wind-turbine-components/*

A nacele é o componente onde diversos outros sistemas são instalados, incluindo o gerador, a caixa de velocidade, o sistema de controle, os sensores de velocidade e direção do vento, e os motores que posicionam a turbina para otimizar a captação do vento, entre outros. A torre tem a função de sustentar e posicionar o rotor e a nacele.

As pás desempenham um papel crucial em uma turbina eólica, pois o desempenho desta depende da eficiência das pás para converter a energia cinética do vento em energia mecânica. Se houver uma caixa multiplicadora, dois eixos são necessários: o eixo de baixa rotação, ligado ao rotor, e o eixo de alta rotação, conectado ao gerador. O gerador elétrico transforma a energia mecânica em energia elétrica por meio de equipamentos de conversão eletromecânica [Letcher, 2023].

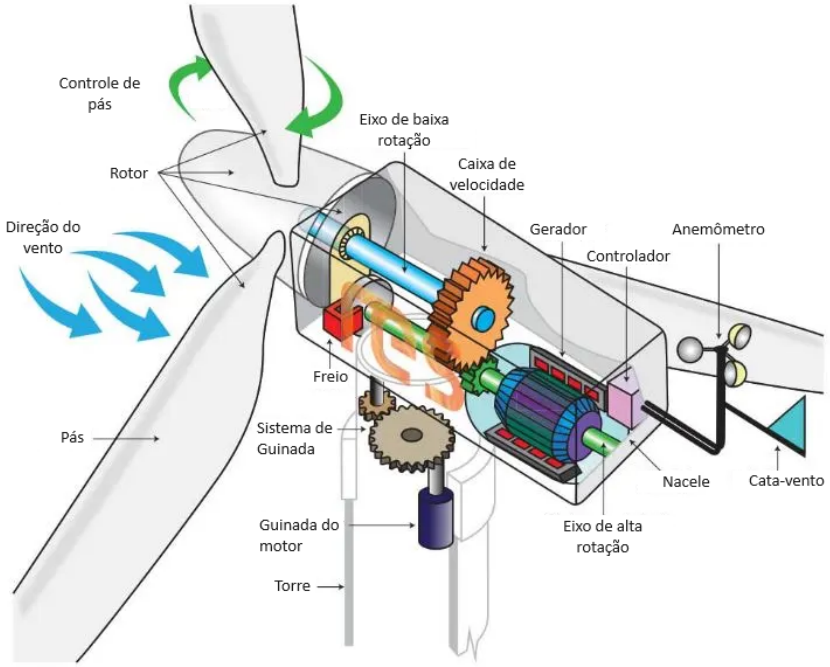


Figura IV.3: Componentes de uma turbina eólica.

Segundo Rezamand et al. [2020], a falha inesperada de componentes das turbinas eólicas pode causar perdas econômicas substanciais. Portanto, é prudente empregar técnicas de diagnóstico e prognóstico das turbinas, que visam reduzir as inspeções custosas e a manutenção baseada em intervalos de tempo, por meio de monitoramento preciso, detecção precoce de falhas e previsão de falhas iminentes, ou seja, a estimativa da Vida Útil Remanescente (RUL, do inglês *Remaining Useful Life*).

Para esse propósito, é essencial um sistema de monitoramento adequado. O Sistema de Supervisão, Controle e Aquisição de Dados (SCADA) é fundamental na detecção de falhas, fornecendo suporte para técnicas de diagnóstico e prognóstico. O SCADA correlaciona múltiplos conjuntos de variáveis, como velocidade do vento e potência, para treinar modelos de estados operacionais normais e utilizar esses modelos para detectar comportamentos anormais e *outliers* [Rezamand et al.,

2020].

Como o conjunto de dados utilizado nesta pesquisa é extraído do sistema SCADA, o interesse do trabalho está no emprego de modelos orientados por dados. Esses modelos buscam transformar os dados fornecidos pelo monitoramento da condição em modelos relevantes do comportamento da degradação, em vez de uma compreensão física dos processos de falha.

IV.3 Base de Dados

A Energias de Portugal (EDP) é uma empresa global do setor energético, presente atualmente em 29 países. A empresa promove periodicamente desafios, disponibilizando dados de suas operações para universidades, pesquisadores e *startups*, com o objetivo de incentivar o desenvolvimento de soluções inovadoras para o mercado. O conjunto de dados utilizado neste trabalho é proveniente da EDP[1], disponibilizado a partir de um desafio proposto pela empresa, cujo objetivo era a detecção de falhas em turbinas eólicas.

De acordo com Mendes et al. [2020], esse é um dos conjuntos de dados gratuitos mais completos disponíveis para análise de recursos eólicos e pesquisa do desempenho de turbinas eólicas. Os registros foram extraídos de um sistema SCADA consistindo em dados de cinco turbinas eólicas medidos nos anos de 2016 e 2017. O conjunto de dados inclui o histórico de falhas, informações meteorológicas e arquivo de registros de eventos.

As informações disponibilizadas pela EDP são descritas a seguir:

- *Metmast*: Conjunto de dados das variáveis do mastro meteorológico, medido a cada 10 minutos. Os dados são extraídos de uma única torre e incluem 40 variáveis relacionadas à velocidade e direção do vento (2 sensores anemométricos), temperatura, pressão atmosférica, umidade, precipitação.

- *Failures*: Conjunto de dados com o registro das ocorrências de falhas dos 5 componentes da turbina eólica, medido no tempo de cada ocorrência. Os componentes com falhas são: Transformador, Rolamento do Gerador, Grupo Hidráulico, Gerador e Caixa de Velocidade. Inclui também a identificação da turbina e o *timestamp*, que registra com precisão de segundos a data da medição.

- *Logs*: Conjunto de dados do histórico dos eventos normais e anormais que ocorreram em cada turbina.

- *Signals*: Conjunto de dados das variáveis dos sensores do sistema SCADA para os componentes e valores de produção mais importantes de cada turbina, com leitura a cada 10 minutos. Inclui 81 variáveis relacionadas à velocidade e direção do vento, gerador, transformador, entre outras.

[1] `https://www.edp.com/en/innovation/open-data/data`, acessado em 15/08/21

- *Locations*: Localização das turbinas, contendo a latitude e longitude. As cinco turbinas estão localizadas em um parque eólico *Offshore* no Golfo da África Ocidental.

A Tabela IV.1 apresenta a quantidade de observações total e de variáveis em cada conjunto de dados. No Apêndice VII, encontram-se as descrições das variáveis dos conjuntos de dados *Metmast* e *Signals* disponibilizadas pela EDP.

Tabela IV.1: Descrição dos conjuntos de dados.

Conjunto de dados	Quantidade de observações	Quantidade de variáveis
Metmast	87.528	41
Failures	28	4
Logs	318.835	5
Signals	498.338	83
Locations	17	3

A Tabela IV.2 apresenta uma amostra do conjunto de dados *Metmast*. Os dados correspondem a registros adquiridos nos anos de 2016 capturados em intervalos de 10 minutos, formando a série temporal multivariada.

Tabela IV.2: Amostra das primeiras observações do dataset *Metmast*.

Timestamp	Windspeed1				Windspeed2			
	Min	Max	Avg	Var	Min	Max	Avg	Var
2016-01-01T00:00:00+00:00	3,70	6,00	5,10	0,21	3,80	6,00	5,10	0,22
2016-01-01T00:10:00+00:00	4,10	6,00	5,10	0,09	4,10	6,00	5,20	0,10
2016-01-01T00:20:00+00:00	4,50	6,70	5,70	0,26	4,40	6,80	5,80	0,30
2016-01-01T00:30:00+00:00	5,10	7,00	6,30	0,11	5,10	7,10	6,40	0,12
2016-01-01T00:40:00+00:00	4,70	7,30	6,20	0,27	4,90	7,40	6,30	0,27
2016-01-01T00:50:00+00:00	4,90	7,40	6,60	0,33	5,00	7,60	6,80	0,35
2016-01-01T01:00:00+00:00	3,90	7,10	5,30	0,33	4,00	7,10	5,40	0,34
2016-01-01T01:10:00+00:00	4,30	6,70	5,70	0,26	4,50	6,90	5,80	0,26
2016-01-01T01:20:00+00:00	4,60	6,50	5,90	0,12	4,80	6,70	6,00	0,13
2016-01-01T01:30:00+00:00	4,40	6,50	5,60	0,16	4,60	6,70	5,70	0,15

O conjunto *Failures* fornece o histórico das falhas que ocorreram nos anos de 2016 e 2017. A Figura IV.4 apresenta as falhas para cada componente das turbinas e, como pode ser visto, o Grupo Hidráulico foi o componente que mais falhou nesse período, totalizando 8 falhas, seguido do Gerador, com 7 falhas.

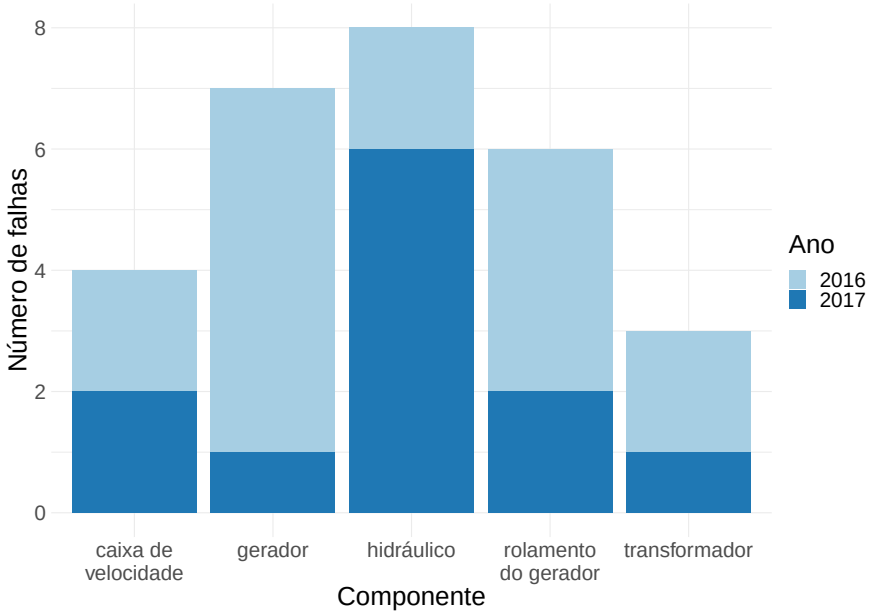


Figura IV.4: Frequência de Falhas nos Componentes da Turbina Eólica.

A Figura IV.5 apresenta as falhas dos componentes para cada uma das 5 turbinas eólicas disponibilizadas pela EDP. A turbina T06 é a que possui o maior número de falhas, com destaque para o componente do Gerador no ano de 2016. Outro componente com alta incidência de falhas foi o Rolamento do Gerador, que registrou 3 falhas no ano de 2016 na turbina T09.

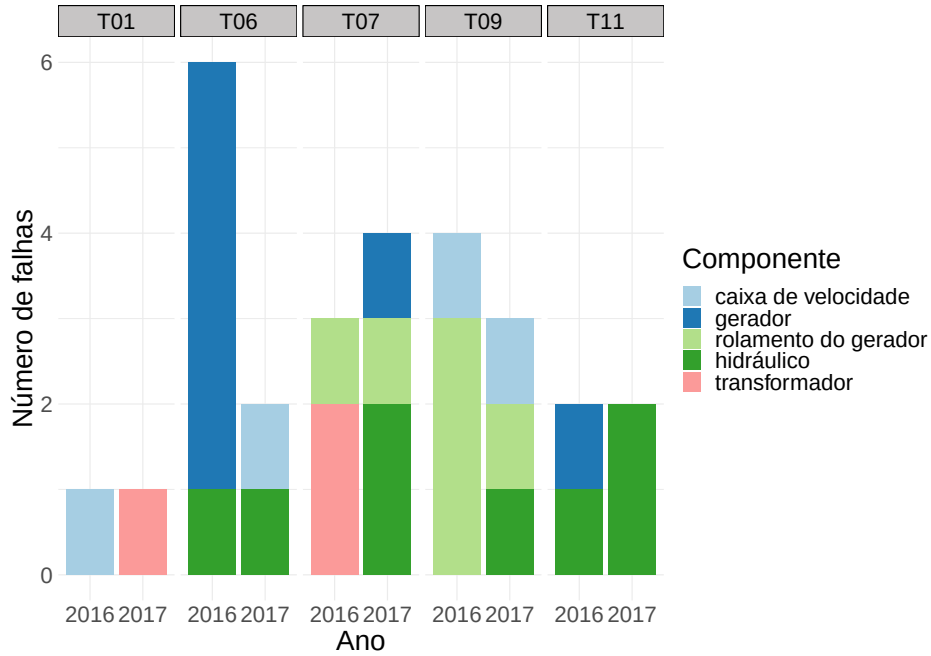


Figura IV.5: Frequência de Falhas por Turbina Eólica.

O conjunto de dados contém medições de cinco turbinas. As medições foram registradas a cada 10 minutos ao longo de dois anos, 2016 e 2017. As falhas fornecidas pela EDP abrangem cinco componentes: Caixa de Velocidades, Gerador, Rolamento do Gerador, Transformador e Grupo

Hidráulico. A Tabela IV.3 resume como as falhas são divididas por componente, ano e turbina.

Tabela IV.3: Falhas das Turbinas por Componente

	2016						2017						
Componente	T01	T06	T07	T09	T11	Total	T01	T06	T07	T09	T11	Total	Total
Caixa de Velocidade	1	0	0	1	0	2	0	1	0	1	0	2	4
Gerador	0	5	0	0	1	6	0	0	1	0	0	1	7
Rolamento do Gerador	0	0	1	3	0	4	0	0	1	1	0	2	6
Grupo Hidráulico	0	1	0	0	1	2	0	1	2	1	2	6	8
Tansformador	0	0	2	0	0	2	1	0	0	0	0	1	3
Total de Falhas	1	6	3	4	2	16	1	2	4	3	2	12	28

Para criar o conjunto de dados com os dados dos sensores (*Signals*) e meteorológicos (*Metmast*), as bases foram combinadas pela variável de tempo de medição (*Timestamp*). Para alguns instantes de tempo em que não havia medição, os dados foram imputados de forma que a série estivesse completa com todas as medições a cada 10 minutos. Essa imputação foi realizada pela repetição dos valores do tempo anterior. Após essa etapa, os dados das falhas foram incluídos pelo código da turbina e tempo de medição menor ou igual ao tempo de falha. Assim, foi construída a base de dados para o aprendizado do modelo.

Como o foco está na tarefa de classificação, foi criado um campo com a diferença entre o tempo de falha e o tempo de medição, atribuindo o valor '1' para as observações dentro do período de 60 dias antes da ocorrência das falhas e '0' para as demais. Esse limite de 60 dias foi escolhido com base na avaliação utilizada pela EDP, onde uma falha era contabilizada como corretamente classificada se fosse prevista com até 60 dias de antecedência da data de ocorrência, conforme exemplificado na Figura IV.6.

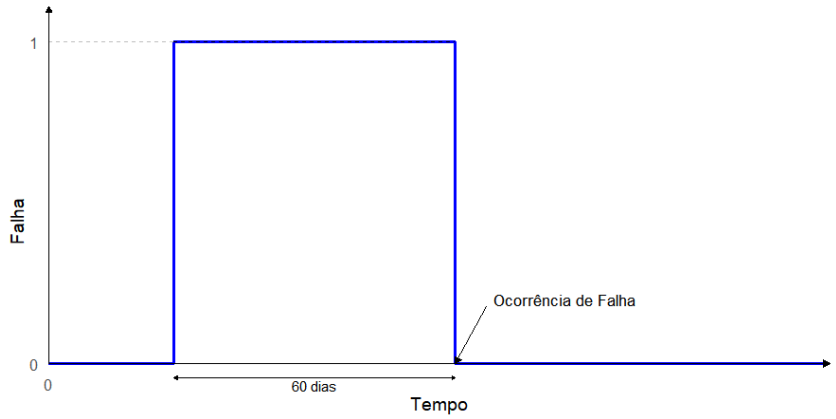


Figura IV.6: Falha

A Figura IV.7 apresenta o percentual de falhas e não falhas em cada componente da turbina eólica, considerando um limite de 60 dias entre o tempo de medição e o tempo de falha para classificar uma observação como falha. Pode-se notar que, para todos os componentes, o percentual de falhas

é baixo, sendo menor que 50%. O Grupo Hidráulico é o componente com o menor percentual de falhas, 12,3%, enquanto o Gerador é o componente com maior percentual de falhas, 22,3%. Embora o Grupo Hidráulico tenha a maior quantidade de falhas, quando é utilizado o critério de 60 dias, também conhecido como vida útil restante, esse componente apresentou falhas em mais turbinas distintas, totalizando 4 turbinas. Portanto, a quantidade de falhas nesse componente não representa uma parte significativa do total.

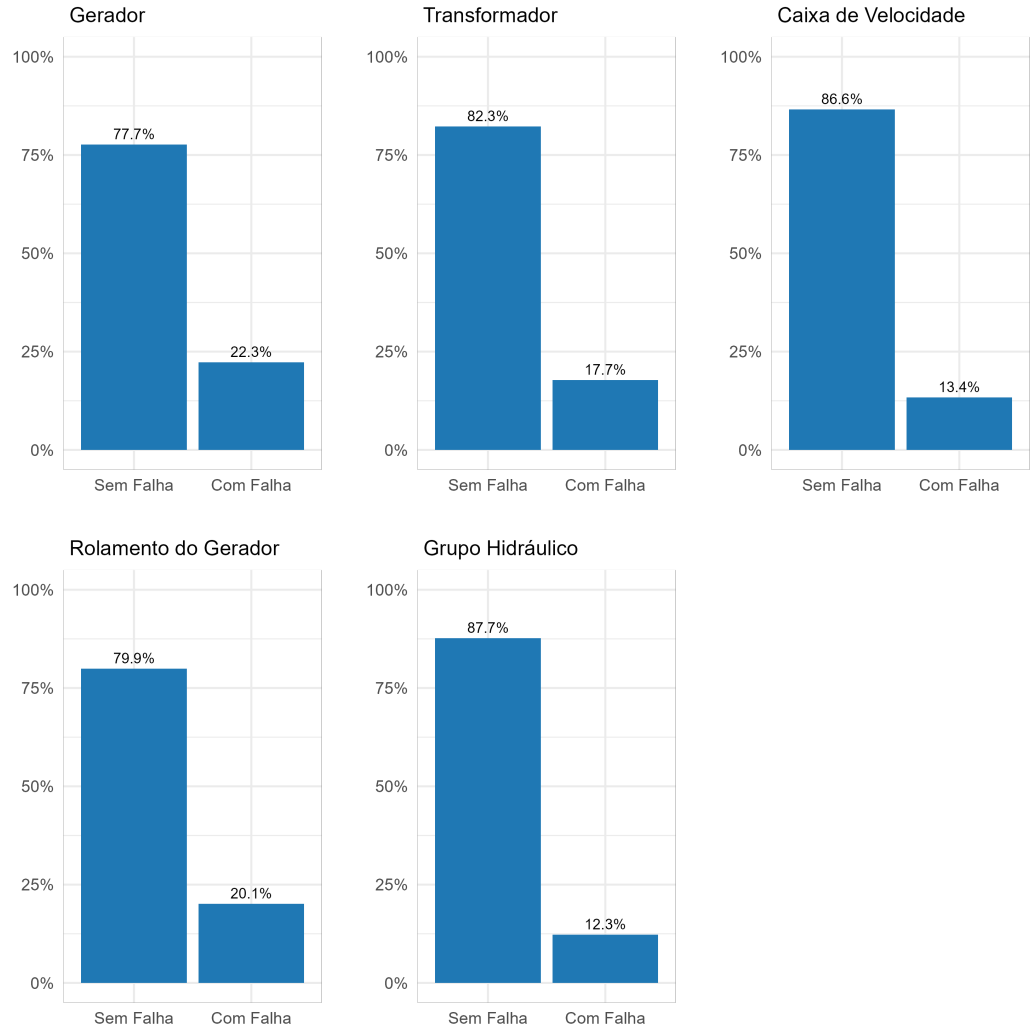


Figura IV.7: Percentual de falhas em cada componente da turbina eólica

A curva de potência da EDP pode ser visualizada na Figura IV.8, a partir dela nota-se que a turbina não produz energia para velocidades do vento abaixo de 4 m/s ou acima de 25 m/s. Este gráfico é essencial para entender o desempenho das turbinas eólicas em diferentes condições de vento e para analisar a eficiência e a capacidade de geração de energia ao longo do tempo.

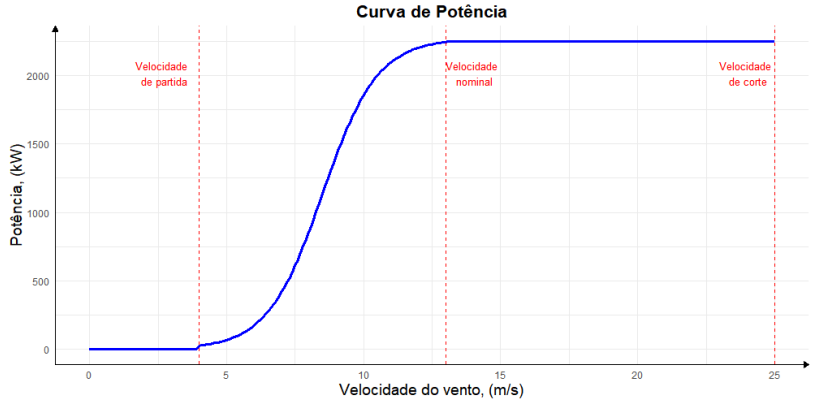


Figura IV.8: Curva de Potência

A seguir, é feita uma análise gráfica detalhada das condições operacionais que antecederam as falhas, destacando a importância de um monitoramento contínuo e preciso para a detecção precoce de anomalias.

No dia 30/04/2016, houve uma falha no rolamento do gerador da turbina T07. A Figura IV.9 evidencia o valor extremo da temperatura média nessa data. A Figura IV.10 mostra a falha em uma janela menor de tempo, onde, no dia 29/04/2016, pode-se observar um aumento da temperatura, indicando uma possível pré-falha. Este aumento na temperatura um dia antes da falha é um sinal importante que poderia ter sido usado para antecipar a necessidade de manutenção preventiva.

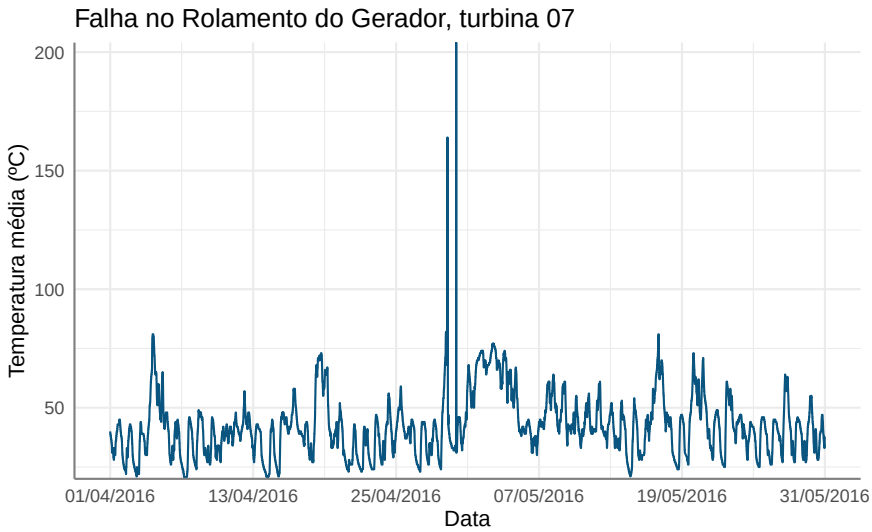


Figura IV.9: Falha Rolamento Gerador T07

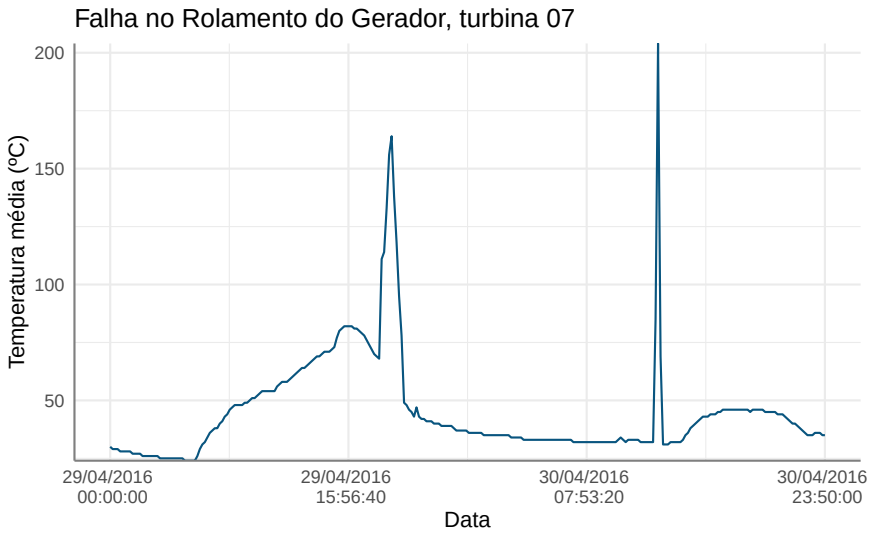


Figura IV.10: Falha Rolamento Gerador T07

No dia 11/07/2016, houve uma falha no gerador da turbina T06. A Figura IV.11 mostra que até o dia 20/07/2016, a medição da velocidade média do vento não foi restabelecida, resultando em 9 dias sem informações sobre essa variável. Esta ausência de dados pode prejudicar o monitoramento contínuo das condições de vento e do desempenho da turbina.

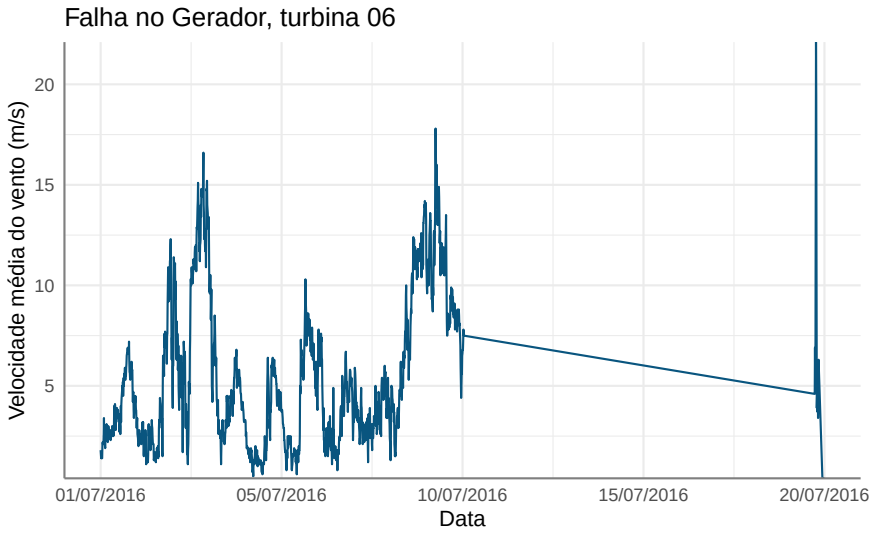


Figura IV.11: Falha Gerador T06

No dia 23/08/2016, houve uma falha no transformador da turbina T07. As Figuras IV.12 e IV.13 mostram, respectivamente, a potência média e a velocidade média da turbina. Na potência média, observa-se a falha iniciando um dia antes, em 22/08/2016, e a velocidade média caindo até próximo de zero no dia da falha, indicando uma perda significativa de desempenho da turbina.

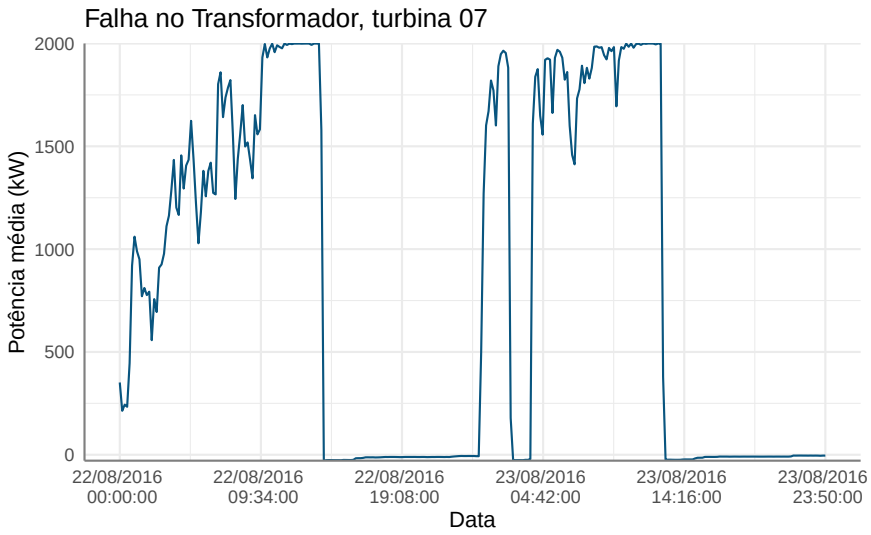


Figura IV.12: Falha Transformador T07

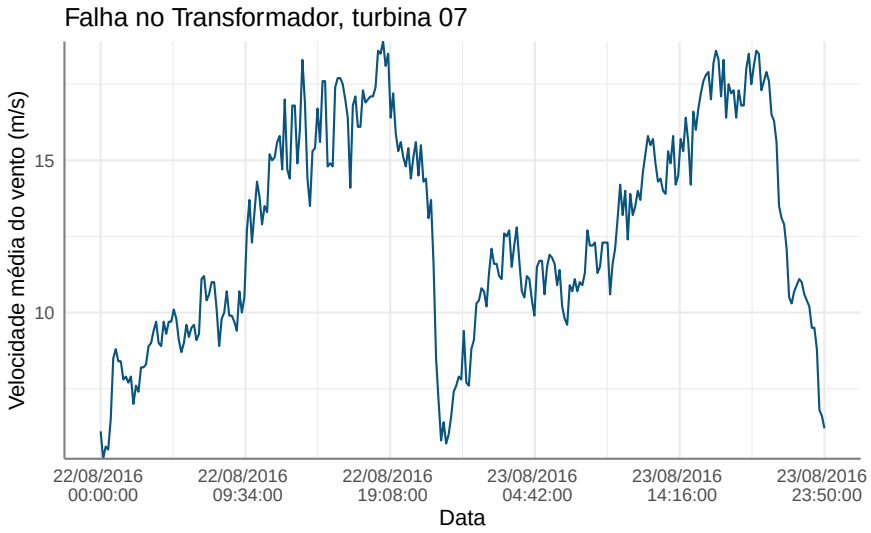


Figura IV.13: Falha Transformador T07

Na Figura IV.14, está a análise temporal das variáveis meteorológicas do conjunto de dados (*Metmast*), resumido pela média diária das observações. A partir das visualizações, pode-se observar que, das 40 variáveis meteorológicas, muitas apresentam variação nula ao longo do tempo. Além disso, nota-se uma interrupção nas medições de fevereiro de 2017 a abril de 2017.

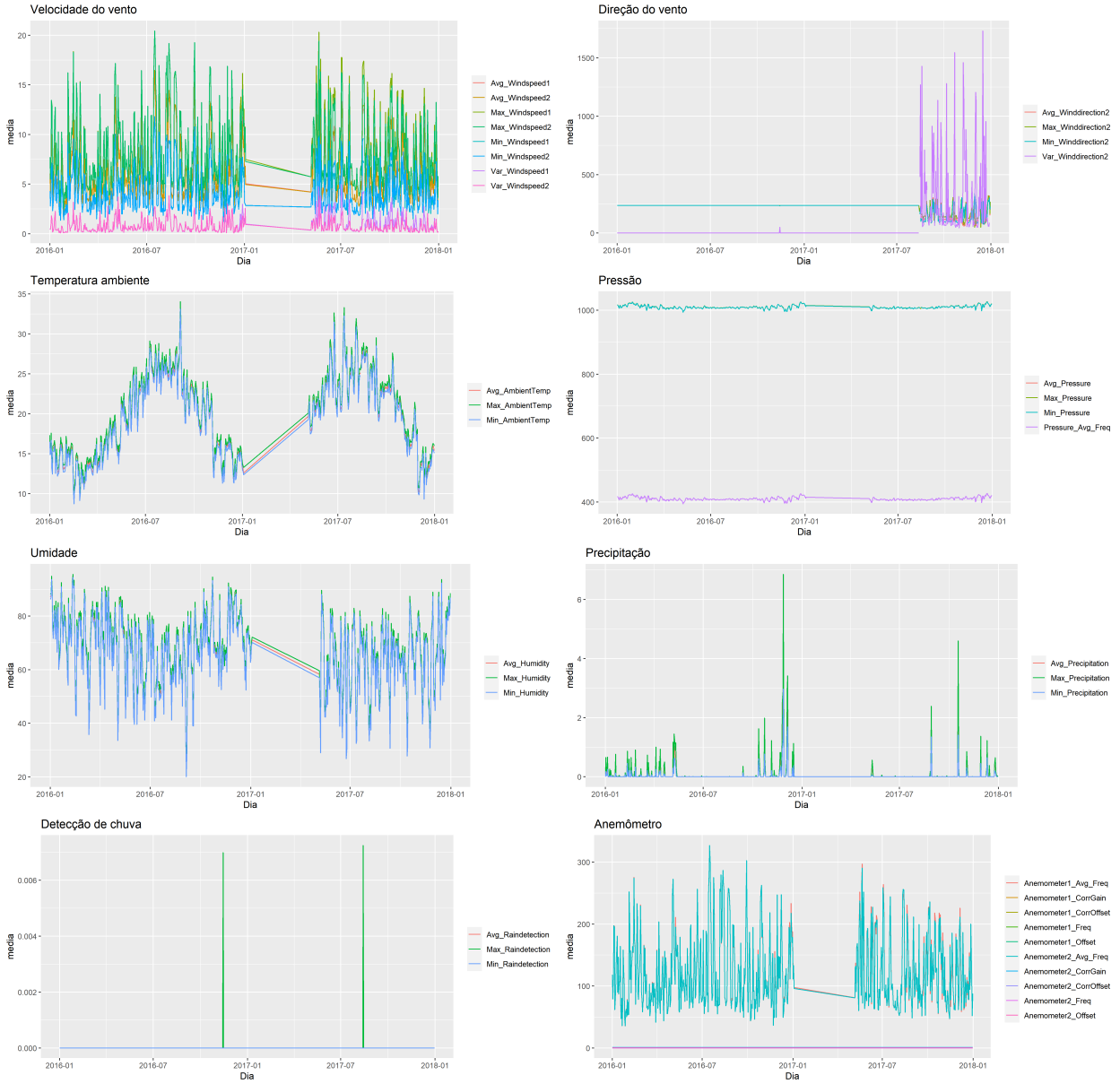


Figura IV.14: Análise temporal das variáveis meteorológicas

Na Figura IV.15, são apresentadas as 15 correlações cruzadas mais relevantes, ou seja, a classificação das variáveis com as correlações mais altas obtidas em uma tabela cruzada. Nota-se a alta correlação entre as variáveis, já que muitas delas são semelhantes, diferenciando-se apenas pela média, máximo ou mínimo.

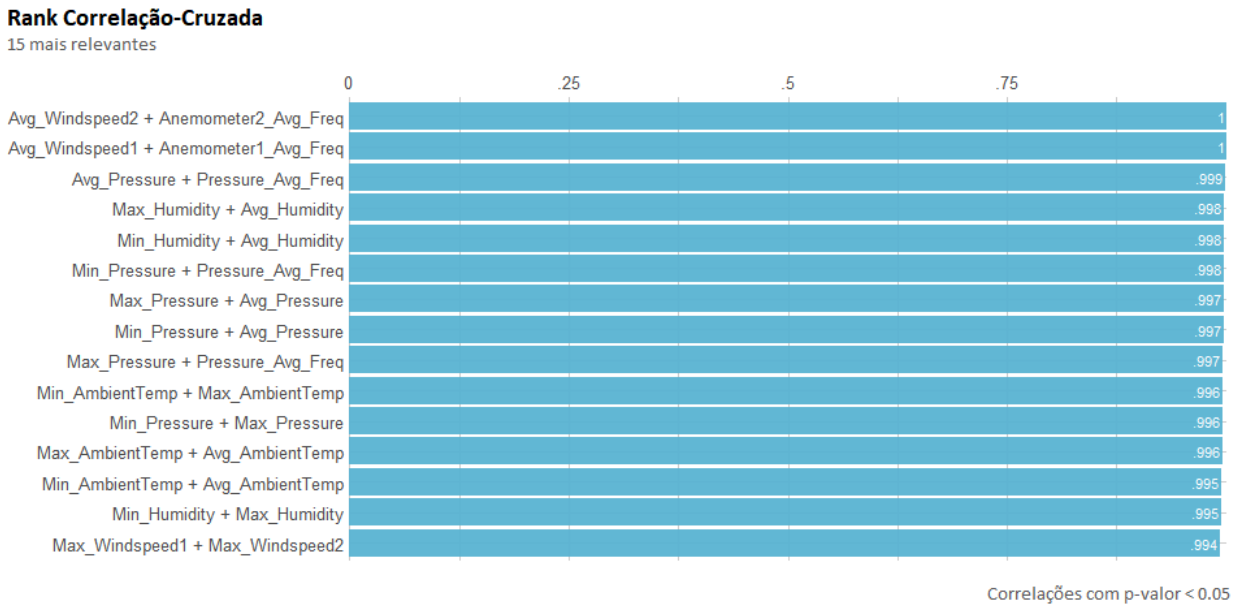


Figura IV.15: Rank das Correlações Cruzada do *Metmast*

No apêndice, está a descrição de cada variável dos conjuntos de dados *Signals* e *Metmast* do sistema SCADA.

Capítulo V Resultados

Neste capítulo são apresentados os resultados do processamento do conjunto de dados utilizando o referencial teórico discutido anteriormente. Este trabalho passou por diversas evoluções à medida que os resultados foram sendo gerados, sendo por isso organizado em diferentes fluxos de trabalho. O *pipeline* adotado em cada um dos fluxos, segue um ciclo centrado em dados, abrangendo as etapas de pré-processamento, engenharia de atributos, construção de modelos e avaliação de desempenho.

A Figura V.1 ilustra as quatro etapas principais dos *pipelines* utilizados. O processo inicia-se com o pré-processamento dos dados, que abrange a imputação de dados faltantes, criação de rótulos, normalização e o tratamento de dados desbalanceados. Em seguida, a engenharia de atributos identifica as variáveis mais relevantes para o problema. Posteriormente, na etapa de construção de modelos, são aplicados algoritmos de aprendizado de máquina para a tarefa de classificação, incluindo o ajuste de hiperparâmetros para otimização dos resultados. Finalmente, os modelos são avaliados com base em métricas de desempenho no conjunto de teste, permitindo a comparação entre diferentes abordagens.

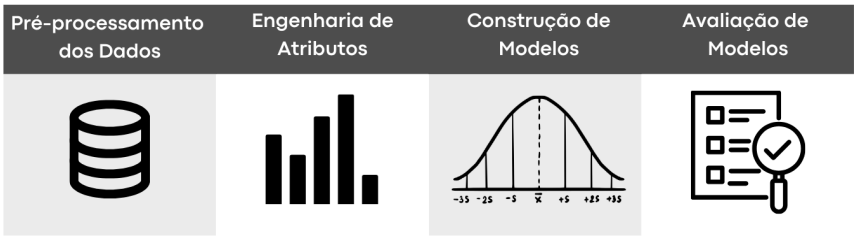


Figura V.1: *Pipeline* da metodologia de aprendizado de máquina adotada.

Os experimentos dos fluxos foram realizados por meio de rotinas computacionais implementadas em Python versão 3, em uma máquina Intel(R) Xeon(R) Gold 5120 CPU 2.20GHz, com 28 núcleos e 192GB de memória RAM. As bibliotecas utilizadas foram *pandas*[1], *numpy*[2] e *scikit-learn*[3].

V.1 Fluxo de Trabalho 1 - Análise das Variáveis do Mastro Meteorológico

Com o propósito de explorar e entender melhor os dados do mastro meteorológico, foi realizada uma Análise de Componentes Principais (PCA). Esse método foi utilizado para reduzir a dimensio-

[1]https://pandas.pydata.org/
[2]https://numpy.org/
[3]https://scikit-learn.org/stable/

nalidade dos dados, facilitando a interpretação das análises e permitindo uma melhor compreensão do comportamento dos dados.

Foram utilizadas as informações de velocidade e direção do vento, temperatura ambiente, pressão, umidade, precipitação, detecção de chuva e do anemômetro. As variáveis que continham apenas um único valor ao longo do tempo, identificadas como "*Offset*", foram desconsideradas da base de dados. Os dados foram normalizados para eliminar a discrepância das unidades de medida entre as variáveis. A seleção do número de componentes principais foi baseada no critério da análise gráfica e no percentual desejado de perda mínima de informação.

A Tabela V.1 apresenta as dez primeiras componentes geradas pelo PCA com o seu respectivo percentual de variância explicada e acumulada. As seis primeiras componentes possuem autovalores maiores que 1 e explicam aproximadamente 85% da variância dos dados. Ou seja, é possível reduzir a dimensionalidade de 28 para 6 componentes, "perdendo" cerca de 15% da variância total.

Tabela V.1: Autovalores e Variância das 10 primeiras componentes.

Componentes	Autovalor	% Variância	% Variância Acumulada
1	9,30	33,20	33,20
2	4,74	16,94	50,14
3	3,99	14,24	64,38
4	2,56	9,16	73,54
5	2,08	7,41	80,95
6	1,21	4,33	85,28
7	1,02	3,64	88,92
8	1,00	3,57	92,48
9	0,93	3,34	95,82
10	0,45	1,60	97,42

A Figura V.2 mostra o PCA em um espaço bidimensional definido pelos dois primeiros componentes principais, PC1 e PC2. Estes componentes explicam, respectivamente, 33,20% e 16,94% da variância total dos dados, totalizando mais de 50% da variância explicada apenas com esses dois componentes, o que é uma boa indicação de que a maioria da informação contida nos dados originais está capturada nesta projeção bidimensional. A coloração por *cos2* (o quadrado do cosseno) mede a importância dos componentes principais para a descrição de uma observação específica e ajuda a identificar quais observações são bem representadas pelos componentes principais e quais podem necessitar de uma maior dimensionalidade para uma representação precisa.

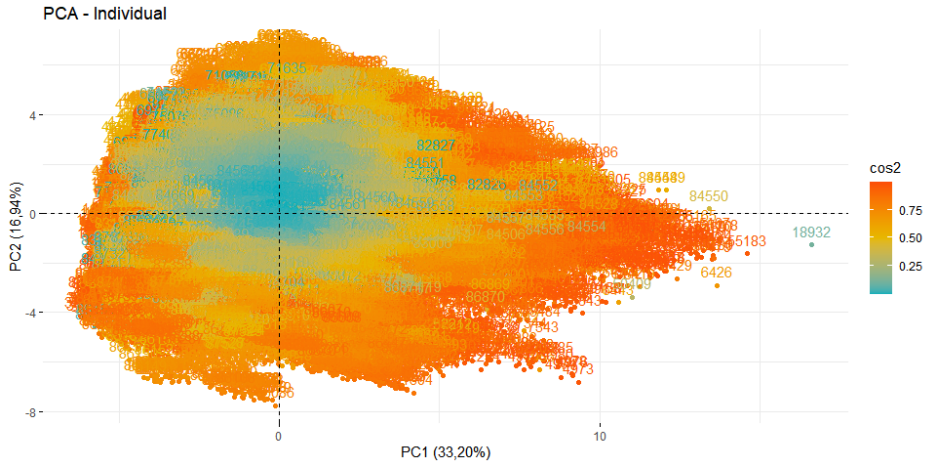


Figura V.2: *Scatter Plot* das 2 primeiras Componentes Principais.

Para entender a importância de cada variável na construção das componentes, são apresentados na Tabela V.2 os coeficientes de ponderação de cada característica. Na primeira componente principal, destacaram-se as variáveis de Velocidade do Vento e Anemômetro, podendo ser chamada de componente de indicador do Vento. A segunda componente principal pode ser chamada de componente da Temperatura Ambiente, a terceira de Umidade, a quarta de Precipitação, a quinta componente ficou com a Direção do Vento e, por último, na sexta componente, destacaram-se conjuntamente as variáveis de Temperatura Ambiente e Umidade.

Tabela V.2: Autovetores das 6 primeiras componentes selecionadas.

Variáveis	Coeficiente de Ponderação					
	PC1	PC2	PC3	PC4	PC5	PC6
Min_Windspeed1	**0,29**	-0,12	0,07	-0,06	0,03	-0,02
Max_Windspeed1	**0,32**	-0,09	0,04	-0,01	-0,02	0,01
Avg_Windspeed1	**0,31**	-0,11	0,05	-0,03	0,00	0,00
Var_Windspeed1	**0,25**	-0,04	0,02	0,05	-0,06	0,03
Min_Windspeed2	**0,29**	-0,12	0,03	-0,04	-0,01	0,00
Max_Windspeed2	**0,32**	-0,09	0,05	-0,02	0,00	0,00
Avg_Windspeed2	**0,32**	-0,10	0,04	-0,03	-0,01	0,00
Var_Windspeed2	**0,23**	-0,03	0,09	0,02	0,00	-0,02
Min_Winddirection2	-0,02	0,05	0,17	-0,20	**0,47**	-0,14
Max_Winddirection2	0,00	0,08	0,18	-0,22	**0,49**	0,10
Avg_Winddirection2	-0,01	0,07	0,20	-0,23	**0,52**	0,03
Var_Winddirection2	-0,02	0,01	-0,04	0,04	-0,04	-0,08
Min_AmbientTemp	0,13	**0,31**	-0,20	0,04	0,05	**0,41**
Max_AmbientTemp	0,13	**0,31**	-0,20	0,04	0,05	**0,41**
Avg_AmbientTemp	0,13	**0,31**	-0,20	0,04	0,05	**0,41**
Min_Pressure	-0,07	-0,36	-0,25	0,06	0,17	0,15
Max_Pressure	-0,07	-0,36	-0,25	0,06	0,16	0,14
Avg_Pressure	-0,07	-0,36	-0,25	0,06	0,17	0,15
Min_Humidity	-0,12	-0,17	**0,35**	-0,11	-0,16	**0,36**
Max_Humidity	-0,12	-0,16	**0,35**	-0,10	-0,16	**0,35**
Avg_Humidity	-0,12	-0,16	**0,35**	-0,10	-0,16	**0,35**
Min_Precipitation	0,01	0,01	0,20	**0,51**	0,15	0,02
Max_Precipitation	0,01	0,01	0,22	**0,51**	0,15	0,02
Avg_Precipitation	0,01	0,01	0,21	**0,52**	0,15	0,02
Max_Raindetection	0,00	0,00	0,00	0,00	-0,01	0,02
Anemometer1_Avg_Freq	**0,31**	-0,11	0,05	-0,03	0,00	0,00
Anemometer2_Avg_Freq	**0,32**	-0,10	0,04	-0,03	-0,01	0,00
Pressure_Avg_Freq	-0,07	-0,36	-0,25	0,06	0,17	0,15

Com base na interpretação das componentes, pode-se observar que as variáveis Velocidade do Vento e Anemômetro, associadas à primeira componente, representam a maior variância dos dados e demonstram um grande potencial de contribuição para o estudo de falhas em turbinas. Dessa forma, a informação contida nas variáveis originais pode ser substituída pela informação presente nas 6 componentes principais não correlacionadas, que explicam 85% da variância total. Isso resulta em uma economia de recursos para trabalhos futuros que utilizem essa mesma base de dados, sem perda significativa de informação.

A análise exploratória dos dados e os resultados obtidos neste fluxo de trabalho foram publicados na Escola Regional de Informática do Rio de Janeiro (ERI) em 2021. Além disso, um resumo deste trabalho foi publicado no VII Congresso Ibero-Americano de Empreendedorismo, Energia, Ambiente e Tecnologia (CIEEMAT) em 2022.

V.2 Fluxo de Trabalho 2 - Comparação de Algoritmos de Aprendizado de Máquina

Neste segundo fluxo de trabalho é realizado uma comparação de três técnicas de aprendizado de máquina supervisionado seguindo uma abordagem centrada em dados para a detecção de falhas em turbinas eólicas extraídos do SCADA.

O interesse está no problema de classificação binária para reconhecer as falhas e as operações sem falhas de uma turbina eólica. O algoritmo de classificação binária lida apenas com duas classes, 0 ou 1. A classe 0 indica uma observação sem falhas (saudável) e a classe 1 indica observações com falhas (defeituosa).

A metodologia adotada consiste em 3 etapas principais, ilustrada na Figura V.3: Pré-processamento de dados, extração de características e treinamento e avaliação dos algoritmos de aprendizado de máquina supervisionado: *Naive Bayes* (NB), Regressão Logística (RL) e K-Vizinhos mais Próximos (KNN).

Figura V.3: *Pipeline* da metodologia adotada no Fluxo de Trabalho 2.

Os dados utilizados compreendem o conjunto de dados dos sensores (*Signals*) e meteorológicos (*Metmast*), incluindo a rotulação das falhas, conforme descrito no Capítulo IV. Como discutido no fluxo de trabalho 1, Seção V.1, muitas variáveis contém informações de valor mínimo, máximo, médio e variância para cada tempo de medição, apresentando alta correlação entre si. Para remover atributos redundantes, foram selecionados apenas os valores médios de cada variável no conjunto de dados, resultando em um total de 60 características.

A etapa de pré-processamento também incluiu a normalização dos dados numéricos, a fim de eliminar a discrepância nas unidades de medida entre as variáveis. Na extração de atributos, a análise de componentes principais (PCA) foi aplicada para eliminar a alta correlação e reduzir a dimensionalidade dos dados multivariados com perda mínima de informação. A seleção do número de componentes principais foi definida mantendo 98% da variância dos dados.

Para a etapa de treinamento do modelo de aprendizado de máquina, os classificadores utilizados foram *Naive Bayes* (NB), Regressão Logística (RL) e K-vizinhos mais próximos (KNN), como uma alternativa aos resultados encontrados em Garan et al. [2022], que aplicaram apenas Árvores de

Decisão (). A base de dados foi dividida em 80% para treinamento e 20% para teste, mantendo a ordem do conjunto de dados. Para chegar ao modelo mais otimizado, foi utilizado o método *Grid Search* com validação cruzada de cinco *folds* para definir os hiperparâmetros.

A Tabela V.3 apresenta as métricas de desempenho dos modelos na base de teste: Acurácia, Precisão, Revocação e F_1-*Score*, além da combinação de valores dos hiperparâmetros de cada modelo que resultou no maior F_1-*Score* e a quantidade de componentes principais selecionada para cada componente da turbina eólica.

Tabela V.3: Métricas dos modelos das componentes da Turbina na base de teste.

Caixa de Velocidade

Modelo	Acurácia	Precisão	Revocação	F_1-*Score*	Hiperparâmetro
Naive Bayes	43,89%	0,03%	0,02%	0,03%	-
Regressão Logística	69,35%	61,80%	21,72%	**32,14%**	penalty='none', solver='newton-cg'
KNN	60,14%	8,93%	2,09%	3,39%	n_neighbors=1, weights='distance'
Número de Componentes Principais: 21					

Rolamento do Gerador

Modelo	Acurácia	Precisão	Revocação	F_1-*Score*	Hiperparâmetro
Naive Bayes	68,60%	73,55%	9,67%	17,09%	-
Regressão Logística	67,90%	62,38%	10,34%	17,73%	penalty='none', solver='lbfgs'
KNN	63,84%	39,35%	14,80%	**21,51%**	n_neighbors=1, weights='distance'
Número de Componentes Principais: 18					

Transformador

Modelo	Acurácia	Precisão	Revocação	F_1-*Score*	Hiperparâmetro
Naive Bayes	76,03%	68,54%	34,94%	**46,29%**	-
Regressão Logística	73,86%	85,56%	13,92%	23,95%	penalty='none', solver='sag'
KNN	70,28%	41,90%	1,38%	2,67%	n_neighbors=9, weights='distance'
Número de Componentes Principais: 19					

Gerador

Modelo	Acurácia	Precisão	Revocação	F_1-*Score*	Hiperparâmetro
Naive Bayes	73,15%	91,42%	3,26%	6,30%	-
Regressão Logística	74,81%	96,99%	9,24%	16,88%	penalty='l2', solver='lbfgs'
KNN	75,69%	66,45%	24,49%	**35,79%**	n_neighbors=1, weights='distance'
Número de Componentes Principais: 18					

Grupo Hidráulico

Modelo	Acurácia	Precisão	Revocação	F_1-*Score*	Hiperparâmetro
Naive Bayes	54,79%	34,03%	8,70%	13,86%	-
Regressão Logística	62,05%	95,86%	9,62%	**17,49%**	penalty='l1', solver='liblinear'
KNN	57,25%	38,06%	3,61%	6,60%	n_neighbors=1, weights='distance'
Número de Componentes Principais: 21					

A escolha do melhor algoritmo testado para cada componente da turbina eólica foi baseada na métrica F_1-*Score*. Os resultados obtidos neste trabalho são comparados com os resultados do artigo *benchmark* de Garan et al. [2022], que também utilizaram um abordagem centrada em dados comparando diferentes técnicas de seleção de atributos para cada componente da turbina com apenas

o classificador de Árvore de Decisão.

Na Tabela V.4 observa-se que para os componentes Transformador e Gerador da turbina eólica, os resultados obtidos superaram ao do estudo de caso, enquanto que para os demais componentes o F_1-*Score* foi menor. A metodologia adotada forneceu resultados em que outros classificadores mais simples são capazes de detectar as falhas dos componentes.

Tabela V.4: Comparação dos resultados com o *benchmark*.

	Benchmark	***Resultado***	
Componente	F_1-*Score*	Modelo Escolhido	F_1-*Score*
Caixa de Velocidade	**37,73%**	Regressão Logística	32,14%
Rolamento do Gerador	**36,29%**	KNN	21,51%
Transformador	8,08%	*Naive Bayes*	**46,29%**
Gerador	9,59%	KNN	**35,79%**
Grupo Hidráulico	**44,85%**	Regressão Logística	17,49%

Os resultados obtidos neste segundo fluxo de trabalho foram publicados na 9ª Conferência Internacional de Ciência Computacional e Inteligência Computacional (CSCI) em 2022.

V.3 Fluxo de Trabalho 3 - Comparação dos Métodos de Otimização de Hiperparâmetros

Uma extensão do Fluxo de Trabalho 2 (Seção V.2) é apresentada nesta seção, comparando três técnicas para ajuste de hiperparâmetros em modelos de aprendizado de máquina supervisionado para detecção de falhas em turbinas eólicas. Esta comparação destaca a importância da otimização dos dados durante o treinamento do modelo, bem como a eficiência no tempo de processamento.

O *pipeline* adotado, ilustrado na Figura V.4, consiste nas seguintes etapas: Pré-processamento de dados, extração de características, treinamento dos algoritmos de aprendizado de máquina supervisionado com ajuste de hiperparâmetros e avaliação de desempenho do modelo.

Como pode ser observado, apenas a última etapa do *pipeline* difere do empregado no Fluxo de Trabalho 2, contendo a comparação dos métodos de otimização de hiperparâmetros e mais três classificadores de aprendizado de máquina: Árvore de Decisão, Floresta Aleatória e Máquina de Vetores de Suporte (SVM).

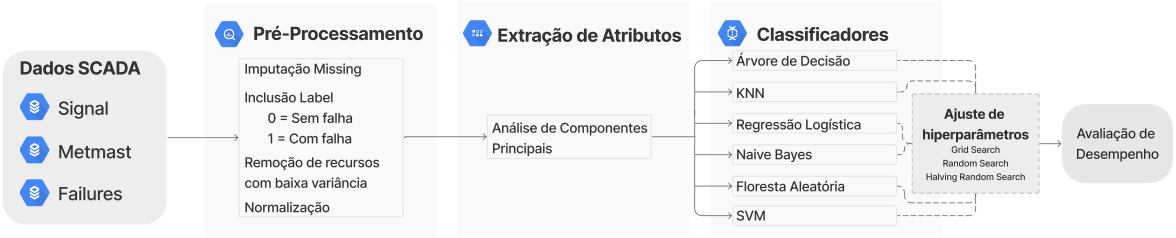


Figura V.4: *Pipeline* da metodologia adotada no Fluxo de Trabalho 3.

Na busca pela melhor combinação de hiperparâmetros, três técnicas de otimização foram avaliadas para medir o tempo de treinamento computacional dos modelos. A Busca em Grade (*Grid Search*) testa todas as possíveis combinações de hiperparâmetros, enquanto a Busca Aleatória (*Random Search*) testa combinações aleatórias de um intervalo de hiperparâmetros. Ao mesmo tempo, o *Halving Random Search* usa uma estratégia de busca aleatória que começa com poucos recursos e, iterativamente, seleciona os melhores candidatos, utilizando cada vez mais recursos.

Para chegar ao modelo mais otimizado, as técnicas de otimização de hiperparâmetros com validação cruzada de cinco dobras foram usadas para definir os hiperparâmetros com o melhor resultado pela métrica F_1-*Score*. O componente Transformador da turbina eólica foi utilizado para avaliar os métodos de otimização.

As Tabelas V.5-V.10 apresentam as métricas de desempenho dos modelos na base de teste: F_1-*Score*, AUC, Acurácia, Precisão, Revocação e Coeficiente de *Matthews* (MCC) para cada um dos classificadores, com os respectivos métodos de otimização de hiperparâmetros, além do tempo de treinamento e da combinação de valores dos hiperparâmetros de cada modelo que resultou no maior F_1-*Score*.

Em todos os modelos, o método *Halving* obteve um baixo custo computacional e um F_1-*Score* semelhante ao *Grid* e *Random Search*. Ou seja, não houve perda no desempenho preditivo do modelo. O modelo SVM com *Halving Search* foi o que mais economizou em processamento, com uma diferença de 9 horas em relação ao *Grid* e *Random Search*. Por outro lado, o *Naive Bayes* não é um método válido para melhorar o ajuste de hiperparâmetros, pois não há hiperparâmetros a serem ajustados da mesma maneira que em outros classificadores de aprendizado de máquina.

Tabela V.5: Resultado da Árvore de Decisão em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica.

Árvore de Decisão

Método	F_1-*Score*	AUC	Acurácia	Precisão	Revocação	MCC	Tempo (s)	Hiperparâmetro
Grid Search	5,06%	50,80%	70,48%	51,45%	2,66%	5,98%	24,15	criterion='entropy', max_depth=5
Random Search	6,19%	51,09%	70,64%	55,82%	3,28%	7,65%	21,78	criterion='log_loss', max_depth=10, max_features='log2'
Halving Random Search	4,91%	50,96%	70,74%	62,61%	2,56%	8,01%	5,55	criterion='log_loss', max_depth=10

Tabela V.6: Resultado do KNN em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica.

K-Vizinhos mais Próximos (KNN)

Método	F_1-*Score*	AUC	Acurácia	Precisão	Revocação	MCC	Tempo (s)	Hiperparâmetro
Grid Search	2,56%	50,27%	70,28%	41,76%	1,32%	2,60%	39,37	n_neighbors=9, weights='distance'
Random Search	2,56%	50,27%	70,28%	41,76%	1,32%	2,60%	63,29	n_neighbors=9, weights='distance'
Halving Random Search	2,56%	50,27%	70,28%	41,76%	1,32%	2,60%	0,59	n_neighbors=9, weights='distance'

Tabela V.7: Resultado da Regressão Logística em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica.

Regressão Logística

Método	F_1-*Score*	AUC	Acurácia	Precisão	Revocação	MCC	Tempo (s)	Hiperparâmetro
Grid Search	23,91%	56,46%	73,85%	85,54%	13,90%	27,56%	165,87	penalty=None, solver='newton-cg'
Random Search	23,91%	56,46%	73,85%	85,54%	13,90%	27,56%	496,40	penalty=None
Halving Random Search	23,95%	56,47%	73,86%	85,56%	13,92%	27,59%	35,58	solver='sag'

Tabela V.8: Resultado do *Naive Bayes* em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica.

Naive Bayes

Método	F_1-*Score*	AUC	Acurácia	Precisão	Revocação	MCC	Tempo (s)	Hiperparâmetro
Grid Search	46,29%	64,10%	76,03%	68,54%	34,94%	35,98%	0,11	-
Random Search	46,29%	64,10%	76,03%	68,54%	34,94%	35,98%	0,14	-
Halving Random Search	46,29%	64,10%	76,03%	68,54%	34,94%	35,98%	0,11	-

Tabela V.9: Resultado da Floresta Aleatória em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica.

Floresta Aleatória

Método	F_1-*Score*	AUC	Acurácia	Precisão	Revocação	MCC	Tempo (s)	Hiperparâmetro
Grid Search	0,71%	50,14%	70,48%	63,27%	0,36%	3,03%	1.950,96	criterion='log_loss', max_depth=10
Random Search	0,71%	50,12%	70,47%	57,41%	0,36%	2,63%	703,59	max_depth=10
Halving Random Search	0,76%	50,16%	70,51%	73,33%	0,38%	3,77%	106,89	criterion='entropy', max_depth=50, max_features='log2'

Tabela V.10: Resultado do SVM em cada método de otimização na base de teste para o componente Transformador da Turbina Eólica.

Máquina de Vetores de Suporte (SVM)

Método	F_1-*Score*	AUC	Acurácia	Precisão	Revocação	MCC	Tempo (s)	Hiperparâmetro
Grid Search	1,10%	50,28%	70,60%	97,96%	0,56%	6,14%	36.167,51	C=0.1
Random Search	1,10%	50,28%	70,60%	97,96%	0,56%	6,14%	33.406,22	C=0.1
Halving Random Search	0,85%	50,17%	70,51%	68,52%	0,43%	3,67%	781,48	C=100

Em Garan et al. [2022], o F_1-*Score* para o componente Transformador também não foi elevado, com um valor de 8,08%. A diferença é que eles utilizaram um modelo de árvore de decisão com um filtro de alta correlação na seleção de características. Uma possível explicação para os baixos valores das métricas obtidas é que, como os dados de falhas em turbinas eólicas se assemelham a eventos raros, pode ser necessário aplicar técnicas especiais para balancear as classes e, assim, melhorar as métricas.

Os resultados obtidos neste terceiro fluxo de trabalho foram apresentados na Conferência Internacional sobre Otimização, Algoritmos de Aprendizagem e Aplicações (OL2A) e publicado na série Comunicações em Ciência da Computação e Informação (CCIS) em 2023.

V.4 Fluxo de Trabalho 4 - Comparação das Técnicas de Seleção e Extração de Atributos

Neste fluxo de trabalho é feita uma comparação entre a técnica de seleção atributos Informação Mútua (MI) e a técnica de extração de atributos Análise de Componentes Principais (PCA). A MI mede a quantidade de informação que uma variável dá sobre a outra, ou seja, mede a dependência entre as variáveis. O PCA é usado para eliminar a alta correlação e reduzir a dimensionalidade dos dados multivariados com perda mínima de informação.

A Figura V.5 ilustra a metodologia adotada: coleta e análise da base de dados, pré-processamento

dos dados, seleção e extração de atributos, treinamento do algoritmo de AM supervisionado junto com o ajuste dos hiperparâmetros e a avaliação de desempenho do modelo.

Figura V.5: *Pipeline* da metodologia adotada no Fluxo de Trabalho 4.

Para a última etapa, utilizou-se os classificadores supervisionados: Regressão Logística (RL), K-Vizinhos mais Próximos (KNN), Árvore de Decisão (AD), Floresta Aleatória (FA) e Máquinas de Vetores de Suporte (SVM), que são alguns dos algoritmos mais usados na tarefa de classificação para aprender com dados de turbinas eólicas [Pandit et al., 2023]. Na busca pela melhor combinação dos hiperparâmetros, são avaliadas duas técnicas de otimização para medir o tempo computacional de treinamento dos modelos, o *Random Search* (RS) e *Halving Random Search* (HRS).

A Tabela V.11 apresenta as métricas de desempenho do melhor classificador na base de teste para cada otimizador com melhor técnica de seleção de atributos. Ela apresenta também o tempo computacional de treinamento e a combinação de valores dos hiperparâmetros de cada modelo com o maior F_1-*Score*. Em todos os modelos das componentes da turbina, o método *Halving* obteve um baixo custo computacional, com praticamente todas as métricas de desempenho ficando semelhantes ao *Random Search*. Para o componente Caixa de Velocidade, o uso do *Halving* reduziu 4 horas de processamento.

Tabela V.11: Resultado para cada otimizador e seleção de atributos na base de teste.

Componente	Modelo	Otimizador	Seleção de Atributos	F_1-*Score*	AUC	Acurácia	Precisão	Revocação	MCC	Tempo Computacional	Hiperparâmetro
Caixa de Velocidade	RL	HRS	PCA	32,2%	57,6%	69,4%	62,3%	21,7%	22,2%	0 hrs 8 min 41 sec	penalty='l1', solver='liblinear'
	RL	RS	PCA	32,2%	57,5%	69,4%	62,2%	21,7%	22,2%	4 hrs 37 min 58 sec	penalty=None, solver='newton-cholesky'
Rolamento do Gerador	RL	HRS	MI	16,7%	52,7%	66,8%	52,5%	9,9%	10,5%	0 hrs 0 min 26 sec	penalty='l1', solver='liblinear'
	RL	RS	MI	18,7%	53,3%	67,2%	55,0%	11,3%	12,4%	0 hrs 10 min 46 sec	penalty=None, solver='newton-cholesky'
Transformador	RL	HRS	PCA	23,9%	56,5%	73,9%	85,6%	13,9%	27,6%	0 hrs 0 min 7 sec	solver='sag'
	RL	RS	PCA	23,9%	56,5%	73,9%	85,5%	13,9%	27,6%	1 hr 31 min 37 sec	penalty=None
Gerador	AD	HRS	MI	31,1%	56,4%	71,1%	45,7%	23,5%	16,4%	0 hrs 0 min 1 sec	criterion='entropy', max_depth=100, max_features='log2'
	AD	RS	MI	39,3%	60,6%	74,1%	55,8%	30,4%	26,5%	0 hrs 1 min 10 sec	max_depth=10, max_features='sqrt'
Grupo Hidráulico	RL	HRS	PCA	17,5%	54,7%	62,0%	95,9%	9,6%	22,9%	0 hrs 0 min 17 sec	penalty='l1', solver='liblinear'
	RL	RS	PCA	16,9%	54,5%	61,9%	95,7%	9,3%	22,5%	0 hrs 20 min 20 sec	penalty='l1', solver='saga'

Para os componentes da turbina eólica Transformador e Gerador, a abordagem apresentada neste fluxo obteve resultados superiores aos do estudo de caso de Garan et al. [2022], enquanto que

para os demais componentes o F_1-*Score* foi menor, conforme pode ser observado na Tabela V.12.

Tabela V.12: Comparação da métrica F_1-*Score* dos resultados para o *benchmark*.

Componente	***Benchmark***	**Resultado**
Caixa de Velocidade	37,7%	32,2%
Rolamento do Gerador	36,3%	18,7%
Transformador	8,1%	**23,9%**
Gerador	9,6%	**39,3%**
Grupo Hidráulico	44,9%	16,9%

Os resultados obtidos neste quarto fluxo de trabalho foram apresentados e publicado no Simpósio Brasileiro de Bancos de Dados (SBBD) em 2023.

V.5 Fluxo de Trabalho 5 - Tratamento de Dados Desbalanceados

Uma extensão do Fluxo de Trabalho 4 (Seção V.4) é apresentada nesta seção, com foco no balanceamento entre as classes de estado saudável da turbina e estado de falha, visando equilibrar a distribuição entre essas duas classes. O balanceamento de classes compreendem a redução dos dados da classe majoritária ou a inclusão de dados da classe minoritária.

Este fluxo pode ser observado na Figura V.6, com a adição do SMOTE e da redução de dados da classes majoritária (*Undersampling*) no pré-processamento dos dados de treinamento do modelo. Na seleção e extração de atributos, foi considerada apenas a técnica de PCA e, para o ajuste de hiperparâmetros, foi utilizado exclusivamente o *Halving Random Search*.

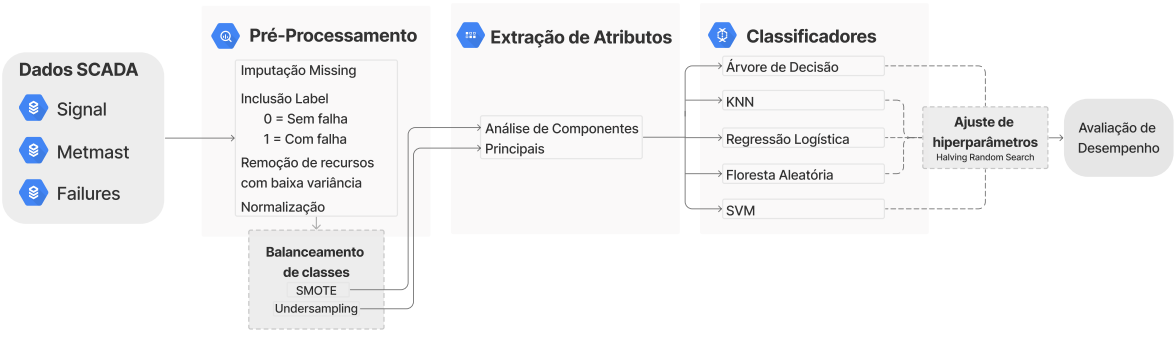


Figura V.6: *Pipeline* da metodologia adotada no Fluxo de Trabalho 5.

A Tabela V.13 apresenta as métricas de desempenho do melhor classificador na base de teste para cada uma das duas técnicas de balanceamento de classes, *Undersampling* e SMOTE, para cada componente da turbina. Além disso, a tabela mostra o tempo computacional de treinamento e a combinação dos valores dos hiperparâmetros de cada modelo com o maior F_1-*Score*. Em todos os componentes da turbina, a técnica SMOTE obteve desempenho preditivo superior ao *Undersam-*

pling.

Tabela V.13: Resultado para cada técnica de balanceamento de classes na base de teste.

Componente	Balanceamento de Classes	Seleção de Atributos	Modelo	F_1-*Score*	AUC	Acurácia	Precisão	Revocação	MCC	Tempo Computacional	Hiperparâmetros
Caixa de Velocidade	*Undersampling*	PCA	RL	22,8%	43,8%	51,1%	24,2%	21,6%	-12,8%	0 hrs 0 min 5 sec	penalty=None, solver='newton-cholesky'
	SMOTE	PCA	AD	**49,0%**	55,2%	49,3%	36,9%	73,0%	10,3%	0 hrs 0 min 9 sec	max_depth=5
Rolamento do Gerador	*Undersampling*	PCA	RL	41,0%	58,7%	66,7%	50,3%	34,6%	19,5%	0 hrs 0 min 3 sec	solver='newton-cholesky'
	SMOTE	PCA	RL	**47,7%**	59,4%	61,8%	44,0%	52,0%	18,1%	0 hrs 0 min 4 sec	penalty=None, solver='newton-cholesky'
Tansformador	*Undersampling*	PCA	RL	23,0%	55,9%	73,2%	75,8%	13,6%	24,0%	0 hrs 0 min 5 sec	penalty=None, solver='sag'
	SMOTE	PCA	SVM	**43,0%**	62,0%	73,4%	58,8%	33,9%	29,1%	5 hrs 32 min 17 sec	C=10, kernel='linear'
Gerador	*Undersampling*	PCA	SVM	46,4%	65,0%	80,2%	92,4%	31,0%	46,3%	4 hrs 32 min 40 sec	C=10, kernel='linear'
	SMOTE	PCA	SVM	**60,0%**	71,7%	82,2%	79,6%	48,2%	52,1%	33 hrs 44 min 11 sec	C=10, kernel='linear'
Grupo Hidráulico	*Undersampling*	PCA	AD	28,7%	52,0%	57,2%	47,2%	20,6%	5,2%	0 hrs 0 min 1 sec	criterion='entropy', max_depth=100, max_features='log2'
	SMOTE	PCA	RL	**39,4%**	50,9%	53,3%	43,1%	36,3%	1,9%	0 hrs 0 min 6 sec	solver='newton-cholesky'

V.6 Fluxo de Trabalho 6 - Classificador de uma única classe OCSVM

Como pode ser observado nos fluxos anteriores, apesar das tentativas de trabalhar com um classificador binário, o problema em questão é de ocorrência rara. Já que a ocorrência de falhas é significativamente menor em comparação ao estado saudável da turbina. Dessa forma, é essencial considerar técnicas que identifiquem padrões em dados que não se conformam ao comportamento comum.

No Fluxo de Trabalho 5, Seção V.5, foi abordado o balanceamento entre as classes, no entanto, esta técnica pode não ser a melhor alternativa, por alterarem o conjunto de dados inicial e distorcer a verdadeira distribuição dos dados. A subamostragem equilibra o conjunto de treinamento eliminando exemplos da classe majoritária, mas pode degradar o desempenho do classificador ao remover informações úteis. A sobreamostragem cria exemplos idênticos da classe minoritária para equilibrar o conjunto de treinamento, o que pode aumentar o tempo computacional e o risco de *overfitting* devido à criação de cópias idênticas da classe minoritária [Maalouf and Trafalis, 2011].

Como alternativa a esses métodos de balanceamento de classes, foi utilizado o algoritmo OCSVM (*One-Class Support Vector Machine*). Este algoritmo é um classificador de uma única classe que visa capturar características das instâncias de treinamento, a fim de ser capaz de distinguir potenciais *outliers* que possam aparecer. O OCSVM é particularmente útil em cenários de detecção de anomalias, onde o interesse principal está em identificar casos raros e excepcionais, como falhas na turbina.

A principal diferença em relação aos algoritmos supervisionados é que ao modelar o algoritmo de uma classe, o conjunto de dados de treinamento possui apenas exemplos da classe normal. A biblioteca *scikit-learn* fornece uma implementação de SVM de uma classe na função *OneClassSVM*.

A Tabela V.14 apresenta as métricas de desempenho do modelo OCSVM na base de teste: F_1-*Score*, Acurácia, Precisão e Revocação, sem otimização dos hiperparâmetros desse algoritmo. É possível observar um resultado superior do F_1-*Score* em todas as componentes da turbina eólica, em

relação aos modelos de classificação *Naive Bayes*, Regressão Logística e K-Vizinhos mais Próximos, Floresta Aleatória, SVM até mesmo da Árvore de Decisão que é o *benchmark*.

Tabela V.14: Métricas do OCSVM sem otimização de hiperparâmetros.

Componente	**F_1-*Score***	**Acurácia**	**Precisão**	**Revocação**
Caixa de Velocidade	62,66%	63,79%	99,97%	45,63%
Rolamento do Gerador	44,91%	44,33%	65,72%	34,10%
Transformador	58,70%	53,76%	79,14%	46,66%
Gerador	39,77%	39,80%	71,98%	27,48%
Grupo Hidráulico	55,93%	57,75%	71,17%	46,06%

Na Tabela V.15, também é apresentada as métricas de desempenho do modelo OCSVM na base de teste: F_1-*Score*, Acurácia, Precisão e Revocação, considerando agora a otimização dos hiperparâmetros desse algoritmo. Estes resultados ressaltam ainda mais a melhora do poder preditivo do OCSVM em comparação aos outros modelos supervisionados testados.

Tabela V.15: Métricas do OCSVM com otimização de hiperparâmetros.

Componente	**F_1-*Score***	**Acurácia**	**Precisão**	**Revocação**	**Hiperparâmetro**	**Tempo Computacional**
Caixa de Velocidade	79,84%	66,46%	66,56%	99,74%	{'nu': 0.001, 'kernel': 'rbf', 'gamma': 0.01}	1 hrs 35 min 11 sec
Rolamento do Gerador	79,77%	66,82%	67,11%	98,33%	{'nu': 0.001, 'kernel': 'rbf', 'gamma': 0.01}	1 hrs 08 min 16 sec
Transformador	82,66%	70,53%	70,58%	99,74%	{'nu': 0.001, 'kernel': 'rbf', 'gamma': 0.01}	0 hrs 27 min 52 sec
Gerador	64,12%	51,26%	68,57%	60,21%	{'nu': 0.01, 'kernel': 'linear', 'gamma': 0.1}	3 hrs 36 min 51 sec
Grupo hidráulico	81,34%	73,69%	69,27%	98,48%	{'nu': 0.001, 'kernel': 'rbf', 'gamma': 0.01}	30 hrs 56 min 17 sec

Os resultados do OCSVM sem otimização de hiperparâmetros foram publicados no Workshop Latino-Americano sobre Fusão de Informação (LAFUSION), focado em apenas uma componente da Caixa de Velocidade. Os resultados de todas as componentes foram publicados no XLII Congresso Nacional de Matemática Aplicada e Computacional (CNMAC), ambos no ano de 2023.

V.7 Sumário

O melhor classificador binário, considerando o F1-*Score* que é uma métrica valiosa para avaliar modelos de classificação, especialmente quando se lida com dados desbalanceados, entre os discutidos nos fluxos de trabalho, foi obtido no fluxo de trabalho 5. Nesse fluxo foi utilizada a técnica de balanceamento de classes SMOTE, o PCA foi aplicado para extração de atributos, e cinco algoritmos de aprendizado de máquina foram testados: Árvore de Decisão (AD), KNN, Regressão Logística (RL), Floresta Aleatória (FA) e SVM.

A Tabela V.16 apresenta os resultados do melhor classificador do fluxo de trabalho 5, comparando-os com os resultados obtidos com o OCSVM. Observa-se que o F1-*Score* do OCSVM são superiores. Este resultado demonstra que o OCSVM é mais eficaz em capturar a variabilidade e complexidade dos dados, proporcionando melhor desempenho na classificação de falhas em componentes de turbinas eólicas.

Tabela V.16: Comparação classificador binário e OCSVM

Componente	**F_1-*Score* Fluxo de Trab. 5 (SMOTE)**	**F_1-*Score* Fluxo de Trab. 6 (OCSVM)**
Caixa de Velocidade	49,00% - AD	79,84%
Rolamento do Gerador	47,70% - RL	79,77%
Transformador	43,00% - SVM	82,66%
Gerador	60,00% - SVM	64,12%
Grupo Hidráulico	39,40% - RL	81,34%

Capítulo VI Considerações Finais

A manutenção preditiva de máquinas que se desgastam com o tempo é crucial para melhorar a eficiência dos processos. As turbinas eólicas são sistemas complexos que demandam manutenção regular. Com o aumento da capacidade de aquisição de dados, surge a possibilidade de aplicar algoritmos de aprendizado de máquina, combinando abordagens centradas em dados para aprimorar a qualidade das informações que alimentam os modelos.

Neste trabalho, a detecção de falhas foi realizada em um conjunto de dados reais do sistema SCADA durante o monitoramento de turbinas eólicas. Uma análise exploratória dos dados foi o passo inicial para compreender a natureza dos dados e realizar o pré-processamento adequado. Essa análise revelou a baixa ocorrência do estado de falha em cada componente da turbina e a alta correlação entre as variáveis do conjunto de dados, apontando desafios significativos para a modelagem preditiva.

Para contornar esses desafios, uma série de seis fluxos de trabalho foi desenvolvida, com o objetivo de explorar, analisar e aprimorar os métodos de detecção de falhas em turbinas eólicas utilizando técnicas de aprendizado de máquina. Cada fluxo de trabalho apresentou uma abordagem distinta, contribuindo para um entendimento mais profundo e um refinamento contínuo das metodologias aplicadas.

No primeiro fluxo de trabalho, a Análise de Componentes Principais (PCA) foi utilizada para reduzir a dimensionalidade dos dados meteorológicos, permitindo uma compreensão mais clara das variáveis mais relevantes. Este fluxo inicial estabeleceu uma base sólida para as análises subsequentes, destacando a importância da redução de dimensionalidade em conjuntos de dados complexos.

O segundo fluxo de trabalho comparou três técnicas de aprendizado de máquina supervisionado, enfatizando a importância da extração de características e do ajuste de hiperparâmetros por pesquisa em grade (*Grid Search*) para a detecção de falhas. Embora os resultados tenham mostrado desempenho inferior ao modelo utilizado como *benchmark*, dois componentes da turbina apresentaram classificadores com desempenho superior, destacando a necessidade de experimentação e ajuste fino dos modelos para cada contexto específico.

No terceiro fluxo de trabalho, a otimização de hiperparâmetros foi abordada com uma comparação entre as técnicas de *Grid Search*, *Random Search* e *Halving Random Search*. Os resultados demonstraram que o *Halving Random Search* pode alcançar desempenho comparável aos outros

métodos com um custo computacional significativamente reduzido, o que é crucial em aplicações onde os recursos computacionais são limitados.

O quarto fluxo de trabalho introduziu uma comparação entre a seleção de atributos baseada em Informação Mútua (MI) e a extração de atributos via PCA. Esta análise mostrou que, dependendo do componente da turbina eólica, uma técnica pode superar a outra em termos de precisão e eficiência, destacando a importância de escolher a técnica adequada para cada tipo de dado.

No quinto fluxo de trabalho, foram exploradas técnicas de balanceamento de classes, como *SMOTE* e *Undersampling*, para melhorar a detecção de falhas em um cenário de dados desbalanceados. Os resultados indicaram que o *SMOTE* é uma técnica mais eficaz para lidar com o desbalanceamento de classes, proporcionando um desempenho preditivo superior em comparação ao *Undersampling*.

Finalmente, o sexto fluxo de trabalho abordou a detecção de falhas raras utilizando o algoritmo *One-Class SVM* (OCSVM). Este fluxo demonstrou que, em cenários onde as falhas são eventos raros, o OCSVM oferece uma solução robusta, superando os classificadores binários tradicionais na tarefa de identificar anomalias.

Como próximos passos, o objetivo é testar outras técnicas de redução de dimensionalidade e métodos de sobreamostragem para balanceamento entre as classes, como o descrito por Moniz et al. [2016], que considera a dependência temporal dos dados e pode fornecer soluções capazes de melhorar significativamente a precisão preditiva de casos raros em tarefas de previsão usando dados de séries temporais desbalanceadas.

Além disso, uma área promissora para trabalhos futuros é verificar a influência do *concept drift*, ou deriva de conceito, nos dados analisados. Em ambientes de dados dinâmicos, as propriedades estatísticas das variáveis podem mudar ao longo do tempo, resultando na necessidade de adaptação dos modelos de aprendizado de máquina. Portanto, validar se os padrões dos dados SCADA mudam rapidamente, é essencial para garantir a eficácia dos modelos de aprendizado de máquina.

Em resumo, o trabalho realizado nesta dissertação representa um passo significativo na aplicação de técnicas de aprendizado de máquina na indústria de energia eólica. A implementação prática dos métodos discutidos pode trazer melhorias consideráveis na manutenção preditiva de turbinas eólicas, resultando em maior capacidade operacional e redução de custos.

Este estudo oferece uma base sólida das técnicas de aprendizado de máquina para futuras pesquisas e desenvolvimentos na área, com potencial para impactar positivamente a sustentabilidade e a eficiência das operações energéticas, garantindo a operação segura e eficiente de parques eólicos.

VI.1 Artigos Publicados

Além das contribuições mencionadas na seção anterior, durante o extenso processo de desenvolvimento das abordagens apresentadas neste trabalho, foram produzidos um total de 9 artigos e publicações, listados abaixo com suas respectivas descrições:

- Escola Regional de Informática do Rio de Janeiro (ERI), 2021 - Visualização de dados de turbinas eólicas baseado na Análise de Componentes Principais.
- *9th International Conference on Computational Science & Computational Intelligence* (CSCI), 2022 - *Fault Identification in Wind Turbines: A Data-Centric Machine Learning Approach.*
- VII Congresso Ibero-Americano de Empreendedorismo, Energia, Ambiente e Tecnologia (CIEEMAT), 2022 - *Wind turbine data visualization based on Principal Component Analysis.*
- XLII Congresso Nacional de Matemática Aplicada e Computacional (CNMAC), 2023 - Estudo sobre Modelos de Aprendizado de Máquina para Detecção de Falhas em Turbinas Eólicas (Qualis B4).
- *International Conference on Optimization, Learning Algorithms and Applications* (OL2A), 2023 - *Fault Classification of Wind Turbine: A Comparison of Hyperparameter Optimization Methods* (publicado em *Communications in Computer and Information Science* - CCIS, Qualis B2).
- Simpósio Brasileiro de Bancos de Dados (SBBD), 2023 - Identificação de Falhas em Turbinas Eólicas Utilizando Abordagens de Aprendizado de Máquina (Qualis A4).
- LV Sociedade Brasileira de Pesquisa Operacional (SBPO), 2023 - NSGA-2 para Seleção de Atributos na Detecção de Falhas em Turbinas Eólicas (Qualis A4).
- Workshop Latino-Americano sobre Fusão de Informação (LAFUSION), 2023 - *Wind turbine fault classification using SCADA and meteorological data fusion.*
- XVIII *Brazilian e-Science Workshop* (BreSci), 2024 - *Aplicação de Modelos Ocultos de Markov para Detecção de Falhas em Componentes de Turbinas Eólicas.*

Referências

ABEEólica. *Boletim Anual - Associação Brasileira de Energia Eólica (ABEEólica)*, 2021.

Agrawal, T. *Hyperparameter optimization in machine learning: make your machine learning and deep learning models more efficient*. Springer, 2021.

Ayman, M., Othman, M., Mahmoud, N., Tamer, Z., Sayed, M., and Hassan, Y. Fault detection in wind turbines using deep learning. *MIUCC 2022 - 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference*, pages 272–278, 2022.

Bilendo, F., Badihi, H., Lu, N., Cambron, P., and Jiang, B. An intelligent data-driven machine learning approach for fault detection of wind turbines. *2021 6th International Conference on Power and Renewable Energy, ICPRE 2021*, pages 444–449, 2021.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*. Springer, 2006.

Blanco-M, A., Sole-Casals, J., Marti-Puig, P., Justicia, J. J. C. I., and Cusido, J. Impact of target variable distribution type over the regression analysis in wind turbine data. *International Work Conference on Bio-Inspired Intelligence: Intelligent Systems for Biodiversity Conservation, IWOBI 2017 - Proceedings*, 2017.

Breiman, L. Random forests. *Machine learning*, 45(1):5–32, 2001.

Corley, B., Koukoura, S., Carroll, J., and McDonald, A. Combination of thermal modelling and machine learning approaches for fault detection in wind turbine gearboxes. *Energies*, 14, 2021.

Correa-jullian, C., Cofre-martel, S., Martin, G., Droguett, E., Leite, G., and Costa, A. Exploring quantum machine learning and feature reduction techniques for wind turbine pitch fault detection. *Energies*, 15, 2022.

Faceli, K., Lorena, A. C., Gama, J., Carvalho, A., et al. Inteligência artificial: Uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, 2:192, 2011.

Feng, B., Zhang, D., Si, Y., Tian, X., and Qian, P. A condition monitoring method of wind turbines based on long short-term memory neural network. *ICAC 2019 - 2019 25th IEEE International Conference on Automation and Computing*, 2019.

Gad, I. and Hassanien, A. A wind turbine fault identification using machine learning approach based on pigeon inspired optimizer. *Proceedings - 2021 IEEE 10th International Conference on Intelligent Computing and Information Systems, ICICIS 2021*, pages 231–235, 2021.

Garan, M., Tidriri, K., and Kovalenko, I. A data-centric machine learning methodology: Application on predictive maintenance of wind turbines. *Energies*, 15:826, 2022.

Grant, M. J. and Booth, A. A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal*, 26(2):91–108, 2009.

GWEC. *Global Wind Report - Global Wind Energy Council (GWEC)*, 2024.

Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M., and Hussain, A. Interpreting black-box models: a review on explainable artificial intelligence. *Cognitive Computation*, 16(1):45–74, 2024.

Helbing, G. and Ritter, M. Deep learning for fault detection in wind turbines. *Renewable and Sustainable Energy Reviews*, 98:189–198, 2018.

Hu, R., Leahy, K., Konstantakopoulos, I., Auslander, D., Spanos, C., and Agogino, A. Using domain knowledge features for wind turbine diagnostics. *Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016*, pages 300–305, 2017.

Japa, L. et al. A population-based hybrid approach for hyperparameter optimization of neural networks. *IEEE Access*, 11:50752–50768, 2023.

Letcher, T. *Wind energy engineering: A handbook for onshore and offshore wind turbines*. Elsevier, 2023.

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., Recht, B., and Talwalkar, A. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246, 2020.

Maalouf, M. and Trafalis, T. B. Rare events and imbalanced datasets: an overview. *International Journal of Data Mining, Modelling and Management*, 3(4):375–388, 2011.

Mammadov, E., Farrokhabadi, M., and Canizares, C. Ai-enabled predictive maintenance of wind generators. *Proceedings of 2021 IEEE PES Innovative Smart Grid Technologies Europe: Smart Grids: Toward a Carbon-Free Future, ISGT Europe 2021*, 2021.

Marsland, S. *Machine learning: an algorithmic perspective*. CRC press, 2015.

Marti-Puig, P., Blanco-M, A., Cárdenas, J., Cusidó, J., and Solé-Casals, J. Effects of the pre-processing algorithms in fault diagnosis of wind turbines. *Environmental Modelling and Software*, pages 119–128, 2018.

Mendes, M., Menezes, D., Almeida, J. A., and Farinha, J. Wind farm and resource datasets: A comprehensive survey and overview. *Energies*, 13, 2020.

Mingoti, S. A. Análise de dados através de métodos estatística multivariada: uma abordagem aplicada. In *Análise de dados através de métodos estatística multivariada: uma abordagem aplicada*, page 295. Editora UFMG, 2007.

Mitchell, T. M. *Machine learning.* McGraw-hill New York, 1997.

Moniz, N., Branco, P., and Torgo, L. Resampling strategies for imbalanced time series. pages 282–291, 2016.

Pandit, R. et al. Scada data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future trends. *Wind Engineering*, 47(2):422–441, 2023.

Pandit, R. and Wang, J. A comprehensive review on enhancing wind turbine applications with advanced scada data analytics and practical insights. *IET Renewable Power Generation*, 2024.

Perera, P., Oza, P., and Patel, V. M. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021.

Provost, F. and Kohavi, R. Glossary of terms. *Journal of Machine Learning*, 30(2-3):271–274, 1998.

Qin, S., Wang, K., Ma, X., Wang, W., and Li, M. Ensemble learning-based wind turbine fault prediction method with adaptive feature selection. *Communications in Computer and Information Science*, 728, 2017.

Rahimilarki, R., Gao, Z., Jin, N., and Zhang, A. Convolutional neural network fault classification based on time-series analysis for benchmark wind turbine machine. *Renewable Energy*, 185:916–931, 2022.

Ramentol, E., Caballero, Y., Bello, R., and Herrera, F. Smote-rs b*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. *Knowledge and information systems*, 33:245–265, 2012.

Raschka, S. Naive bayes and text classification i-introduction and theory. *arXiv preprint arXiv:1410.5329*, 2014.

Raschka, S. *Python machine learning.* Packt publishing ltd, 2015.

Rezamand, M., Kordestani, M., Carriveau, R., Ting, D. S.-K., Orchard, M. E., and Saif, M. Critical wind turbine components prognostics: A comprehensive review. *IEEE Transactions on Instrumentation and Measurement*, 69(12):9306–9328, 2020.

Russell, S. J. and Norvig, P. *Artificial intelligence a modern approach*. London, 2010.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., and Aroyo, L. M. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., and Platt, J. Support vector method for novelty detection. *NIPS*, 12:582–588, 1999.

Soper, D. S. Hyperparameter optimization using successive halving with greedy cross validation. *Algorithms*, 16(1), 2023.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J., and Nenadic, G. Machine learning methods for wind turbine condition monitoring: A review. *Renewable energy*, 133:620–635, 2019.

Suthaharan, S. Machine learning models and algorithms for big data classification. *Integr. Ser. Inf. Syst*, 36:1–12, 2016.

Trizoglou, P., Liu, X., and Lin, Z. Fault detection by an ensemble framework of extreme gradient boosting (xgboost) in the operation of offshore wind turbines. *Renewable Energy*, 179:945–962, 2021.

Velandia-Cardenas, C., Vidal, Y., and Pozo, F. Wind turbine fault detection using highly imbalanced real scada data. *Energies*, 14, 2021.

Vergara, J. R. and Estévez, P. A. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24:175–186, 2014.

Waqas Khan, P. and Byun, Y.-C. Multi-fault detection and classification of wind turbines using stacking classifier. *Sensors*, 22(18), 2022.

Witten, D. and James, G. *An introduction to statistical learning with applications in R*. springer publication, 2013.

Yi, H., Jiang, Q., Yan, X., and Wang, B. Imbalanced classification based on minority clustering synthetic minority oversampling technique with wind turbine fault detection application. *IEEE Transactions on Industrial Informatics*, 17:5867–5875, 2021.

Zhang, G. and Li, Y. A fault diagnosis methodfor wind turbines of new wind farm based on joint matching adaptive network. *Proceedings of 2021 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control, SDPC 2021*, pages 240–245, 2021.

Zhang, G., Li, Y., Jiang, W., and Shu, L. A fault diagnosis method for wind turbines with limited labeled data based on balanced joint adaptive network. *Neurocomputing*, 481:133–153, 2022.

Capítulo VII Apêndice A

Abaixo está a descrição de cada variável dos conjuntos de dados *Metmast* e *Signals* usadas no sistema SCADA.

Variável	**Descrição**
TimeStamp	Data e hora da medição
Min_Windspeed1 [m/s]	Velocidade mínima do vento - sensor 1
Max_Windspeed1 [m/s]	Velocidade máxima do vento - sensor 1
Avg_Windspeed1 [m/s]	Velocidade média do vento - sensor 1
Var_Windspeed1 [m/s]	Variação da velocidade do vento - sensor 1
Min_Windspeed2 [m/s]	Velocidade mínima do vento - sensor 2
Max_Windspeed2 [m/s]	Velocidade máxima do vento - sensor 2
Avg_Windspeed2 [m/s]	Velocidade média do vento - sensor 2
Var_Windspeed2 [m/s]	Variação da velocidade do vento - sensor 2
Min_Winddirection2 [º]	Direção mínima do vento - sensor 2
Max_Winddirection2 [º]	Direção máxima do vento - sensor 2
Avg_Winddirection2 [º]	Direção média do vento - sensor 2
Var_Winddirection2 [º]	Variação da direção do vento - sensor 2
Min_AmbientTemp [ºC]	Temperatura ambiente mínima
Max_AmbientTemp [ºC]	Temperatura ambiente máxima
Avg_AmbientTemp [ºC]	Temperatura ambiente média
Min_Pressure [hPa]	Pressão mínima
Max_Pressure [hPa]	Pressão máxima
Avg_Pressure [hPa]	Pressão média
Min_Humidity [%]	Umidade mínima
Max_Humidity [%]	Umidade máxima
Avg_Humidity [%]	Umidade média
Min_Precipitation [mm]	Precipitação mínima
Max_Precipitation [mm]	Precipitação máxima
Avg_Precipitation [mm]	Precipitação média
Min_Raindetection	Detecção mínima de chuva
Max_Raindetection	Detecção máxima de chuva
Avg_Raindetection	Detecção média de chuva
Anemometer1_Freq [Hz]	Frequência de amostragem do anemômetro - sensor 1
Anemometer1_avg_Freq [Hz]	Taxa de amostragem média do anemômetro - sensor 1
Anemometer1_offset [m/s]	Erro de desvio do sensor do anemômetro - sensor 1
Anemometer1_corrGain	Fator de correção de ganho do anemômetro - sensor 1
Anemometer1_corrOffset [m/s]	Correção de desvio do anemômetro - sensor 1
Anemometer2_Freq [Hz]	Frequência de amostragem do anemômetro - sensor 2
Anemometer2_avg_Freq [Hz]	Taxa de amostragem média do anemômetro - sensor 2
Anemometer2_offset [m/s]	Erro de desvio do sensor do anemômetro - sensor 2
Anemometer2_corrGain	Fator de correção de ganho do anemômetro - sensor 2
Anemometer2_corrOffset [m/s]	Correção de desvio do anemômetro - sensor 2
AirPressureSensorZeroOffset [hPa]	Desvio zero do sensor de pressão
Pressure_avg_freq [Hz]	Taxa de amostragem média do sensor de pressão

Tabela VII.1: Descrição das variáveis do conjunto de dados *Metmast*

Variável	Descrição	Componente
Identificador de turbina	ID da turbina eólica	Geral
TimeStamp	Data e hora da medida	Geral
Gen_RPM_Max [rpm]	RPM máxima do gerador	Gerador
Gen_RPM_Min [rpm]	RPM mínima do gerador	Gerador
Gen_RPM_Avg [rpm]	RPM média do gerador	Gerador
Gen_RPM_Std [rpm]	Desvio padrão do RPM do gerador	Gerador
Gen_Bear_Temp_Avg [ºC]	Temp. média no rolamento do gerador 1	Gerador
Gen_Phase1_Temp_Avg [ºC]	Temp. média nos enrolamentos do estator fase 1	Gerador
Gen_Phase2_Temp_Avg [ºC]	Temp. média nos enrolamentos do estator fase 2	Gerador
Gen_Phase3_Temp_Avg [ºC]	Temp. média nos enrolamentos do estator fase 3	Gerador
Hyd_Oil_Temp_Avg [ºC]	Temp. média do óleo no grupo hidráulico	Hidráulico
Gear_Oil_Temp_Avg [ºC]	Temp. média do óleo na caixa de engrenagens	Caixa de Velocidade
Gear_Bear_Temp_Avg [ºC]	Temp. média no rolamento da caixa de engrenagens	Caixa de Velocidade
Nac_Temp_Avg [ºC]	Temp. média na nacele	Nacele
Rtr_RPM_Max [rpm]	RPM máxima do rotor	Rotor
Rtr_RPM_Min [rpm]	RPM mínima do rotor	Rotor
Rtr_RPM_Avg [rpm]	RPM média do rotor	Rotor
Amb_WindSpeed_Max [m/s]	Velocidade máxima do vento	Ambiente
Amb_WindSpeed_Min [m/s]	Velocidade mínima do vento	Ambiente
Amb_WindSpeed_Avg [m/s]	Velocidade média do vento	Ambiente
Amb_WindSpeed_Std [m/s]	Desvio padrão da velocidade do vento	Ambiente
Amb_WindDir_Relative_Avg [º]	Direção relativa média do vento	Ambiente
Amb_WindDir_Abs_Avg [º]	Direção absoluta média do vento	Ambiente
Amb_Temp_Avg [ºC]	Temp. média ambiente	Ambiente
Prod_LatestAvg_ActPwrGen0 [Wh]	Potência ativa - gerador desconectado	Produção
Prod_LatestAvg_ActPwrGen1 [Wh]	Potência ativa - gerador conectado em delta	Produção
Prod_LatestAvg_ActPwrGen2 [Wh]	Potência ativa - gerador conectado em estrela	Produção
Prod_LatestAvg_TotActPwr [Wh]	Potência ativa total	Produção
Prod_LatestAvg_ReactPwrGen0 [VArh]	Potência reativa - gerador desconectado	Produção
Prod_LatestAvg_ReactPwrGen1 [VArh]	Potência reativa - gerador conectado em delta	Produção
Prod_LatestAvg_ReactPwrGen2 [VArh]	Potência reativa - gerador conectado em estrela	Produção
Prod_LatestAvg_TotReactPwr [VArh]	Potência reativa total	Produção
HVTrafo_Phase1_Temp_Avg [ºC]	Temp. média no transformador de AT fase L1	Transformador
HVTrafo_Phase2_Temp_Avg [ºC]	Temp. média no transformador de AT fase L2	Transformador
HVTrafo_Phase3_Temp_Avg [ºC]	Temp. média no transformador de AT fase L3	Transformador
Grd_InverterPhase1_Temp_Avg [ºC]	Temp. média no inversor do lado da rede	Rede
Cont_Top_Temp_Avg [ºC]	Temp. média no controlador da nacele superior	Controlador
Cont_Hub_Temp_Avg [ºC]	Temp. média no controlador do hub	Controlador
Cont_VCP_Temp_Avg [ºC]	Temp. média na placa VCP	Controlador
Gen_SlipRing_Temp_Avg [ºC]	Temp. média na câmara do anel dividido	Gerador
Spin_Temp_Avg [ºC]	Temp. média no cone do nariz	Spinner
Blds_PitchAngle_Min [º]	Ângulo mínimo	Lâminas
Blds_PitchAngle_Max [º]	Ângulo máximo	Lâminas
Blds_PitchAngle_Avg [º]	Ângulo médio	Lâminas
Blds_PitchAngle_Std [º]	Desvio padrão do ângulo	Lâminas
Cont_VCP_ChokcoilTemp_Avg [ºC]	Temp. média nas bobinas de estrangulamento na seção VCS	Controlador
Grd_RtrInvPhase1_Temp_Avg [ºC]	Temp. média no inversor do lado do rotor 1	Rede
Grd_RtrInvPhase2_Temp_Avg [ºC]	Temp. média no inversor do lado do rotor 2	Rede
Grd_RtrInvPhase3_Temp_Avg [ºC]	Temp. média no inversor do lado do rotor 3	Rede
Cont_VCP_WtrTemp_Avg [ºC]	Temp. média na água de resfriamento do VCS	Controlador
Grd_Prod_Pwr_Avg [kW]	Potência média	Rede
Grd_Prod_CosPhi_Avg	Deslocamento de fase real médio	Rede
Grd_Prod_Freq_Avg [Hz]	Frequência média	Rede
Grd_Prod_VoltPhse1_Avg [V]	Tensão média na fase 1	Rede
Grd_Prod_VoltPhse2_Avg [V]	Tensão média na fase 2	Rede
Grd_Prod_VoltPhse3_Avg [V]	Tensão média na fase 3	Rede
Grd_Prod_CurPhse1_Avg [A]	Corrente média na fase 1	Rede
Grd_Prod_CurPhse2_Avg [A]	Corrente média na fase 2	Rede
Grd_Prod_CurPhse3_Avg [A]	Corrente média na fase 3	Rede
Grd_Prod_Pwr_Max [kW]	Potência máxima	Rede
Grd_Prod_Pwr_Min [kW]	Potência mínima	Rede
Grd_Busbar_Temp_Avg [ºC]	Temp. média na seção do barramento	Rede
Rtr_RPM_Std [rpm]	Desvio padrão do RPM do rotor	Rotor
Amb_WindSpeed_Est_Avg [m/s]	Velocidade média estimada do vento	Ambiente
Grd_Prod_Pwr_Std [kW]	Desvio padrão da potência	Rede
Grd_Prod_ReactPwr_Avg [kVAr]	Potência reativa média	Rede
Grd_Prod_ReactPwr_Max [kVAr]	Potência reativa máxima	Rede
Grd_Prod_ReactPwr_Min [kVAr]	Potência reativa mínima	Rede
Grd_Prod_ReactPwr_Std [kVAr]	Desvio padrão da potência reativa	Rede
Grd_Prod_PsblePwr_Avg [kW]	Potência ativa média possível	Rede
Grd_Prod_PsblePwr_Max [kW]	Potência ativa máxima possível	Rede
Grd_Prod_PsblePwr_Min [kW]	Potência ativa mínima possível	Rede
Grd_Prod_PsblePwr_Std [kW]	Desvio padrão da potência ativa possível	Rede
Grd_Prod_PsbleInd_Avg [kVAr]	Potência reativa indutiva média possível	Rede
Grd_Prod_PsbleInd_Max [kVAr]	Potência reativa indutiva máxima possível	Rede
Grd_Prod_PsbleInd_Min [kVAr]	Potência reativa indutiva mínima possível	Rede
Grd_Prod_PsbleInd_Std [kVAr]	Desvio padrão da potência reativa indutiva possível	Rede
Grd_Prod_PsbleCap_Avg [kVAr]	Potência reativa capacitiva média possível	Rede
Grd_Prod_PsbleCap_Max [kVAr]	Potência reativa capacitiva máxima possível	Rede
Grd_Prod_PsbleCap_Min [kVAr]	Potência reativa capacitiva mínima possível	Rede
Grd_Prod_PsbleCap_Std [kVAr]	Desvio padrão da potência reativa capacitiva possível	Rede
Gen_Bear2_Temp_Avg [ºC]	Temp. média no rolamento do gerador 2	Gerador
Nac_Direction_Avg [º]	Direção média da nacele	Nacele

Tabela VII.2: Descrição das variáveis do conjunto de dados *Signals*