
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

CAMILLA MAHON CAMPELLO DE SOUZA

# MÉTODOS DE APRENDIZADO DE MÁQUINA PARA LIMPEZA AUTOMÁTICA DE CURVAS DE POTÊNCIA UTILIZANDO *AUTOENCODERS* E REDE NEURAL KOLMOGOROV-ARNOLD

Recife
2025

CAMILLA MAHON CAMPELLO DE SOUZA

# MÉTODOS DE APRENDIZADO DE MÁQUINA PARA LIMPEZA AUTOMÁTICA DE CURVAS DE POTÊNCIA UTILIZANDO *AUTOENCODERS* E REDE NEURAL KOLMOGOROV-ARNOLD

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

Área de concentração: Energia.

Orientador (a): Prof. Dr. Alex Maurício Araújo

Orientador (a): Prof. Dr. Frederico Menezes

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Souza, Camilla Mahon Campello de.
Métodos de aprendizado de máquina para limpeza automática de curvas de potência utilizando autoencoders e rede neural Kolmogorov-Arnold / Camilla Mahon Campello de Souza. - Recife, 2025.
170f.: il.

Dissertação (Mestrado)- Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Programa de Pós Graduação em Engenharia Mecânica, 2025.
Orientação: Alex Maurício Araújo.
Coorientação: Frederico Duarte de Menezes.

1. Energia eólica; 2. Dados SCADA; 3. Limpeza de curvas de potência; 4. Autoencoders; 5. Rede Kolmogorov Arnold. I. Araújo, Alex Maurício. II. Menezes, Frederico Duarte de. III. Título.

UFPE-Biblioteca Central

CAMILLA MAHON CAMPELLO DE SOUZA

# MÉTODOS DE APRENDIZADO DE MÁQUINA PARA LIMPEZA AUTOMÁTICA DE CURVAS DE POTÊNCIA UTILIZANDO *AUTOENCODERS* E REDE NEURAL KOLMOGOROV-ARNOLD

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do título de Mestre em Engenharia Mecânica.

Área de concentração: Energia.

**BANCA EXAMINADORA**

_____

Prof. Dr. Frederico Menezes (Orientador)
Universidade Federal de Pernambuco

_____

Prof. Dr. Ângelo Peixoto (Examinador Interno)
Universidade Federal de Pernambuco

_____

Prof. Dr. Leandro Almeida (Examinador Externo)
Universidade Federal de Pernambuco

Dedico esse trabalho a todos que tornaram sua realização possível.

## AGRADECIMENTOS

Agradeço, em primeiro lugar, aos meus pais, Ricardo Menezes Campello de Souza e Marcia Mahon Campello de Souza, que, além de professores por profissão, foram os primeiros e mais importantes mestres da minha vida. Foi graças ao apoio constante e ao amor incondicional que sempre me ofereceram que pude trilhar este caminho e chegar até aqui.

Estendo meus agradecimentos aos familiares e amigos, pelo apoio ao longo da jornada e pela compreensão nas ausências que este trabalho exigiu.

Aos colegas da DNV, pelo companheirismo, pelo acolhimento nos momentos de desafio e pelas trocas de ideias, que muito contribuíram para este processo.

Ao professor Alex Maurício, meu primeiro orientador, agradeço pelo empenho, pela dedicação em acompanhar o desenvolvimento do trabalho e pelo incentivo nas etapas iniciais desta trajetória.

Por fim, ao professor Frederico Menezes, meu orientador, agradeço de forma especial pela orientação, pelas discussões enriquecedoras ao longo de toda a trajetória e pela constante disponibilidade. Este trabalho só foi possível graças às suas contribuições e comprometimento, inclusive nos momentos mais desafiadores.

“Se pude enxergar mais longe, foi porque me apoiei em ombros de gigantes.”

(Isaac Newton)

**RESUMO**

A energia proveniente da fonte eólica tem cada vez mais confirmado a sua importância na produção de energia elétrica renovável no Brasil e no mundo. Em tempos de transição energética, desastres climáticos e metas cada vez mais ambiciosas na redução de emissão de gases de efeito estufa, a eólica tem se firmado como uma opção não só viável, como essencial, na produção de energia limpa. Uma das formas de se avaliar o desempenho de uma turbina eólica se dá através da análise de dados SCADA, essenciais no monitoramento, tanto da condição, quanto da performance da turbina, fornecendo informações cruciais aos operadores. Também é possível utilizá-los para estimar a curva de potência histórica das turbinas de um parque eólico e fazer previsões futuras da produção de energia. Para que isso seja possível de ser realizado, a limpeza dos dados é essencial; tanto de dados espúrios, quanto para isolar problemas como indisponibilidade e problemas de desempenho, como por exemplo no sistema de *pitch* e de *yaw*. A depender da quantidade de dados a serem avaliados, esta tarefa pode ser exaustiva e computacionalmente custosa. Este trabalho apresenta uma nova metodologia para a limpeza automática de curvas de potência de turbinas eólicas, utilizando técnicas de aprendizado de máquina que ainda são pouco exploradas nesse contexto. A pesquisa começou com um algoritmo de agrupamento para identificar anomalias nos dados, mas os resultados iniciais mostraram limitações na separação clara entre diferentes tipos de falhas. Para superar esse desafio, foram desenvolvidos dois novos modelos baseados na combinação de *autoencoders* com uma rede neural inspirada na teoria de Kolmogorov-Arnold, denominados AE-KAN e VAE-KAN. Ambos os modelos conseguiram classificar melhor os tipos de falhas e se destacaram por detectar com mais sensibilidade os casos mais raros, superando outros métodos já consagrados na literatura. Como referência, os dados utilizados foram rotulados por um especialista da área, com base em uma ferramenta usada na indústria eólica, reforçando o potencial de aplicação prática da metodologia proposta.

.

Palavras-chave: Energia eólica; Dados SCADA; Limpeza de curvas de potência; *Autoencoders*; Rede *Kolmogorov-Arnold*.

**ABSTRACT**

Wind energy has increasingly proven its importance in renewable electricity generation in Brazil and worldwide. In times of energy transition, climate disasters, and increasingly ambitious targets for reducing greenhouse gas emissions, wind power has established itself not only as a viable option but as an essential component of clean energy production. One of the key methods for assessing the performance of a wind turbine is through the analysis of SCADA data, which is crucial for monitoring both the turbine's condition and performance, providing valuable insights to operators. In addition to evaluating turbine performance, SCADA data can also be used to estimate a wind farm's historical power curve and make future energy production forecasts. However, for these analyses to be reliable, proper data cleaning is essential—both to remove spurious data and to isolate issues such as unavailability, curtailments, and performance problems, including malfunctions in the *pitch* and yaw systems. Depending on the volume of data to be analyzed, this task can be both exhaustive and computationally demanding. This work proposes a new methodology for the automatic cleaning of wind turbine power curves, using machine learning techniques that are still underexplored in this context. The research began with a clustering algorithm to identify anomalies in the data, but initial results revealed limitations in separating different types of failures. To overcome this challenge, two new models were developed, combining autoencoders with a neural network inspired by Kolmogorov-Arnold theory — resulting in the AE-KAN and VAE-KAN approaches. Both models proved effective in distinguishing between different failure types and showed superior sensitivity in detecting rare cases, outperforming widely used methods in the literature. The dataset was labeled by an industry expert using a tool commonly applied in the wind energy sector, highlighting the practical applicability of the proposed methodology.

Keywords: Wind energy; SCADA data; Power curve cleaning; Autoencoders; Kolmogorov-Arnold Network.

## LISTA DE FIGURAS

**LISTA DE TABELAS**

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| ABEEólica | Associação Brasileira de Energia Eólica e Novas Tecnologias |
| ACR | Ambiente de Contratação Regulada |
| AM | Aprendizado de Máquina |
| AE | *Autoencoder* |
| ANEEL | Agência Nacional de Energia Elétrica |
| ANN | *Artificial Neural Network* |
| AUC | *Area Under Curve* |
| CFD | *Computational Fluid Dynamics* |
| CAGR | *Compound Annual Growth Rate* |
| CGH | Central Geradora Hidrelétrica |
| COP-28 | Conferência das partes 28 |
| CMS | C*ondition Monitoring System* |
| CNN | *Convolutional Neural Network* |
| DBSCAN | *Density-Based Spatial Clustering of Applications with Noise* |
| DNV | *Det Norske Veritas* |
| ECMI | *Empirical Copula-Based Mutual Information* |
| *et al.* | e outro |
| FPR | *False Positive Rate* |
| GAN | *Generative Adversarial Network* |
| GMM | *Gaussian Mixture Model* |
| GWEC | *Global Wind Energy Council* |
| GWO | *Grey Wolf Optimizer* |

| | |
|---|---|
| IA | Inteligência Artificial |
| IBAMA | Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis |
| IEC | *International Electrotechnical Commission* |
| KAN | *Kolmogorov-Arnold Network* |
| KNN | *K-Nearest Neighbors* |
| LBFGS | *Limited-memory Broyden–Fletcher–Goldfarb–Shanno* |
| LOF | *Local Outlier Factor* |
| LSTM | *Long Short-Term Memory* |
| MAE | *Mean Absolute Error* |
| MG | *Magnetic Positioning* |
| MLP | *Multilayer Perceptron* |
| MP | *Magnetic Position* |
| NB | *Nave Bayes* |
| NaN | *Not a Number* |
| PCA | *Principal Component Analysis* |
| PCH | Pequena Central Hidrelétrica |
| PG | Processo Gaussiano |
| PPA | *Purchase Power Agreement* |
| RBM | *Restricted Boltzmann Machine* |
| RF | *Random Forest* |
| ROC | *Receiver Operating Characteristic* |
| RMSE | *Root Mean Square Error* |
| SCADA | *Supervisory Control and Data Acquisition System* |
| SGBRT | *Stochastic Gradient Boosting Regression tree* |

| | |
|---|---|
| SMOTE | *Synthetic Minority Over-sampling Technique* |
| SVM | *Support Vector Machine* |
| TPR | *True Positive Rate* |
| TLBO-DL | *Teaching-Learning-Based Optimization with Deep Learning* |
| TTLOF | *Thompson Tau-Local Outlier Factor* |
| TSR | *Tip Speed Ratio* |
| VAE | *Variational autoencoder* |
| XAI | *Explainable AI* |

## LISTA DE SÍMBOLOS

| | |
|---|---|
| $P$ | Potência de saída |
| $C_P$ | Coeficiente de potência |
| $\rho$ | Densidade do ar |
| $A$ | Área do rotor |
| $U$ | Velocidade do vento |
| $a$ | Fator de indução axial |
| $p$ | Pressão |
| $g$ | Aceleração da gravidade |
| $h$ | Altura do fluido |
| $T$ | Força de Thrust |
| $C_T$ | Coeficiente de Thrust |
| $\lambda$ | Velocidade de ponta de pá |
| $R$ | Distância da ponta da pá ao centro do cubo |
| $\Omega$ | Velocidade angular do rotor |
| $x_i$ | Dados de entrada |
| $h_j$ | Nós ocultos |
| $w_{nj}$ | Pesos |
| $b_j$ | Viés |
| $f_j$ | Saída do nó j |
| $g()$ | Função de ativação |
| $\rho_q$ | Erro quadrático |
| $o_k$ | Rótulos verdadeiros |
| $\tilde{o}_k$ | Rótulos previstos |

| | |
|---|---|
| $X$ | Dado de entrada do *autoencoder* variacional |
| μ | Média da distribuição de probabilidade do *autoencoder* variacional |
| σ | Desvio padrão da distribuição de probabilidade do *autoencoder* variacional |
| $P(x)$ | Probabilidade de geração de X |
| $Z$ | Variável latente |
| $dz$ | Diferencial da variável latente |
| $f(x\|z)$ | Distribuição dos dados condicionada à variável latente $z$ |
| $P(z)$ | Distribuição sobre $z$ |
| $Q(x\|z)$ | Distribuição aproximada de $z$ |
| D | Divergência de Kullback-Leibler |
| E | Valor esperado |
| $J_{VAE}$ | Função de perda |
| $f(x_i)$ | Função multivariável |
| $\Phi_q$, $\Phi_{q,p}$ | Funções univariadas |
| $c_i$ | Coeficientes da função *spline* |
| $B_i(x)$ | Funções *splines* |
| TP | Verdadeiros positivos |
| TN | Verdadeiros negativos |
| FP | Falsos positivos |
| FN | Falsos negativos |

**SUMÁRIO**

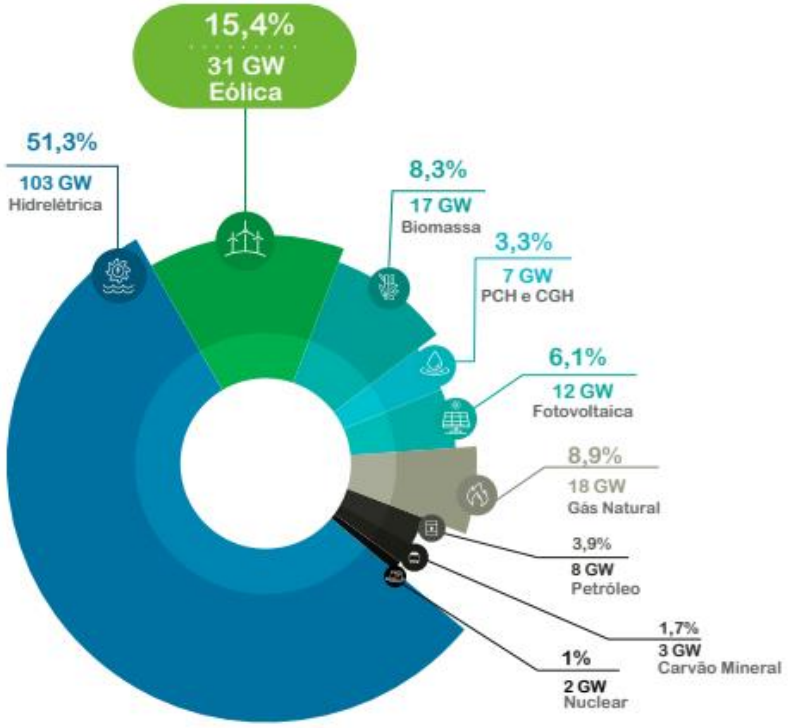
# 1 INTRODUÇÃO

Um dos principais objetivos da sociedade consiste na produção sustentável e de forma segura de energia, uma vez que ela é uma propulsora indispensável ao desenvolvimento econômico e social. Ao longo dos anos, o uso extensivo dos combustíveis fósseis contribuiu diretamente para o agravamento do aquecimento global e o aumento na emissão de gases do efeito estufa e, infelizmente, os mesmos ainda têm sido amplamente utilizados e são empregados em vários setores para suprir a necessidade de geração de energia. A grande escala de esgotamento da energia não renovável ocorreu devido à rápida utilização desses recursos, o que também causou efeitos adversos como mudanças climáticas e aquecimento global devido à alta emissão de gases de efeito estufa. Esses efeitos podem causar problemas inevitáveis, como elevação do nível do mar, derretimento de geleiras, destruição de florestas, poluição do ar, diminuição da camada de ozônio, uso de água e terra, emissões radioativas, precipitação ácida, perda de vida selvagem e danos à ecologia, ameaçando significativamente a humanidade (Bennagi et al., 2024).

Diante deste cenário, uma das maiores preocupações atuais está relacionada às mudanças climáticas e seus impactos no planeta. A 28ª edição da Conferência das Partes (COP-28), ocorrida em novembro de 2023, em Dubai, nos Emirados Árabes Unidos, foi um dos maiores eventos já realizados sobre o tema, reunindo 198 partes (197 países e a União Europeia). A COP-28 foi especificamente marcante, pois representou o primeiro chamado “*global stockage*”: processo que consiste em verificar onde os países estão progredindo em relação ao que foi definido no acordo de Paris e onde eles não estão. Uma das regras definidas neste acordo foi, por exemplo, a limitação do aumento da temperatura média mundial a 1,5°C até 2050, em relação aos níveis pré-industriais, e a redução à metade das emissões de gases do efeito estufa até 2030 (*United Nations Climate Change, 2024*). Para que possam avançar nesta direção, durante a COP-28, ficou definido que os países devem reduzir as emissões de gases do efeito estufa em pelo menos 45% até 2030, em relação aos níveis de 2010 (Alba energia, 2023). Apesar de serem metas bastante ambiciosas, isto significa um compromisso cada vez maior com a transição energética, progressivamente abandonando todo um sistema baseado em combustíveis fósseis para outro majoritariamente renovável.

O Brasil, que possui participação relativa de 84,4% de fontes renováveis em sua matriz energética, conforme apresentado na Figura 1-1 (Infovento, 2024), representa uma parte importante na transição energética global. Nos últimos anos, o país viu um crescimento significativo das fontes eólicas e solar, chegando a 31 GW de capacidade instalada da primeira fonte e 12 GW da segunda, para geração centralizada, e mais de 27,7 GW em geração distribuída. Baseado nos contratos viabilizados em leilões já realizados e no mercado livre, há uma expectativa de que mais de 22 GW de energia proveniente da fonte eólica *onshore* sejam instalados entre 2025 e 2030 (Figura 1-2).

**Figura 1-1 Matriz energética brasileira.**

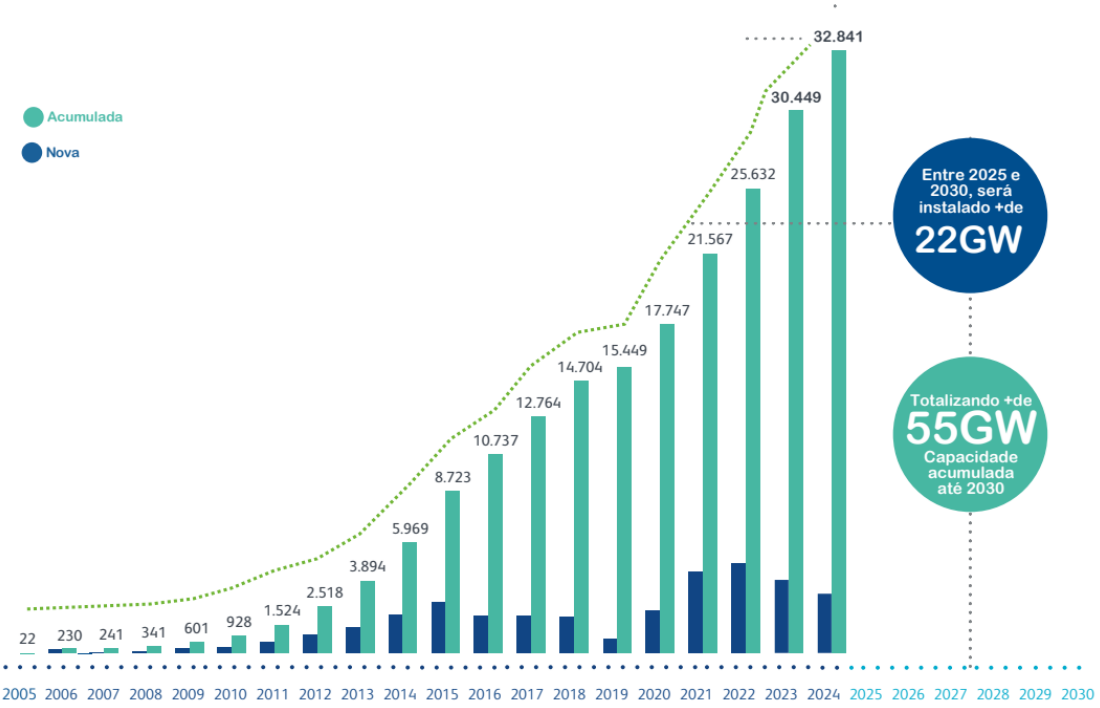



Fonte: Infovento (2024)

Além disso, o IBAMA já recebeu mais de 170 GW em projetos propostos para energia eólica *offshore*; praticamente o mesmo valor de toda a capacidade da matriz elétrica brasileira atual, mostrando, portanto, o apetite dos investidores e confirmando o grande potencial de energia proveniente da fonte eólica *offshore* antes previsto. Com cerca de 8000 km de costa, o Brasil tem o potencial de instalar mais de 1200 GW de eólica *offshore*, de acordo com um estudo realizado pelo Banco Mundial (World bank group, 2020). A ABEEólica (Associação Brasileira de

Energia Eólica e Novas Tecnologias) tem se mostrado otimista em relação ao mercado *offshore*, principalmente após a aprovação no congresso nacional e sanção do presidente da república, resultando na publicação da Lei n° 15.097 de 10 de janeiro de 2025 – o Marco Legal das Eólicas *Offshore* no Brasil (Gomes et al., 2025).

**Figura 1-2 Evolução da capacidade instalada da fonte eólica onshore.**



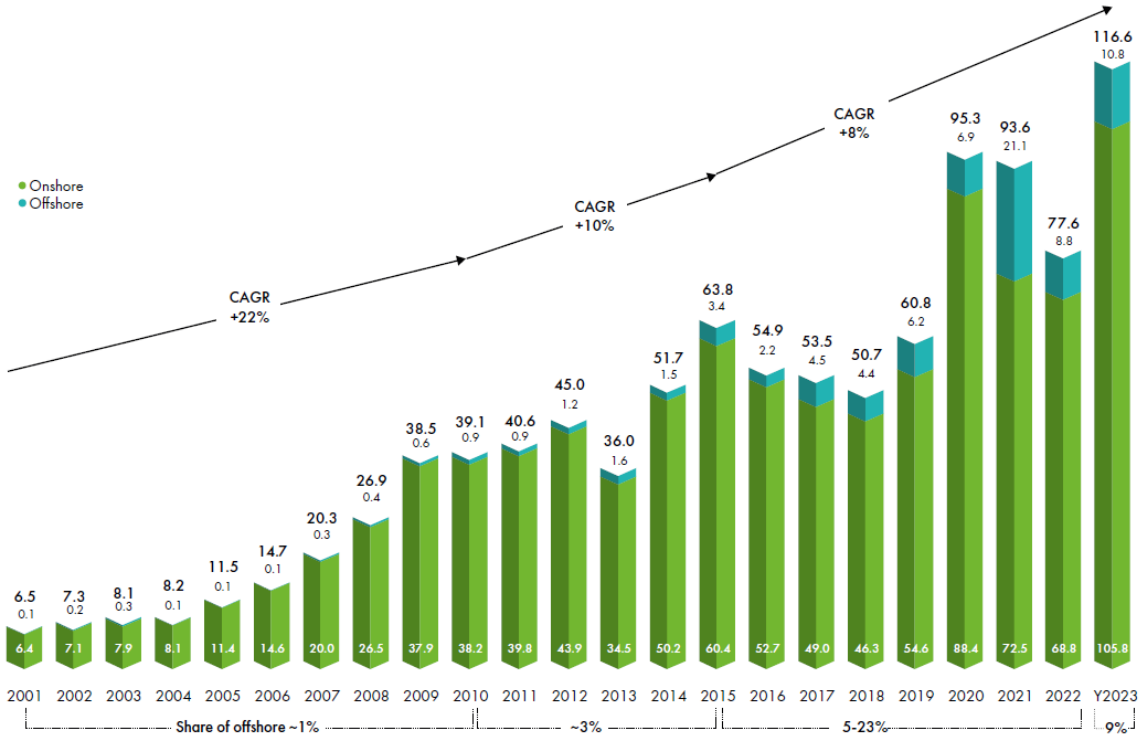

Fonte: Infovento (2024)

A fonte eólica, que é a segunda maior na matriz elétrica brasileira, com participação, em março/2024, de 15,4% e potência instalada de 31 GW, teve o seu desenvolvimento impulsionado por importantes políticas públicas federais, estaduais e de instituições de fomento, sendo responsável em 2023, por abastecer mais de 47 milhões de residências brasileiras (Infovento, 2024). Em um estudo feito para ABEEólica, estima-se que para cada um real investido em energia eólica, há um aumento de R$2,90 no PIB (Borges, 2022). O setor eólico no Brasil tem consolidado seu crescimento através do mercado livre, se vendo mais distante do ACR (Ambiente de Contratação Regulada) e seguindo cada vez mais na direção dos PPAs (*Power Purchase Agreements*) corporativos, o que lhe confere uma maior resiliência (GWEC, 2024).

Globalmente, este cenário não é muito diferente. A energia eólica é um dos pilares da transição energética, sendo uma das fontes mais competitivas, com o

mercado amadurecido e a tecnologia consolidada. De toda energia gerada no mundo, 6% são dependentes da fonte eólica. Em alguns países esse número é ainda muito mais expressivo. A Dinamarca gera mais de 50% de sua eletricidade a partir da energia eólica e na Alemanha essa participação chega próxima aos 30%. Em 2023, o mundo bateu um novo recorde em termos de acréscimo em capacidade instalada, como apresentado na Figura 1-3 (GWEC, 2024).

**Figura 1-3 Histórico de novas capacidades instaladas no planeta, em GW.**




Fonte: GWEC (2024)

De acordo com o Global Wind Energy Council, todos os planos de desenvolvimento indicam que as novas potências instaladas devem quadruplicar anualmente, quando comparadas aos níveis atuais, para que se atinja a neutralidade das emissões de carbono até 2030. Até 2050, a eólica deverá fornecer mais de 35% da energia elétrica mundial, em comparação aos 6% que se tem hoje (GWEC, 2024). Em um período marcado por uma crise climática evidente, com desastres ambientais frequentes, como o ocorrido no Rio Grande do Sul em maio de 2024, a urgência pela transição energética se torna mais clara. Já estamos atrasados para abandonar as fontes de energia que emitem gases de efeito estufa e adotar alternativas renováveis. Catástrofes como as que vivenciamos em 2024, antes

previstas pelos cientistas para ocorrerem mais adiante, já são realidade. Portanto, a busca por um futuro mais sustentável é essencial para a sobrevivência da humanidade.

Embora a energia eólica apresente diversas vantagens, como ser renovável e possuir um mercado consolidado globalmente, sua natureza intermitente gera incertezas significativas nos sistemas de gestão de energia, afetando a programação de despacho e, consequentemente, a confiabilidade da rede elétrica (BILENDO et al., 2023). Essa questão motiva pesquisadores a desenvolver soluções específicas, muitas das quais dependem de uma estimativa precisa da curva de potência da turbina. Tal estimativa é essencial tanto para o gerenciamento operacional da energia eólica quanto para o monitoramento de desempenho da turbina.

Como será abordado na fundamentação teórica, a curva de potência representa a relação entre a velocidade do vento e a potência elétrica gerada por uma turbina eólica — ou seja, indica quanta energia a turbina entrega em função da velocidade do vento. Essa curva pode ser estimada a partir de grandes volumes de dados operacionais registrados automaticamente por sensores instalados nas próprias turbinas. Esses sensores monitoram continuamente variáveis como velocidade do vento, potência gerada, ângulo de inclinação das pás (*pitch*), posição da nacele (*yaw*), temperatura de componentes e o estado geral de operação. Os dados coletados são organizados por sistemas de supervisão e aquisição, usualmente conhecidos como sistemas SCADA (do inglês *Supervisory Control and Data Acquisition System*), que armazenam milhares de pontos de medição ao longo do tempo. A análise dessas informações permite não apenas estimar a curva de potência em condições reais de operação, mas também identificar falhas, eventos de indisponibilidade, cortes de produção (*curtailments*) e anomalias de desempenho. Tais eventos compõem o que se convencionou chamar de pontos de operação anômalos, cuja correta identificação constitui a chamada limpeza da curva. Essa etapa é essencial para garantir que a curva de potência estimada represente com precisão o comportamento da turbina, servindo como base para decisões técnicas, operacionais e até comerciais. No entanto, devido ao volume massivo de dados gerados diariamente por cada turbina, esse processo de limpeza pode se tornar bastante trabalhoso e computacionalmente custoso, especialmente quando realizado manualmente ou com abordagens pouco eficientes. Esse desafio reforça a

importância de se desenvolver metodologias automatizadas, robustas e confiáveis para o tratamento desses dados.

## 1.1. OBJETIVOS

### 1.1.1. Objetivo Geral

Este trabalho tem como objetivo desenvolver uma nova metodologia para a filtragem automática de curvas de potência de turbinas eólicas – ou mais comumente conhecida no setor eólico como limpeza da curva de potência - utilizando técnicas de Aprendizado de Máquina (AM), incluindo métodos de agrupamento de dados e classificação.

### 1.1.2 Objetivos Específicos

Dentre os objetivos específicos, pode-se citar:

a) Implementar algoritmos de agrupamento de dados e de classificação com o objetivo de identificar dados anômalos e dados normais em curvas de potência de turbinas eólicas;
b) Avaliar diferentes técnicas de aprendizado de máquina e determinar a mais eficiente para limpeza de curvas de potência;
c) Aplicar a metodologia desenvolvida em dados de turbinas eólicas reais;
d) Comparar a metodologia desenvolvida com outras técnicas de aprendizado de máquina já consolidadas na literatura analisando vantagens e limitações;
e) Validar a metodologia com a limpeza manual conduzida por um especialista no setor.

## 1.2. JUSTIFICATIVAS

A energia eólica atualmente se destaca como uma das principais fontes de geração de energia elétrica no mundo, com grande potencial de expansão. A medida que mais turbinas eólicas são instaladas, torna-se cada vez mais relevante garantir o seu bom desempenho ao longo do tempo. A curva de potência da turbina é uma das principais ferramentas para esse monitoramento. Desvios em relação à curva esperada podem ocorrer por diversos motivos, incluindo períodos de indisponibilidade parcial, cortes programados de geração (*curtailments*), limitações

temporárias no controle da máquina, falhas em sensores ou até problemas de comunicação de dados. Embora nem sempre representem falhas definitivas ou degradação física, esses desvios impactam diretamente a produção de energia e dificultam a avaliação precisa do desempenho da turbina.

Nesse contexto, os dados operacionais registrados durante a vida útil das turbinas assumem um papel fundamental. Eles possibilitam a estimativa da curva de potência em condições reais, a identificação de comportamentos anômalos e a antecipação de situações que possam comprometer a performance do parque. No entanto, o grande volume de dados gerados, aliado à sua complexidade e à presença recorrente de registros inconsistentes, torna o processo de limpeza uma tarefa trabalhosa e custosa do ponto de vista computacional. Para lidar com esse desafio, surgem como alternativa as técnicas de aprendizado de máquina, capazes de reconhecer padrões complexos e auxiliar na separação automática entre pontos normais e anômalos. Com a aplicação dessas técnicas, torna-se possível construir modelos mais robustos para o monitoramento da curva de potência, contribuindo diretamente para previsões mais precisas de geração, avaliações operacionais confiáveis e decisões estratégicas sobre manutenção, operação e gestão de ativos.

Esta dissertação tem como objetivo investigar os dados de curvas de potência de turbinas eólicas, com foco nos desafios enfrentados por engenheiros na análise e no processamento dessas informações. São exploradas diferentes abordagens para a limpeza automática da curva de potência, avaliando suas limitações e potencial de aplicação. Como principal contribuição, propõe-se um novo modelo híbrido que combina autoencoders com redes neurais baseadas na teoria Kolmogorov-Arnold, buscando maior precisão na identificação de anomalias e melhor separação entre os diferentes tipos de desvios operacionais.

## 1.3. ESTRUTURA DO TRABALHO

Esta dissertação está estruturada da seguinte forma: o primeiro capítulo apresenta a introdução, incluindo a contextualização do tema, a definição dos objetivos gerais e específicos e a justificativa do estudo. Em seguida, o segundo capítulo aborda o referencial teórico, fornecendo a base conceitual necessária para o estudo desenvolvido, tanto no contexto da energia eólica quanto no do aprendizado de máquina. O terceiro capítulo corresponde à revisão de literatura,

onde são explorados os avanços e estudos mais recentes na área, destacando o estado da arte sobre o tema.

No quarto capítulo, é apresentada a metodologia, detalhando-se os dados utilizados e os algoritmos implementados. O quinto capítulo expõe os resultados, trazendo análises em formato de tabelas e figuras para ilustrar os testes conduzidos e os achados da análise. Por fim, o sexto capítulo traz as conclusões, com uma síntese dos principais resultados obtidos, e por fim, as referências bibliográficas utilizadas ao longo do trabalho.

# 2 REFERENCIAL TEÓRICO

## 2.1 FUNCIONAMENTO DE UMA TURBINA EÓLICA

### 2.1.2 Princípios básicos

Turbinas eólicas ou moinhos, como eram chamados, têm sido utilizados há séculos para extrair energia do vento. Uma turbina eólica é uma máquina que converte a energia cinética do vento em torque e velocidade angular no seu eixo e, posteriormente, em energia elétrica. A potência de saída P é dada pela conhecida Equação (1) (*BURTON et al. (2011)*):

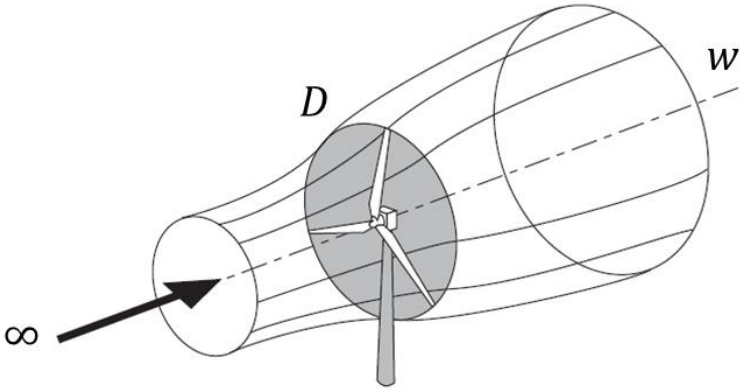
$$P = \frac{1}{2} C_P \rho A U^3, \quad \textbf{(1)}$$

em que $\rho$ é a densidade do ar (massa específica), $C_P$ é o coeficiente de potência, $A$ é a área do rotor e $U$ é a velocidade do vento não perturbada. O coeficiente de potência representa a fração da potência do vento que pode ser convertida pela turbina em taxa de trabalho mecânico. Possui um limite máximo teórico de 0,593, demonstrado mais adiante, chamado de o “limite de Betz”. O físico alemão Albert Betz, em um artigo publicado em 1920 na revista *Journal of Turbine Science,* provou que no máximo 59,3% da energia cinética contida em um escoamento que está em um tubo de corrente de mesma seção transversal de um disco atuador (que simula o rotor de uma turbina) pode ser convertido em trabalho útil pelo disco (Okulov & Kuik, 2009). Na prática valores sempre menores do que este são atingidos.

A teoria do disco atuador explica o processo da extração de energia de uma turbina eólica. Por conservação de energia, ao remover a energia cinética contida no vento, a velocidade da massa de ar que passa pelo disco atuador é reduzida. Antes do disco, a área de seção transversal do tubo de corrente é menor do que a do disco e se torna maior à jusante (Figura 2-1). Essa expansão acontece porque a mesma quantidade de ar deve passar por cada seção e a massa que passa pela seção transversal do tubo, por unidade de tempo, é dada por $\rho A U$ (vazão mássica). Logo, mantendo a densidade do ar constante (escoamento incompressível), ao reduzir a velocidade, a área deve ser maior. A taxa do fluxo de massa deve ser a mesma ao longo do tubo de corrente, então,

$$\rho A_\infty A_U = \rho A_D U_D = \rho A_w U_w. \quad \textbf{(2)}$$

Figura 2-1 Tubo de escoamento da extração de energia de uma turbina eólica.




Fonte: Adaptado de BURTON et al. (2011)

O símbolo ∞ se refere à condição *upstream* (anterior ao disco – escoamento não perturbado), D se refere às condições no disco e w às condições *downstream* (na esteira/wake). Se considera que a presença do disco atuador induz uma redução da velocidade do vento livre, dada por $-\boldsymbol{aU}_{\infty}$, em que $\boldsymbol{a}$ é chamado de fator de indução axial. No disco, portanto, a velocidade na direção do escoamento é dada por

$$U_D = U_{\infty}(1-a). \quad \textbf{(3)}$$

O fluxo de ar sofre uma mudança resultante de velocidade de $U_{\infty} - U_w$. A taxa de mudança do momento linear é dada pela mudança da velocidade vezes a taxa do fluxo de massa, e assim,

$$Taxa\ de\ mudança\ do\ momento = (U_{\infty} - U_w)\rho A_D U_D. \quad \textbf{(4)}$$

A força que causa a mudança de momento advém da diferença de pressão no entorno do disco atuador (o tubo de corrente é cercado por ar a pressão atmosférica, então a força resultante é zero). Então, tem-se que

$$(p_D^+ - p_D^-)A_D = (U_{\infty} - U_w)\rho A_D U_{\infty}(1-a). \quad \textbf{(5)}$$

Para calcular a diferença de pressão, a Equação de Bernoulli é aplicada, de forma separada, antes e após o disco atuador (energias separadas são necessárias porque a energia é diferente antes e após). A energia total do escoamento, que

consiste em energia cinética, pressão estática e gravitacional, deve permanecer constante, visto que nenhum trabalho é realizado pelo fluido. Assim, para um volume de ar, tem-se que

$$\frac{1}{2}\rho U^2 + p + \rho g h = constante. \quad \textbf{(6)}$$

*Upstream* temos

$$\frac{1}{2}\rho_\infty U_\infty^2 + p_\infty + \rho_\infty g h_\infty = \frac{1}{2}\rho_D U_D^2 + p_D^+ + \rho_D g h_D, \quad \textbf{(7)}$$

e assumindo que o escoamento é incompressível ($\rho_\infty = \rho_D$) e horizontal ($h_\infty = h_D$), então:

$$\frac{1}{2}\rho U_\infty^2 + p_\infty = \frac{1}{2}\rho U_D^2 + p_D^+. \quad \textbf{(8)}$$

Analogamente, *downstream*

$$\frac{1}{2}\rho U_w^2 + p_\infty = \frac{1}{2}\rho U_D^2 + p_D^-. \quad \textbf{(9)}$$

Subtraindo a equação (9) da (8) chegamos em

$$p_D^+ - p_D^- = \frac{1}{2}\rho(U_\infty^2 - U_w^2) \quad \textbf{(10)}$$

e substituindo a (10) na (5)

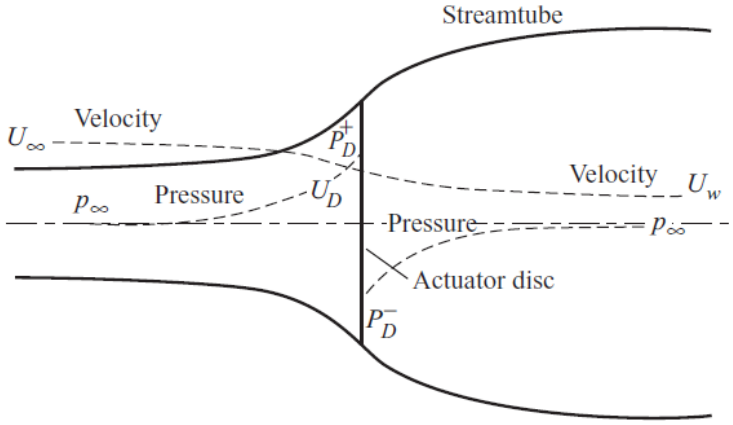
$$\frac{1}{2}\rho(U_\infty^2 - U_w^2)A_D = (U_\infty - U_w)\rho A_D U_\infty(1-a), \quad \textbf{(11)}$$

o que resulta em

$$U_w = (1-2a)U_\infty. \quad \textbf{(12)}$$

As variáveis das equações (1) a (12) são ilustradas na Figura 2-2, que é uma representação esquemática do modelo do disco atuador. As curvas na Figura demonstram como a velocidade do escoamento diminui e a pressão sofre uma queda ao atravessar o disco, simulando a extração de energia pela turbina.

Figura 2-2 Variação das grandezas de velocidade e pressão em diferentes regiões do escoamento: à montante (*upstream*), à jusante (*downstream*) e próximas ao disco simulando o rotor.




Fonte: BURTON et al. (2011)

Da equação (5), a força atuante é dada por

$$T = (p_D^+ - p_D^-)A_D = 2\rho A_D U_\infty^2 a(1-a). \quad \textbf{(13)}$$

A potência produzida pelo fluxo de ar no disco atuador é $TU_D$. Logo, tem-se:

$$P = TU_D = 2\rho A_D U_\infty^2 a(1-a)^2. \quad \textbf{(14)}$$

Da equação (1), o coeficiente de potência é definido por

$$C_P = \frac{P}{\frac{1}{2}\rho A U^3}. \quad \textbf{(15)}$$

Substituindo a Equação (14) na Equação (15), tem-se que

$$C_P = 4a(1-a)^2. \quad \textbf{(16)}$$

O máximo valor de $C_P$ ocorre quando

$$\frac{dC_P}{da} = 0 \ \therefore 4(1-a)(1-3a) = 0. \quad \textbf{(17)}$$

Resultando em $a = \frac{1}{3} \therefore C_{Pmáx} \approx 0.593$. Esse é o limite obtido por Betz, representando o valor teórico máximo de energia cinética que uma turbina eólica pode extrair do vento. Este fator é importante na estimativa da curva de potência da turbina. A força no disco atuador causada pela diminuição da pressão, dada pela Equação (13) também pode ser adimensionalizada, fornecendo o coeficiente de Thrust,

$$C_T = \frac{Thrust}{\frac{1}{2}\rho U_\infty^2 A_D} = 4(1-a). \quad \textbf{(18)}$$

Quanto maior o valor de $C_T$, maior a resistência imposta pela turbina ao escoamento, permitindo a extração de mais energia. Em contrapartida, essa maior energia extraída resulta em uma menor velocidade à jusante e maior o efeito esteira provocado pela turbina.

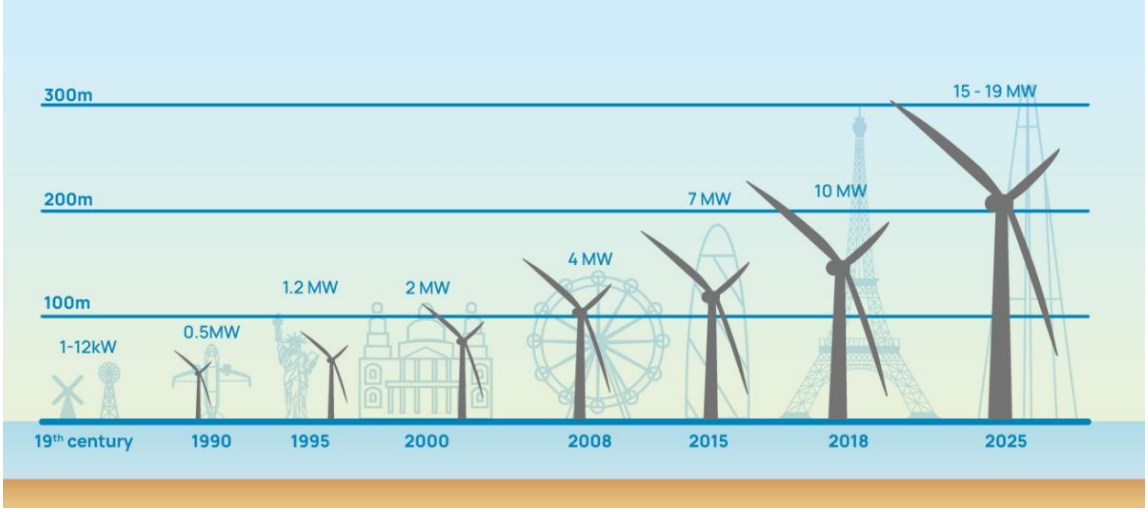
#### 2.1.2.1 Tecnologia de uma turbina eólica

Em 1983, uma turbina com potência nominal de 55kW e diâmetro de rotor de cerca de 15 m era referência comercial no mercado europeu ocidental (Heinzelmann, 2019). Ainda segundo a autora, pouco mais de 10 anos depois, a potência nominal média multiplicou-se por 10 e as dimensões dos rotores aerodinâmicos tornaram-se duas a três vezes maiores. Nos dias de hoje temos turbinas centenas de vezes mais potentes. Considerando a interdisciplinaridade que existe em uma turbina eólica e toda a cadeia de suprimentos envolvida, desde a concepção do projeto, fabricação, construção e manutenção, considera-se esse um desenvolvimento bastante célere.

Em 2020, em novas instalações na Europa, as turbinas tinham em torno de 8,2 MW, chegando até 10,4 MW. O modelo da GE, Haliade-X, estabeleceu um novo recorde, chegando a 14 MW de capacidade. Esta turbina recebeu uma certificação independente da consultora norueguesa DNV para operar até 14,7 MW e um protótipo segue operando em um porto de Roterdã, onde foi instalada em outubro/2021 (Marinho, 2022).

A Figura 2-3 mostra uma comparação do tamanho das turbinas com prédios famosos, em que se é possível perceber a evolução destas. Num futuro muito próximo, esperam-se turbinas de 15 MW provenientes da Siemens Gamesa e Vestas. Além destas, a chinesa MingYang já anunciou o seu novo modelo de 16 MW e 242 m de diâmetro de rotor. A indústria prevê que em 2030 haverá máquinas de 20 MW de capacidade e 275 m de rotor.

**Figura 2-3 - Evolução do tamanho de turbinas eólicas em comparação a edificações históricas.**




Fonte: MEGAWIND (2024)

Como observado na Figura 2-3, as turbinas alcançaram tamanhos impressionantes, sendo, portanto, as maiores máquinas rotativas do mundo. São necessários três A380s (o maior avião de transporte civil do planeta) para equivaler ao comprimento do diâmetro de rotor da GE Haliade-X (vide Figura 2-4). Mas diferentemente dos aviões, as turbinas são projetadas para operar de maneira totalmente autônoma, em um ambiente bastante insalubre, com muitas cargas externas, com o mínimo de manutenção e máximo de disponibilidade possível, acumulando cerca de 100 milhões de ciclos de fadiga ao longo dos seus 20 anos de operação (Veers et al., 2023)

**Figura 2-4 Dimensões de um A380 e de uma turbina GE Haliade-X 12-14MW.**

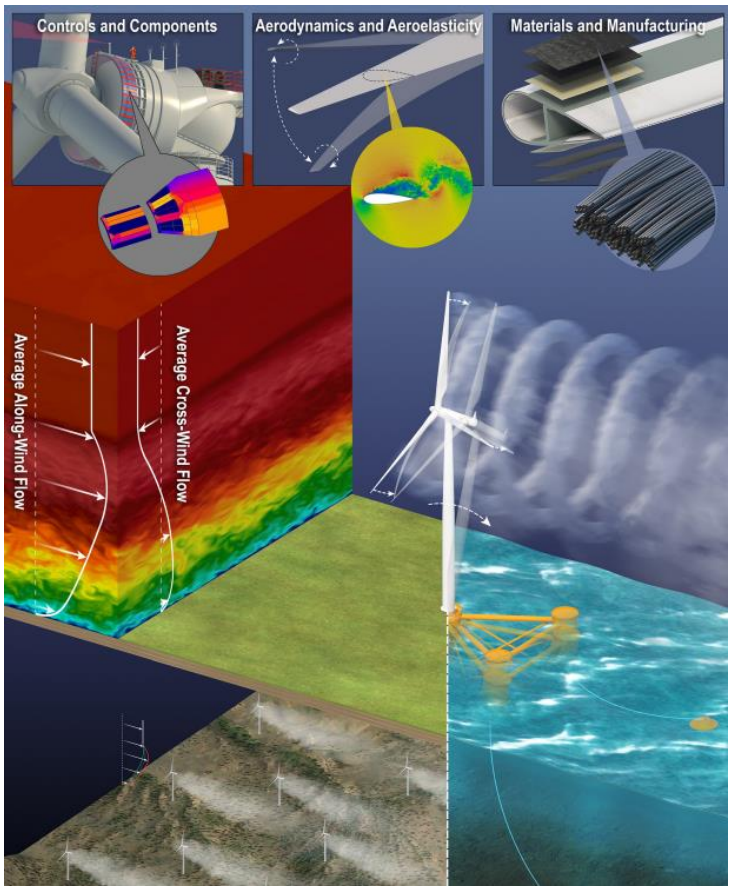

Fonte: VEERS et al. (2023)

As turbinas eólicas modernas possuem pás tão extensas que ultrapassam a camada limite atmosférica. Essas estruturas são longas e flexíveis, interagindo de forma dinâmica com o escoamento do ar no local. Além do carregamento causado pelo escoamento livre, as turbinas também estão sujeitas aos carregamentos provenientes das esteiras de outras turbinas eólicas. Esse vento perturbado, de baixa velocidade e alta turbulência, exerce um impacto adicional significativo nessas estruturas. A Figura 2-5 apresenta as variáveis e toda complexidade envolvida na análise de uma turbina eólica, seja no ambiente *onshore* ou *offshore*.

A complexidade aumenta no ambiente *offshore*, onde várias cargas atuam na turbina: além do vento em velocidades mais altas, há também a influência de correntes e marés. Nesse cenário, a fundação assume um papel crucial. No caso da fundação monopilar, a mais utilizada em todo o mundo, a turbina e a fundação são consideradas como uma única unidade estrutural. Isso resulta em um sistema mais esbelto e dinamicamente suscetível às cargas externas, devido à sua frequência natural estar próxima das frequências de excitação. A interação do solo com a turbina também se torna crítica devido aos grandes carregamentos (CAMPELLO DE SOUZA; RIBEIRO, 2017). Além da fundação monopilar, existem outras soluções na eólica *offshore*, como as fundações do tipo jaqueta, gravidade e ainda as flutuantes, que estão em constante movimento.

Sistemas de controle avançados, juntamente com sensores de monitoramento, são empregados para melhorar a gestão dessas máquinas. Ferramentas de simulação aeroelástica precisam capturar a grande escala, a flexibilidade aumentada e os carregamentos complexos, frequentemente utilizando modelos de alta fidelidade. Nos processos de fabricação, espera-se a adoção de novos materiais com maior resistência e menor peso, além de melhorias na qualidade para comprimentos superiores a 100 metros. As turbinas eólicas representam um campo de alta complexidade e contínua evolução. A pesquisa contínua é fundamental para aprofundar o entendimento das dinâmicas envolvidas e impulsionar inovações que garantirão a eficiência, sustentabilidade e resistência dessas gigantes energéticas frente aos desafios ambientais e operacionais.

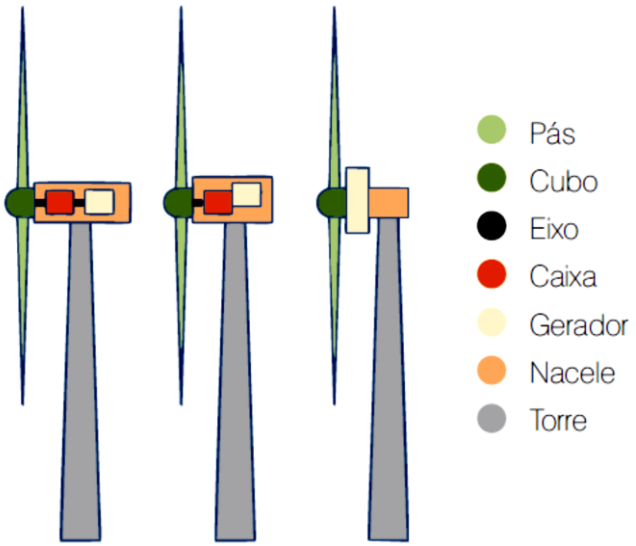
**Figura 2-5 Componentes da natureza e parâmetros físicos de uma turbina eólica.**




Fonte: VEERS et al. (2023).

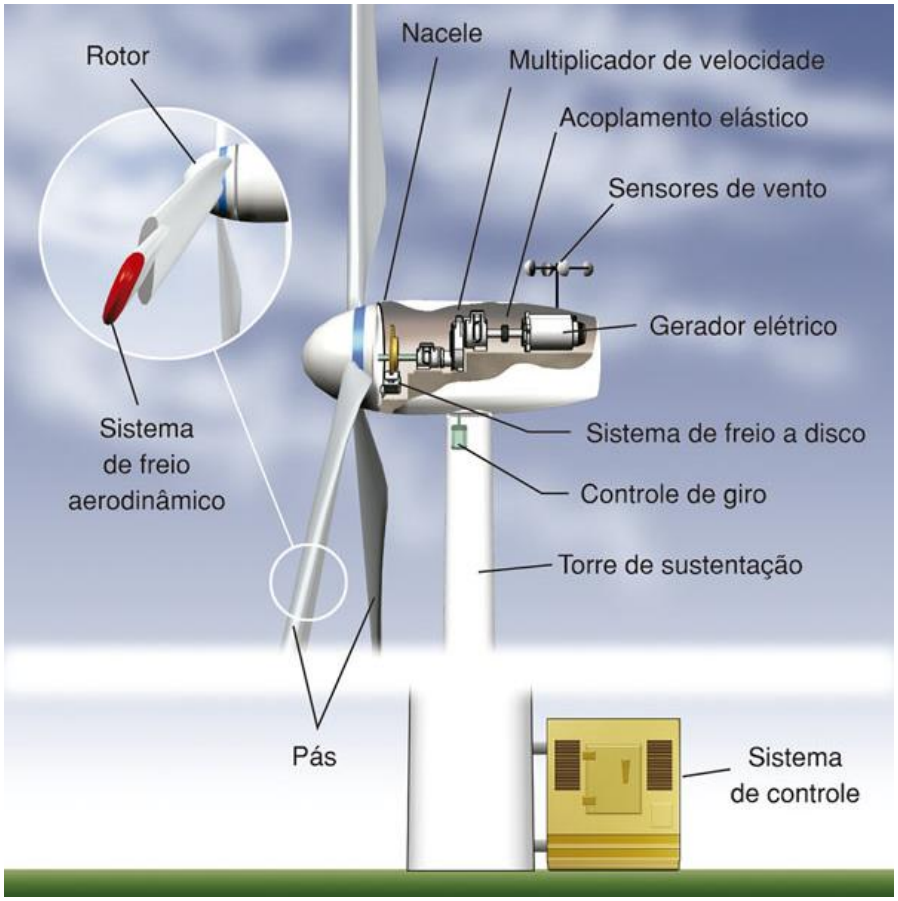
A Figura 2-6 e a Figura 2-7 apresentam, respectivamente, os componentes básicos de uma turbina de eixo horizontal e um exemplo de um típico modelo, com seus respectivos componentes.

**Figura 2-6 Componentes básicos de turbinas de eixo horizontal.**




Fonte: NERY et al (2014).

**Figura 2-7 Esquema dos componentes principais de uma turbina eólica.**




Fonte: PIRES (2018)

A Tabela 2-1 Principais características mecânicas de uma turbina eólica.Tabela 2-1 apresenta as principais características mecânicas de uma turbina eólica e os respectivos tipos em cada característica.

Tabela 2-1 Principais características mecânicas de uma turbina eólica.

| | |
|---|---|
| Princípio aerodinâmico de conversão de energia | • Baseado na força de sustentação<br>• Baseado na força de arrasto |
| Posição do rotor em relação à torre | • À barlavento<br>• À sotavento |
| Número de pás | • Pá única<br>• Duas pás<br>• Três pás<br>• Múltiplas pás |
| Princípio aerodinâmico de controle de torque | • Estol<br>• Estol ativo<br>• Controle de passo ativo (*pitch*) |
| Sistema de orientação do rotor em relação à direção do vento (*yaw*) | • Ativo<br>• Passivo |
| Velocidade de rotação do rotor | • Constante<br>• Variável |
| Eixo de acionamento mecânico | • Expandido<br>• Semicompacto<br>• Compacto |
| Conversão de velocidade de rotação | • Com caixa de engrenagem<br>• Sem caixa de engrenagem *(Direct-drive)* |

Fonte: Adaptado de HEINZELMANN (2019).

A nacele é uma estrutura, em formato de “caixa”, que abriga os principais componentes de uma turbina eólica. As turbinas eólicas comerciais conectadas na rede elétrica, para geração centralizada, possuem as seguintes características:

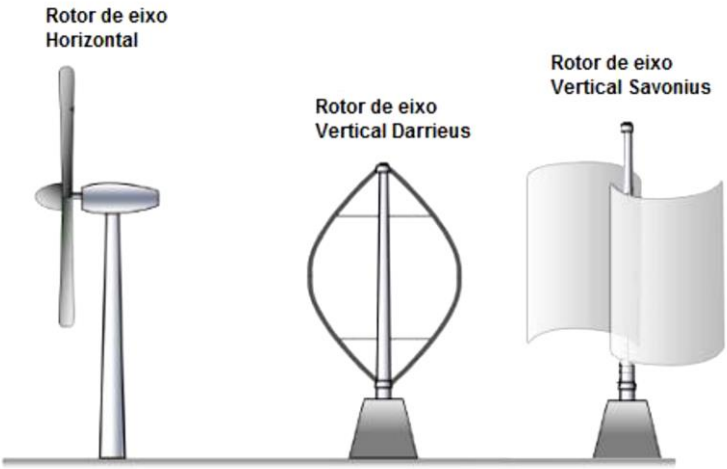
- Princípio de extração de energia baseado na força de sustentação aerodinâmica;
- Eixo de acionamento horizontal;
- Rotor de três pás, à barlavento;
- *Yaw* ativo.

Quanto às outras características mecânicas, descritas na Tabela 2-1, estas ainda podem variar de acordo com a máquina escolhida.

As turbinas de eixo horizontal funcionam mediante a força de sustentação, enquanto as de eixo vertical geralmente são atreladas à força de arrasto. No

entanto, algumas turbinas de eixo vertical também podem ter seu funcionamento mediante a força de sustentação, como por exemplo, a *Darrieus* e a *Savonius* (GASCH; TWELE, 2012), (Hau, 2013). A Figura 2-8 ilustra diferentes tipos de rotores.

**Figura 2-8 Tipos de rotores.**




Fonte: EPE (2016)

Apesar de não necessitarem de direcionamento do rotor em relação à direção predominante do vento e o fato da casa de máquinas ficar no solo (facilitando a manutenção), os custos de produção e fabricação das pás das turbinas *Darrieus* são cerca de 30% maiores do que as de eixo horizontal. O desempenho aerodinâmico e a eficiência energética também são inferiores, o que faz com que as turbinas de rotores verticais não sejam competitivas comercialmente frente as de eixo horizontal (HEINZELMANN, 2019). Dentre as vantagens de uma turbina eólica de eixo horizontal, podemos citar (KUSUMA et al., 2024):

1. Captura de ventos de altas velocidades: devido à sua elevada altura, essas turbinas conseguem capturar velocidades mais altas. Em alguns locais, o coeficiente de cisalhamento (*wind shear*) é relativamente alto, podendo aumentar a velocidade em 20% a cada 10 metros de altura, resultando em um aumento de 34% na potência de saída.
2. Grande área varrida pelas pás: a ampla área varrida pelas pás permite a captura de mais vento, o que eleva a eficiência dessas máquinas para níveis geralmente superiores a 70%.

3. Uso comercial: devido à sua alta eficiência, essas turbinas são amplamente utilizadas comercialmente, havendo uma grande disponibilidade no mercado.
4. Sistemas variáveis de *pitch*: muitos modelos possuem um sistema variável de *pitch*, permitindo que as pás se posicionem em um ângulo de ataque ótimo, melhorando ainda mais a eficiência na captação de energia.

A posição à barlavento consiste no rotor estar posicionado anteriormente à torre e à sotavento, o oposto. Na posição à sotavento, o escoamento passa pela torre antes de chegar ao rotor sofrendo perturbação devido à interação com a estrutura; perdendo, portanto, energia, provocando maior impacto aerodinâmico e ainda gerando maiores emissões acústicas. Sendo assim, a configuração à barlavento é a adotada comercialmente.

Considerando a turbina comercial com eixo horizontal e rotor à barlavento, faz-se necessário um sistema de orientação do rotor em relação à velocidade predominante do vento; que no caso da configuração padrão, é sempre ativo. Esse sistema é composto por uma roda dentada, motores elétricos de passo e sistemas de frenagem com pastilha que fornecem o torque necessário para rotacionar a nacele e mantê-la alinhada com o vento (KARAKASIS et al., 2016). Existe ainda o freio do *yaw*, que trava a nacele quando ela está corretamente orientada, evitando giros desnecessários.

A escolha pelo número ótimo de pás está ligada à razão da velocidade de ponta de pá $\lambda$ (*Tip Speed Ratio*, ou, TSR), dada pela Equação (19)

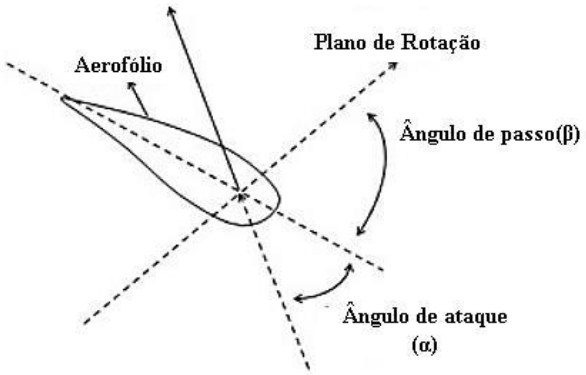
$$\lambda = \frac{R\Omega}{U_\infty}, \tag{19}$$

em que $R$ é a distância da ponta da pá ao centro do cubo, $\Omega$ é a velocidade angular do rotor ($R\Omega$ é, portanto, a velocidade linear da ponta de pá), e $U_\infty$ é a velocidade do vento não perturbado. Um outro conceito importante na definição do número de pás é a solidez, que é a razão entre a área sólida de pás e a área circular definida pelo extremo da pá em rotação. Uma baixa solidez significa que a pá tem uma área de suporte menor, o que pode levar a vibrações e instabilidade na rotação, reduzindo a vida útil da turbina. Porém, uma baixa solidez também indica um alto TSR, o que aumenta a eficiência na conversão de energia. Por outro lado, uma alta solidez, reduz a eficiência da turbina, pois se tem um menor TSR; entretanto, elas são mais estáveis mecanicamente. O equilíbrio entre esses dois conceitos, levou a um número ótimo de pás como sendo menor do que cinco (Fadigas, 2011). Uma turbina

de quatro pás tem a desvantagem de ser mais custosa do que soluções com menos pás (o rotor corresponde a cerca de 20% do custo de uma turbina e cada pá a cerca de 6% do total), não compensada pelo aumento na eficiência. Um compromisso ótimo, portanto, entre custo, eficiência energética e mecânica da estrutura chega a um número de 3 pás (Heinzelmann, 2019).

Em relação ao controle do torque, uma turbina pode fazer mediante estol passivo, *pitch* e estol ativo. O controle por estol passivo, como o nome sugere, é um sistema de controle passivo que responde à velocidade do vento. As pás do rotor são fixadas em um ângulo de passo β específico (Figura 2-9), escolhido para que o escoamento de ar ao redor do perfil aerodinâmico se desprenda da superfície, não girando em torno do eixo longitudinal. Para velocidades de vento superiores à nominal, o efeito estol reduz as forças de sustentação e aumenta as forças de arrasto. Por isso, as pás são projetadas para que esse efeito ocorra pelo menos parcialmente. Menores forças de sustentação e maiores forças de arrasto contrabalançam o aumento da potência do rotor, e uma pequena torção longitudinal é feita nas pás para evitar que o efeito ocorra simultaneamente em todas as posições radiais (Adaramola, 2014).

**Figura 2-9 Aerofólio com respectivo ângulo de passo e de ataque.**

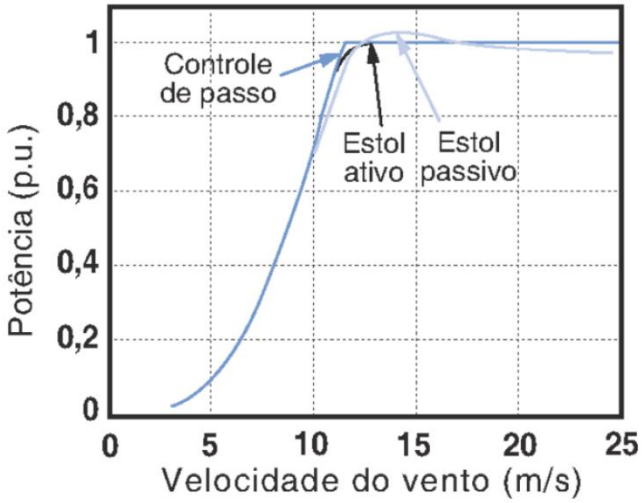



Fonte: CARVALHO (2003).

Já o controle por *pitch* é um sistema de controle ativo que requer informações de um controlador. Quando este indica que a potência nominal do gerador foi ultrapassada, as pás alteram seu ângulo de passo β, girando em torno do eixo longitudinal para reduzir o ângulo de ataque α e, assim, a potência extraída. Para velocidades de vento superiores à nominal, o ângulo é ajustado para que a turbina produza apenas a potência nominal. Em todas as condições de vento, o escoamento

de ar ao redor dos perfis das pás do rotor permanece aderente à superfície, gerando sustentação aerodinâmica e forças de arrasto reduzidas. No mecanismo por *pitch*, cada pá possui ajuste individual, mas as pás são sempre ajustadas de um modo síncrono em um mesmo ângulo. Por último, o controle por estol ativo combina os sistemas de estol e de passo. Nesse caso, o ângulo de passo da pá do rotor é ajustado ativamente na direção do estol (ou seja, aumentando o ângulo de ataque) e não na direção da posição de embandeiramento (menor sustentação), como nos sistemas de passo convencionais. Esta variante foi apresentada comercialmente por um curto período e existe em algumas turbinas de classe 2MW – 3MW. A Figura 2-10 apresenta curvas de potência de turbinas controlada por *pitch* e por estol.

**Figura 2-10 Curvas de potência de turbinas controlada por passo (*pitch*) e por estol ativo e passivo.**




Fonte: LIBERADO (2020).

As turbinas que operam usando o mecanismo de estol estão sujeitas a cargas aerodinâmicas não completamente previsíveis e, portanto, pás e rotores são dimensionados de maneira mais conservadora. Em função disso, rotores “estol” apresentam uma região cilíndrica (Figura 2-11).

**Figura 2-11 Turbina eólica com região cilíndrica na base da pá.**

As turbinas com *pitch*, que dominam o mercado atual, permitem um dimensionamento de pá mais delgado, o que contribui para uma maior eficiência da turbina (Heinzelmann, 2019). Ainda segundo a autora, as soluções técnico-conceituais com controle por *pitch*, velocidade do rotor variável, conexão à rede de forma indireta, via conversor de frequência, são o estado da arte das turbinas eólicas.

O eixo de acionamento de uma turbina eólica é composto por um eixo de transmissão principal, que será acoplado ou integrado ao gerador, com o objetivo de transmitir o torque gerado pelo rotor aerodinâmico ao gerador elétrico. Dependendo da configuração, alguns componentes podem ou não estar presentes, como por exemplo, a caixa de engrenagens (Heinzelmann, 2019).

Inicialmente, muitas soluções adotavam um eixo de acionamento expandido, que oferece a vantagem de melhor acesso às peças e subsistemas, facilitando a montagem, inspeção, manutenção e substituição de componentes. Entretanto, tanto para o eixo de acionamento expandido, quanto para o semi-compacto, a caixa de engrenagens está presente (no semi-compacto, o rolamento posterior do eixo principal está integrado à caixa de engrenagens).

No início do desenvolvimento de turbinas eólicas, ambas as configurações exigiam que o rotor fosse desmontado e o eixo mecânico retirado e levado para a fábrica para a substituição da *gearbox*. A substituição deste grande componente era muito comum, uma vez que as cargas aerodinâmicas a que as turbinas estavam expostas ainda não eram completamente compreendidas e definidas, impedindo o correto dimensionamento. Somando-se a erros na montagem e produção, em vez de um esperado ciclo de vida de 20 anos, o setor confrontava-se com a substituição de engrenagens após apenas 3 a 5 anos. Este tipo de parada impacta diretamente na disponibilidade da turbina, interferindo na produção de energia do parque.

A partir dessa custosa experiência, para turbinas de classes de potência mais elevadas, especialmente para as turbinas *offshore*, as soluções técnico-conceituais foram revistas e reprojetadas, incluindo não apenas o apoio posterior para o eixo de acionamento, mas também um guincho e uma abertura na nacele, que possibilitaram a inserção e retirada de peças e componentes.

Por outro lado, outros fabricantes optaram pela solução do eixo de acionamento compacto, visando reduzir o peso da nacele e eliminar a necessidade da caixa de engrenagens. Um exemplo são as turbinas da empresa alemã Enercon (no Brasil conhecida como Wobben). A Figura 2-2 apresenta exemplos de eixo de acionamento semi-compacto e compacto.

**Figura 2-2 À esquerda, um modelo com eixo de acionamento semi-compacto e, à direita, um modelo compacto.**

Fontes: HINE (2020), ENERCON (2021).

Em relação à evolução das pás de turbinas eólicas, houve um grande progresso em termos de projeto, dimensionamento, pesquisa e produção em série nos últimos anos. A evolução técnica e computacional permitiu que, além do uso de túneis de vento com infraestrutura de teste para pás de grandes dimensões e softwares para cálculos aerodinâmicos bidimensionais de perfis, softwares altamente especializados, comerciais e de domínio livre e de simulação CFD (*Computational Fluid Dynamics*), também se tornassem ferramentas essenciais para o estudo, melhoramento e otimização dos projetos de pás (Heinzelmann, 2019). No entanto, esses avanços técnicos não evitam diversos problemas que ocorrem com muitos fabricantes. Um exemplo é a Siemens Gamesa, que tem enfrentado problemas

significativos com suas turbinas eólicas no Brasil. Os defeitos, identificados principalmente nas pás, foram detectados nas novas plataformas 4.X e 5.X, resultando em incidentes como a quebra de pás e até fogo em uma turbina, gerando um custo estimado de 1 bilhão de euros. Empresas como AES Brasil e Engie foram impactadas, mas estão investigando as causas e avaliando soluções em colaboração com a Siemens Gamesa (Eixos, 2023).

#### 2.1.2.2 Curva de potência

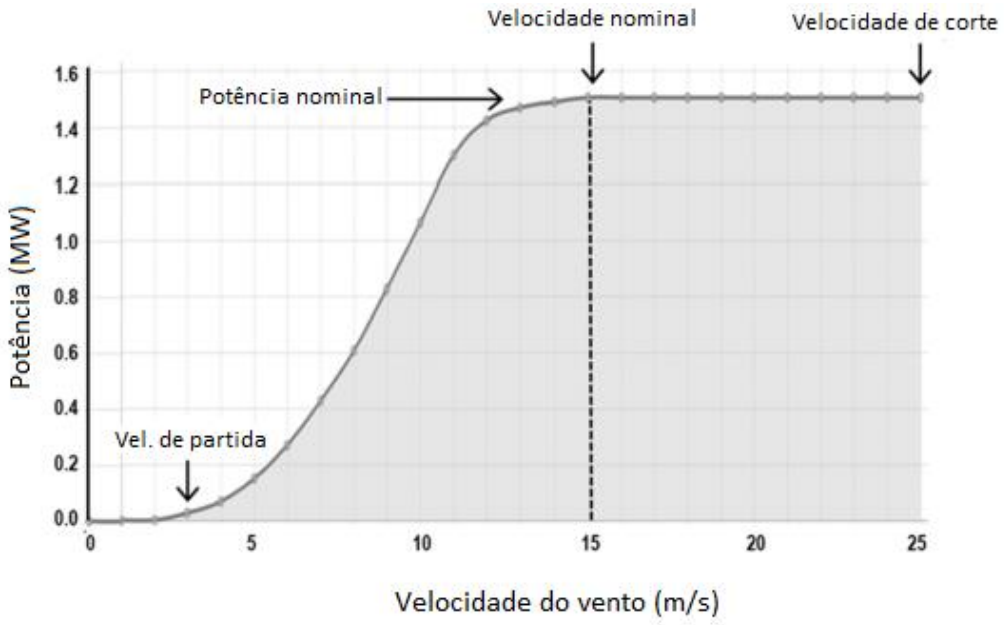
A relação entre potência de saída e velocidade do vento na altura do cubo é representada pela curva de potência da turbina. A menor velocidade na qual uma turbina consegue dar partida é chamada de velocidade de partida (ou *cut-in*). A velocidade nominal é a velocidade na qual a máquina atinge a potência nominal. Ao atingir esta velocidade, o sistema de controle é acionado com o objetivo de manter a potência de saída constante. A velocidade de corte, ou *cut-out* é aquela na qual a máquina se desliga, sendo um valor definido em projeto que visa a proteção da turbina contra carregamentos extremos.

Em algumas turbinas, há a chamada estratégia de histerese, na qual ao invés de haver um desligamento abrupto, a potência é progressivamente reduzida com o aumento da velocidade, até que se atinja a velocidade de corte. Ainda existe a chamada velocidade de "*recut-in*", que é a velocidade na qual a turbina volta a funcionar após ser desligada na velocidade de corte. A velocidade de *recut-in* é tipicamente menor do que a de *cut-out*. A Figura 2-12 ilustra uma curva de potência típica, com as velocidades de partida, nominal e de corte.

Tem-se, portanto, três regiões características em uma curva de potência:

1. potência igual a zero abaixo da velocidade de partida;
2. potência proporcional ao cubo da velocidade (na chamada parte cúbica da curva), advinda da Equação (1);
3. potência constante e igual à nominal.

**Figura 2-12 Curva de potência típica de uma turbina eólica.**




Fonte: adaptado de PAIK; CHUNG; KIM (2023)

Da Equação (1), tem-se que a potência de saída ao longo do tempo irá depender essencialmente da densidade do ar, da velocidade do vento e do coeficiente de potência. Considerando que a densidade do ar permanece praticamente constante na altura do cubo, resta apenas a velocidade e o coeficiente de potência, este dependendo do TSR e do ângulo de *pitch*.

Ao longo do desenvolvimento de um projeto eólico, em suas diversas fases, estimativas de produção de energia serão conduzidas para que se avalie a viabilidade técnica e financeira deste, e para que se cumpram as exigências regulatórias na emissão das outorgas. Para tais estimativas, duas informações são cruciais: a primeira é a velocidade do vento do local e a segunda é a curva de potência da turbina.

Quanto à primeira, medições da velocidade, direção, pressão, temperatura e umidade relativa do ar são registrados, de preferência em vários pontos do local do projeto, por equipamentos de medição confiáveis, ao longo de, no mínimo, 3 anos, conforme exigido pela ANEEL. Esses dados, em resolução de 10 minutos, serão limpos e manipulados, com o objetivo de se obter um recurso eólico representativo dos 20 anos de operação, no local das turbinas, na altura do cubo. O recurso, ao ser cruzado com a informação da curva de potência, fornecerá a energia bruta do parque eólico.

A curva de potência é obtida pela fabricante da turbina, em um local de testes, com uma turbina em funcionamento e uma torre anemométrica à montante em relação à direção predominante do vento, no intuito de capturar o escoamento livre. Em teoria, a curva de potência é definida em função do vento livre; porém, na prática, não é possível que esta medição seja feita. A norma IEC61400-12-1:2005 estabelece que, em uma medição de curva de potência, a torre anemométrica esteja entre 2 e 4 diâmetros de rotor (indicado como D) de distância da turbina, sendo perto o suficiente para que o escoamento esteja bem correlacionado com as condições da turbina, mas distante para que a influência da indução da turbina seja desprezível. Apesar dessas restrições, há evidências que mostram que a presença da turbina perturba o escoamento à montante: é o chamado efeito de bloqueio.

Em 2018, a DNV, com base em medições realizadas em três parques eólicos onshore e em simulações complementares, identificou que as velocidades do vento medidas a 2 D a montante dos parques eólicos apresentaram uma redução média de 3,4% após o início da operação das turbinas. As reduções observadas foram significativamente superiores ao que seria esperado apenas pela indução de uma única turbina, o que levou à conclusão de que outras turbinas do parque também contribuíram para esse efeito. Dessa forma, concluiu-se que o efeito de bloqueio não apenas reduz a velocidade do vento a montante do parque, mas também impacta as velocidades do vento incidentes nas turbinas posicionadas nessa região, fazendo com que, em geral, produzam menos do que produziriam se estivessem operando isoladamente (Bleeg et al., 2018). Até este momento, os chamados modelos de baixa fidelidade aplicados na indústria apenas consideravam os efeitos de esteira; a partir deste momento, fica constatada a importância de também se contabilizar o efeito de bloqueio tanto na curva de potência quanto no cálculo de produção de energia.

Em resumo, a medição da curva de potência é realizada em um local diferente do parque e por um período específico, com medições impactadas pelo efeito de bloqueio. Idealmente, seria necessária uma curva de potência obtida no local do projeto, representativa dos 20 anos de operação e baseada na velocidade do vento livre. Dessa forma, para que a curva de potência possa ser utilizada de forma precisa no cálculo de produção de energia, são necessários ajustes para compensar o efeito de bloqueio, as condições ambientais locais e a degradação ao longo do tempo.

O cenário descrito nos parágrafos anteriores ilustra o que ocorre antes do parque eólico entrar em operação, ainda no campo das estimativas, que servirão de referência para o modelo financeiro do projeto. Considerando as diversas incertezas inerentes à estimativa da curva de potência na fase pré-construtiva, é apenas na fase operacional que a curva de potência real poderá ser modelada, utilizando os dados de potência e velocidade capturados pelo SCADA. É por isso que a modelagem e a limpeza dessa curva ao longo da vida útil do parque são fundamentais, assegurando que a análise de desempenho reflita com precisão as condições operacionais reais.

#### 2.1.2.3 Medição de uma curva de potência a partir da IEC 61400-12-1:2005

Dada a importância da estimativa da curva de potência “mais próxima da real” de uma turbina, existe, ainda, um procedimento frequentemente executado no local do projeto durante a operação do parque: a medição ou teste da curva de potência segundo a norma IEC. A IEC 61400-12-1 fornece orientações para medição da curva, incluindo requisitos para os equipamentos utilizados, posicionamento dos mesmos, bem como critérios que devem ser atendidos durante a medição (Zou; Djokic, 2020). O procedimento descrito em seguida está de acordo com o que enuncia a norma IEC 61400-12-1, primeira edição, do ano de 2005.

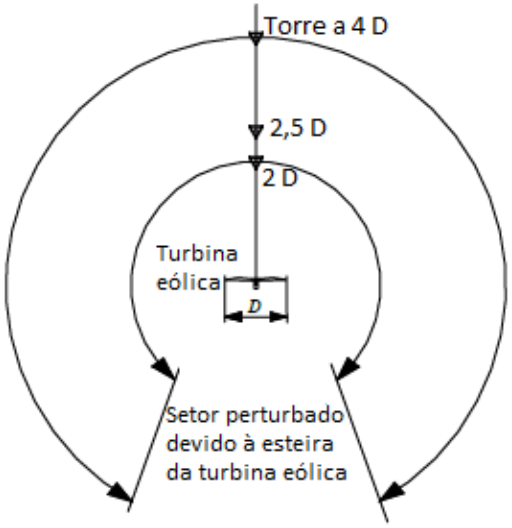
O objetivo de uma medição de curva de potência é coletar dados que atendam a critérios previamente definidos, garantindo quantidade e qualidade suficientes para determinar as características de desempenho da geração de energia de uma turbina eólica. Para isso, uma torre anemométrica deve ser posicionada entre 2 e 4 diâmetros de rotor (Figura 2-13). Este intervalo é para que a torre não seja posicionada muito perto da turbina (onde sofre pelo efeito de bloqueio), nem muito longe, onde o vento capturado pela torre já não correlacione com o vento experienciado pela turbina.

Além disso, de acordo com a IEC, as características do local podem influenciar significativamente o desempenho medido da turbina, especialmente devido a possíveis distorções no escoamento de vento, que podem causar diferenças entre a velocidade registrada na torre e aquela que realmente atinge a turbina. Por isso, é fundamental avaliar o local considerando fatores como topografia, presença de outras turbinas e obstáculos, como edifícios e árvores. Essa

análise permite posicionar corretamente a torre, aplicar correções ao escoamento e definir um setor de medição confiável para reduzir incertezas.

Durante o período de medição, a turbina eólica deve operar normalmente, sem alterações na configuração da máquina. O status operacional deve ser registrado por meio de sinais de status, e a manutenção regular deve ser realizada durante todo o período, com os trabalhos registrados em um “diário de teste”. Qualquer manutenção especial, como por exemplo lavagem das pás para melhorar o desempenho, deve ser devidamente anotada, mas só deve ser realizada se acordada previamente entre as partes contratantes.

**Figura 2-13 Distância da torre anemométrica à turbina eólica de 2 D a 4 D. Distância recomendada de 2,5 D.**




Fonte: adaptado de IEC 61400-12-1:2005.

Para garantir que apenas os dados obtidos durante a operação normal da turbina sejam utilizados na análise e que os dados não sejam corrompidos, devem ser excluídos os conjuntos de dados nos seguintes casos:

- Condições externas, exceto a velocidade do vento, fora da faixa operacional da turbina;
- Falha na turbina que impeça sua operação;
- Desligamento manual ou operação em modo de teste ou manutenção;
- Falha ou degradação dos equipamentos de teste (por exemplo, devido à formação de gelo);
- Direção do vento fora do setor de medição definido;

- Direções do vento fora dos setores válidos e completos de calibração do local.

Os conjuntos selecionados devem ser organizados utilizando o "método de bins". Os dados devem abranger, no mínimo, uma faixa de velocidade do vento que se estende desde 1 m/s abaixo da velocidade de entrada em operação (*cut-in*) até 1,5 vezes a velocidade do vento correspondente a 85% da potência nominal da turbina eólica. O banco de dados será considerado completo quando atender aos seguintes critérios:

- Cada bin deve conter, no mínimo, 30 minutos de dados amostrados;
- O banco de dados deve incluir, no total, no mínimo 180 horas de dados amostrados.

A curva de potência medida de acordo com a norma IEC tem, no entanto, suas limitações. Ela pode servir de referência para a turbina na qual está sendo medida, porém não é tão simples extrapolar para outras turbinas do parque, ainda mais em locais com elevada complexidade do terreno e do escoamento. Além disso, as possíveis causas para o desempenho subótimo não são exploradas na norma (Astolfi; De Caro; Vaccaro, 2023). É aí que entra a avaliação do especialista e a utilização dos dados SCADA.

## 2.2 MONITORAMENTO DE UMA CURVA DE POTÊNCIA DE UMA TURBINA EÓLICA OPERACIONAL

Ao se monitorar uma curva de potência se tem alguns objetivos principais (Lydia et al., 2014):

- Cálculo da produção de energia histórica e previsão futura;
- Monitoramento de performance com a avaliação de problemas de desempenho e disponibilidade;
- Monitoramento da condição, incluindo o controle preditivo e otimização da operação.

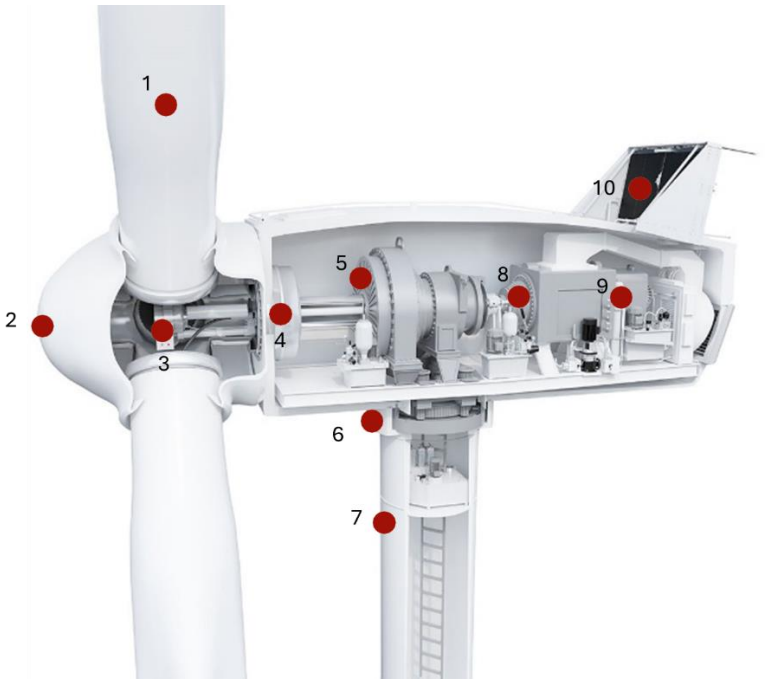
### 2.2.1. Dados SCADA

O sistema SCADA (*Supervisory Control and Data Acquisition System*) é uma solução composta por hardware e software voltada à supervisão, aquisição e

monitoramento de dados em tempo real. Esse sistema permite a coleta de informações e o controle de variáveis e dispositivos em sistemas industriais, viabilizando o gerenciamento eficiente de processos automatizados.

Sua arquitetura é formada por componentes como Unidades Terminais Remotas (RTUs), Controladores Lógico-Programáveis (CLPs) e uma interface gráfica que facilita a visualização e a interação com os dados operacionais. Além disso, o SCADA pode operar com protocolos de comunicação proprietários ou abertos, o que garante compatibilidade com equipamentos e softwares de diferentes fabricantes. O sistema pode ser instalado em um único computador ou distribuído em diversas máquinas, conforme a complexidade e as necessidades da planta. É comumente executado em computadores convencionais e utiliza sistemas operacionais amplamente conhecidos, como o Windows (Cravo, 2024). A Figura 2-14 traz uma ilustração do posicionamento de alguns sensores em uma turbina eólica e as respectivas grandezas medidas. Tipicamente mais de 300 variáveis são monitoradas (Marti-Puig et al., 2021).

**Figura 2-14 Posicionamento de alguns sensores para monitoramento SCADA em uma turbina eólica.**




Fonte: adaptado de HINE (2020).

| Posição | Variável |
|---|---|
| 1 | Ângulo de pitch |
| 2 | Velocidade do rotor |

| | |
|---|---|
| 3 | Pressão do atuador |
| 4 | Temperatura do rolamento principal |
| 5 | Temperatura do óleo da *gearbox* |
| 6 | Ângulo de y*aw* |
| 7 | Temperatura ambiente |
| 8 | Rotação do eixo do gerador |
| 9 | Corrente/Voltagem nos enrolamentos do gerador |
| 10 | Velocidade do vento |

### 2.2.2. Cálculo da produção de energia

O cálculo da produção histórica de energia de um parque eólico em operação baseia-se na curva de potência derivada de dados SCADA. Para obter essas curvas de referência, é essencial processar e limpar esses dados, excluindo registros inválidos e identificando corretamente, para posterior remoção, os pontos que indicam operações anômalas da turbina. Considerando-se uma amostra representativa de longo prazo, essa curva de potência histórica pode ser usada como referência para a operação normal da máquina ao longo de todo o período operacional. A estimativa da produção de energia de longo prazo de um parque eólico operacional conterá menos incerteza, se comparada a uma estimativa pré-construtiva. Além de refletir de forma mais precisa o desempenho real do ativo, essa estimativa funciona como uma espécie de recertificação para o operador, permitindo a atualização do planejamento financeiro para os anos remanescentes do projeto. Também é uma ferramenta estratégica para embasar decisões relacionadas à expansão do parque, à renegociação de contratos de energia ou à avaliação de viabilidade em processos de compra e venda de ativos.

### 2.2.3. Monitoramento de performance e de condição

Embora o sistema SCADA não tenha sido originalmente projetado para monitoramento por condição, a utilização de seus dados para avaliar a saúde das turbinas tornou-se uma prática amplamente adotada à medida que a otimização da manutenção passou a ser uma prioridade na indústria eólica (Tautz-Weinert; Watson, 2016). Uma das principais estratégias para reduzir os prejuízos financeiros de um parque eólico é a contenção dos custos de operação e manutenção (O&M), que podem representar até 30% dos custos totais ao longo da vida útil de um parque *onshore (May; McMillan; Thöns, 2015)*, e ainda mais nos casos de parques *offshore*. Isso destaca a crescente atenção dada ao monitoramento de performance e

condição das turbinas eólicas (Stetco et al., 2019), uma tarefa desafiadora devido à complexidade das turbinas e sua exposição a condições operacionais não estacionárias.

A previsão de falhas em estágio incipiente é desejável na redução de custos da operação e manutenção. A manutenção preditiva baseia-se no monitoramento de condição, fornecendo informações sobre equipamentos e componentes que provavelmente falharão e os substituindo no momento certo. A manutenção preditiva ajuda os gestores de ativos a preencherem a lacuna entre a manutenção reativa e a manutenção programada, realizando a manutenção não muito tarde nem muito cedo, mas no momento ideal. A manutenção preditiva pode ajudar a estimar o tempo até a falha (vida útil restante), detectar problemas em equipamentos (detecção de anomalias) e ajudar a identificar quais partes precisam ser consertadas (diagnóstico de tipos de falhas) (Udo; Muhammad, 2021).

Um dos sistemas muito utilizados é o CMS (*Condition Monitoring System*). O CMS monitora diversos parâmetros chave incluindo vibrações dos componentes da nacele, qualidade do óleo e temperatura em alguns conjuntos principais. Sistemas como estes são frequentemente implementados como complementos à configuração padrão das turbinas eólicas. No entanto, o aumento nos custos de operação e manutenção resultante dessa instalação desencorajou alguns operadores, apesar dos benefícios da detecção precoce de falhas por meio do CMS já terem sido provados (Yang; Court; Jiang, 2013).

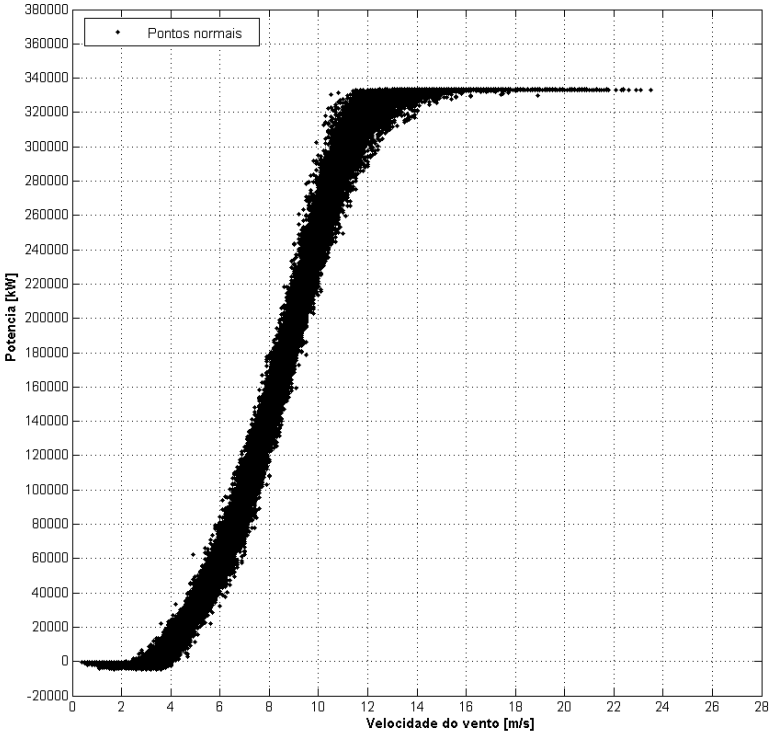
Todas as turbinas eólicas de grande escala já possuem um sistema SCADA padrão, que é utilizado principalmente para o monitoramento de desempenho. O uso de dados SCADA para o monitoramento de condições representa uma alternativa menos custosa, que não requer dados adicionais. Diversas metodologias baseadas nesses dados têm sido desenvolvidas ao longo dos últimos anos (Tautz-Weinert; Watson, 2016). Neste contexto, a curva de potência gerada a partir dos dados SCADA se destaca como uma ferramenta valiosa para a análise de desempenho, pois esses dados são facilmente acessíveis e oferecem uma abordagem mais econômica (Gonzalez et al., 2017).

No presente estudo, definimos como 'anormais' as instâncias que divergem do padrão predominante e, portanto, que se situam fora da trajetória principal da curva de potência. A identificação e remoção dessas anomalias são essenciais para evitar vieses nas análises realizadas (Morrison; Liu; Lin, 2022). Ressalta-se que esses

*outliers* nem sempre representam dados inválidos; em condições adversas, eles podem refletir o comportamento esperado da máquina e, portanto, são fisicamente coerentes. Entre os *outliers*, também se incluem dados efetivamente inválidos, que não atendem aos critérios de operação da turbina. Os pontos avaliados são divididos em quatro categorias:

1. Operação normal da turbina: são os pontos que seguem o padrão esperado de uma curva de potência (Figura 2-12). Podem apresentar pequenas variações ao longo da operação, resultando em uma curva mais ou menos dispersa. Um exemplo é mostrado na Figura 2-15.

**Figura 2-15 Pontos normais de uma curva de potência com dados SCADA.**

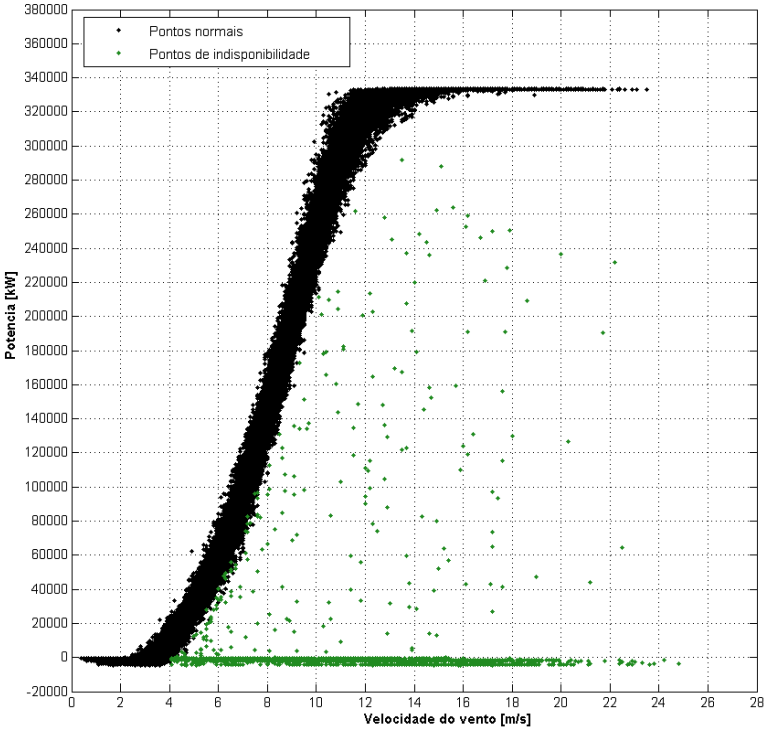



Fonte: A autora (2024).

2. Indisponibilidade: abrange tanto a indisponibilidade total (potência igual ou menor que zero – correspondendo ao consumo da turbina – e velocidade acima da velocidade de corte) quanto a parcial (pontos dispersos à direita da curva). A indisponibilidade parcial indica que, em um intervalo de 10 minutos, a turbina esteve parcialmente indisponível, resultando em um valor de potência integralizado entre zero e a potência esperada. Uma

curva de potência com exemplos de indisponibilidade total e parcial é apresentada na Figura 2-16.

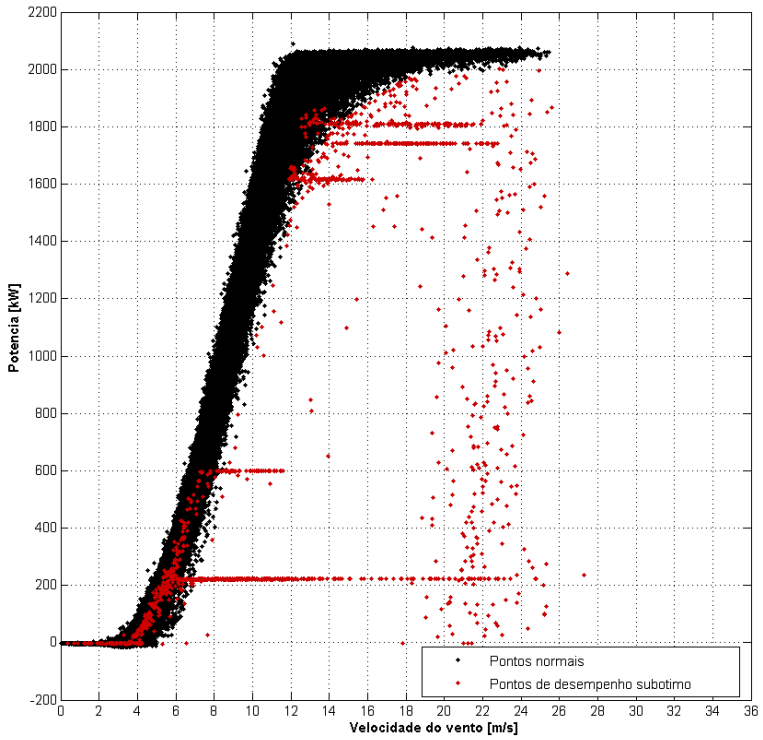
**Figura 2-16 Pontos normais e de indisponibilidade na curva de potência.**




Fonte: A autora (2024).

3. Desempenho subótimo (ou subdesempenho): inclui limitações de potência (*curtailments*) impostas pelo operador do sistema ou por restrições internas a nível de parque ou turbina, além de problemas no sistema de controle (*pitch* ou *yaw*), estratégias de desligamento por setor de direção do vento (*wind sector management*), desligamento por altas temperaturas (*temperature derating*), histerese (atraso na resposta da turbina no desligamento e religamento a mudanças nas condições de vento), dentre outros fatores de subdesempenho. O exemplo da Figura 2-17 apresenta limitações fixas de potência e histerese.

**Figura 2-17 Pontos normais e de subdesempenho na curva de potência.**

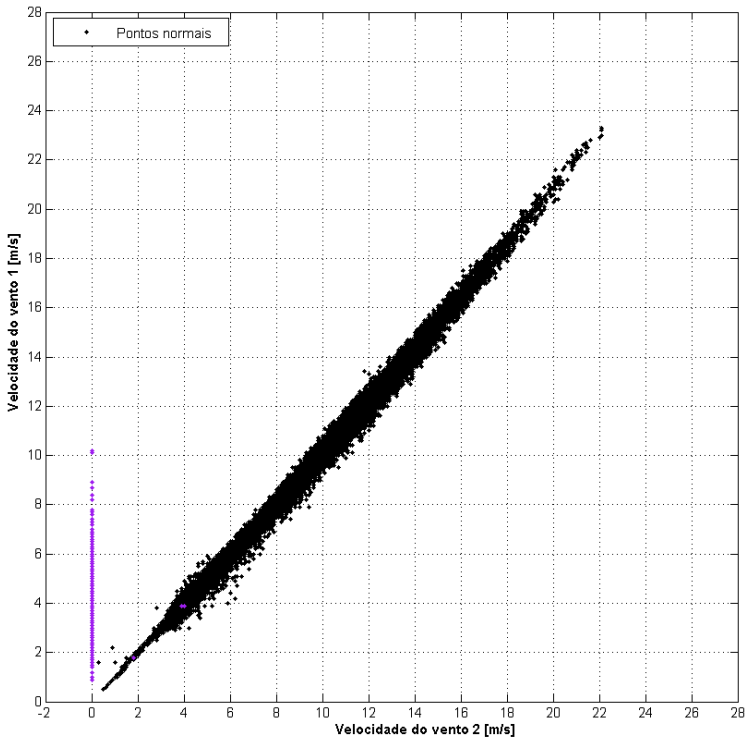



Fonte: A autora (2024).

4. Dados espúrios: registros decorrentes de erros no processamento e armazenamento de dados ou mau funcionamento de sensores. São considerados inválidos e devem ser excluídos. Podem existir dados espúrios de qualquer sinal. Instâncias espúrias podem ser observadas nos seguintes casos:
    a. Potência acima de zero para velocidades abaixo do *cut-in*;
    b. Valores fora de bandas aceitáveis, por exemplo:
        i. Velocidade negativa ou acima de 40 m/s;
        ii. Potência muito negativa (abaixo de -300kW) ou mais do que 10% acima da potência nominal;
    c. Velocidade do vento fixa em determinado valor enquanto a potência ou outro sensor de velocidade varia.

O exemplo da Figura 2-18 apresenta o item “c” descrito acima. Neste caso, dois anemômetros da nacele estão presentes nos dados SCADA, o que permite estabelecer a correlação entre os sinais.

**Figura 2-18 Pontos normais e espúrios (em roxo) da velocidade do vento 2.**

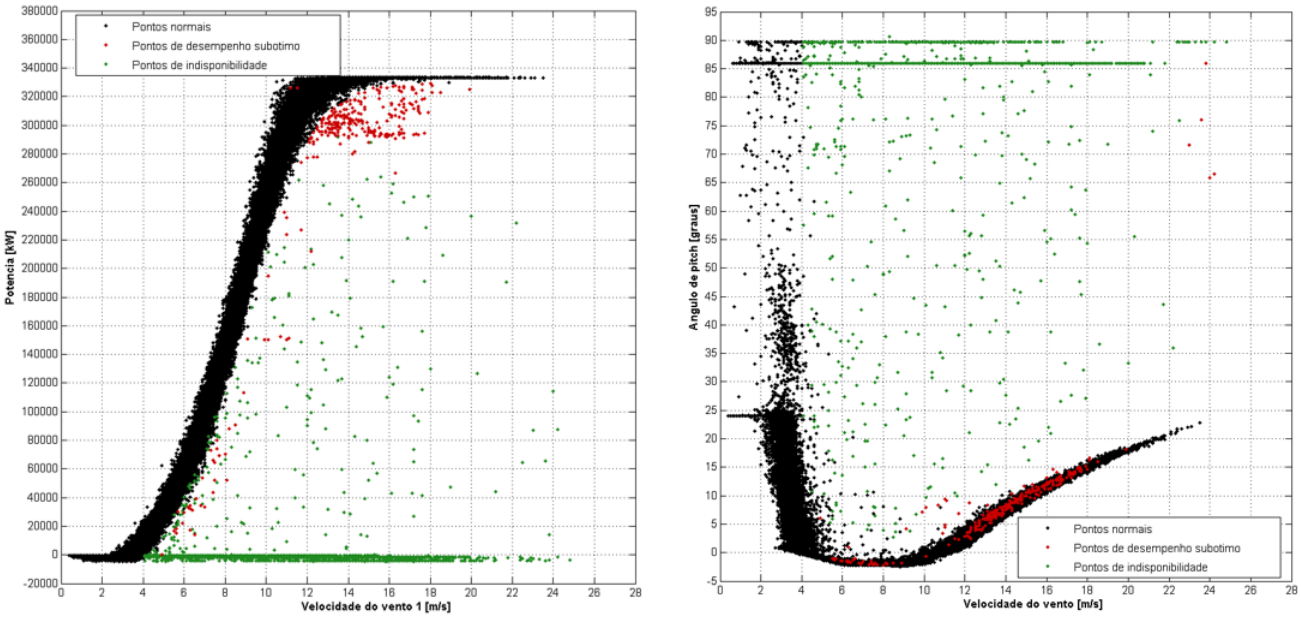



Fonte: A autora (2024).

Caso não existam dois sensores, é possível que a velocidade do anemômetro da nacele seja comparada com a velocidade média do parque, por exemplo.

Vale destacar que a distinção entre indisponibilidade e desempenho subótimo nem sempre é clara, já que alguns pontos podem facilmente ser confundidos, introduzindo uma incerteza inerente à classificação. Para reduzir essa incerteza, podem ser utilizados sinais adicionais do sistema SCADA, como velocidade do rotor, velocidade do gerador, direção da nacele, temperatura ambiente e ângulo de *pitch*, entre outros. Gráficos como potência versus velocidade do rotor ou ângulo de *pitch* versus velocidade do vento exibem comportamentos característicos que ajudam nessa distinção. A Figura 2-19 ilustra uma curva de potência e um gráfico de ângulo de *pitch* versus velocidade do vento com instâncias de indisponibilidade e subdesempenho. Os pontos no gráfico inferior auxiliam na distinção entre essas classificações: indisponibilidade geralmente é caracterizada por valores mais elevados de *pitch*, enquanto subdesempenho pode ocorrer em valores normais ou até mais baixos de *pitch*.

**Figura 2-19 Pontos de indisponibilidade e de subdesempenho na curva de potência (à esquerda) e *pitch* versus velocidade do vento (à direita)**




Fonte: A autora (2024).

Registros de paradas, mau funcionamento, falhas e períodos de manutenção das turbinas eólicas podem servir como informações auxiliares na identificação de anomalias (WANG et al., 2019).

A limpeza de uma curva de potência consiste, portanto, em classificar corretamente os dados em pontos normais ou anômalos — sendo estes últimos associados a condições de indisponibilidade, subdesempenho ou registros espúrios.

## 2.3 APRENDIZAGEM DE MÁQUINA

Aprendizado de Máquina (AM) é um ramo da Inteligência Artificial (IA) que tem como foco capacitar computadores e máquinas a imitarem a forma como os humanos aprendem, permitindo que realizem tarefas de maneira autônoma e melhorem seu desempenho e precisão com base na experiência e na exposição a novos dados (UC Berkeley, 2022).

O sistema de aprendizado de um algoritmo de AM pode ser dividido em três etapas (UC Berkeley, 2022):

1. Processo de decisão: em geral, os algoritmos são usados para fazer previsões ou classificações. Com base em dados de entrada (que podem ser rotulados ou não) o algoritmo gera uma estimativa sobre um padrão presente nesses dados;

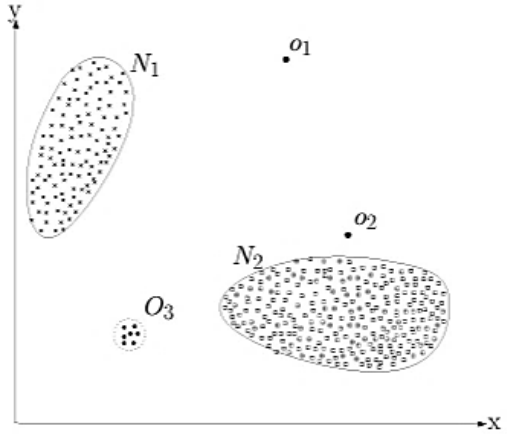
2. Função de erro: a função de erro avalia a previsão feita pelo modelo. Quando há exemplos conhecidos, essa função realiza uma comparação entre o resultado previsto e o valor real, permitindo medir a precisão do modelo.
3. Processo de otimização do modelo: caso o modelo possa se ajustar melhor aos dados do conjunto de treinamento, os pesos são modificados para reduzir a diferença entre os exemplos conhecidos e as estimativas do modelo. Esse processo iterativo de avaliar e otimizar é repetido diversas vezes, com o algoritmo atualizando os pesos de forma autônoma até atingir um nível de precisão satisfatório.

Entre as diversas aplicações do Aprendizado de Máquina, destaca-se a detecção de anomalias. Modelos de AM são particularmente eficazes nesse tipo de tarefa justamente por sua capacidade de aprender padrões complexos e reconhecer desvios sutis, muitas vezes imperceptíveis por métodos tradicionais de análise.

### 2.3.1 Algoritmos de detecção de anomalia

Uma anomalia pode ser entendida como uma mudança inesperada que exibe comportamentos significativamente divergentes em comparação com outras observações dentro de um determinado período (Ersoy; Erşahin; Kılınç, 2021). Em outras palavras, a detecção de anomalias consiste em identificar *outliers* em um conjunto de dados que apresentam características consideravelmente diferentes dos demais pontos, categorizando-os como desvios em relação ao padrão normal. A Figura 2-20 ilustra anomalias em um conjunto de dados bidimensional. Os dados normais possuem duas regiões, $N_1$ e $N_2$, já que a maior parte das observações caem nessas regiões. Pontos que estão suficientemente longe dessas regiões, como os pontos $o_1$ e $o_2$ e os da região $O_3$ são anomalias.

**Figura 2-20 Exemplo de anomalias em um conjunto de dados bidimensional.**




Fonte: CHANDOLA; BANERJEE; KUMAR (2009)

A obtenção de dados rotulados, sejam eles normais ou anômalos, de forma precisa e representativa, é frequentemente um processo caro e trabalhoso, especialmente por depender da expertise de especialistas humanos. Além disso, rotular instâncias anômalas é particularmente desafiador, pois as anomalias tendem a ser dinâmicas, com novos tipos surgindo sem registros prévios. Em contextos críticos, como a segurança aérea, as anomalias geralmente correspondem a eventos raros e catastróficos, o que dificulta ainda mais sua identificação e rotulação (Chandola; Banerjee; Kumar, 2009). Ainda segundo os autores, a rotulação pode operar nos três modos seguintes:

1. **Algoritmos supervisionados**: requer dados rotulados para classes normais e anômalas. Modelos preditivos são treinados para diferenciar as classes. Porém, há dois desafios principais:
   - Desequilíbrio entre instâncias normais (mais numerosas) e anômalas.
   - Dificuldade em obter rótulos representativos para anomalias. Algumas técnicas usam anomalias artificiais para contornar essa limitação.

   Dentre os algoritmos conhecidos de aprendizado supervisionado para detecção de anomalia podemos citar SVMs (*Support Vector Machines*), CNNs (*Convolutional Neural Networks*), RNNs (*Recurrent Neural Networks*), LSTM (*Long Short-Term Memory*), regressão logística, NB (*Nave Bayes*), KNNs (*K-Nearest Neighbors*) supervisionado, RF (*Random Forests*), árvores de decisão, etc (Kwon et al., 2019).
2. **Algoritmos semi-supervisionados:** treina o modelo apenas com instâncias normais, sendo mais flexível que a abordagem supervisionada. O modelo identifica

desvios do comportamento normal como anomalias. Rótulos de anomalias não são necessários, mas isso limita a capacidade de capturar comportamentos anômalos mais complexos. Como exemplos podemos citar o SVM de classe única, *autoencoders*, *isolation forest* adaptado, dentre outros.

3. **Algoritmos não-supervisionados:** não requer dados rotulados, sendo a mais amplamente aplicável. Assume que instâncias normais são muito mais frequentes que as anômalas. Se essa suposição estiver errada, ocorre alta taxa de alarmes falsos. Técnicas semi-supervisionadas podem ser adaptadas para operar de forma não-supervisionada, desde que as anomalias sejam raras no conjunto de dados. Podemos subdividir os exemplos em:

   - Algoritmos de agrupamento de dados: DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) e o *K-means*;
   - Modelos estatísticos: GMMs (*Gaussian Mixture Models);*
   - Redes neurais: *Autoenconders* e GANs (*Generative Adversarial Networks*);
   - Modelos baseados em distância: KNN e *Isolation Forest.*

No presente trabalho, o foco é dado ao DBSCAN e aos *autoencoders*, visto que são os algoritmos utilizados.

#### 2.3.1.1 Agrupamento de dados

Algoritmos de agrupamento de dados são um tipo de técnica de aprendizagem de máquina não supervisionada usada para agrupar pontos similares em *clusters*. O objetivo principal é encontrar agrupamentos naturais dos dados sem conhecimento prévio algum de classificação ou de categorias às quais os pontos pertençam (Paik; Chung; Kim, 2023)

Agrupamento de dados é amplamente utilizado para detecção de anomalias em diferentes contextos e fazem as seguintes suposições (Toshniwal et al., 2020):

1. Pertinência a *clusters*: instâncias de dados normais pertencem a um *cluster*, enquanto anomalias não pertencem a nenhum *cluster* nos dados.
2. Proximidade ao centroide: instâncias normais estão próximas ao centroide do *cluster* mais próximo, enquanto as anomalias estão distantes de seu centroide mais próximo.
3. Tamanho e densidade do *cluster*: instâncias normais pertencem a *clusters* grandes e densos, enquanto as anomalias pertencem a *clusters* pequenos ou esparsos.

Existem diversos algoritmos de agrupamento de dados, sendo os mais utilizados aqueles baseados em densidade e partição. Entre eles, destacam-se o DBSCAN e o *K-means*, respectivamente (Paik; Chung; Kim, 2023). Ainda segundo os autores, o DBSCAN é um algoritmo baseado na densidade de pontos capaz de identificar agrupamentos com formatos arbitrários sem a necessidade de especificar previamente o número de *clusters*. Ele funciona agrupando pontos que estão próximos uns dos outros e separando os *outliers*, com base na definição de uma vizinhança em torno dos pontos e na densidade local. O algoritmo possui dois parâmetros principais:

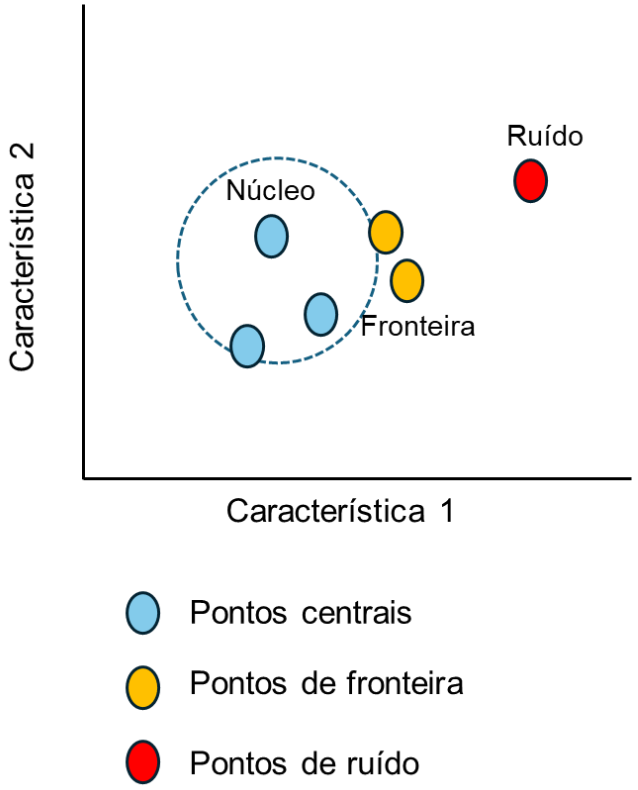
- Épsilon (ε): determina o raio da vizinhança ao redor de cada ponto.
- *Min_samples*: define o número mínimo de pontos necessários para que uma região seja considerada densa e, consequentemente, formar um *cluster*.

Em resumo, o DBSCAN se baseia em três conceitos principais:

- Pontos centrais: são pontos que possuem pelo menos um número mínimo de vizinhos (*min_samples*) dentro de um raio especificado (épsilon).
- Pontos de fronteira: são pontos que estão dentro da distância ε de um ponto central, mas que, por si só, não possuem vizinhos suficientes para serem considerados pontos centrais.
- Pontos de ruído: são pontos que não são nem centrais nem de fronteira. Eles estão distantes de qualquer cluster e, portanto, não são incluídos em nenhum agrupamento.

A Figura 2-21 ilustra esses três conceitos.

**Figura 2-21 Exemplo do agrupamento com o DBSCAN.**




Fonte: adaptado de KUMAR (2024).

### 2.3.2 Breve introdução às Redes Neurais Artificiais

As Redes Neurais Artificiais (ANNs - *Artificial Neural Networks*) são ferramentas bastante consolidadas, com origens que remontam a década de 1950 (Rosenblatt, 1958). São modelos computacionais projetados para capturar relações não-lineares complexas entre variáveis, utilizando conjuntos de dados de treinamento. Sua arquitetura básica é composta por uma camada de entrada, um número variável de camadas ocultas e uma camada de saída. Cada camada é formada por um conjunto de neurônios, que recebem entradas provenientes diretamente dos dados ou das ativações de neurônios em camadas anteriores (Tautz-Weinert; Watson, 2016).

As ANNs apresentam uma ampla variedade de tipos, mas todas compartilham o fato de serem algoritmos de aprendizagem de máquina utilizados para tarefas, a princípio, de regressão e classificação (aprendizado supervisionado) ou para aprendizado de representações (não supervisionado) (Helbing; Ritter, 2018). Os tipos mais comuns de ANNs para aprendizado supervisionado são a Rede Neural Multicamadas (MLP - *Multilayer Perceptron*), a Rede Neural Convolucional (CNN) e

a Rede Neural Recorrente (RNN). Para aprendizado não supervisionado, os tipos mais comuns são a Máquina de Boltzmann Restrita (RBM - *Restricted Boltzmann Machine*) e o *autoencoder*. Além disso, a literatura apresenta várias combinações e subtipos dessas redes (Lecun; Bengio; Hinton, 2015).

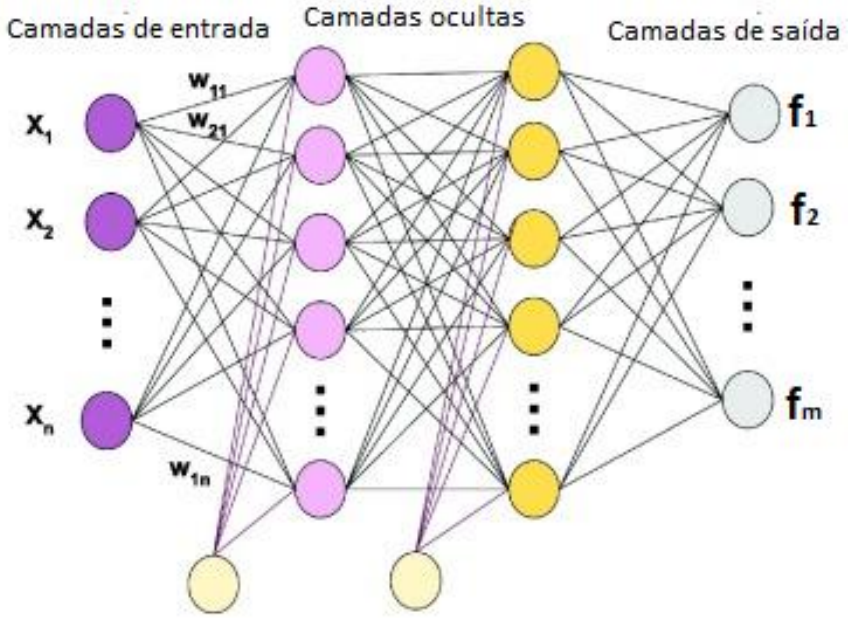
O foco geral do aprendizado de máquina está na representação dos dados de entrada e na generalização dos padrões aprendidos para uso em dados futuros ainda não observados (Najafabadi et al., 2015).

O processo de aprendizado da rede envolve o ajuste dos pesos associados às conexões entre os neurônios, normalmente através de algoritmos como o *backpropagation*, que atualiza os pesos com base no gradiente do erro calculado na saída. Cada neurônio utiliza uma função de transferência linear ou não-linear para combinar as entradas recebidas e aplica uma função de ativação, como ReLU, sigmoide ou tangente hiperbólica, para determinar a saída a ser propagada para a próxima camada. Entre as arquiteturas mais comuns estão as redes *feedforward*, caracterizadas pelo fluxo unidirecional de informações, da entrada para a saída. Estas contrastam com as RNNs, que possuem conexões retroalimentadas, permitindo o processamento de dados sequenciais ou com dependências temporais.

#### 2.3.2.1 MLP

Um *MultiLayer Perceptron* é uma rede *feedforward*, composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída (Haykin, 1994). Cada camada é formada por vários nós, cada um deles conectados a todos os nós da camada subsequente por meio de ligações ponderadas com números reais, conforme ilustrado na Figura 2-22.

**Figura 2-22 Esquema de um MLP.**




Fonte: Adaptado de CHAN et al. (2023).

Os dados de entrada $x_i$ são transmitidos para os nós das camadas ocultas através de conexões ponderadas. Em cada nó oculto $h_j$, as entradas recebidas são somadas, cada uma multiplicada por um peso específico. Isto corresponde ao teorema da aproximação universal e pode ser expresso por

$$f_j(x_1, \dots, x_n;\ w_{ij}, \dots, w_{nj}, b_j) = \sum_{i=1}^{n} w_{ij}\, x_i + \ b_j, \qquad \textbf{(20)}$$

em que $f_j$ é a saída do nó j, $x_i$ são suas entradas, $w_{ij}$ são os pesos das conexões com a camada anterior e $b_j$ é o viés. Os resultados dessas transformações passam por uma função de ativação não-linear, geralmente uma função sigmoide, tangente hiperbólica ou a chamada unidade linear retificada (*ReLU*). Por exemplo, a função sigmoide, frequentemente usada, transforma a saída do nó j conforme a Equação (21):

$$g(f_j) = \frac{e^{f_j}}{1+e^{f_j}}. \qquad \textbf{(21)}$$

Após essa transformação, os valores são encaminhados para todos os nós da camada subsequente por meio de conexões ponderadas, e o processo continua até que os dados transformados alcancem a camada de saída e passem pela última função de ativação. Esses valores constituem a saída da rede. Em resumo, um MLP é uma função não-linear $f: R^n \to R^m$, em que $n$ é a dimensão dos dados de entrada

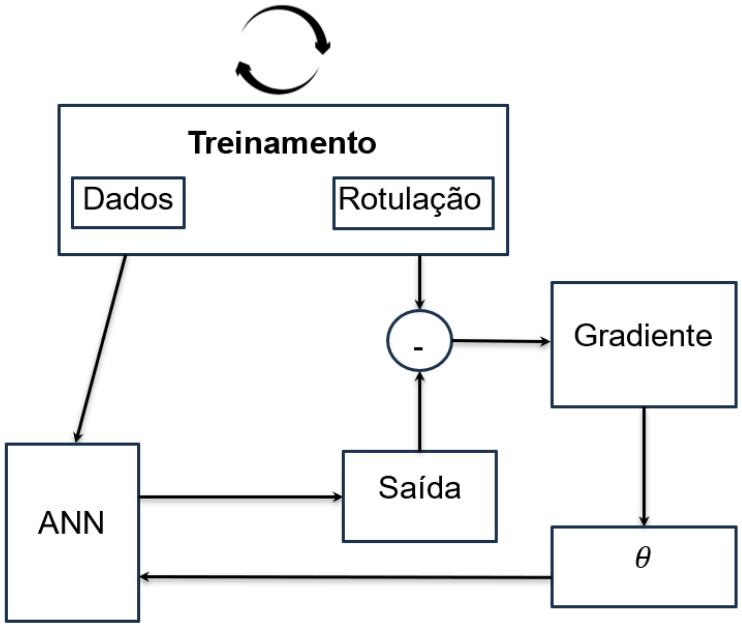
e $m$ a dimensão dos dados de saída. Treinar um MLP significa ajustar seus pesos e vieses para que a saída da rede sobre um conjunto de treinamento aproxime os valores verdadeiros (as rotulações) o máximo possível (Helbing; Ritter, 2018). Normalmente o erro quadrático $\rho_q$ é utilizado como medida de erro de predição. Dado o rótulo $o\ (x_1, \ldots, x_n) \in R^n$ , o erro quadrático $\rho_q$ da saída $\tilde{o}\ (x_1, \ldots, x_m) \in R^m$ é calculado como

$$\rho_q = \frac{1}{2}\sum_{k=1}^{m}(o_k - \tilde{o}_k)^2, \quad \textbf{(22)}$$

em que $o_k$ representa os valores verdadeiros, $\tilde{o}_k$ as previsões da rede e o fator de $\frac{1}{2}$ é usado para facilitar na derivação de certas propriedades.

O algoritmo de *backpropagation* é utilizado para calcular o gradiente do erro quadrático em relação aos pesos e vieses da rede. Esses gradientes são então aplicados em um algoritmo de otimização, como o gradiente descendente estocástico, para ajustar os pesos e minimizar o erro de forma eficiente. Um desenho esquemático sobre o treinamento de um MLP é apresentado na Figura 2-23.

Figura 2-23 Diagrama de um processo de treinamento de um MLP.




Fonte: adaptado de CHAN et al. (2023)

O treinamento ocorre ao alimentar a ANN com uma sequência de entradas de dados de forma iterativa. No aprendizado supervisionado, cada entrada inclui os

dados e seu rótulo correspondente. Já no aprendizado não supervisionado, o rótulo é simplesmente o próprio dado de entrada. A saída da rede é comparada ao rótulo, e a diferença é processada por uma função de perda. Os gradientes dessa perda são calculados para cada parâmetro $\theta$ usando *backpropagation*. Por fim, os parâmetros $\theta$ são ajustados, geralmente com métodos como o gradiente descendente estocástico (Helbing; Ritter, 2018).

Frequentemente, os pesos do MLP são inicializados com valores aleatórios no início do treinamento e, em seguida, otimizados iterativamente. No entanto, descobriu-se que esse procedimento leva a resultados progressivamente piores à medida que o MLP se torna mais profundo (ou seja, com mais camadas). Isso ocorre devido à natureza do algoritmo do *backpropagation*, no qual os gradientes tendem a diminuir quanto mais distante sua camada está da camada de saída. Este fenômeno é chamado de o problema dos "*vanishing gradients*" (Schmidhuber, 2015). Essa dificuldade em treinar MLPs com mais de algumas camadas pode explicar por que muitas aplicações utilizam apenas uma camada oculta.

Em resumo, os MLPs possuem a capacidade de representação de funções não-lineares, sendo fundamentos no teorema da aproximação universal (Hornik; Stinchcome; White, 1989). O teorema da aproximação universal estabelece que uma rede neural com pelo menos uma camada oculta e um número suficiente de neurônios, utilizando uma função de ativação não-linear pode aproximar, com um grau arbitrário de precisão, qualquer função contínua definida em um espaço de dimensão finita. Essa propriedade torna os MLPs ferramentas extremamente poderosas, pois são capazes de capturar a complexidade de relações não lineares em dados reais, independentemente do formato ou da origem dos dados. No entanto, o teorema não fornece garantias sobre a eficiência computacional ou o número de neurônios necessários para alcançar essa aproximação, o que é um ponto crítico na prática.

#### 2.3.2.2 *Autoenconders*

*Autoencoders* são um tipo especial de rede neural *feedforward*, podendo ser semi-supervisionada ou não supervisionada, composta por uma camada de entrada e uma camada oculta, ambas totalmente conectadas, como apresentado na Figura 2-24. A principal aplicação do *autoencoder* é capturar aspectos chave dos dados fornecidos. Assim, ele é treinado para reconstruir os dados de entrada, e, para isso,

estes são mapeados para a camada oculta (ou seja, os dados são "codificados"). Esta camada normalmente contém menos nós do que a camada de entrada; portanto, há uma compressão dos dados.

Dada a função de ativação $g$ e o vetor de entrada $x$ de dimensão $n$, a codificação $h(x)$, de dimensão $m$, é calculada como:

$$h_j(x) = g(\sum_{i=1}^{n} w_{ij}x_i + b_j), \qquad j \in 1, \dots, m, \tag{23}$$

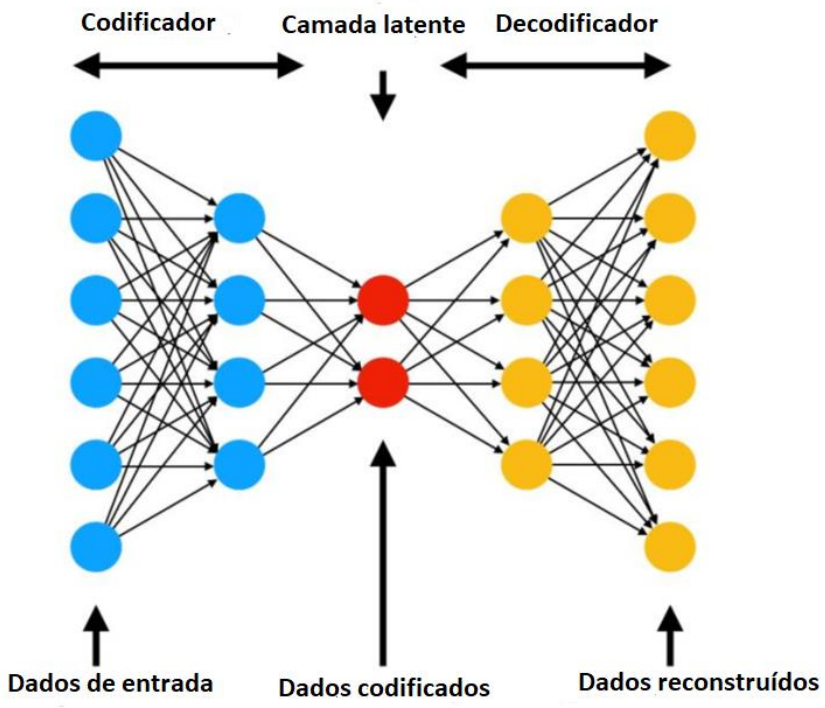
em que $w_{ij}$ são os pesos das conexões com a camada anterior e $b_j$ é o viés. Da camada latente, os dados transformados vão até a camada de saída, onde os dados são retransformados (decodificados) e o erro quadrático de reconstrução entre a entrada e a saída é computado. Assim, a saída do *autoenconder* durante o treinamento pode ser calculada por

$$\tilde{o}_i(x) = g(\sum_{j=1}^{m} w_{ji}{}' h_j(x) + b_i{}'), \; i \in 1, \dots, n, \tag{24}$$

e o erro de reconstrução é dado por

$$\rho_q = \sum_{i=1}^{n} (\tilde{o}_i(x) - x)^2, \tag{25}$$

**Figura 2-24 Esquema de um *autoenconder*.**



Adaptado de AGGARWAL (2023).

Durante o processo de treino, o *autoenconder* aprende a comprimir os dados de entrada de modo a preservar o máximo de informação possível. Deste modo, o *autoenconder* é uma variante não-linear do algoritmo de Análise de Componentes Principais (PCA). Vale notar que a composição dos codificadores tem a mesma estrutura de um MLP, mas a rede é treinada de maneira incremental e sem rótulos (Helbing; Ritter, 2018).

Existem diversos pacotes de software que permitem implementar as ANNs “rasas” em linguagens de programação populares. Exemplos incluem o pacote do R (Günther; Fritsch, 2010), o módulo Scikit-learn do Python (Pedregosa et al., 2011) e o Neural Network Toolbox do Matlab (Hudson et al., 1992).

Por outro lado, as aplicações de *Deep Learning* demandam mais recursos computacionais, pois consistem em muitos neurônios interconectados e geralmente requerem grandes volumes de dados para treinamento. Para atender a essas demandas, surgiram *frameworks* especializados. A maioria desses *frameworks* utiliza um *backend* em C++ combinado com APIs para linguagens amplamente usadas, como Python, permitindo que os analistas de dados se concentrem na modelagem, sem se preocupar com detalhes técnicos, como o uso de GPUs via APIs como CUDA (Helbing; Ritter, 2018).

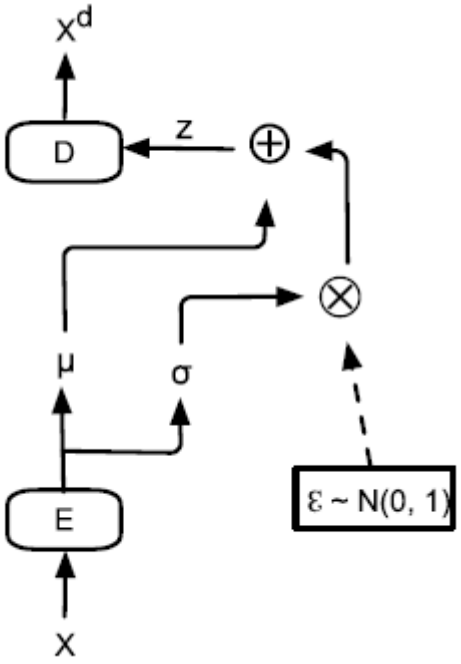
No presente trabalho, módulos como o Scikit-learn do Python e os *frameworks* do TensorFlow, Pytorch e Keras são utilizados. A plataforma utilizada para execução dos códigos em Python foi o VSCode e o Google Colab, esta última sendo uma plataforma baseada na nuvem, que oferece acesso a GPUs e TPUs.

3.3.2.2.1. *Autoencoders* variacionais

O *Autoencoder* Variacional (VAE) foi proposto por KINGMA & WELLING (2013) e é baseado na inferência variacional Bayesiana. Seu princípio fundamental é mapear um conjunto de dados para uma distribuição Gaussiana por meio de um codificador (*encoder*). A partir dessa distribuição, novas amostras são geradas e utilizadas como entrada para o decodificador (*decoder*), que tem a função de reconstruir os dados originais.

A estrutura do VAE é ilustrada na Figura 3.23, em que E e D representam o *encoder* e o *decoder*, respectivamente.

**Figura 2-25 Estrutura de um *autoencoder* variacional.**




Fonte: LI; PEI; LI (2023)

Os símbolos ⊗ e ⊕ representam, respectivamente, a multiplicação e a adição elemento a elemento entre vetores. O codificador recebe X como entrada para calcular μ e σ, introduzindo uma distribuição Gaussiana ε para obter a codificação probabilística Z. Em seguida, o decodificador processa Z para reconstruir X (Li; Pei; Li, 2023).

Supondo que existe um conjunto de funções capazes de gerar X a partir de Z (em que cada função é determinada a partir de um parâmetro θ), o objetivo da otimização do VAE é maximizar a probabilidade P(x) de geração de X, ajustando θ sob a premissa de que Z é amostrado. P(x) é dado por

$$P(x) = \int f(x|z)P(z)dz. \quad \textbf{(26)}$$

O VAE obtém a distribuição de probabilidade da variável latente Z ao adicionar uma rede de codificação que atua como um mecanismo de inferência porque aproxima a relação entre os dados observáveis (X) e as variáveis latentes (Z) escondidas no modelo. Para isso, é introduzida a função Q(z|x), responsável por atuar como a rede de codificação. O objetivo dessa função é determinar a distribuição da variável latente Z que permite reconstruir X, dado X como entrada.

Queremos que Q(z|x) seja o mais próximo possível da distribuição ideal P(z|x). Para medir essa similaridade, utiliza-se a divergência de *Kullback-Leibler*, representada por D, conforme apresentado na Equação (27).

$$D[Q((z|x) \,||\, \mathrm{P}((z|x)] \;= E_{Q((Z|X))}[\log Q\big((z|x)\big) - logP((z|x))] \qquad \textbf{(27)}$$

Em seguida, P(x|z) é expandido utilizando a fórmula de Bayes, resultando na fórmula (28) após simplificações. A partir disso, obtém-se a função de perda do VAE, apresentada na Equação (29),

$$logP(x) - \; D[Q((z|x) \,||\, \mathrm{P}((z|x)] \;=\; E_{Q((Z|X))}[\log P((x|z)) - D[Q((z|x) \,||\, \mathrm{P}(z)] \qquad \textbf{(28)}$$

$$J_{VAE} = \; E_{Q((Z|X))}[\log P((x|z)) - D[Q((z|x) \,||\, \mathrm{P}(z)] \;. \qquad \textbf{(29)}$$

A função de perda do VAE é composta por duas partes: a primeira impõe uma restrição à variável latente Z, garantindo que siga uma distribuição padrão; a segunda busca minimizar a diferença entre os dados reconstruídos e os dados de entrada, tornando o resultado final o mais próximo possível dos dados originais.

#### 2.3.2.3 Redes Kolmogorov-Arnold

Ao longo dos últimos anos, diversos autores têm proposto alternativas aos MLPs, cada uma projetada para lidar com tipos específicos de problemas e dados, ampliando a aplicabilidade das redes neurais tradicionais. Entre essas alternativas destacam-se as já mencionadas CNNs e as RNNs. As CNNs são projetadas para explorar as relações espaciais entre os dados de entrada, sendo amplamente utilizadas em tarefas como reconhecimento de imagens e análise de vídeos (Krizhevsky; Sutskever; Hinton, 2013). Por outro lado, as RNNs são especialmente adequadas para o processamento de dados sequenciais, como séries temporais ou textos em linguagem natural, devido à sua capacidade de capturar dependências temporais entre os elementos da sequência (Graves; Mohamed; Hinton, 2013).

Em 2024, LIU et al. propuseram uma nova arquitetura de redes neurais, denominada Redes Kolmogorov-Arnold (KANs), que se destacam como uma abordagem inovadora frente aos modelos tradicionais, como os MLPs. Diferentemente dessas redes especializadas, as KANs oferecem uma abordagem

mais geral, fundamentada no teorema de Kolmogorov-Arnold, que endereça a representação de funções multivariáveis usando funções mais simples, de apenas uma variável. O teorema de Vladimir Arnold e Andrey Kolmogorov estabelece que sendo $f$ uma função contínua, multivariável em um domínio fechado, então $f$ pode ser escrita como uma composição finita de funções contínuas de uma única variável e a operação da adição (Kolmogorov, 1957). Mais especificamente, sendo $f: [0,1] \rightarrow R$,

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^{n} \Phi_{q,p}(x_p) \right), \quad \textbf{(30)}$$

em que $\Phi_{q,p}: [0,1] \rightarrow R$ e $\Phi_q: R \rightarrow R$.

No campo do aprendizado de máquina, a aproximação de funções desempenha um papel importante, e o teorema de Kolmogorov-Arnold poderia, em teoria, ter aplicações significativas. No entanto, na prática, as funções univariadas resultantes da decomposição podem ser não suaves ou até mesmo apresentar um comportamento irregular, tornando-as extremamente difíceis de aprender. Por essa razão, apesar de sua robustez teórica, o teorema foi amplamente considerado impraticável para aplicações em aprendizado de máquina, sendo efetivamente relegado a um papel marginal na área (Girosi; Poggio, 1989). Todavia, para resolver tais limitações LIU et al. (2024) não aderiram estritamente à formulação original. Inicialmente, eles partiram do princípio de que em um problema de aprendizado supervisionado, tem-se pares de entrada-saída $\{x_i, y_i\}$, em que se quer encontrar uma função $f$ tal que $y_i \approx f(x_i)$. De acordo com a Equação (30), isto pode ser feito caso se consiga determinar funções univariadas apropriadas $\Phi_{q,p}$ e $\Phi_q$. Com base nisso, a ideia dos autores foi a de projetar uma rede neural onde todas as funções a serem aprendidas seriam univariadas e parametrizadas como uma curva *B-spline*, com coeficientes ajustáveis, seguindo o teorema de Kolmogorov-Arnold. Uma curva *B-spline* pode ser representada por

$$spline\ (x) = \sum_i c_i B_i(x), \quad \textbf{(31)}$$

em que $c_i$ são coeficientes que determinam o peso de cada função base e $B_i(x)$ são funções que determinam como cada intervalo do domínio contribui para a curva final. Essas funções base são definidas em termos de nós, que dividem o domínio em intervalos específicos. Entretanto, como mencionado, tal rede seria muito simples para aproximar funções arbitrárias com apenas *splines.*

Eis que entra a contribuição dos autores ao fazer uma analogia entre MLP e KAN. No MLP uma camada é composta por transformações lineares seguidas por funções de ativação não-lineares, podendo-se empilhar várias camadas para tornar a rede mais profunda. Uma camada KAN com entradas de dimensões $n_{in}$ e saídas de dimensões $n_{out}$ é representada por uma matriz de funções univariadas,

$$\Phi = \{\Phi_{q,p}\}, p = 1,2, \dots, n_{in}, \qquad q = 1,2, \dots, n_{out}, \tag{32}$$

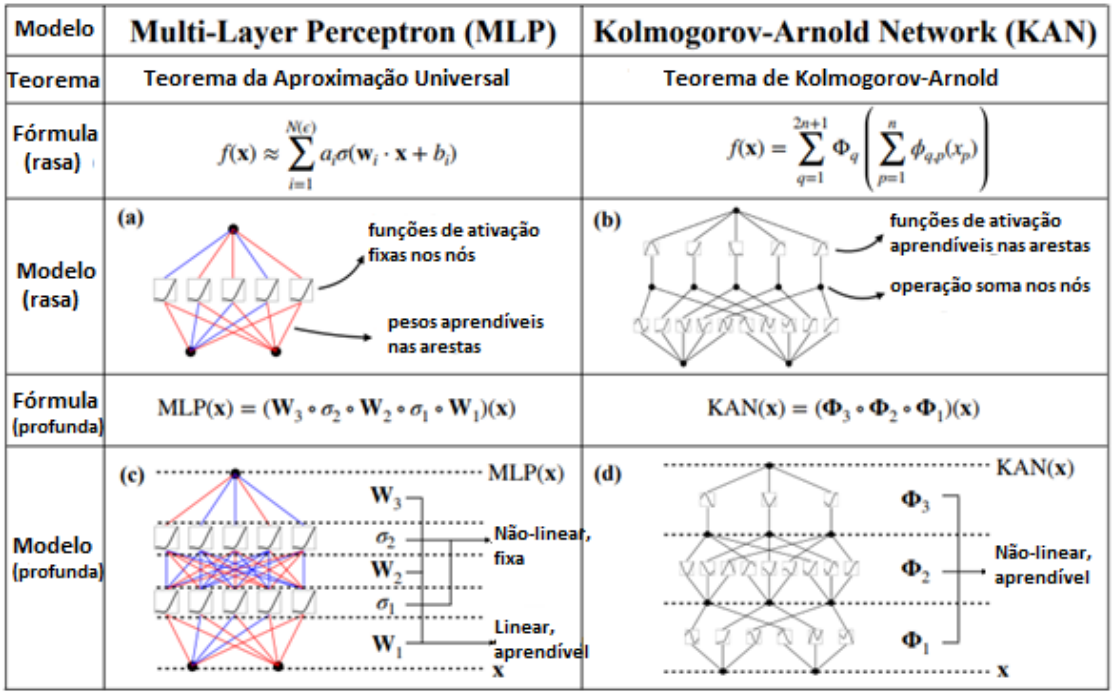
em que as funções $\Phi_{q,p}$ possuem parâmetros treináveis. Sendo assim,

- As funções internas formam uma camada KAN com $n_{in} = n$ e $n_{out} = 2n + 1$;
- As funções externas formam outra camada KAN com $n_{in} = 2n + 1$ e $n_{out} = 1$.

Isto significa que as representações descritas na Equação (30) podem ser vistas como a composição de duas camadas KAN. Os autores propõem então generalizar a rede para largura e profundidades arbitrárias, ao invés de duas camadas e um número de termos $2n + 1$.

Em essência a KAN é uma rede neural que aplica funções de ativação aprendíveis nas arestas, ao invés de funções de ativação fixas nos nós, como nos MLPs. Isto permite que qualquer parâmetro de peso seja substituído por uma função univariada. Cada nó no KAN soma as funções sem aplicar nenhuma transformação não-linear, ao contrário do MLP. Além disso, a flexibilidade das *splines* permite modelar de maneira adaptativa complexas relações nos dados, ajustando a forma para minimizar o erro de aproximação e, consequentemente, melhorando a capacidade da rede de aprender determinados padrões de alta dimensão. A Figura 2-26 apresenta as principais diferenças entre o MLP e a KAN.

**Figura 2-26 Principais diferenças entre MLP e KAN.**




Fonte: adaptado de LIU et al. (2024)

### 2.3.2.4. Métricas de algoritmos de classificação

Neste trabalho, a KAN é empregada como algoritmo de classificação e seu desempenho é avaliado por meio de diferentes métricas. Além da acurácia, são consideradas outras métricas relevantes, com o objetivo de proporcionar uma análise mais abrangente da performance do modelo. As métricas adotadas são descritas a seguir.

1. Acurácia

A acurácia mede, de forma simples, a proporção de previsões corretas realizadas por um modelo. Ela é definida como a razão entre o número de acertos e o total de previsões realizadas, e é dada por

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN}, \quad \textbf{(33)}$$

em que:

- TP - verdadeiro positivo: prevê positivo e é positivo;
- TN - verdadeiro negativo: prevê negativo e é negativo;
- FP - falso positivo: prevê positivo e é negativo;
- FN – falso negativo: prevê negativo e é positivo.

A acurácia é uma métrica adequada para conjuntos de dados balanceados, mas pode ser enganosa em cenários desbalanceados, pois tende a mascarar o real desempenho do modelo. Por exemplo, em um problema binário com 99 instâncias da classe 0 e apenas 1 da classe 1, um modelo que classifica todas as amostras como pertencentes à classe 0 atingirá 99% de acurácia. Embora esse valor pareça alto, o modelo falha completamente em identificar a classe 1 — que, em aplicações reais, costuma representar eventos críticos, como falhas, fraudes com cartão de crédito ou spam em e-mails. Nesses casos, outras métricas são mais indicadas para avaliar a performance do modelo de forma mais apropriada.

2. Precisão

Explica quantos dos casos corretamente previstos como TP de fato se tornaram positivos. É a razão entre os verdadeiros positivos e o total de instâncias classificadas como positivas,

$$Precisão = \frac{TP}{TP + FP}. \quad \textbf{(34)}$$

Essa métrica é especialmente relevante em cenários em que o custo de uma falsa detecção positiva é elevado, como em sistemas de detecção de fraudes.

3. *Recall* (sensibilidade)

Explica quantos casos de verdadeiros positivos foram corretamente identificados. É uma métrica interessante para quando falsos negativos são mais preocupantes do que falsos positivos, como por exemplo em diagnósticos médicos. Prever que um paciente está com uma doença que ele não está é menos crítico do que deixar passar uma doença que existe. *Recall* é dada por

$$Recall = \frac{TP}{TP + FN}. \quad \textbf{(35)}$$

4. *F1-score*

Combina precisão e *recall*, e é dada pela expressão

$$F1 - score = 2\frac{Precisão\ .\ Recall}{Precisão + Recall}. \quad \textbf{(36)}$$

O *F1-score* pune valores extremos. Geralmente é relevante quando falsos negativos e falsos positivos são igualmente custosos.

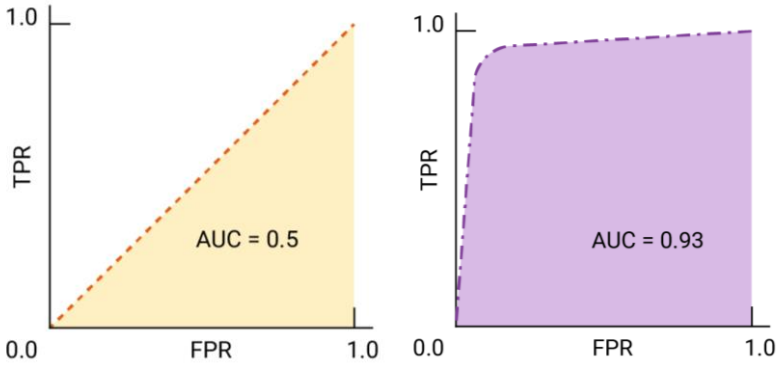
5. AUC (área sob a curva) ROC

A curva ROC (*Receiver Operating Characteristic*) é uma representação gráfica que avalia o desempenho de um modelo de classificação, plotando a taxa de verdadeiros positivos (sensibilidade) contra a taxa de falsos positivos em diferentes limiares de decisão. A métrica associada à curva ROC é a AUC *(Area Under the Curve*), que quantifica a capacidade do modelo de distinguir entre as classes. Por exemplo:

- Um modelo com AUC = 1 possui separação perfeita entre as classes.
- Um AUC = 0,5 indica desempenho equivalente ao acaso, ou seja, o modelo não tem capacidade discriminativa.
- Já um AUC = 0 representa um classificador que inverte totalmente as previsões (erra tudo).

A Figura 2-27 ilustra dois exemplos de área sob a curva ROC, com valores de 0,5 e 0,93, evidenciando a diferença entre um modelo sem capacidade de discriminação e outro com bom desempenho.

**Figura 2-27 Exemplos de curva ROC e respectivos valores de AUC.**




Fonte: Classification: ROC and AUC | Machine Learning | Google for Developers (2024)

### 2.3.3 Considerações da fundamentação teórica

Este capítulo apresentou os principais conceitos teóricos que sustentam a proposta desenvolvida nesta dissertação. Inicialmente, foram discutidos os fundamentos da energia eólica e a importância das curvas de potência como ferramenta essencial para o monitoramento e avaliação do desempenho de turbinas, com destaque para o papel dos sistemas de aquisição de dados (SCADA). A partir

daí, exploraram-se os paradigmas do Aprendizado de Máquina e suas aplicações na detecção de anomalias. Por fim, foram apresentados os principais modelos de redes neurais utilizados em tarefas de classificação e modelagem de comportamento. Todos esses conceitos convergem para a construção da solução proposta neste trabalho, que visa automatizar a limpeza de curvas de potência por meio de técnicas de aprendizado de máquina.

# 3 REVISÃO DA LITERATURA

## 3.1 LIMPEZA DA CURVA DE POTÊNCIA

O SCADA desempenha um papel crucial no monitoramento da condição e do desempenho das turbinas eólicas (CAMBRON et al., 2016).De acordo com KUSIAK (2016), o monitoramento orientado por dados pode reduzir os custos de manutenção de um parque eólico em até 10%. O autor defende a importância do acesso aberto a dados sobre o desempenho das turbinas eólicas para otimizar o funcionamento dos parques através da mineração de dados. KUSIAK enfatiza que a indústria energética poderia melhorar significativamente sua eficiência e inovação ao permitir que pesquisadores tenham acesso aos dados.

No geral, um processo de limpeza de uma curva de potência envolve distinguir corretamente os dados considerados normais dos anômalos, de forma eficiente, classificando os últimos corretamente (WANG et al., 2019). Diversos estudos têm explorado métodos variados para detecção e limpeza de dados anômalos em uma curva de potência. Nas próximas seções, os estudos são divididos em três grandes grupos: aqueles que aplicam métodos baseados em regressão e modelos estatísticos, os que utilizam algoritmos baseados em agrupamento de dados e análise de distância e os que utilizam como base métodos de aprendizado de máquina.

- **Métodos baseados em regressão e modelos estatísticos**

TASLIMI-RENANI et al. (2016) propuseram um modelo paramétrico baseado na tangente hiperbólica (MHTan) e empregaram o erro quadrático mínimo e a estimativa de máxima verossimilhança para estimar os parâmetros. Também avaliaram a utilização de outros modelos paramétricos e não-paramétricos e compararam o desempenho de todos os modelos com dados reais coletados de um parque eólico do Irã.

VILLANUEVA & FEIJÓO (2018) fizeram comparações entre diferentes funções logísticas, variando a quantidade de parâmetros utilizados, para modelagem de curvas de potência comerciais. Cada função foi testada com sete turbinas diferentes. Erros percentuais absolutos, erros quadráticos médios e erros absolutos foram calculados. As funções com 3 e 5 parâmetros demonstraram ser o melhor compromisso entre quantidade de parâmetros e erros calculados.

WANG et al. (2018) propuseram dois modelos de regressão para modelagem de curvas de potência: HSRM (*Heteroscedastic Spline Regression Model*) e RSRM *(Robust Spline Regression Model*). Para avaliação dos modelos, dados de duas turbinas foram utilizados, em duas estações do ano, usando como métrica o erro médio absoluto e o erro médio quadrático. O desempenho dos modelos propostos foi comparado com outros modelos da literatura e apresentou erros menores.

MEHRJOO; JAFARI JOZANI; PAWLAK (2020) propuseram dois métodos baseados no método de inclinação e no método de regressão por *splines* monotônica para modelar a curva. Os algoritmos foram testados com dados de quatro turbinas de um parque eólico em Manitoba, no Canadá. Utilizando-se de métricas como o erro quadrático médio e o erro médio normalizado percentual absoluto, concluiu-se que o método de regressão por *splines* teve melhor desempenho.

MARČIUKAITIS et al. (2017) apresentaram um modelo de regressão não-linear e usaram validação cruzada para estimar a precisão. Este modelo foi aplicado a uma turbina do parque eólico Seirjai na Lituânia.

JAVADI et al. (2018) empregaram um algoritmo linear por partes, baseado no programa *Statistical Analysis Software* para descrever a curva de potência e eliminar os dados anômalos. O algoritmo foi testado usando dados de uma turbina eólica real.

QIAO et al. (2024) propõem uma metodologia de modelagem multivariada de curvas de potência de turbinas eólicas que considera as diferenças de controle por segmentos e a autodependência de curto prazo dos parâmetros ambientais. Inicialmente, é apresentada uma técnica de limpeza de dados anômalos baseada em correspondência temporal e algoritmo de quartis bidirecional. Em seguida, é construído um modelo multivariado baseado na regressão de *piece-wise* de múltiplos parâmetros ambientais, aplicado à avaliação de degradação de desempenho da turbina. Os resultados indicam que a abordagem de limpeza proposta é eficaz na identificação de regiões de transição entre dados normais e anômalos, e que o modelo multivariado melhora a acurácia da modelagem e da avaliação de desempenho sob diferentes condições de recurso eólico.

- **Métodos baseados em agrupamento de dados e análise de distância**

KUSIAK & VERMA (2013) construíram uma curva de referência baseada em 5 anos de dados. O terceiro e quarto momento estatístico (curtose e *skewness)* foram calculados como métricas para descrever o formato das curvas. Para identificação de *outliers* sugeriram um algoritmo multivariável baseado em agrupamento *k-means* e distância de Mahalanobis.

YESILBUDAK (2016) desenvolveu um método para detecção de *outliers* em três níveis: agrupamento de dados por *k-means*, baseada na distância Euclidiana ao quadrado e de Manhattan, cálculo da forma da curva para comparação das duas clusterizações e filtragem dos dados usando a distância de Mahalanobis como limiar. A distância Euclidiana ao quadrado resultou em um coeficiente de *Silhouette* maior quando comparado ao de Manhattan, mas ao final dos três níveis, o autor foi bem-sucedido ao obter as curvas de referência.

Há na literatura, ainda, o caso da modelagem do formato dos *outliers*, ao invés da curva. É o caso de SHEN; FU; ZHOU (2019), que classificaram os *outliers* da curva de potência em quatro categorias: os da base da curva, os do meio, os fixos do topo e os esparsos. A partir das formas e distribuições desses *outliers*, o algoritmo do *changepoint* e do quartil são aplicados.

LUO et al. (2021) empregaram diferentes algoritmos para cada forma de *outlier* e validaram seu método usando dados de diferentes parques eólicos. Dentre os algoritmos utilizados incluíam agrupamento de dados, extração de contorno e regularização de contorno. Os resultados indicaram que os modelos de curva de potência foram, no geral, eficazes na limpeza dos *outliers*, mas enfrentaram dificuldades em reconhecer dados anômalos gerados por *curtailment*.

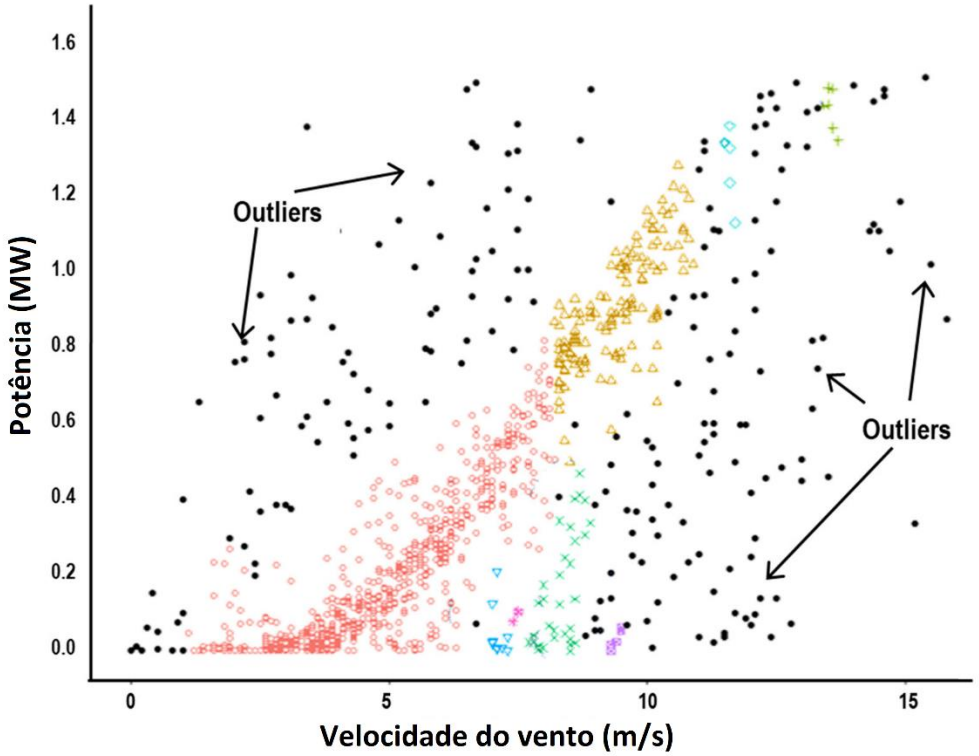
ZHENG; HU; MIN (2015) utilizaram o algoritmo LOF (*Local Outlier Factor*) em combinação com a avaliação do grau de similaridade em dados de vento de turbinas eólicas para calcular um fator de *outlier*. O método foi testado em dados de um parque eólico no nordeste da China.

ZHAO et al. (2018) propuseram um método de limpeza de dados anômalos que combina agrupamento de dados em quartis e densidade de pontos. Primeiramente o método do quartil é utilizado duas vezes para eliminação de *outliers* esparsos e em seguida o algoritmo DBSCAN é usado para eliminação de pontos com a potência fixa. Um estudo de caso em um parque eólico com 20 turbinas foi conduzido e o método se provou eficaz e com baixo custo computacional. Também

se mostrou que o método é insensível aos parâmetros utilizados, sendo, portanto, capaz de ser utilizado em diferentes turbinas eólicas sem a necessidade de calibração prévia.

É amplamente reconhecido que as técnicas de agrupamento de dados baseadas em densidade são mais eficientes do que as técnicas baseadas em partição quando se trata de identificar *clusters* de formas arbitrárias ou detectar anomalias (Hossain, 2017). A Figura 3-1 apresenta um exemplo de agrupamento por agrupamento de dados com o DBSCAN.

**Figura 3-1 Outilers identificados com o DBSCAN.**




Fonte: Adaptado de PAIK; CHUNG; KIM (2023).

PAIK; CHUNG; KIM (2023) propuseram um novo procedimento para a identificação e remoção de *outliers* na estimativa de curvas de potência de parques eólicos, utilizando algoritmos de agrupamento de dados baseados em quantização vetorial no DBSCAN. A metodologia é aplicada e validada em turbinas individuais de um parque eólico na Coreia, testando diferentes modelos paramétricos para a curva de potência.

- **Métodos baseados em aprendizado de máquina**

MANOBEL et al. (2018) apresentaram um método de modelagem baseado em Processo Gaussiano (PG) e em redes neurais. Inicialmente, os dados foram filtrados através do PG e, em seguida, esses dados “limpos” são utilizados como dados de treinamento da rede neural. Como dados de entrada se utilizam da velocidade e direção do vento para obter como saída a potência. Por fim, os autores utilizam o erro quadrático médio entre a potência gerada e a potência esperada como métrica de desempenho do algoritmo, comparando o erro do método desenvolvido com outros da literatura.

DONG et al. (2022) utilizaram aprendizagem semi-supervisionada e o algoritmo *Robust Random Cut Forest*. Para isso, selecionaram os dados considerados normais e, a cada nova amostra, inseriram esses dados no modelo. A alteração na complexidade do modelo foi então comparada com um limite dinâmico, permitindo a identificação de dados anômalos. Para minimizar a dependência dos dados normais rotulados na modelagem, foi proposta uma estratégia de atualização em tempo real baseada em auto-treinamento semi-supervisionado. Os resultados experimentais indicam que a precisão de detecção do método proposto pode atingir 95% com 1000 grupos de dados normais rotulados, e o tempo de detecção de uma única amostra é de 50 ms.

ZHANG; HU; YANG (2022) propuseram um método de detecção e diagnóstico de anomalias baseado em um *denoising autoencoder* com LSTM (LSTM-SDAE) e *XGBoost*. Primeiramente um algoritmo de reconhecimento de dados anômalos baseado no LOF e *k-means* adaptativo foi desenvolvido para fazer o pré-processamento e eliminar ruído. O modelo LSTM-SDAE foi estabelecido para obter uma relação temporal não-linear entre variáveis. Em seguida, a distância de Mahalanobis foi calculada baseada em uma técnica de janela deslizante para detecção de anomalias em tempo real. Para testar o método proposto, dados SCADA reais de um parque eólico localizado no nordeste da China foram utilizados.

MORRISON; LIU; LIN (2022) conduziram uma análise comparativa de quatro métodos de detecção de anomalias, o iForest, LOF, GMM e k-NN, com e sem filtragem. A avaliação foi baseada no erro de previsão, nas taxas de remoção de dados e na preservação das características estatísticas do vento. Os resultados mostraram que a filtragem melhorou o desempenho de todos os métodos, com o

GMM demonstrando precisão favorável enquanto ainda mantinha a variabilidade do vento.

KHAN; YEUN; BYUN (2023) apresentaram uma abordagem de aprendizado em conjunto, baseada em algoritmos genéticos, para detectar anomalias em turbinas eólicas usando dados SCADA. O método proposto combina *XGBoost*, *random forest* e modelos de árvore extra, enquanto emprega um limiar de erro quadrático médio para identificação de anomalias. A principal desvantagem desses modelos não paramétricos é o alto custo computacional.

YAO et al. (2023) empregaram uma abordagem abrangente composta por duas etapas principais para a limpeza da curva de potência. Primeiro, usaram uma técnica de pré-processamento para remover *outliers* com base no mecanismo operacional da máquina. Em seguida, propuseram um novo método de limpeza de dados chamado TTLOF (*Thompson Tau-Local Outlier Factor*), que utiliza ECMI (*Empirical Copula-Based Mutual Information*) para seleção de limiares de parâmetros de correlação e limpeza fina por segmentação (reduzindo a complexidade da limpeza) a fim de identificar características anômalas nos dados de curva de potência. Por fim, o método LSTM é usado para avaliar a eficácia do método.

LETZGUS & MÜLLER (2024) propõem uma metodologia baseada em inteligência artificial explicável (*XAI*) para avaliação de modelos de curvas de potência de turbinas eólicas gerados por aprendizado de máquina. Com o objetivo de complementar as métricas tradicionais de erro, introduzem uma métrica que quantifica o alinhamento dos modelos com princípios físicos do sistema. A análise é conduzida utilizando uma variedade de abordagens, que vão desde modelos físicos simplificados até métodos de aprendizado supervisionado mais complexos, incluindo regressões lineares segmentadas, regressões polinomiais, *Random Forests*, ANNs e SVMs. O trabalho investiga como essas diferentes escolhas influenciam a capacidade de generalização e a robustez dos modelos em ambientes dinâmicos.

YIN et al. (2025) propõem uma abordagem multivariada para previsão de curvas de potência de turbinas eólicas, integrando técnicas de aprendizado de máquina avançadas. O método combina regressão por árvores impulsionadas por gradiente estocástico (SGBRT) e otimização por matilha de lobos cinzentos (GWO), aliados a etapas inovadoras de pré-processamento de dados e seleção de variáveis. A limpeza dos dados é realizada em um espaço bidimensional de Cópula, utilizando a velocidade do rotor como critério auxiliar para lidar com incertezas e dependências

não lineares. A seleção de variáveis é feita com base na análise da informação mútua parcial (PMI), resultando na escolha de oito parâmetros significativos. O modelo SGBRT tem seus hiperparâmetros otimizados via GWO, considerando uma função de ajuste baseada em RMSE, MAE e $R^2$. A validação com dados SCADA reais demonstra que o modelo proposto supera métodos existentes em termos de acurácia, eficiência e robustez.

## 3.2 REDES KOLMOGOROV-ARNOLD

Até o momento da escrita desta dissertação, poucos trabalhos utilizando KAN haviam sido publicados.

No campo da energia eólica, apenas um artigo foi identificado. MUBARAK et al. (2024) avaliaram o desempenho da KAN e MLP em previsões de produção de energia de seis parques eólicos na China. A KAN supera limitações do MLP, como escalabilidade e interpretabilidade, utilizando funções de ativação *B-Spline* e otimização pelo algoritmo LBFGS. Técnicas de pré-processamento, como *Interquartile Range* para tratar *outliers* e *K-Nearest Neighbor* para imputação de dados, também foram aplicadas. A KAN demonstrou desempenho superior, com erro médio quadrático de 0,0039 no melhor local.

SULAIMAN et al. (2024) propuseram o uso da KAN para modelar as relações não lineares dos dados de consumo de um edifício comercial. Comparando o desempenho da KAN com MLP e um algoritmo híbrido TLBO-DL (*Teaching-Learning-Based Optimization with Deep Learning*), o KAN demonstrou superioridade. A pesquisa destaca a aplicação inovadora do KAN em previsões energéticas, com maior precisão e eficiência computacional, contribuindo para a gestão energética em sistemas reais.

GAO et al. (2025) propõem o uso da KAN como uma solução para melhorar a interpretabilidade e o desempenho preditivo da radiação solar e temperatura externa. Os autores conduziram estudos de caso com dados do Observatório Meteorológico de Tóquio, a KAN mostrou-se capaz de reduzir o erro médio quadrático em 75,33% em relação a modelos recorrentes tradicionais, mesmo com apenas um neurônio oculto na previsão de radiação solar.

GAO; KONG (2025) propõem uma abordagem para sistemas de posicionamento espacial, especialmente no caso de cápsulas médicas, com o uso da tecnologia *Magnetic Positioning* (MP) combinada à KAN. O algoritmo demonstrou

bom desempenho, com erro máximo de posicionamento de 0,24 mm e erro relativo variando de 0,25% a 5,72%, mantendo precisão constante independente da distância entre o alvo e o sistema de medição.

## 3.3 PERSPECTIVAS DO ESTUDO

A partir da revisão apresentada, observa-se que a detecção e a remoção de anomalias em curvas de potência de turbinas eólicas têm sido amplamente estudadas por meio de diferentes abordagens, incluindo modelos estatísticos, técnicas de agrupamento e algoritmos de aprendizado de máquina. Métodos baseados em regressão apresentam boa capacidade de ajuste, mas são sensíveis à presença de outliers e requerem suposições sobre a forma da curva. Técnicas de agrupamento e análise de distância demonstram eficácia na identificação de padrões anômalos sem a necessidade de rótulos, porém podem apresentar limitações em cenários com estruturas de dados mais complexas ou com ruídos sobrepostos. Já os métodos baseados em aprendizado de máquina, especialmente os não supervisionados ou semi-supervisionados, oferecem maior flexibilidade e capacidade de generalização, mas ainda enfrentam desafios relacionados à interpretabilidade dos modelos e ao custo computacional.

Além disso, embora muitos estudos foquem na remoção de pontos inconsistentes ou ruídos, poucos abordam de forma clara a separação entre diferentes tipos de anomalias, como eventos de indisponibilidade, nos quais a turbina está fora de operação, e situações de subdesempenho, em que a turbina permanece operando, porém com rendimento inferior ao esperado. Essa distinção é fundamental, pois impacta diretamente na estimativa de produção, nas análises de disponibilidade e nos relatórios técnicos de desempenho. A correta identificação dessas condições exige modelos capazes de capturar nuances nos dados e interpretar diferentes padrões de desvio em relação à curva de potência ideal.

Diante desse cenário, observa-se uma lacuna na aplicação de modelos que aliem previsão e identificação precisa de diferentes anomalias, além de interpretabilidade. Em especial, observa-se a ausência de estudos que explorem o uso de redes Kolmogorov-Arnold (KAN) nesse contexto. A aplicação que existe está relacionada à previsão da produção de energia, não contemplando seu potencial para a classificação de dados operacionais. Assim, a presente dissertação propõe o desenvolvimento de um modelo híbrido, baseado na combinação de *autoencoders* e

redes KAN, com o objetivo de automatizar a limpeza da curva de potência ao mesmo tempo em que diferencia, de forma confiável, dados normais, eventos de indisponibilidade e casos de subdesempenho.

# 4 METODOLOGIA

Este trabalho tem como objetivo propor uma nova metodologia de limpeza automática de curvas de potência, explorando abordagens computacionais modernas para identificação e remoção de *outliers*. A limpeza consiste em categorizar os dados de uma curva de potência em duas principais rotulações:

1. Dados normais;
2. Dados anômalos, que incluem:
   a. Indisponibilidade;
   b. Subdesempenho.

As duas categorias dos dados anômalos seguem as definições previamente apresentadas na seção 2.2. Importante mencionar que se assume a classe 0 como pontos normais, classe 1, indisponibilidade e classe 2, subdesempenho. As rotulações categorizadas automaticamente são comparadas com a rotulação realizada manualmente por um especialista da área.

## 4.1 DADOS DE TURBINAS EÓLICAS UTILIZADOS

No setor eólico, os dados SCADA são geralmente confidenciais e de propriedade da operadora do parque. Apesar disto, existem algumas iniciativas e conjuntos de dados disponíveis publicamente para pesquisa. A Tabela 4-1 apresenta os dados utilizados no presente trabalho.

**Tabela 4-1 Dados SCADA públicos utilizados no presente trabalho.**

| Parque eólico / Empresa | Localização | Quantidade de turbinas | Dados | Resolução temporal | Fonte |
|---|---|---|---|---|---|
| Kelmarsh / Cubico | Northamptonshire – Reino Unido | 6 | • SCADA<br>• Produção de energia da subestação<br>• Log de alarmes<br>• Layout<br>• Turbina | 10 minutos | https://zenodo.org/records/5841834#.YgpBQ_so-V7 |

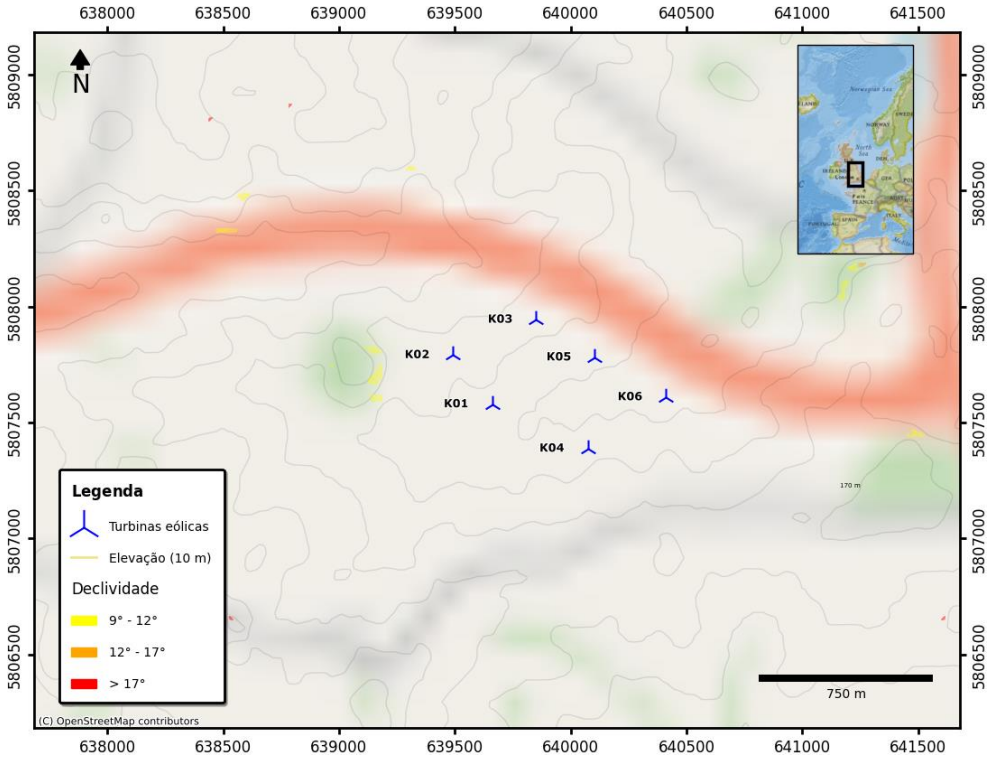
A seguir, é feita uma breve descrição sobre o conjunto de dados utilizado.

## 1. Kelmarsh e Pernmanshiel

Os dados foram disponibilizados pela *Cubico Sustainable Investments Ltd* em 2022 com o objetivo de ampliar o acesso a informações do setor e incentivar o envolvimento de profissionais e pesquisadores em desafios inovadores. Foi criado o espaço "*Cubico Open Data Exploration*", liderado por Charlie Plumley, que lançou o primeiro desafio: "*Operational Energy Yield Analysis Using Open Data*". O objetivo desse desafio era prever a produção de energia ao longo de 20 anos e as incertezas associadas para o parque eólico (WeDoWind, 2023).

Os conjuntos de dados abrangem o período de 2016 a 2021, totalizando 6 anos de informações. Eles incluem dados SCADA, medições de energia na subestação, layout dos parques, especificações das turbinas e as respectivas datas de entrada em operação. Considerando todo o período, a quantidade de pontos dos dados SCADA gira em torno de 200.000 a 210.000. A Figura 4-1 ilustra o layout do parque eólico Kelmarsh, bem como curvas de nível e declividade do terreno.

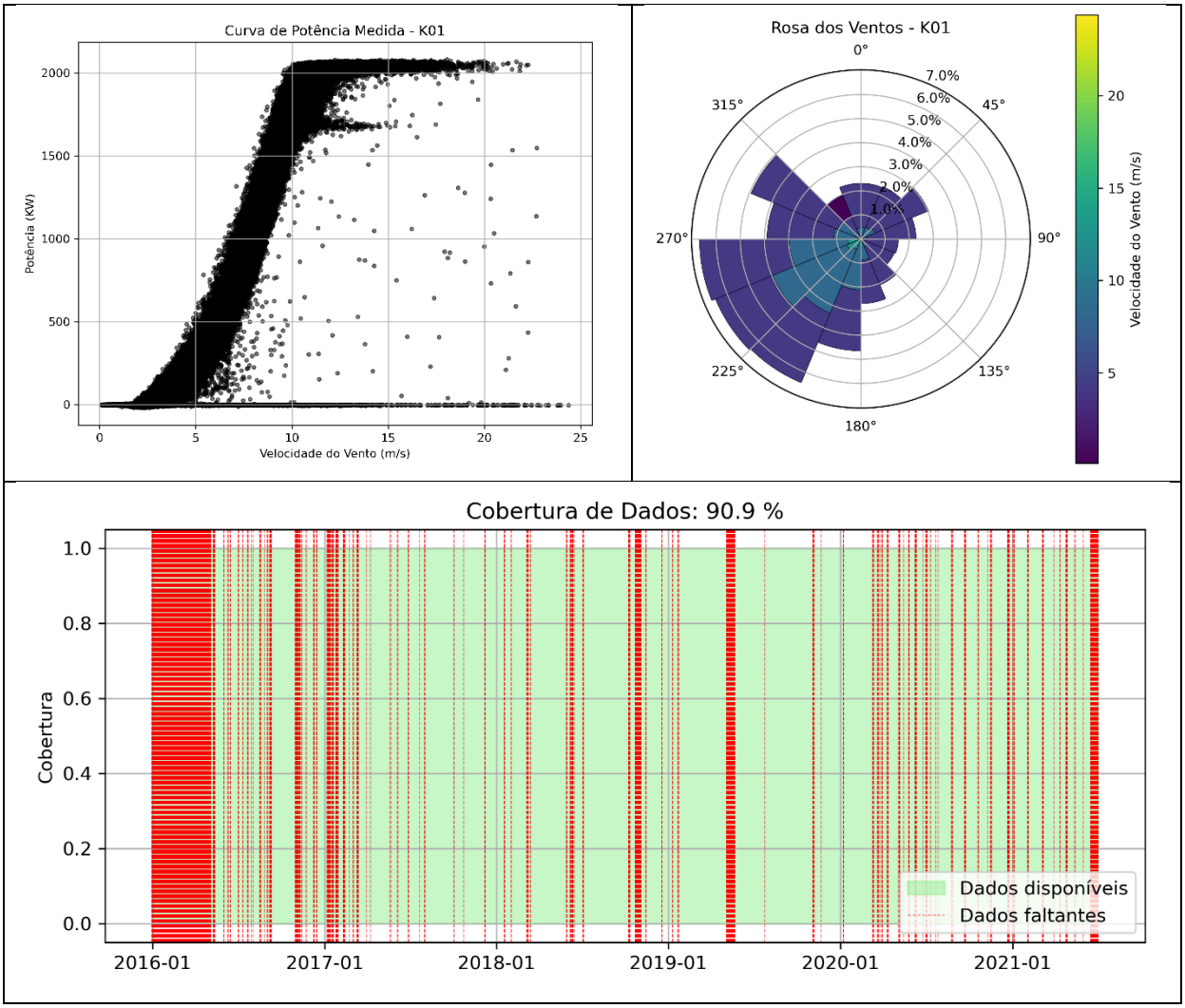
**Figura 4-1 Parque eólico Kelmarsh.**




Fonte: a Autora (2024)

A Figura 4-2 apresenta a curva de potência, rosa dos ventos e cobertura de dados de uma turbina do parque.

**Figura 4-2 Curva de potência, rosa dos ventos e cobertura da turbina K01.**




Fonte: a Autora (2024)

## 4.2 VARIÁVEIS SCADA CONSIDERADAS

Na limpeza de uma curva de potência, a princípio, o foco principal está na relação entre potência e velocidade do vento. No entanto, conforme destacado na seção 2.2.3, sinais auxiliares podem contribuir significativamente para a classificação dos pontos associados à indisponibilidade e ao subdesempenho. Por esse motivo, além da potência e da velocidade do vento, também foram consideradas as variáveis velocidade do rotor, ângulo de *pitch* e direção da nacele.

## 4.3 METODOLOGIA DE ANÁLISE E ALGORITMOS EMPREGADOS

Nesta seção, são detalhados os procedimentos adotados, estruturados em quatro etapas principais: pré-processamento dos dados, testes de algoritmos e implementação, avaliação dos resultados e comparação com algoritmos bem estabelecidos na área de aprendizado de máquina. Os códigos foram rodados em uma máquina com as seguintes características:

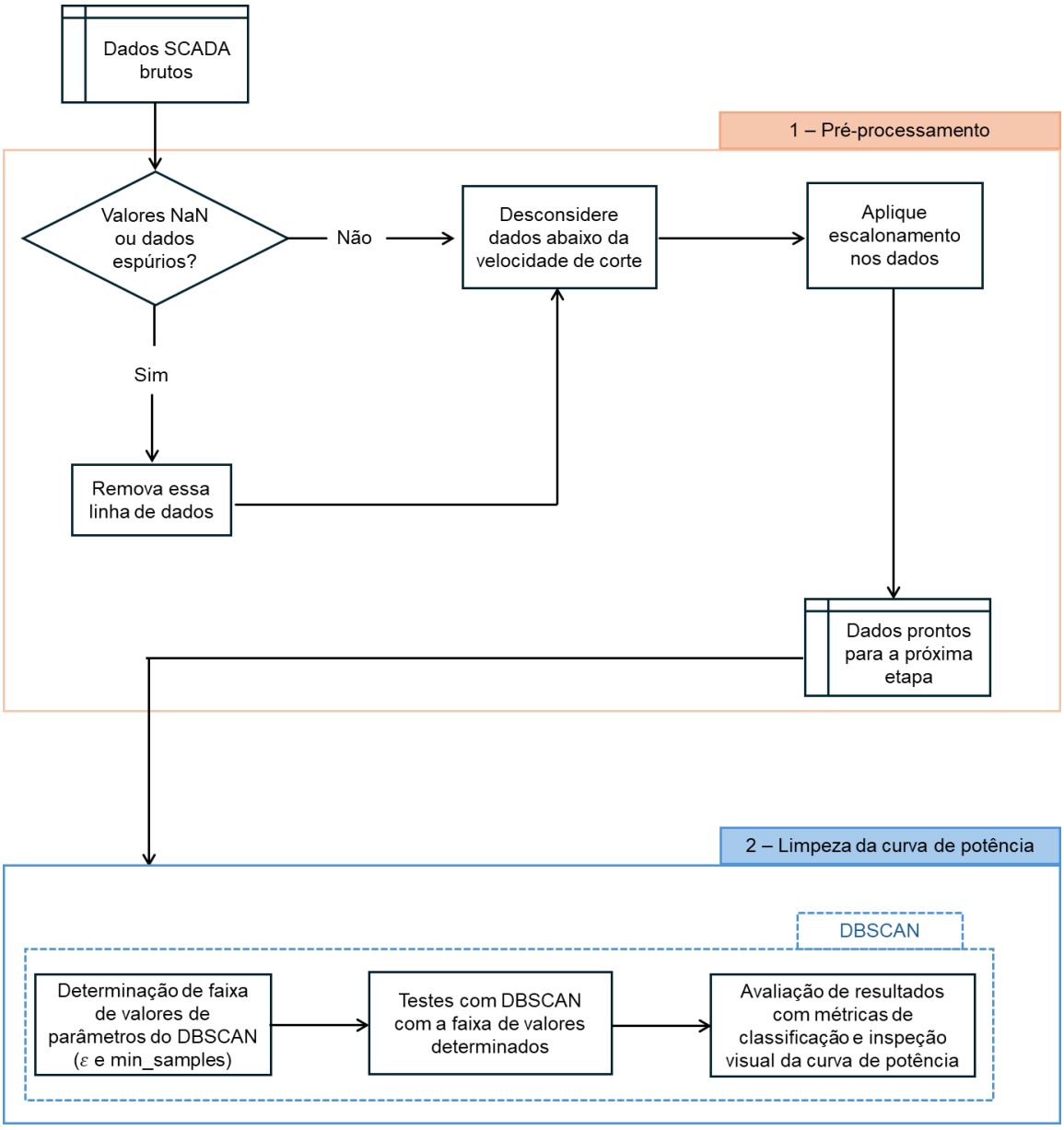
- Processador: Intel Core i7 (2 núcleos físicos, 4 núcleos lógicos, 2.7 GHz);
- Memória RAM:16 GB;
- Placa de vídeo: NVIDIA GeForce 940MX (4 GB VRAM).

É importante ressaltar que algumas abordagens foram testadas até a obtenção dos modelos finais, sendo elas:

1. DBSCAN;
2. DBSCAN com janela deslizante e parâmetros estatísticos;
3. *Autoconder* com KAN;
4. *Autoencoder* variacional com KAN.

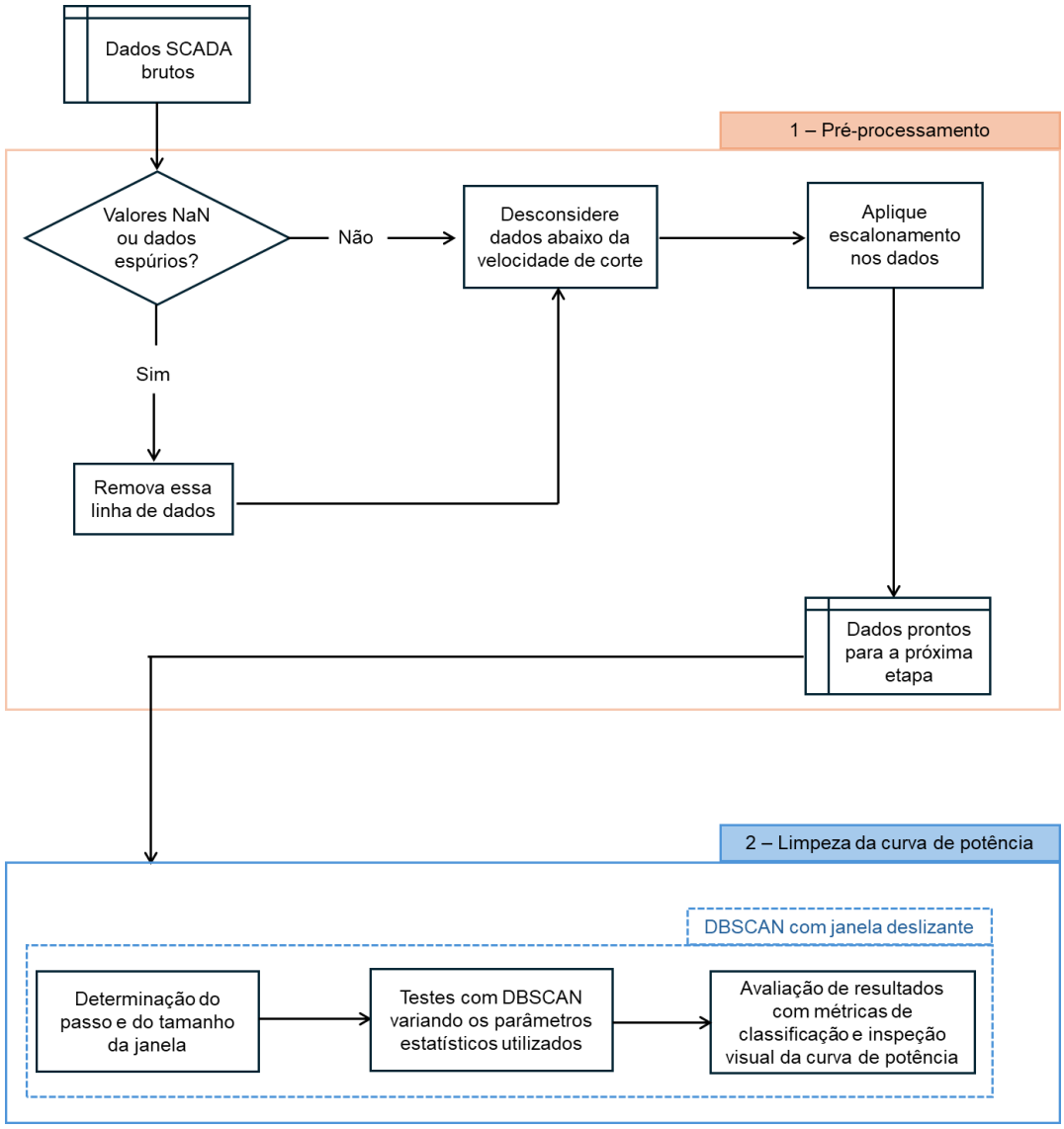
A etapa de pré-processamento é comum a todas as metodologias avaliadas, enquanto a etapa seguinte, referente à limpeza da curva de potência, foi testada e explorada com diferentes abordagens até a definição do algoritmo selecionado. A Figura 4-3 e a Figura 4-4 ilustram a metodologia dos modelos 1 e 2, respectivamente. A Figura 4-5 apresenta as etapas dos modelos 3 e 4 (modelos finais), que vão do pré-processamento até a comparação com algoritmos bem estabelecidos na área.

**Figura 4-3 Fluxograma da metodologia DBSCAN.**



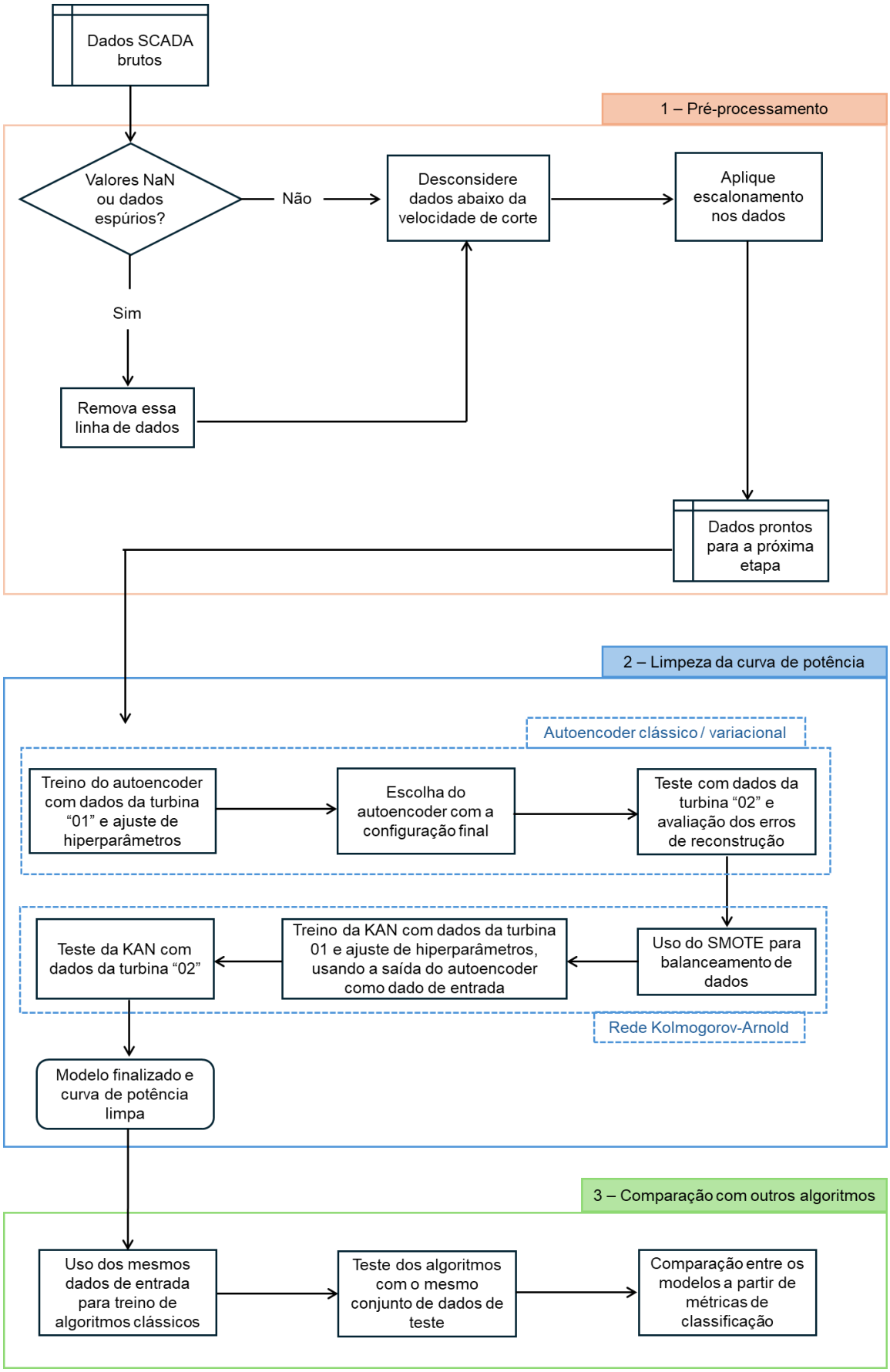

Fonte: A autora (2024).

**Figura 4-4 Fluxograma da metodologia DBSCAN com parâmetros estatísticos e janela deslizante.**




Fonte: A autora (2024).

**Figura 4-5 Fluxograma da metodologia final utilizada.**




Fonte: a Autora (2024)

### 4.3.1 Pré-processamento

O pré-processamento de dados SCADA é fundamental, pois os dados brutos frequentemente contêm registros inválidos que precisam ser filtrados. Sensores podem registrar leituras incorretas devido a falhas operacionais e lacunas nos dados podem surgir por interrupções na comunicação do sistema SCADA. Além disso, algoritmos em aprendizagem de máquina podem ser sensíveis a escalas entre diferentes variáveis e uma boa prática é a normalização ou escalonamento. O pré-processamento é dividido basicamente em três etapas:

1. Remoção de dados *NaN*: registros cujos *timestamps* contenham valores *NaN* são eliminados. Esta abordagem foi adotada neste estudo, ao invés do preenchimento dos valores faltantes, pois o preenchimento estaria apenas adicionando incerteza e ruído e os dados remanescentes são suficientes para manter a robustez do modelo, pela grande quantidade de dados.
2. Eliminação de dados espúrios: a remoção de leituras incorretas é fundamental para garantir a qualidade dos dados. Os seguintes critérios foram estabelecidos:
    a. Velocidade do vento inferior a 0 m/s ou superior a 40 m/s;
    b. Velocidade do vento com cinco ou mais repetições consecutivas;
    c. Potência, ângulo de *pitch* e velocidade do vento com mais de cinco repetições consecutivas, simultaneamente. É interessante que as repetições de *pitch* e potência estejam correlacionadas entre si e com a velocidade do vento, pois, isoladamente, sensores podem apresentar valores repetidos plausíveis.
3. Escalonamento dos dados: algoritmos de aprendizagem de máquina são sensíveis a diferenças de magnitude entre variáveis. Para garantir consistência e evitar que características com valores maiores dominem a análise, os dados foram normalizados.

#### 4.3.1.1. Balanceamento de classes em algoritmos de classificação

Muitos conjuntos de dados do mundo real apresentam desbalanceamento e isso pode levar a um desempenho enviesado do modelo, já que os algoritmos de aprendizado de máquina tendem a classificar corretamente a classe majoritária, enquanto cometem erros na classificação da classe minoritária. Entretanto, em

problemas de classificação desbalanceada, geralmente há interesse em classificar corretamente as amostras da classe minoritária, pois são elas que representam os casos mais importantes, como por exemplo falhas ou anomalias. Ainda assim, os algoritmos de aprendizado de máquina utilizados para classificação binária ou multiclasse são, em sua maioria, projetados para trabalhar com conjuntos de dados balanceados, otimizando métricas igualmente distribuídas entre as classes (Galli, 2023).

Uma forma de lidar com o desbalanceamento é fazer um "*resampling*" (reamostragem) dos dados de treino. No presente trabalho, se utilizou da abordagem SMOTE, sigla para *Synthetic Minority Over-sampling Technique*, que é uma técnica de oversampling, que cria dados sintéticos para a classe minoritária. O funcionamento do SMOTE baseia-se na criação de amostras sintéticas ao longo das linhas que conectam os vizinhos mais próximos. Geram-se novas amostras da classe minoritária dando pequenos passos a partir de uma instância existente em direção a um de seus k vizinhos mais próximos, sendo k um parâmetro do algoritmo. Para isso, o algoritmo seleciona aleatoriamente um dos k vizinhos mais próximos e gera uma nova amostra ao adicionar uma pequena perturbação vetorial ao ponto de origem, interpolando entre ele e o vizinho escolhido. Dessa forma, as novas amostras sintéticas mantêm características similares às amostras reais da classe minoritária, mas não são cópias exatas, aumentando assim a diversidade da base de dados.

### 4.3.2 Testes de algoritmos e implementação

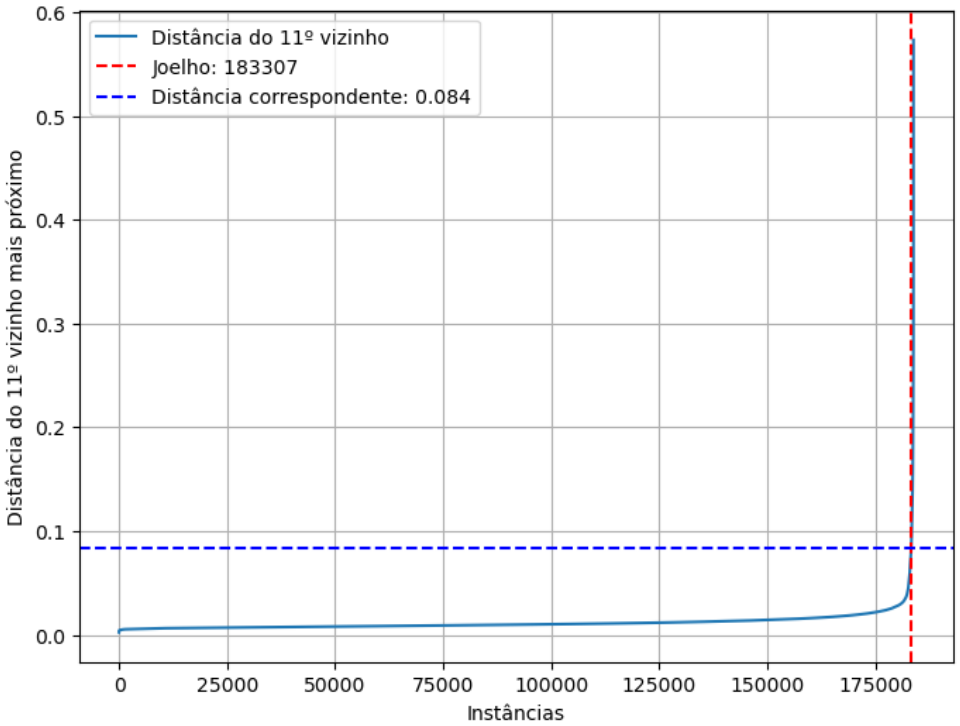
Nesta seção são apresentados os algoritmos testados durante a elaboração deste trabalho, descrevendo-se a metodologia utilizada em cada um deles. Para fins de ilustração de resultados, estes são mostrados para a turbina K01 do parque eólico Kelmarsh.

#### 4.3.2.1 DBSCAN

Conforme apresentado na seção 3.1, diversos autores empregam métodos de agrupamento de dados para limpeza de curvas de potência, abordagem que serviu como ponto de partida para este trabalho. Os parâmetros principais a serem determinados são o "Épsilon" e o "*Min_samples*".

Para determinação do primeiro, foi utilizado o algoritmo *KneeLocator*, que identifica um ponto de inflexão na curva de distâncias entre vizinhos mais próximos. A premissa é que, em um agrupamento, espera-se que os pontos centrais apresentem distâncias menores entre si, os pontos na borda do cluster ainda permaneçam dentro de um intervalo razoável, e os pontos considerados ruídos tendem a apresentar distâncias significativamente maiores. Dessa forma, o ponto de joelho na curva representa a transição entre regiões e pode fornecer um valor adequado para ε. Isto é ilustrado na Figura 4-6 para uma investigação considerando 11 vizinhos mais próximos.

**Figura 4-6 Obtenção do joelho da curva com o *KneeLocator***




Fonte: a Autora (2024).

Neste exemplo, o joelho ocorre na instância 183307, correspondendo a uma distância de 0,084. No presente trabalho, na realização de alguns testes exploratórios, notou-se que o valor de Épsilon igual ao previsto pelo *KneeLocator* era conservador. Isto acontece, pois quando os *outliers* são raros ou mais dispersos, pontos mais afastados podem ser incorretamente incorporados ao cluster, ao invés de serem identificados como ruído. Para mitigar este efeito, no presente trabalho, optou-se por ajustar o ε para metade do valor obtido com o *KneeLocator*.

Quanto ao *Min_samples*, existem algumas regras gerais que se pode adotar na escolha (Sefidian, 2023):

- Quanto maior o conjunto de dados, maior deve ser o valor de *Min_samples*;
- Se o conjunto de dados for ruidoso, é preferível um valor maior de *Min_samples*;
- Geralmente, *Min_samples* deve ser maior ou igual à dimensionalidade do conjunto de dados
- Se os dados tiverem mais de duas dimensões, escolha *Min_samples* = 2*dimensão (Sander et al., 1998)

No presente trabalho, a dimensão corresponde a 5. Logo, a princípio, é adotado como valor inicial *Min_samples* = 10. No entanto, considerando o grande volume de dados disponíveis, foi realizada uma investigação dos valores *Min_samples*, permitindo a avaliação de valores maiores, de modo a otimizar a segmentação dos *clusters*.

#### 4.3.2.2 DBSCAN com parâmetros estatísticos e janela deslizante

Como segunda abordagem a ser testada, foi utilizado um algoritmo de DBSCAN considerando-se como dados de entrada parâmetros estatísticos em janela deslizante, para processar os dados em segmentos sobrepostos. Isto é muito útil em séries temporais, pois se permite dividir os dados em janelas móveis, analisando padrões e tendências ao longo do tempo. As anomalias, em geral, possuem uma dependência temporal, pois períodos de falha ou baixo desempenho da turbina costumam se estender por um período de tempo. Dessa forma, pontos que mantêm um comportamento anômalo de forma consistente são mais facilmente identificáveis e a janela deslizante é uma forma de incluir esta dependência temporal.

Os primeiros parâmetros a serem definidos são o tamanho da janela e o passo. Neste estudo, utilizou-se uma janela de 6 horas e um passo de 2 horas. Esses valores foram determinados a partir de testes exploratórios, nos quais se variou o tamanho da janela e o passo para avaliar em quais configurações a curva de potência média preservava seu comportamento normal. Com essa parametrização, um novo *dataframe* segmentado foi criado, no qual os dados foram

organizados em janelas de 6 horas, com um deslocamento de 2 horas entre cada janela. O *timestamp* original, registrado a cada 10 minutos, foi mantido, mas agora os valores eram agrupados dentro das janelas de 6 horas. Como resultado, uma mesma ocorrência poderia aparecer repetidamente em diferentes janelas, já que pertencia a vários segmentos, devido à sobreposição das janelas móveis.

Como parâmetros de entrada para o algoritmo DBSCAN, são utilizados dados estatísticos extraídos por meio da biblioteca *tsfresh*. Entre as estatísticas calculadas incluem-se mediana, média, desvio padrão, mínimo, máximo, amplitude, primeira e última posição dos valores extremos, coeficientes da tendência linear (inclinação, *offset* e coeficiente de correlação), curtose e assimetria. Essas métricas permitem capturar padrões importantes da série temporal, facilitando a identificação de pontos com características similares. Diferentes parâmetros são testados a fim de verificar o desempenho do modelo. Os resultados são apresentados no capítulo 5.

#### 4.3.2.3 *Autoencoder* clássico com KAN (AE-KAN)

Como terceira metodologia a ser testada, adotam-se algoritmos de classificação. Para avaliar o desempenho da rede neural Kolmogorov-Arnold, ainda pouco explorada, a mesma foi escolhida como principal método de teste. Além disso, optou-se por utilizar um *autoencoder* padrão como etapa de pré-processamento, com o objetivo de aprimorar os dados de entrada para a KAN. Este algoritmo híbrido é nomeado no presente trabalho de AE-KAN.

Um *autoencoder* padrão possui os seguintes parâmetros:

1. Dimensão de entrada (*input_dim*): define o número de neurônios da camada de entrada e depende da quantidade de variáveis dos dados originais;
2. Dimensão do espaço latente (*encoding_dim*): número de neurônios da camada latente (compactação ou expansão da informação);
3. Arquitetura da rede: define a quantidade de camadas e neurônios do codificador e decodificador;
4. Função de ativação: controla a não linearidade entre as camadas (funções ReLU, sigmóide, tanh, etc);
5. Função de perda: mede a diferença entre a entrada original e a saída reconstruída;
6. Otimizador: ajusta os pesos para minimizar a função de perda;

7. Taxa de aprendizado: define o tamanho do passo que o otimizador dá na atualização dos pesos;
8. Número de épocas: quantas vezes o modelo vê os dados de treinamento e aprende com eles.

Para treinamento do *autoencoder*, apenas os pontos considerados normais (classe 0) são utilizados como dados de entrada para que o modelo aprenda a representação dos dados normais. Para treino e ajuste, são utilizados dados da turbina K01 do parque Kelmarsh. Os hiperparâmetros base são apresentados na Tabela 4-2.

**Tabela 4-2 Parâmetros base considerados no *autoencoder*.**

| Parâmetro | Valor considerado |
|---|---|
| Input_dim | 4 |
| Encoding_dim | 3 |
| Arquitetura da rede | 3 camadas no codificador + 3 camadas no decodificador |
| Função de ativação | ReLU |
| Função de perda | MSELoss |
| Otimizador | Adam |
| Taxa de aprendizado | 0,001 |
| Número de épocas | 100 |

Para o treinamento do *autoencoder*, é feito um ajuste dos hiperparâmetros a serem considerados, conforme Tabela 4-3. Os parâmetros são combinados, formando um total de 24 possibilidades. Em relação ao custo computacional, contabilizou-se em torno de 15 minutos para cada rodada do ajuste, totalizando um tempo aproximado de 6 horas.

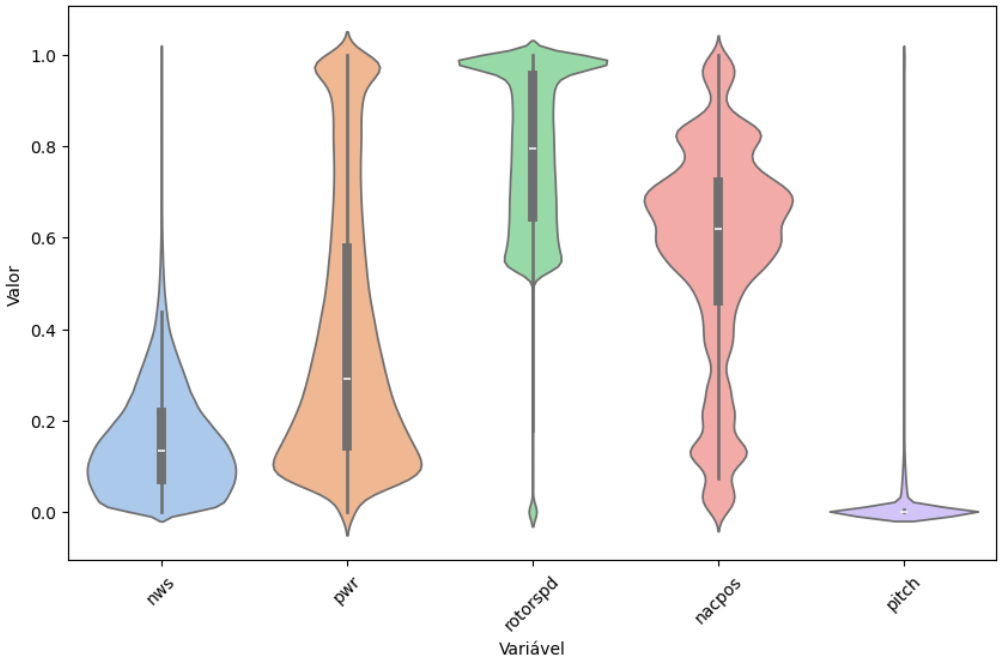
**Tabela 4-3 Parâmetros para treinamento do *autoencoder*.**

| Número de camadas | Dimensão latente | Otimizador | Número de épocas |
|---|---|---|---|
| 3+3 | 3 | Adam | 100 |
| 4+4 | 4 | RMSProp | 500 |
| | | AdamW | |

Inicialmente, as variáveis consideradas são as mesmas utilizadas para o agrupamento de dados: velocidade do vento, potência, posição da nacele, velocidade do rotor e ângulo de *pitch*. No entanto, com a introdução do *autoencoder*, torna-se interessante avaliar a influência de cada variável na reconstrução dos dados. Analisar a variância dos parâmetros ajuda a identificar se todas as variáveis são relevantes ou se algumas podem ser removidas sem comprometer a qualidade

da representação. Para isso, gera-se um gráfico de violino, permitindo visualizar a distribuição dos dados e identificar padrões de variação, conforme ilustrado na Figura 4-7.

**Figura 4-7 Gráfico violino com os valores das variáveis consideradas.**




Fonte: a Autora (2024)

No gráfico tem-se:

- Nws: velocidade do vento;
- Pwr: potência;
- Rotorspd: velocidade do rotor;
- Nacpos: posição da nacele;
- *Pitch*: ângulo de *pitch*.

Pode-se observar na Figura 4-7 que o ângulo de *pitch* apresenta baixa variação, indicando que sua inclusão na reconstrução dos dados pelo *autoencoder* tende a ter um impacto mínimo. Portanto, ela não é considerada como variável para treinamento.

Após ajuste de hiperparâmetros do autoencoder com a turbina K01 e escolha da configuração do autoencoder, os dados da turbina K02 são passados pelo autoencoder treinado.

No caso da KAN, os seguintes parâmetros devem ser avaliados:

1. Tamanho das camadas ocultas (*hidden_layer_size*): número de neurônios das camadas ocultas;
2. Regularização da ativação (*regularize_activation*): controla a penalização na ativação dos neurônios para evitar *overfitting*;
3. Regularização da entropia (*regularize_entropy*): regula a entropia para suavizar o comportamento da rede;
4. Regularização *ridge* (*regularize_ridge*): adiciona penalização para evitar pesos excessivamente grandes;
5. Ordem do *spline*: determina a ordem da interpolação *spline* usada na modelagem.

A Tabela 4-4 apresenta os parâmetros base utilizados no treinamento da rede Kolmogorov-Arnold.

**Tabela 4-4 Hiperparâmetros base da KAN.**

| Tamanho da camada | Regularização de ativação | Regularização da entropia | Regularização *ridge* | Ordem do *spline* |
|---|---|---|---|---|
| 32 | 0.3 | 0.3 | 0.5 | 3 |

Para otimização dos hiperparâmetros, rodadas de treino foram executadas variando-se os valores utilizados. Os valores são mostrados na Tabela 4-5. Para cada rodada de ajustes de hiperparâmetros, estima-se 10 minutos de custo computacional, totalizando em torno de 2h e 40 minutos de tempo. Os resultados são apresentados mais adiante.

**Tabela 4-5 Hiperparâmetros da KAN.**

| Rodada | Tamanho da camada | Regularização de ativação | Regularização da entropia | Regularização *ridge* | Ordem do *spline* |
|---|---|---|---|---|---|
| 1 | 32 | 0.3 | 0.3 | 0.5 | 3 |
| 2 | 32 | 0.3 | 0.3 | 0.7 | 3 |
| 3 | 32 | 0.3 | 0.5 | 0.5 | 3 |
| 4 | 32 | 0.3 | 0.5 | 0.7 | 3 |
| 5 | 32 | 0.5 | 0.3 | 0.5 | 3 |
| 6 | 32 | 0.5 | 0.3 | 0.7 | 3 |
| 7 | 32 | 0.5 | 0.5 | 0.5 | 3 |
| 8 | 32 | 0.5 | 0.5 | 0.7 | 3 |
| 9 | 64 | 0.3 | 0.3 | 0.5 | 3 |
| 10 | 64 | 0.3 | 0.3 | 0.7 | 3 |

| | | | | | |
|---|---|---|---|---|---|
| 11 | 64 | 0.3 | 0.5 | 0.5 | 3 |
| 12 | 64 | 0.3 | 0.5 | 0.7 | 3 |
| 13 | 64 | 0.5 | 0.3 | 0.5 | 3 |
| 14 | 64 | 0.5 | 0.3 | 0.7 | 3 |
| 15 | 64 | 0.5 | 0.5 | 0.5 | 3 |
| 16 | 64 | 0.5 | 0.5 | 0.7 | 3 |

Após o ajuste de hiperparâmetros o modelo AE-KAN é testado com a turbina K02 do parque eólico Kelmarsh. Uma vez que o autoencoder está treinado, o custo computacional para teste é de algo em torno de 3 minutos. Para a KAN, temos o tempo de 2 minutos. Os resultados do treino e teste são apresentados no capítulo 5.

#### 4.3.2.4 *Autoencoder* variacional com KAN (VAE-KAN)

Para avaliar possíveis melhorias nos resultados (mais especificamente na identificação da classe 2, como é discutido mais adiante), um *autoencoder* variacional (VAE) é testado no pré-processamento em substituição ao *autoencoder* padrão. A principal diferença do VAE é que a saída da camada latente não apenas representa uma codificação comprimida dos dados, mas também incorpora uma distribuição probabilística, permitindo maior flexibilidade na modelagem das variações dentro dos dados. Nesse contexto, a saída da camada latente é utilizada como informação adicional. O objetivo é avaliar se os resultados são aprimorados quanto à separabilidade das classes e potencial redução de falsos positivos na classe 2. O modelo é denominado VAE-KAN.

Em relação aos hiperparâmetros utilizados, alguns são iguais aos já listados no *autoencoder* padrão, como o *input_dim*, arquitetura do codificador e decodificador e a função de perda. Além destes, tem-se:

1. Tamanho das camadas ocultas (*hidden_dim*): quantidade de neurônios das camadas ocultas;
2. Tamanho do espaço latente (*latent_dim*): número de variáveis compactadas no espaço latente;
3. Parâmetros da distribuição latente (fc_mu e fc_logvar): camadas que aprendem média e logaritmo da variância da distribuição latente;
4. Reparametrização: garante que a distribuição latente seja amostrada de forma contínua;

5. Função de perda com o *KL divergence*: regularização do espaço latente, forçando-o a se aproximar de uma distribuição normal padrão.

Para otimização dos hiperparâmetros, assim como realizado nos outros casos, diferentes valores são testados com o objetivo de obter a configuração ótima. Os valores utilizados são apresentados na Tabela 4-6, totalizando 16 combinações. Assim como no caso do autoencoder clássico, o tempo para cada rodada de treino é em torno de 15 minutos, o que totaliza 4 horas.

**Tabela 4-6 Hiperparâmetros do *autoencoder* variacional.**

| Tamanho do espaço latente | Épocas | Tamanho das camadas ocultas | Otimizador |
|---|---|---|---|
| 3 | 20 | 8 | Adam |
| 4 | 30 | 16 | RMSProp |

Assim como no caso do autoencoder clássico, após ajuste e treino com os dados da turbina K01, o teste é feito com os dados da turbina K02. De forma análoga, a saída do autoencoder variacional é usada como entrada para a KAN, que é treinada novamente. Os hiperparâmetros são ajustados conforme apresentado na Tabela 4-5 e o custo computacional é o mesmo. De forma análoga, executa-se o teste para a turbina K02. O tempo do teste é virtualmente o mesmo do caso AE-KAN.

De modo a resumir, a Tabela 4-7 e a Tabela 4-8 apresentam os dados de entrada utilizados nos modelos AE-KAN e VAE-KAN para treino e teste de uma turbina arbitrária. No presente trabalho correspondem às turbinas K01 e K02 do parque eólico Kelmarsh. Vale ressaltar que as variáveis utilizadas compreendem a potência, velocidade do vento, velocidade do rotor e posição da nacele e que o SMOTE é apenas utilizado do conjunto de dados de treino. O treino e teste são conduzidos em turbinas distintas.

**Tabela 4-7 Dados de entrada do modelo AE-KAN.**

| Treino – turbina arbitrária 01 | Teste – turbina arbitrária 02 |
|---|---|
| Variáveis originais | Variáveis originais |
| Variáveis reconstruídas | Variáveis reconstruídas |
| Erro de reconstrução | Erro de reconstrução |

**Tabela 4-8 Dados de entrada do modelo VAE-KAN**

| Treino – turbina arbitrária 01 | Teste – turbina arbitrária 02 |
|---|---|
| Variáveis originais | Variáveis originais |
| Variáveis reconstruídas | Variáveis reconstruídas |
| Saída da camada latente | Saída da camada latente |
| Erro de reconstrução | Erro de reconstrução |

Com base nos experimentos definidos na metodologia, espera-se que a abordagem com DBSCAN, utilizada como ponto de partida, permita uma identificação inicial de *outliers* com base na densidade dos dados. A seguir, a introdução de janelas deslizantes e variáveis estatísticas visa incorporar a dimensão temporal e melhorar a detecção de padrões anômalos persistentes, especialmente associados à indisponibilidade e ao subdesempenho. Na sequência, os modelos baseados em *autoencoders*, AE-KAN e VAE-KAN, são esperados apresentar desempenho superior, com maior capacidade de representar a estrutura dos dados normais e identificar desvios com maior precisão. Por fim, espera-se que os modelos generalizem adequadamente entre turbinas distintas, mantendo coerência nos resultados e boa correspondência com a rotulação manual feita pelo especialista.

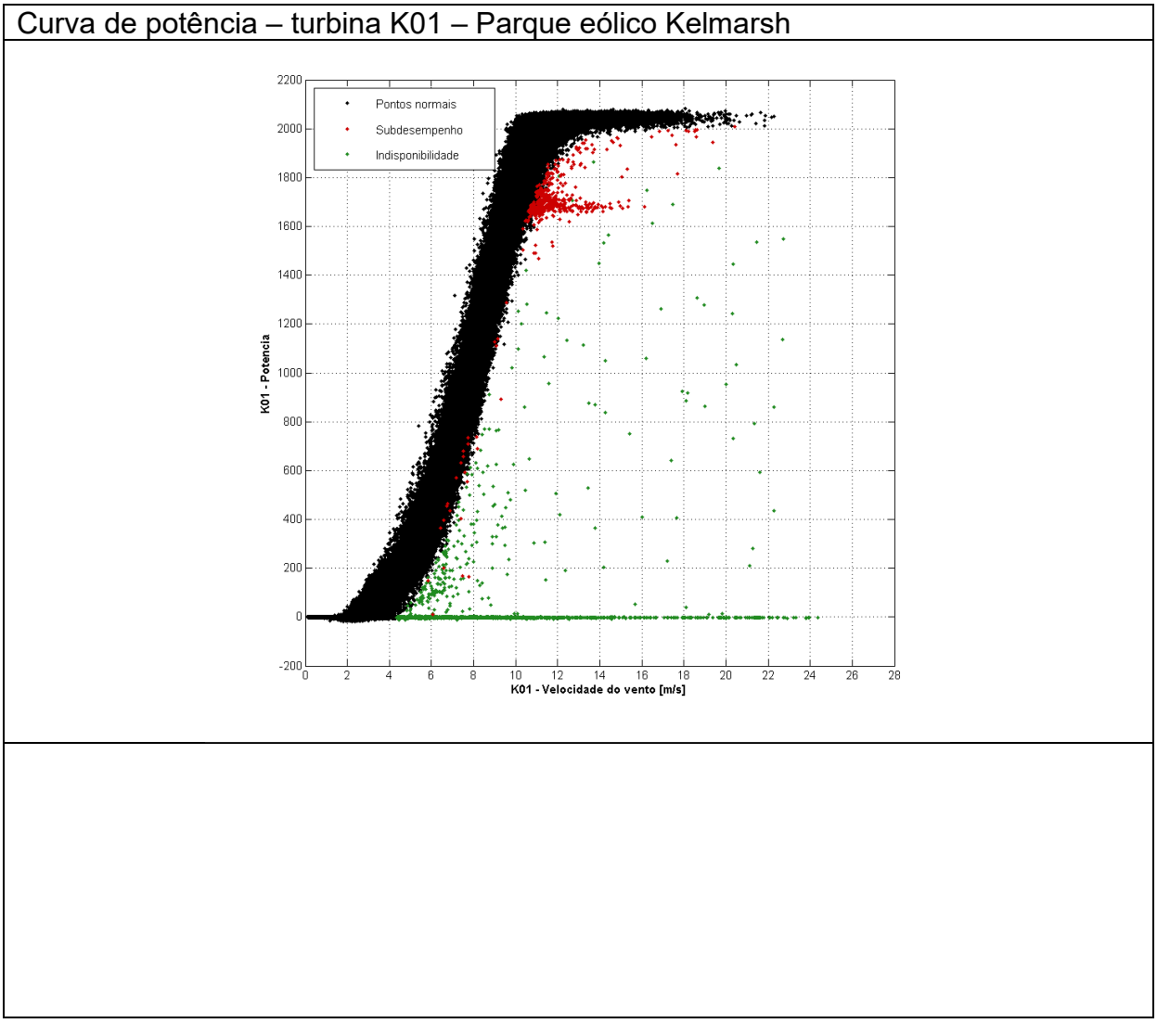
# 5 RESULTADOS

## 5.1 LIMPEZA DA CURVA DE POTÊNCIA PELO ESPECIALISTA

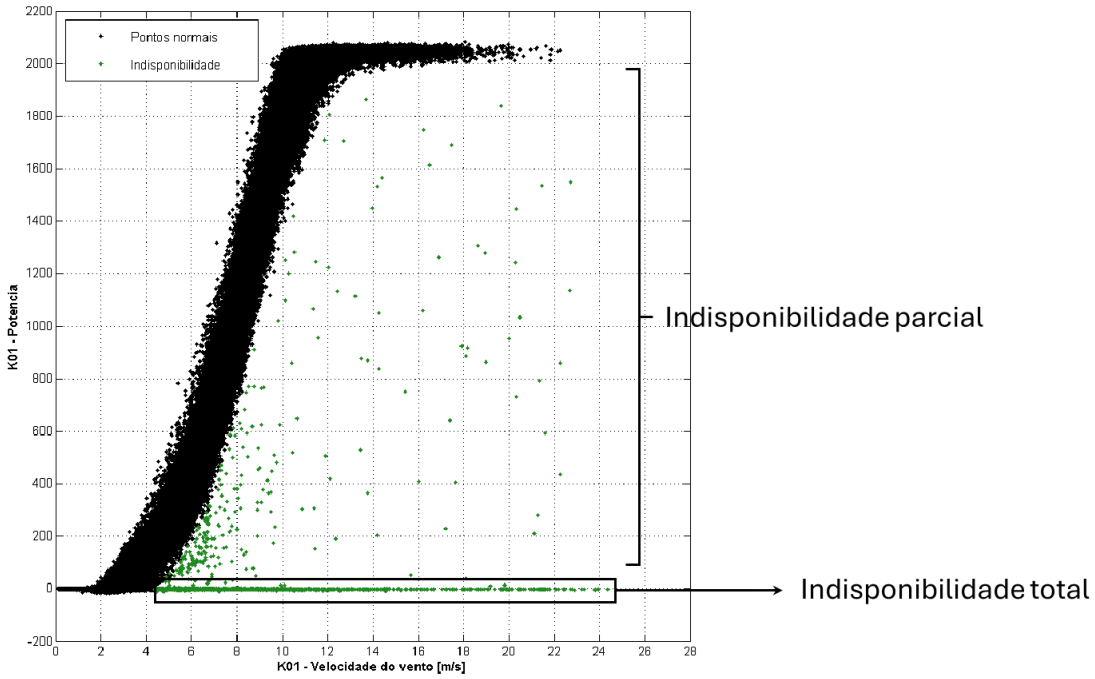
A limpeza das curvas de potência foi realizada por um software interno da companhia de certificação e classificação DNV, amplamente validado e utilizado globalmente. Este programa permite a visualização de qualquer sinal na resolução temporal desejada, além da marcação manual de pontos específicos. Além dos sinais principais de velocidade do vento e potência, sinais auxiliares de ângulo de *pitch*, velocidade do rotor e ângulo da nacele foram utilizados. Importante mencionar que a classe 0 denota pontos normais, classe 1, indisponibilidade e classe 2, subdesempenho.

A Figura 5-1 e a Figura 5-2 apresentam, respectivamente, as marcações de pontos das turbinas utilizadas para treino e teste.

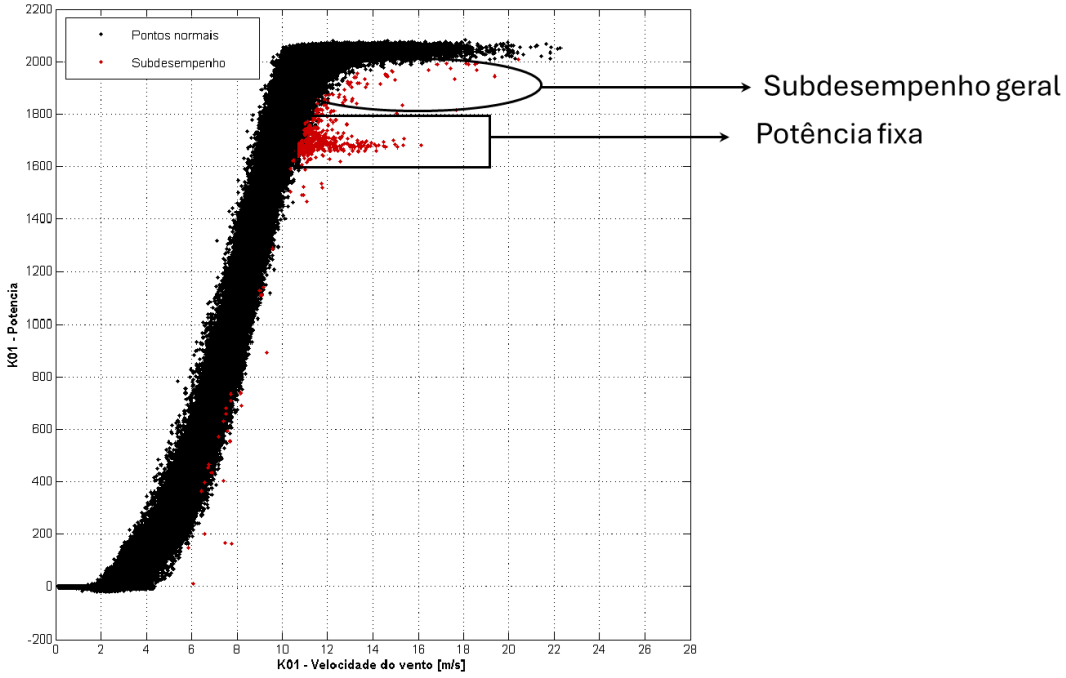
**Figura 5-1 Curva de potência da turbina K01 do parque Kelmarsh manualmente limpa.**



Indisponibilidade



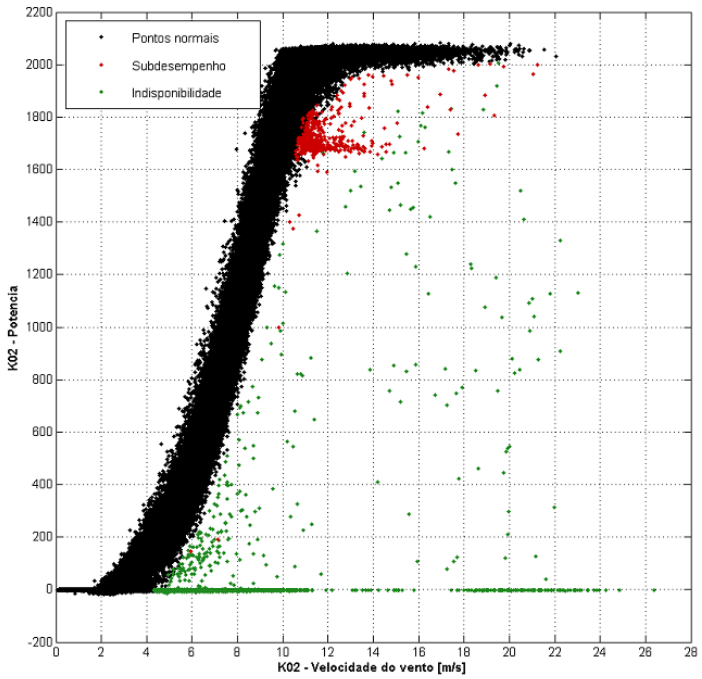

Subdesempenho



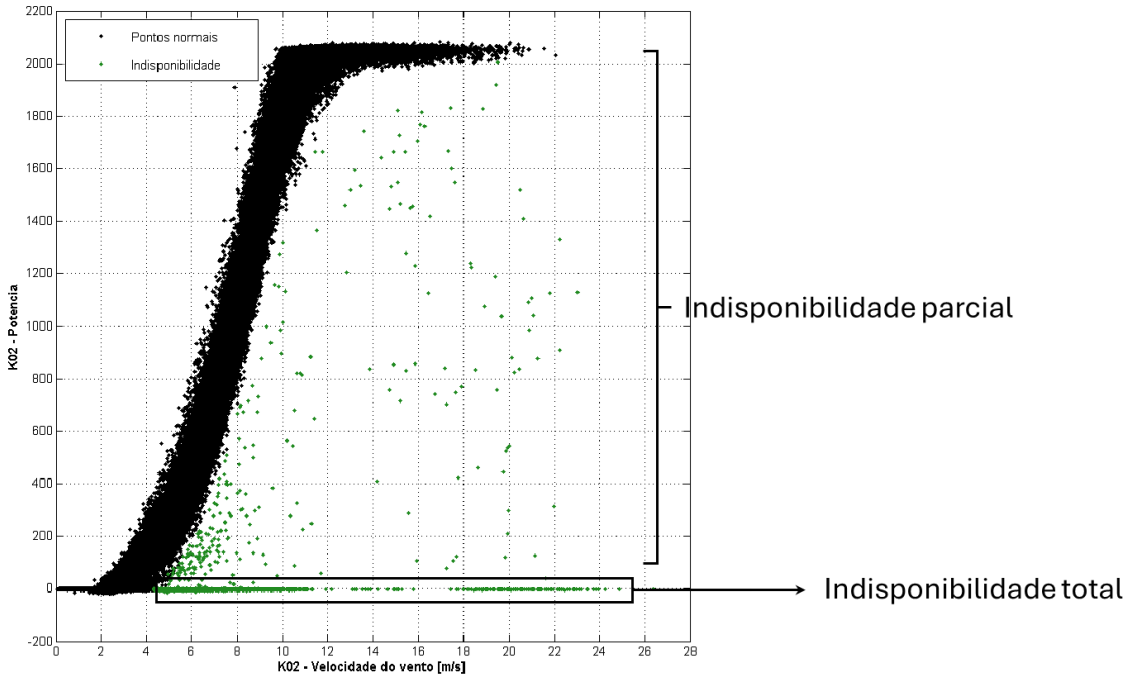

Fonte: a Autora (2024).

**Figura 5-2 Curva de potência da turbina K02 do parque Kelmarsh manualmente limpa.**

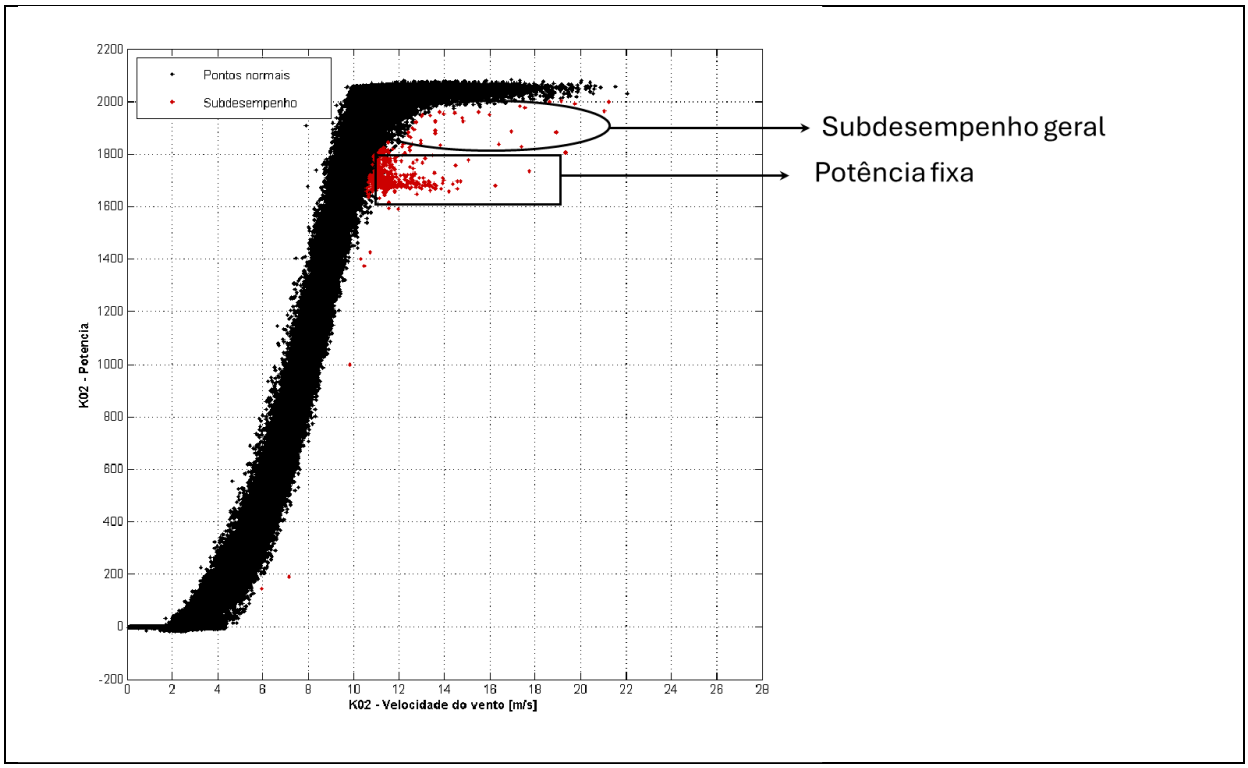
Curva de potência – turbina K02 – Parque eólico Kelmarsh




Indisponibilidade




Subdesempenho



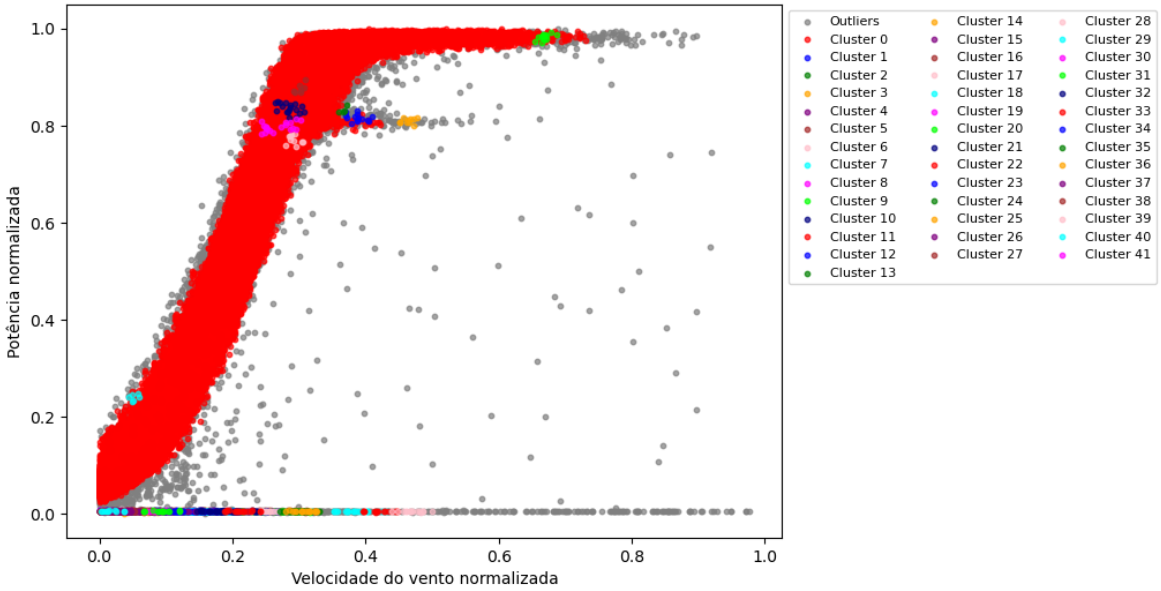

Fonte: a Autora (2024).

## 5.2 DBSCAN

Para detecção automática de *outliers*, o primeiro algoritmo utilizado foi a clusterização com o DBSCAN. A metodologia empregada é a descrita na seção 4.3.2.1. Os resultados são apresentados variando-se a quantidade de vizinhos mais próximos K e o valor de *Min_samples*. Para avaliação dos resultados considerou-se a classe 0 como pontos normais e a 1 como anômalos. As Figuras a seguir apresentam os resultados encontrados. Para interpretação dos resultados, considera-se que o *cluster* 0 denota os pontos normais e os demais clusters, usualmente fora da curva de potência, por simplificação, foram considerados como dados da classe 1. As métricas de precisão, *recall* e *F1-score* foram calculadas baseadas nesta consideração.

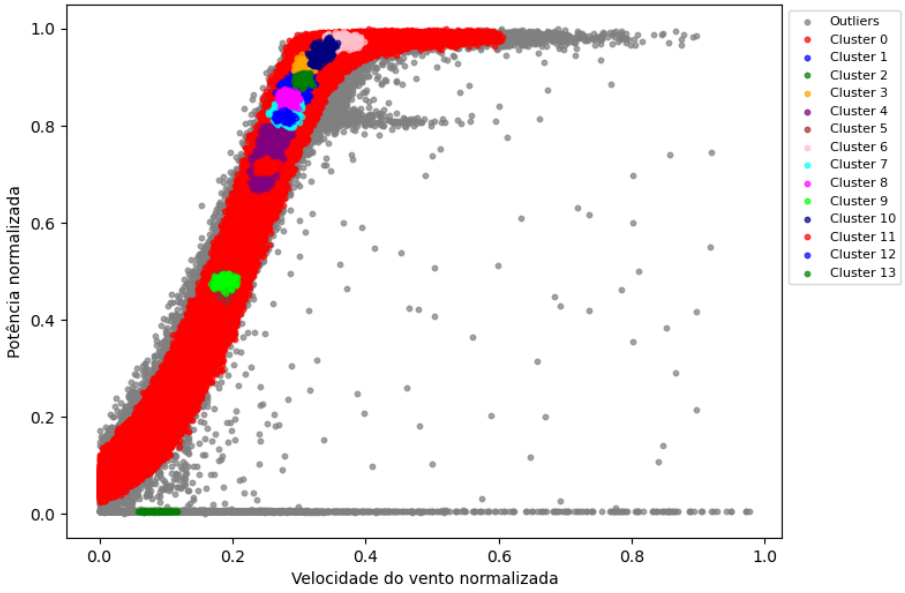
**Figura 5-3 Resultados do DBSCAN com K = 7**

Min_samples = 10



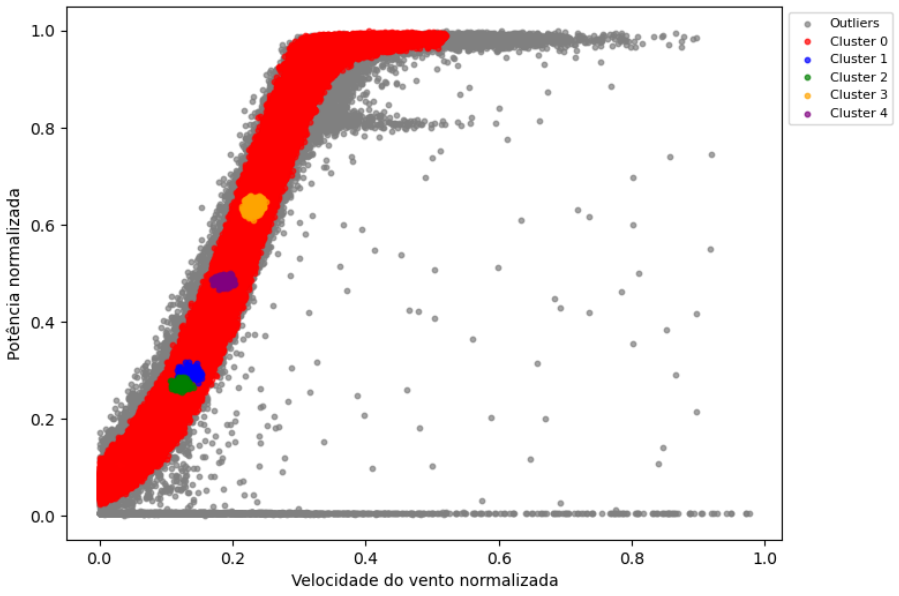

| Classe | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| 0 | 0.99 | 0.98 | 0.99 |
| 1 | 0.25 | 0.45 | 0.32 |

Min_samples = 50



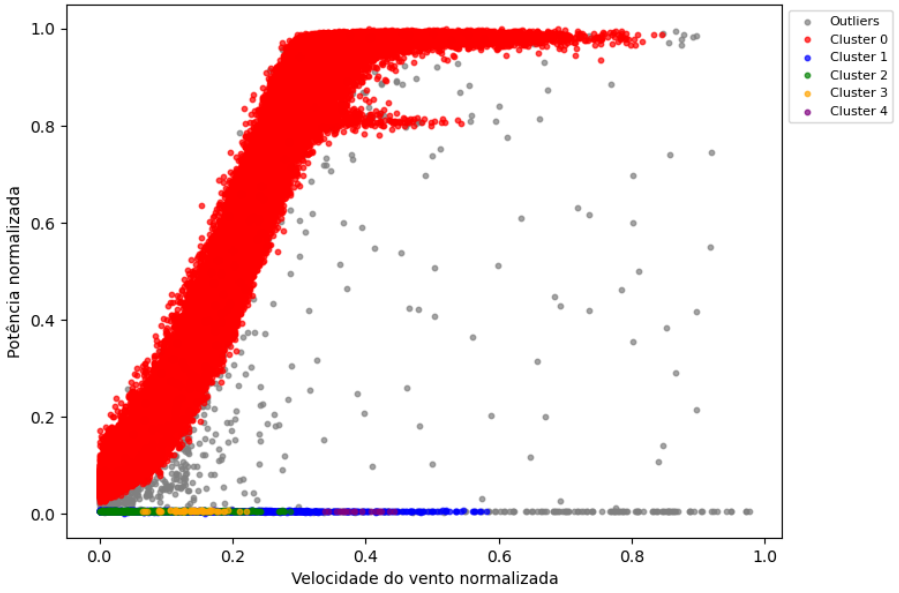

| Classe | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| 0 | 1.00 | 0.90 | 0.95 |
| 1 | 0.12 | 0.97 | 0.22 |

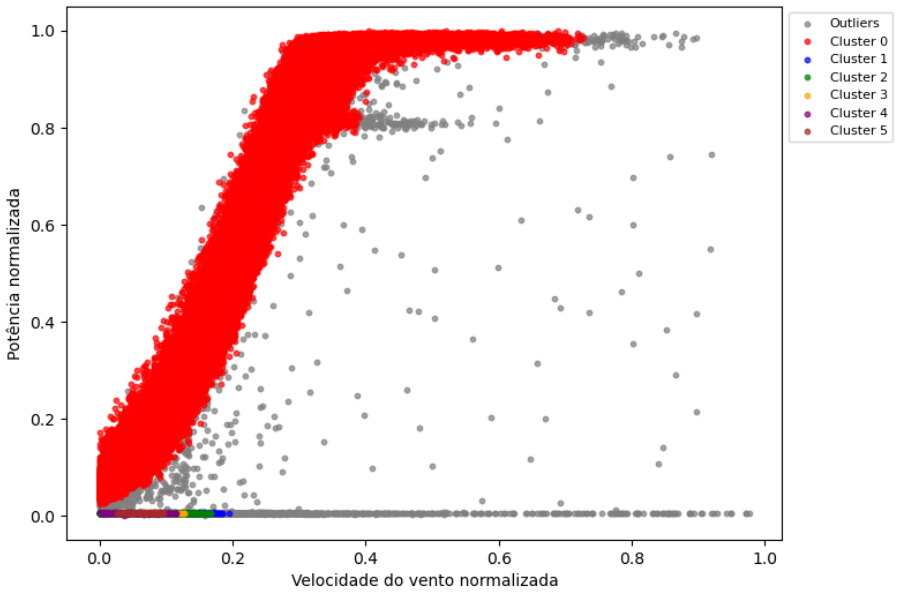
Min_samples = 100



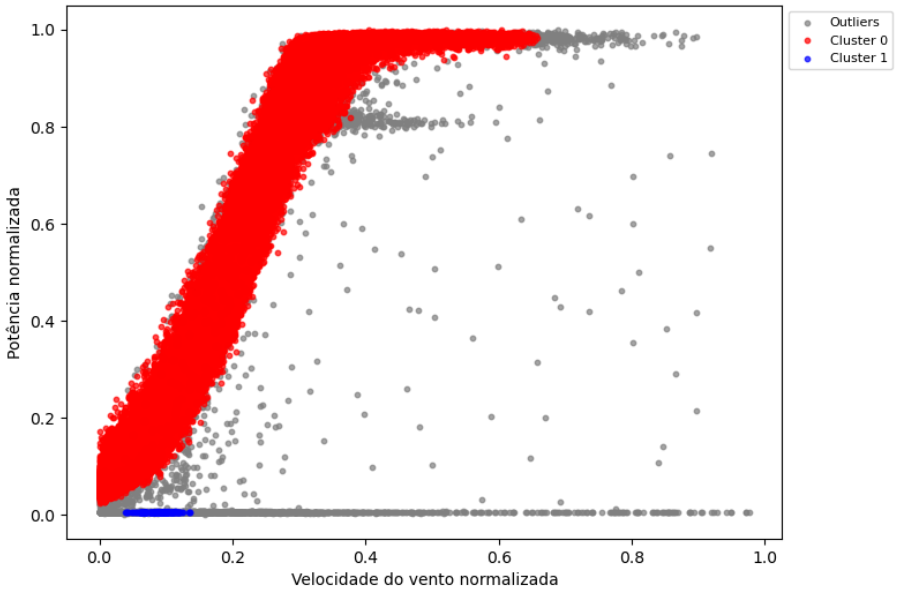

| Classe | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| 0 | 1.00 | 0.80 | 0.89 |
| 1 | 0.07 | 1.00 | 0.13 |

**Figura 5-4 Resultados com o DBSCAN utilizando K = 9.**

| Min_samples = 10 | | | |
|---|---|---|---|
|  | | | |
| Classe | Precisão | *Recall* | *F1-score* |
| 0 | 0.99 | 1.00 | 0.99 |
| 1 | 0.43 | 0.18 | 0.25 |
| Min_samples = 50 | | | |



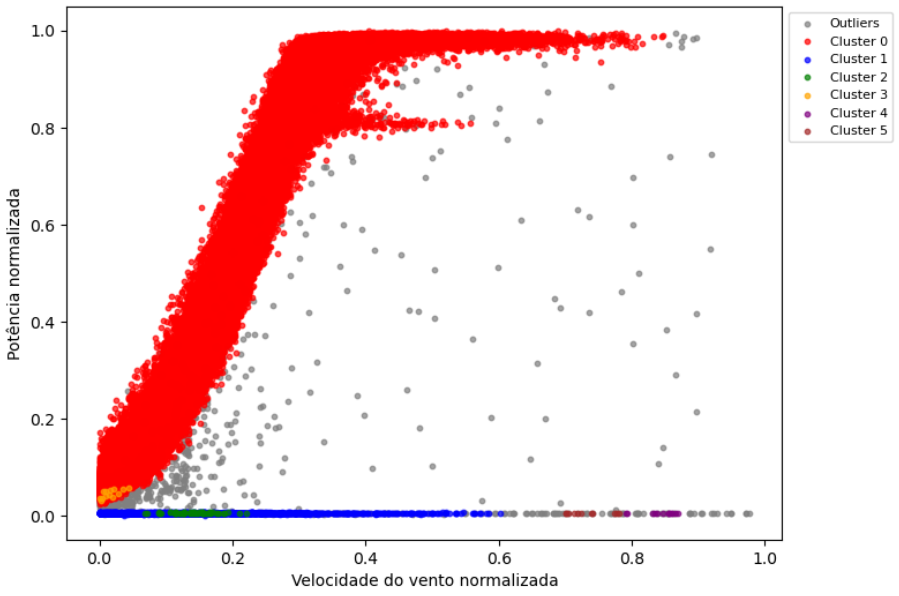

| Classe | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| 0 | 0.99 | 0.99 | 0.99 |
| 1 | 0.57 | 0.56 | 0.57 |
| Min_samples = 100 | | | |



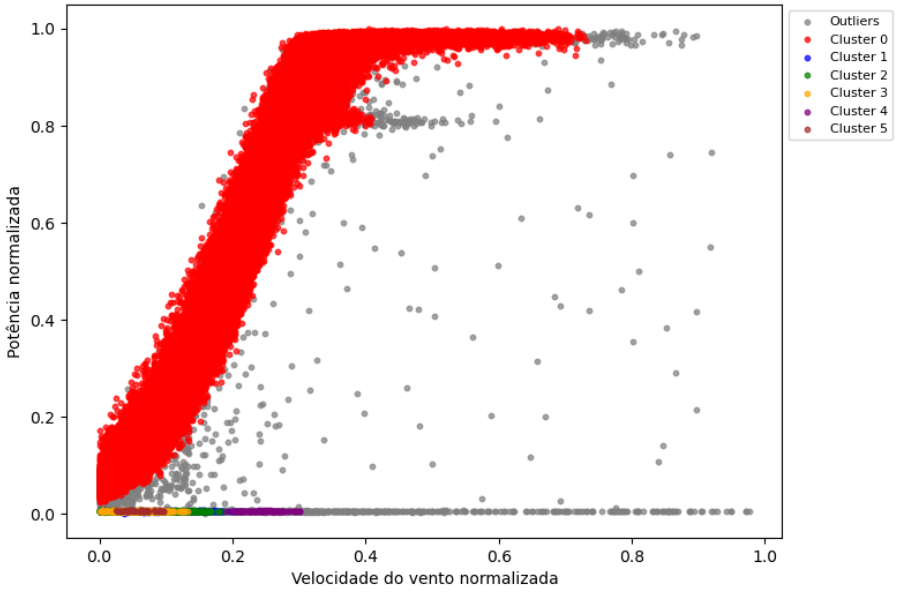

| Classe | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| 0 | 1.00 | 0.98 | 0.99 |
| 1 | 0.43 | 0.89 | 0.58 |

**Figura 5-5 Resultados com o DBSCAN para K = 11.**

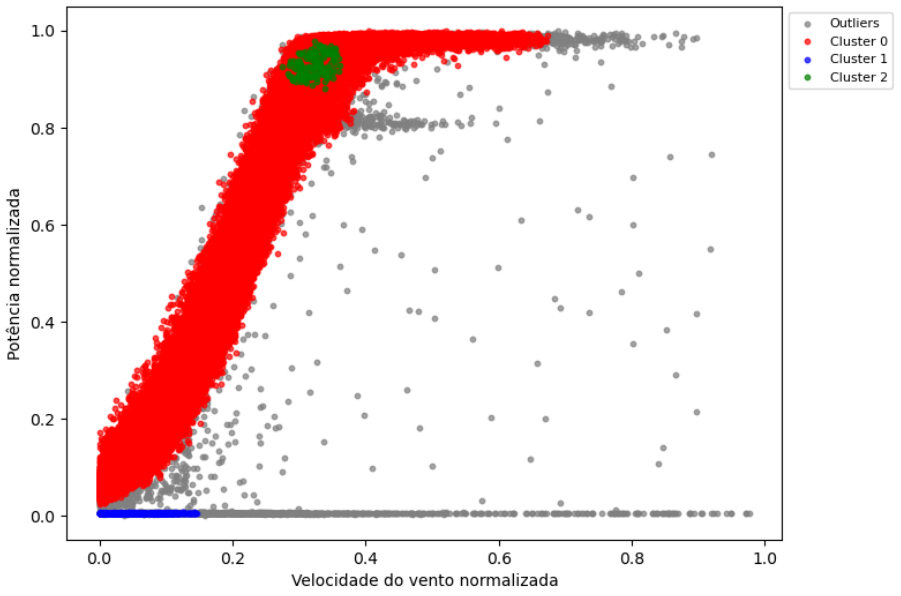
| Min_samples = 10 |
|---|




| Classe | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| 0 | 0.99 | 1.00 | 0.99 |
| 1 | 0.42 | 0.16 | 0.23 |
| Min_samples = 50 | | | |




| Classe | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| 0 | 0.99 | 1.00 | 0.99 |
| 1 | 0.59 | 0.47 | 0.52 |
| Min_samples = 100 | | | |




| Classe | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| 0 | 1.00 | 0.99 | 0.99 |
| 1 | 0.53 | 0.81 | 0.65 |

As imagens apresentadas demonstram o desempenho do algoritmo DBSCAN na classificação dos dados. Observa-se que o mesmo teve um excelente desempenho na identificação da classe 0, com métricas de precisão, *recall* e *F1-score* praticamente iguais a 1. Além disso, na detecção de dados anômalos, os resultados foram satisfatórios em alguns casos, considerando o total de verdadeiros positivos, com *recall* chegando a 0,81.

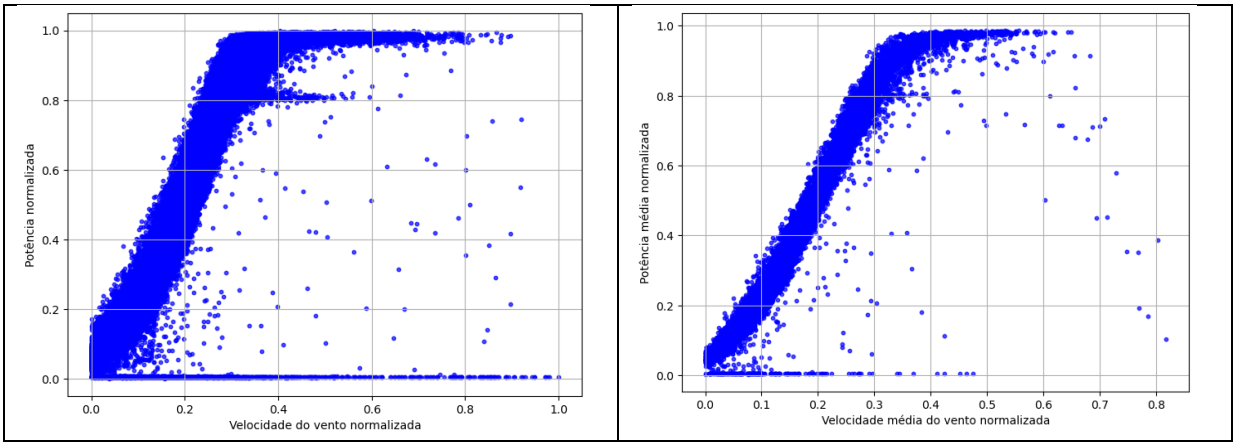
No entanto, nota-se que o algoritmo enfrentou dificuldades na subdivisão dos *outliers*, não conseguindo distinguir de forma clara entre indisponibilidade e subdesempenho. Esse comportamento indica que a divisão dessas categorias poderia exigir ajustes nos parâmetros do DBSCAN ou a combinação com outras abordagens para melhorar a diferenciação.

## 5.3 DBSCAN COM PARÂMETROS ESTATÍSTICOS E JANELA DESLIZANTE

Com o objetivo de aprimorar a metodologia baseada no DBSCAN, foi utilizado um algoritmo que incorpora parâmetros estatísticos como dados de entrada. Além disso, implementou-se uma abordagem com janela deslizante, permitindo capturar a dependência temporal dos dados anômalos. O objetivo da adaptação é melhorar a identificação de padrões e a divisão dos *outliers* em indisponibilidade e subdesempenho.

Conforme mencionado na seção 4.3.2.2, foi adotado um comprimento de janela de 6 horas com um passo de 2 horas, com o objetivo de se obter uma curva de potência próxima da real a partir dos valores médios. Para cada janela temporal, parâmetros estatísticos são extraídos a partir do *tsfresh*. A Figura 5-6 apresenta a curva de potência original versus a curva de potência média, construída a partir dos valores médios obtidos pelo *tsfresh*. Cada ponto no gráfico à direita representa um intervalo de 6 horas.

**Figura 5-6 À esquerda, a curva de potência original da turbina K01. À direita, a curva de potência com pontos médios, advindos do tsfresh.**




É importante mencionar que para o DBSCAN são utilizados, além da média, parâmetros estatísticos que também servem de auxílio na detecção de anomalias. Então, mesmo que algum ponto tenha sido suavizado pela média, o desvio padrão, por exemplo, pode ajudar na identificação de pontos discrepantes. A amplitude, que é a diferença entre o maior e o menor valor dentro da janela, também pode ajudar na identificação de picos que apareçam, mesmo que na média o valor tenha sido diluído.

Os parâmetros estatísticos calculados pelo *tsfresh* são testados no intuito de se avaliar o desempenho do DBSCAN. Os testes podem ser divididos em três grupos:

1. Grupo 1: apenas valores de média e de desvio padrão;
2. Grupo 2: média, desvio padrão e parâmetros da correlação linear;
3. Grupo 3: média, desvio padrão, parâmetros da correlação linear, mínimo, máximo e posições de mínimo e máximo.

O valor de Épsilon é de metade do valor do joelho calculado, assim como para o primeiro caso do DBSCAN e o Min_samples é igual a 100, visto que este foi o valor

que apresentou o melhor desempenho para o DBSCAN. Os resultados para cada um dos testes são apresentados na Figura 5-7, Figura 5-8 e Figura 5-9.

**Figura 5-7 Agrupamento em *clusters* utilizando DBSCAN com parâmetros estatísticos em janela deslizante para o Grupo 1.**

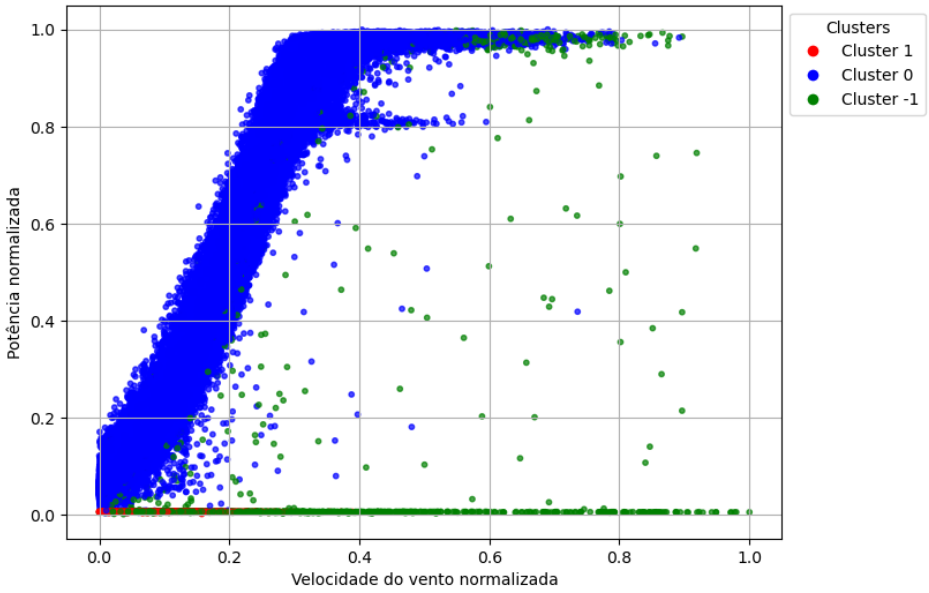



**Figura 5-8 Agrupamento em *clusters* utilizando DBSCAN com parâmetros estatísticos em janela deslizante para o Grupo 2.**

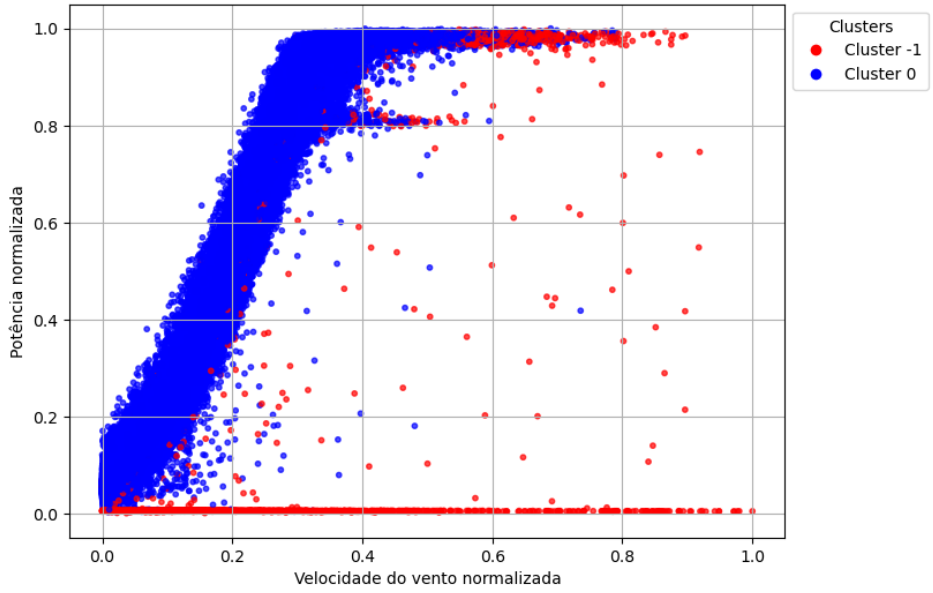
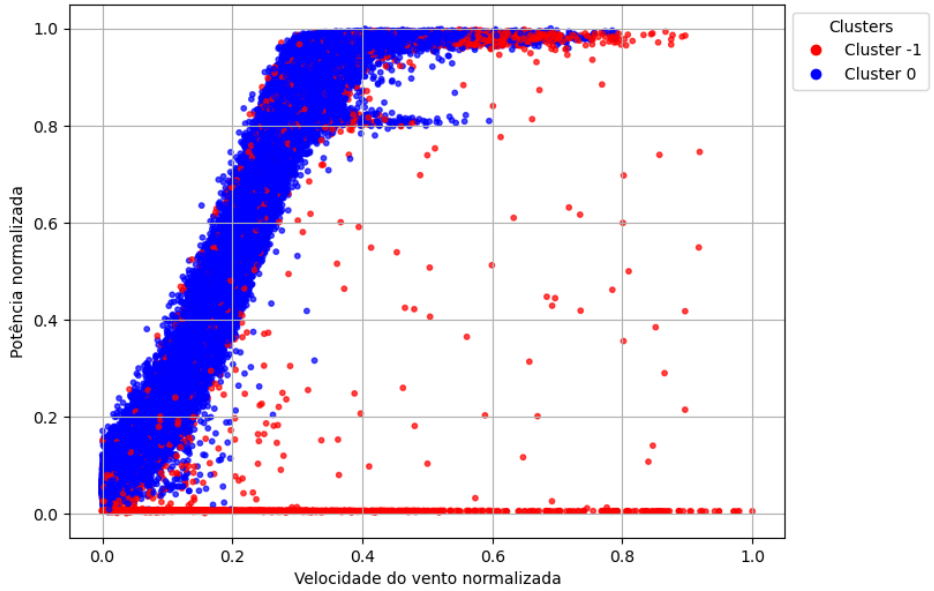


**Figura 5-9 Agrupamento em *clusters* utilizando DBSCAN com parâmetros estatísticos em janela deslizante para o Grupo 3.**




Pelas Figuras apresentadas, pode-se observar que mais uma vez o algoritmo foi bem sucedido na detecção da indisponibilidade total e em pontos esparsos abaixo da curva. Porém, novamente, não foi capaz de separar pontos de indisponibilidade e de subdesempenho. A Tabela 5-1, Tabela 5-2, Tabela 5-3 apresentam os resultados de precisão, *recall* e *F1-score,* considerando cluster 0 como pontos normais e diferente de 0 como pontos anômalos para estimativa de métricas de precisão, *recall* e *F1-score* para cada grupo de variáveis.

**Tabela 5-1 Resultados do DBSCAN com parâmetros estatísticos e janela deslizante para o grupo 1.**

| | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| Classe 0 | 1,00 | 0,99 | 0,99 |
| Classe 1 | 0,50 | 0,79 | 0,62 |

**Tabela 5-2 Resultados do DBSCAN com parâmetros estatísticos e janela deslizante para o grupo 2.**

| | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| Classe 0 | 1,00 | 0,99 | 0,99 |
| Classe 1 | 0,51 | 0,80 | 0,62 |

**Tabela 5-3 Resultados do DBSCAN com parâmetros estatísticos e janela deslizante para o grupo 3.**

| | Precisão | *Recall* | *F1-score* |
|---|---|---|---|
| Classe 0 | 0,99 | 0,94 | 0,97 |
| Classe 1 | 0,17 | 0,86 | 0,28 |

Assim como nos primeiros testes com o DBSCAN, o algoritmo apresenta um bom desempenho na identificação da classe majoritária. No entanto, para os dados anômalos, os grupos 1 e 2 são melhores, ambos com um *F1-score* de 0,62, enquanto o grupo 3 tem um desempenho inferior, com um *F1-score* de apenas 0,28. Embora o grupo 3 apresente um *recall* mais alto, identificando uma maior quantidade de pontos anômalos, sua precisão é muito baixa. Isso indica que muitos falsos positivos (ou seja, falhas detectadas erroneamente) foram classificados, o que impactou negativamente o *F1-score*. E assim, como acontece no DBSCAN clássico, o DBSCAN com janela deslizante também falha na diferenciação entre as classes.

## 5.4 *AUTOENCODER* CLÁSSICO COM KAN (AE-KAN)

Conforme descrito na seção 4.3.2.3, a terceira metodologia a ser testada é o *autoencoder* combinado com KAN (AE-KAN). A Tabela 5-4 apresenta os parâmetros utilizados em cada rodada de treino do *autoencoder* para o ajuste de hiperparâmetros e o respectivo erro de reconstrução.

**Tabela 5-4 Parâmetros para ajustes de hiperparâmetros do *autoencoder*.**

| Rodada | Número de camadas | Dimensão latente | Otimizador | Número de épocas | Erro de reconstrução |
|---|---|---|---|---|---|
| 1 | 3+3 | 3 | Adam | 100 | 0,19 |
| 2 | 3+3 | 4 | Adam | 100 | 0,21 |
| 3 | 3+3 | 3 | Adam | 500 | 0,12 |
| 4 | 3+3 | 4 | Adam | 500 | 0,07 |
| 5 | 3+3 | 3 | RMSProp | 100 | 0,15 |
| 6 | 3+3 | 4 | RMSProp | 100 | 0,08 |
| 7 | 3+3 | 3 | RMSProp | 500 | 0,09 |
| 8 | 3+3 | 4 | RMSProp | 500 | 0,03 |
| 9 | 3+3 | 3 | AdamW | 100 | 0,19 |
| 10 | 3+3 | 4 | AdamW | 100 | 0,21 |
| 11 | 3+3 | 3 | AdamW | 500 | 0,12 |
| 12 | 3+3 | 4 | AdamW | 500 | 0,07 |
| 13 | 4+4 | 3 | Adam | 100 | 0,18 |
| 14 | 4+4 | 4 | Adam | 100 | 0,10 |

| | | | | | |
|---|---|---|---|---|---|
| 15 | 4+4 | 3 | Adam | 500 | 0,06 |
| 16 | 4+4 | 4 | Adam | 500 | 0,05 |
| 17 | 4+4 | 3 | RMSProp | 100 | 0,09 |
| 18 | 4+4 | 4 | RMSProp | 100 | 0,06 |
| 19 | 4+4 | 3 | RMSProp | 500 | 0,03 |
| 20 | 4+4 | 4 | RMSProp | 500 | 0,03 |
| 21 | 4+4 | 3 | AdamW | 100 | 0,18 |
| 22 | 4+4 | 4 | AdamW | 100 | 0,10 |
| 23 | 4+4 | 3 | AdamW | 500 | 0,06 |
| 24 | 4+4 | 4 | AdamW | 500 | 0,05 |

A curva de potência com os dados originais e reconstruídos de cada rodada é apresentada no Apêndice A. Observa-se que as rodadas 08, 19 e 20 são as que apresentam menor erro de reconstrução. Avaliando também a curva de potência, o treino de número 20 apresentou o melhor desempenho e, portanto, seus resultados são usados como dados de entrada para a KAN. São usados tanto os dados da turbina K01, para treino da KAN, quanto os dados da turbina K02, para teste da KAN.

Na rede Kolmogorov-Arnold, os hiperparâmetros são ajustados e as métricas de classificação – acurácia global, AUC ROC, precisão, *recall* e *F1-score* - são calculadas a fim de se comparar o desempenho de cada um dos treinos. Os resultados são apresentados na Tabela 5-5, em que P denota precisão, R, *recall* e F1, o *F1-score*, seguidos de um hífen e da respectiva classe.

**Tabela 5-5 Acurácia, AUC-ROC, precisão, *recall*, *F1-score* para cada uma das classes durante o ajuste de hiperparâmetros da KAN.**

| Rodada | Acurácia | AUC ROC | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1,00 | 0,83 | 1,00 | 0,83 | 0,09 | 0,98 | 0,99 | 0,74 | 0,99 | 0,90 | 0,16 |
| 2 | 1,00 | 0,84 | 1,00 | 0,85 | 0,15 | 0,99 | 0,98 | 0,84 | 0,99 | 0,91 | 0,25 |
| 3 | 1,00 | 0,85 | 1,00 | 0,84 | 0,13 | 0,99 | 0,99 | 0,87 | 0,99 | 0,91 | 0,23 |
| 4 | 1,00 | 0,85 | 1,00 | 0,83 | 0,09 | 0,98 | 0,98 | 0,81 | 0,99 | 0,90 | 0,17 |
| 5 | 1,00 | 0,84 | 1,00 | 0,82 | 0,05 | 0,97 | 0,99 | 0,65 | 0,99 | 0,89 | 0,09 |
| 6 | 1,00 | 0,85 | 1,00 | 0,83 | 0,14 | 0,99 | 0,99 | 0,79 | 0,99 | 0,90 | 0,23 |
| 7 | 1,00 | 0,83 | 1,00 | 0,85 | 0,06 | 0,97 | 0,99 | 0,69 | 0,99 | 0,91 | 0,11 |
| 8 | 1,00 | 0,84 | 1,00 | 0,82 | 0,16 | 0,99 | 0,99 | 0,80 | 0,99 | 0,89 | 0,26 |
| 9 | 1,00 | 0,85 | 1,00 | 0,84 | 0,12 | 0,99 | 0,99 | 0,75 | 0,99 | 0,90 | 0,20 |
| 10 | 1,00 | 0,86 | 1,00 | 0,83 | 0,26 | 0,99 | 0,99 | 0,85 | 1,00 | 0,90 | 0,40 |
| 11 | 1,00 | 0,85 | 1,00 | 0,84 | 0,13 | 0,99 | 0,98 | 0,78 | 0,99 | 0,91 | 0,22 |
| 12 | 1,00 | 0,86 | 1,00 | 0,84 | 0,07 | 0,98 | 0,99 | 0,64 | 0,99 | 0,91 | 0,12 |
| 13 | 1,00 | 0,85 | 1,00 | 0,83 | 0,20 | 0,99 | 0,99 | 0,80 | 1,00 | 0,90 | 0,31 |
| 14 | 1,00 | 0,85 | 1,00 | 0,85 | 0,06 | 0,98 | 0,99 | 0,65 | 0,99 | 0,91 | 0,11 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 1,00 | 0,87 | 1,00 | 0,84 | 0,13 | 0,99 | 0,99 | 0,76 | 0,99 | 0,90 | 0,23 |
| 16 | 1,00 | 0,85 | 1,00 | 0,81 | 0,07 | 0,98 | 0,99 | 0,67 | 0,99 | 0,89 | 0,13 |

Observa-se que, para a classe 0, os resultados são excelentes, com métricas próximas de 1. Para a classe 1, o desempenho também é satisfatório, com o *F1-score* atingindo 0,91. No caso da classe 2, há uma baixa precisão e um *recall* razoável, indicando que, apesar do alto número de falsos positivos, o modelo ainda consegue identificar boa parte dos verdadeiros positivos. Na identificação de anomalias, é preferível um maior número de falsos positivos do que de falsos negativos, garantindo que menos falhas reais passem despercebidas.

O critério para a seleção dos hiperparâmetros foi o valor do *recall* para a classe 2, uma vez que os resultados para as classes 0 e 1 são muito semelhantes. Com base nisso, os hiperparâmetros da rodada 3 são escolhidos para teste. O modelo é testado em outras turbinas do parque, a K02, K03 e K04. A Tabela 5-6, Tabela 5-7 e Tabela 5-8 apresentam os resultados dos testes.

**Tabela 5-6 Acurácia, AUC ROC, precisão, *recall* e *F1-score* do teste com a turbina K02 – modelo AE-KAN.**

| Acurácia | AUC ROC | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,98 | 1,00 | 1,00 | 0,85 | 0,09 | 0,98 | 0,99 | 0,69 | 0,99 | 0,91 | 0,17 |

**Tabela 5-7 Acurácia, AUC ROC, precisão, *recall* e *F1-score* do teste com a turbina K03 – modelo AE-KAN.**

| Acurácia | AUC ROC | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,97 | 1,00 | 1,00 | 0,88 | 0,06 | 0,97 | 0,99 | 1,00 | 0,99 | 0,93 | 0,11 |

**Tabela 5-8 Acurácia, AUC ROC, precisão, *recall* e *F1-score* do teste com a turbina K04 – modelo AE-KAN.**

| Acurácia | AUC ROC | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,97 | 1,00 | 1,00 | 0,88 | 0,07 | 0,97 | 1,00 | 0,97 | 0,98 | 0,94 | 0,12 |

Durante o processo de treinamento, o modelo apresentou um desempenho consistente na distinção entre as classes 0 e 1, resultado que também se confirmou na etapa de teste. O AE-KAN teve capacidade razoável de reconhecer corretamente instâncias da classe 2, com *recall* chegando a 1,00 para a turbina K03. Contudo, sua

maior fragilidade está relacionada à baixa precisão nessa classe, já que uma quantidade considerável de dados normais foi equivocadamente rotulada como pertencente à classe 2. Isto é apresentado na matriz de confusão, para cada uma das turbinas, ilustradas na Figura 5-10, Figura 5-11 e Figura 5-12.

**Figura 5-10 Matriz de confusão do modelo AE-KAN testado na turbina K02.**

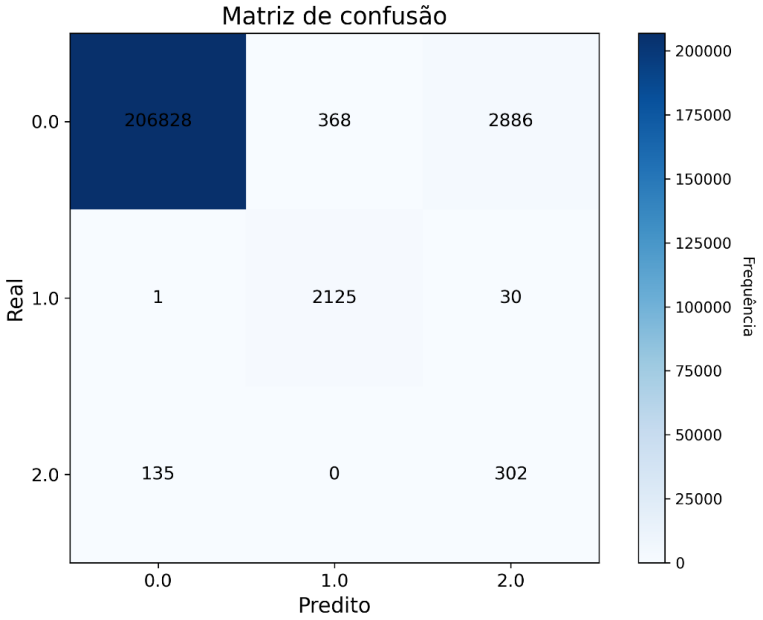



**Figura 5-11 Matriz de confusão do modelo AE-KAN testado na turbina K03.**

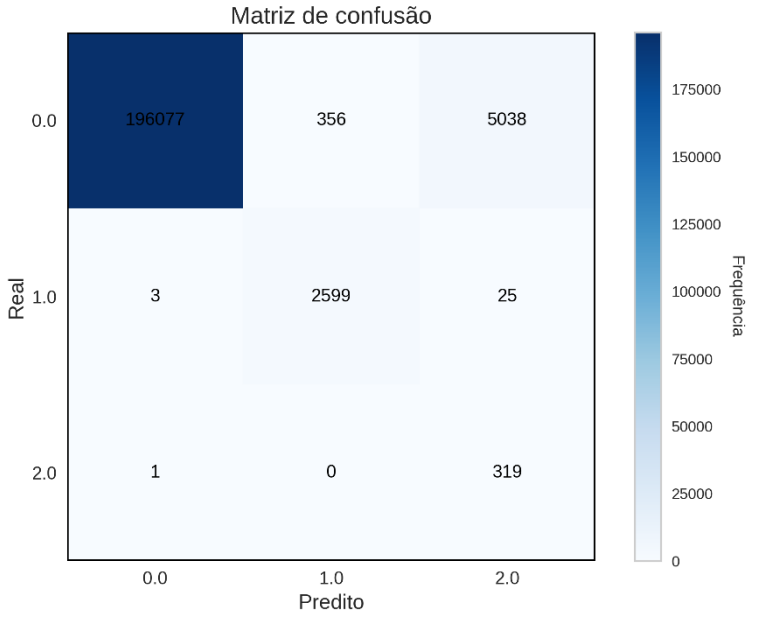
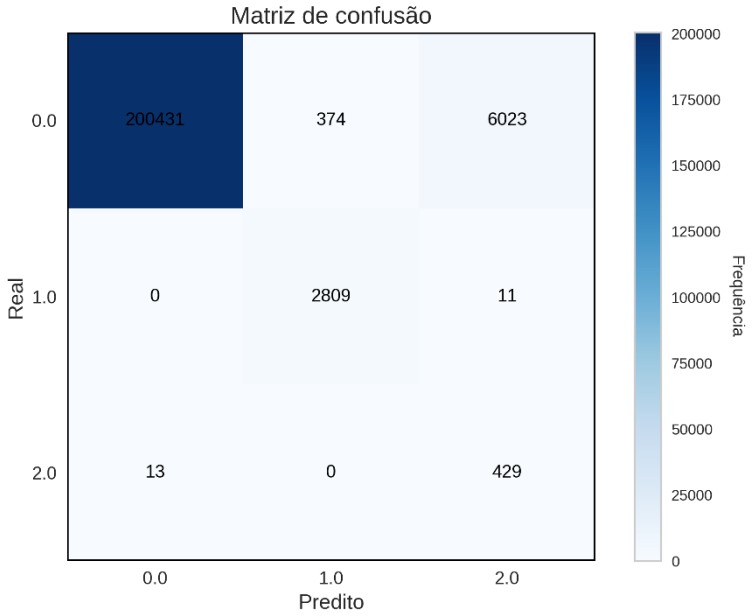


**Figura 5-12 Matriz de confusão do modelo AE-KAN testado na turbina K04.**




A Figura 5-13, Figura 5-14, Figura 5-15 apresentam os gráficos da área sob a curva ROC para as três classes, para as turbinas K02, K03 e K04, respectivamente.

**Figura 5-13 Área sob a curva ROC, para cada classe - modelo AE-KAN testado na turbina K02.**

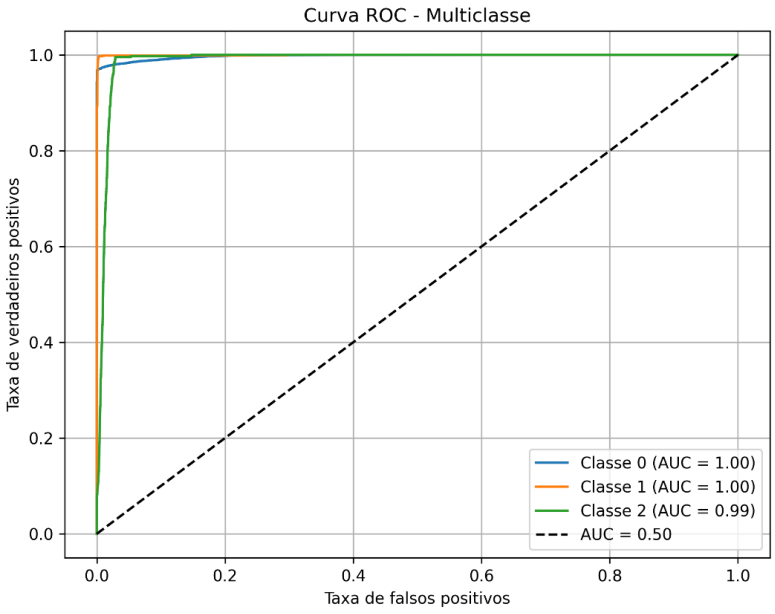


**Figura 5-14 Área sob a curva ROC, para cada classe - modelo AE-KAN testado na turbina K03.**

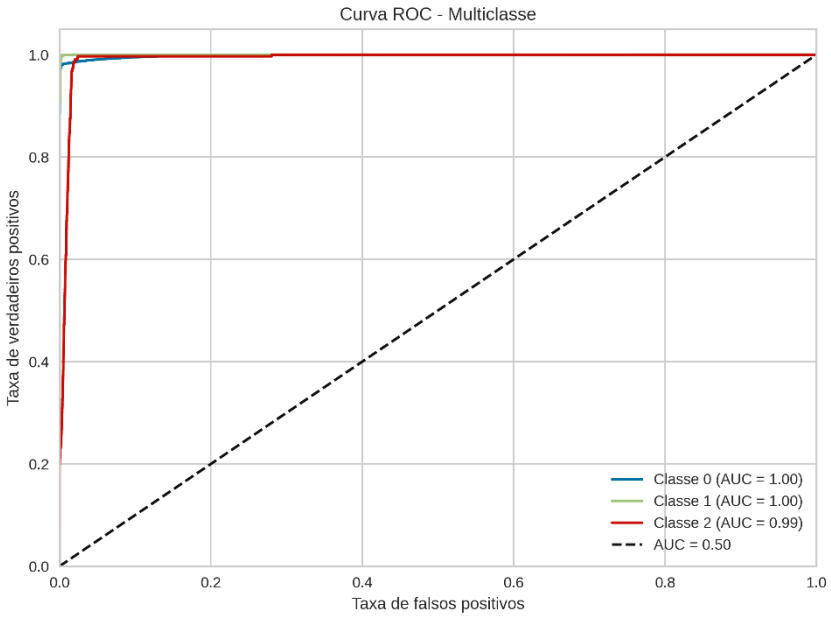
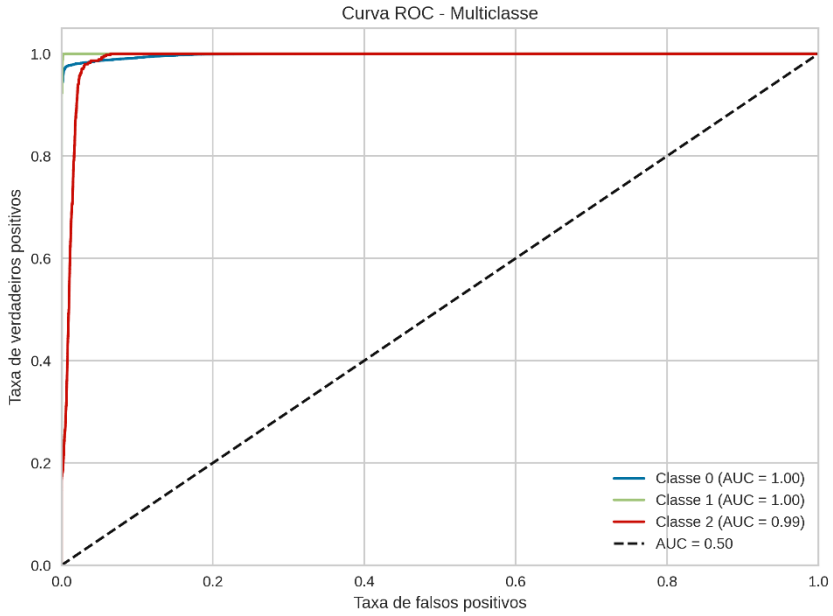



**Figura 5-15 Área sob a curva ROC, para cada classe - modelo AE-KAN testado na turbina K04.**




A área sob a curva ROC indica que o modelo teve um desempenho excepcional na segmentação entre as classes 0, 1 e 2, mostrando a eficácia do modelo testado.

A Figura 5-16, Figura 5-25, Figura 5-28 mostram a curva de potência limpa, ou seja, com a classificação em pontos normais, indisponibilidade e subdesempenho para as turbinas K02, K03 a K04, respectivamente.

**Figura 5-16 Curva de potência limpa, com pontos classificados em normais, indisponíveis e subdesempenho para a turbina K02 - modelo AE-KAN.**

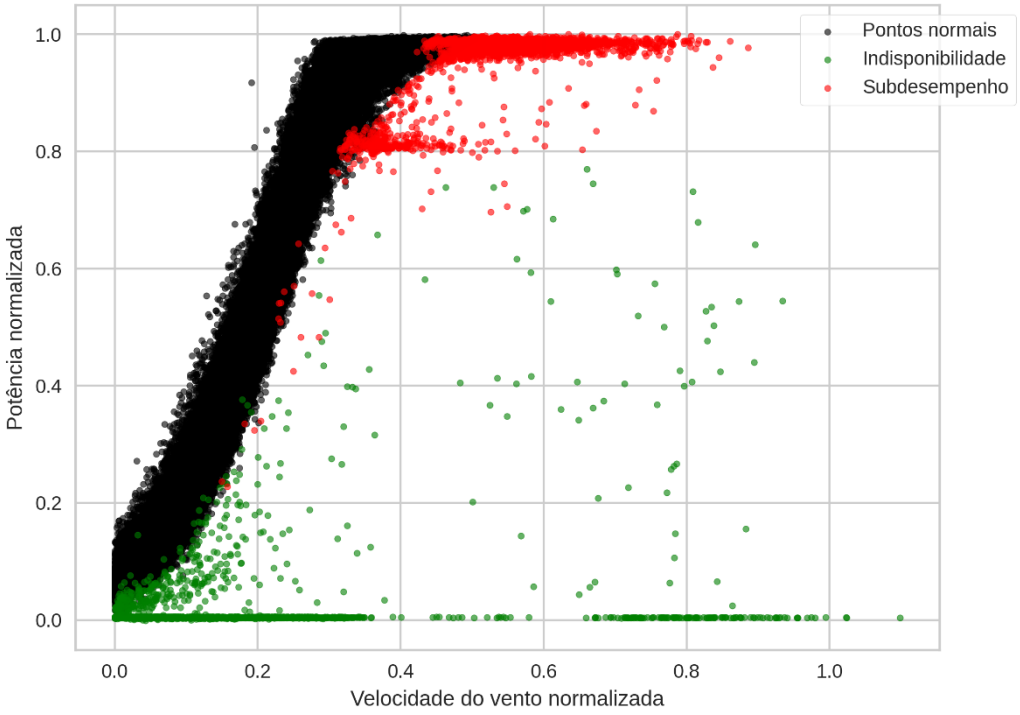



**Figura 5-17 Curva de potência limpa, com pontos classificados em normais, indisponíveis e subdesempenho para a turbina K03 - modelo AE-KAN.**

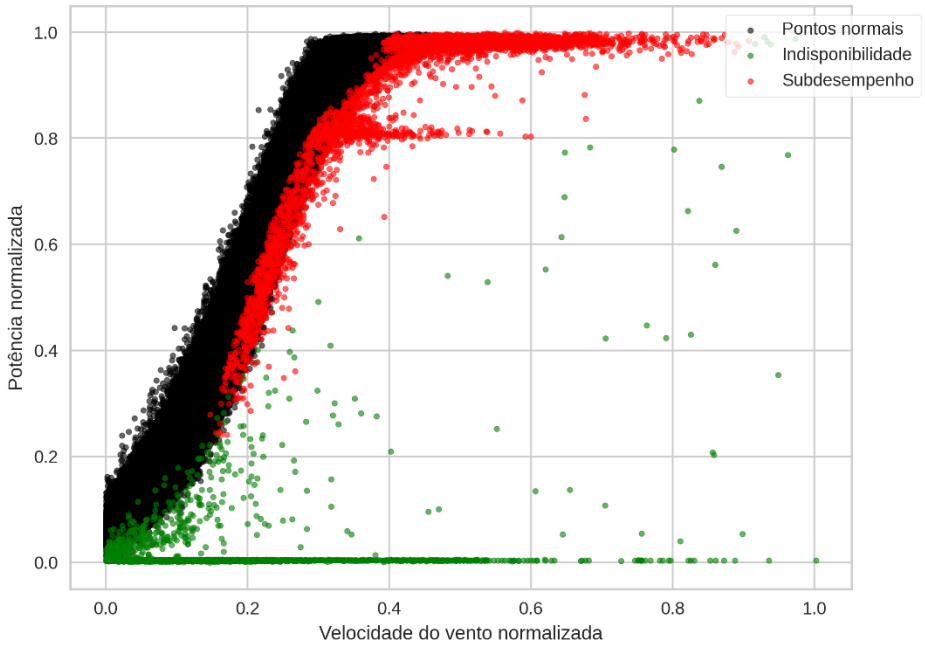
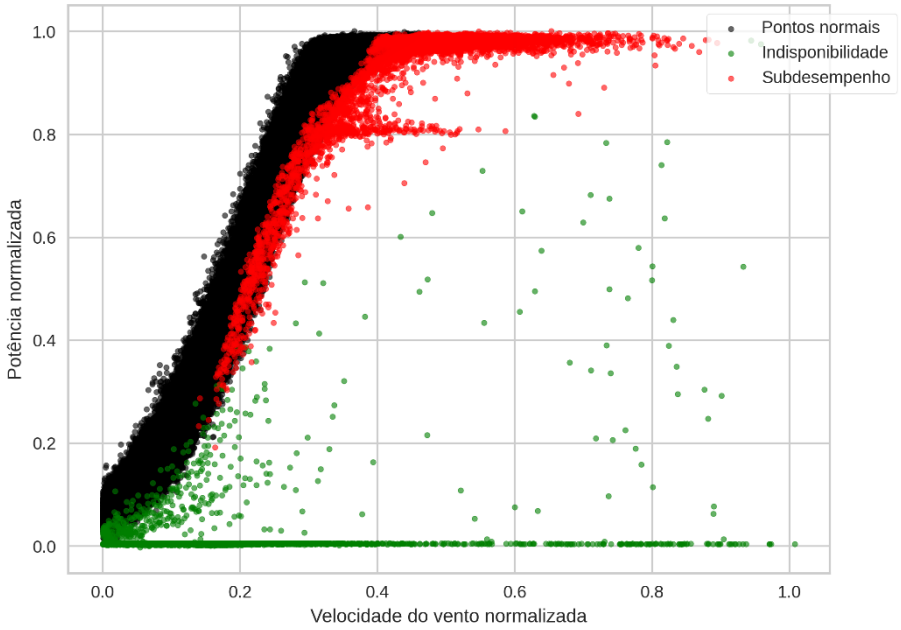


**Figura 5-18 Curva de potência limpa, com pontos classificados em normais, indisponíveis e subdesempenho para a turbina K04 - modelo AE-KAN.**




Apesar dos resultados do *autoencoder* padrão combinados com a rede KAN terem sido satisfatórios, uma nova metodologia é testada, no intuito de melhorar o desempenho da identificação da classe 2. Por este motivo, um *autoencoder* variacional é utilizado no lugar do *autoencoder* padrão e novos testes são feitos. Os resultados são apresentados a seguir.

## 5.5 *AUTOENCODER* VARIACIONAL COM KAN (VAE-KAN)

Conforme descrito na seção 4.3.2.4, um *autoencoder* variacional é testado. Combina-se o *autoencoder* variacional (VAE) com a KAN. A Tabela 5-9 apresenta os valores utilizados no ajuste de hiperparâmetros do *autoencoder* e o respectivo erro de reconstrução de cada rodada.

**Tabela 5-9 Hiperparâmetros do *autoencoder* variacional e o erro de reconstrução.**

| Rodada | Tamanho do espaço latente | Épocas | Tamanho das camadas ocultas | Otimizador | Erro de reconstrução |
|---|---|---|---|---|---|
| 1 | 3 | 20 | 8 | Adam | 0,109 |
| 2 | 3 | 20 | 8 | RMSProp | 0,108 |
| 3 | 3 | 20 | 16 | Adam | 0,109 |
| 4 | 3 | 20 | 16 | RMSProp | 0,111 |
| 5 | 3 | 30 | 8 | Adam | 0,111 |
| 6 | 3 | 30 | 8 | RMSProp | 0,164 |
| 7 | 3 | 30 | 16 | Adam | 0,106 |
| 8 | 3 | 30 | 16 | RMSProp | 0,107 |

| | | | | | |
|---|---|---|---|---|---|
| 9 | 4 | 20 | 8 | Adam | 0,106 |
| 10 | 4 | 20 | 8 | RMSProp | 0,165 |
| 11 | 4 | 20 | 16 | Adam | 0,109 |
| 12 | 4 | 20 | 16 | RMSProp | 0,107 |
| 13 | 4 | 30 | 8 | Adam | 0,106 |
| 14 | 4 | 30 | 8 | RMSProp | 0,108 |
| 15 | 4 | 30 | 16 | Adam | 0,165 |
| 16 | 4 | 30 | 16 | RMSProp | 0,111 |

Como os valores dos erros estão ainda mais próximos do que para o *autoencoder* clássico, optou-se por reportar três casas decimais neste caso, para que as rodadas pudessem ser mais bem discernidas. As curvas de potência com os pontos originais e reconstruídos são mostradas no Apêndice B. A rodada de número 13 foi a que apresentou o menor erro de reconstrução, sendo, portanto, escolhida como dado de entrada para a KAN.

De forma similar, gera-se os resultados do ajuste de hiperparâmetros da KAN com o *autoencoder* variacional. Os mesmos parâmetros utilizados anteriormente, em cada rodada, conforme mostrado na Tabela 4-5, se mantêm. Para avaliação, as métricas de acurácia global, AUC ROC, precisão, *recall* e *f1-score* são calculadas. Os resultados são apresentados na Tabela 5-10.

**Tabela 5-10 Acurácia, AUC-ROC, precisão, *recall*, *F1-score* para cada uma das classes durante o ajuste de hiperparâmetros do modelo VAE-KAN.**

| Rodada | Acurácia | AUC ROC | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0,99 | 1,00 | 1,00 | 0,87 | 0,04 | 0,95 | 0,98 | 0,94 | 0,98 | 0,92 | 0,08 |
| 2 | 0,98 | 1,00 | 1,00 | 0,86 | 0,04 | 0,95 | 0,98 | 0,94 | 0,98 | 0,92 | 0,08 |
| 3 | 0,99 | 1,00 | 1,00 | 0,86 | 0,07 | 0,97 | 0,98 | 0,97 | 0,99 | 0,92 | 0,13 |
| 4 | 0,99 | 1,00 | 1,00 | 0,86 | 0,05 | 0,96 | 0,98 | 0,96 | 0,98 | 0,92 | 0,10 |
| 5 | 0,99 | 1,00 | 1,00 | 0,87 | 0,04 | 0,95 | 0,98 | 0,94 | 0,98 | 0,92 | 0,08 |
| 6 | 0,97 | 0,99 | 1,00 | 0,86 | 0,06 | 0,97 | 0,98 | 0,96 | 0,98 | 0,92 | 0,11 |
| 7 | 0,98 | 0,99 | 1,00 | 0,87 | 0,04 | 0,95 | 0,98 | 0,95 | 0,98 | 0,92 | 0,08 |
| 8 | 0,98 | 1,00 | 1,00 | 0,86 | 0,07 | 0,97 | 0,98 | 0,97 | 0,99 | 0,91 | 0,14 |
| 9 | 0,98 | 1,00 | 1,00 | 0,86 | 0,07 | 0,97 | 0,98 | 0,95 | 0,99 | 0,92 | 0,12 |
| 10 | 0,99 | 1,00 | 1,00 | 0,86 | 0,07 | 0,97 | 0,98 | 0,97 | 0,99 | 0,92 | 0,13 |
| 11 | 0,99 | 1,00 | 1,00 | 0,85 | 0,08 | 0,97 | 0,98 | 0,96 | 0,99 | 0,91 | 0,14 |
| 12 | 0,99 | 1,00 | 1,00 | 0,87 | 0,05 | 0,96 | 0,98 | 0,95 | 0,98 | 0,92 | 0,09 |
| 13 | 0,98 | 1,00 | 1,00 | 0,87 | 0,09 | 0,98 | 0,98 | 0,97 | 0,99 | 0,92 | 0,17 |
| 14 | 0,98 | 1,00 | 1,00 | 0,86 | 0,12 | 0,98 | 0,98 | 0,97 | 0,99 | 0,92 | 0,21 |
| 15 | 0,99 | 1,00 | 1,00 | 0,85 | 0,12 | 0,98 | 0,98 | 0,97 | 0,99 | 0,91 | 0,21 |
| 16 | 0,99 | 1,00 | 1,00 | 0,86 | 0,06 | 0,97 | 0,98 | 0,95 | 0,98 | 0,91 | 0,11 |

Tanto o modelo AE-KAN quanto o VAE-KAN demonstraram sucesso na diferenciação entre pontos normais, indisponibilidade e subdesempenho, superando os modelos de agrupamento de dados testados. Comparativamente, o VAE-KAN apresentou R-2 mais alto para a turbina K02, enquanto o AE-KAN obteve maior acurácia e P-2. As métricas para as classes 0 e 1 foram, de modo geral, semelhantes entre os dois modelos.

Embora a classe 2 apresente baixa precisão e, consequentemente, um *F1-score* reduzido, o VAE-KAN se destaca pelo maior *recall* nessa classe. Esse fator o torna mais eficiente na identificação de verdadeiros positivos da classe 2, caracterizando-o como o modelo de melhor desempenho nesse critério.

Para teste, utiliza-se o modelo com os hiperparâmetros otimizados com uma segunda turbina. Aplica-se, pois, o modelo treinado nas turbinas K02, K03 e K04. A Tabela 5-11, Tabela 5-12 e Tabela 5-15 apresentam os resultados dos testes.

**Tabela 5-11 Acurácia, AUC ROC, precisão, *recall* e *F1-score* do teste com a turbina K02 – modelo VAE-KAN.**

| Acurácia | AUC ROC | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,96 | 0,99 | 1,00 | 0,87 | 0,05 | 0,96 | 0,98 | 0,95 | 0,98 | 0,92 | 0,10 |

**Tabela 5-12 Acurácia, AUC ROC, precisão, *recall* e *F1-score* do teste com a turbina K03 – modelo VAE-KAN.**

| Acurácia | AUC ROC | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,92 | 0,99 | 1,00 | 0,86 | 0,02 | 0,92 | 0,99 | 0,92 | 0,96 | 0,92 | 0,03 |

**Tabela 5-13 Acurácia, AUC ROC, precisão, *recall* e *F1-score* do teste com a turbina K04 – modelo VAE-KAN.**

| Acurácia | AUC ROC | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,91 | 0,99 | 1,00 | 0,87 | 0,02 | 0,91 | 0,99 | 0,91 | 0,95 | 0,93 | 0,04 |

Conforme observado durante o treinamento, o modelo demonstrou excelente desempenho na identificação das classes 0 e 1, comportamento que se manteve no teste. Além disso, o VAE-KAN mostrou-se eficaz na identificação de verdadeiros positivos da classe 2. No entanto, sua principal limitação está na precisão dessa classe, uma vez que classifica erroneamente muitos pontos da classe normal como

pertencentes à classe 2. Esse comportamento pode ser visualizado na Figura 5-19, Figura 5-20 e Figura 5-21, que apresentam as matrizes de confusão das turbinas K02, K03 e K04, respectivamente.

**Figura 5-19 Matriz de confusão do modelo VAE-KAN testado na turbina K02.**

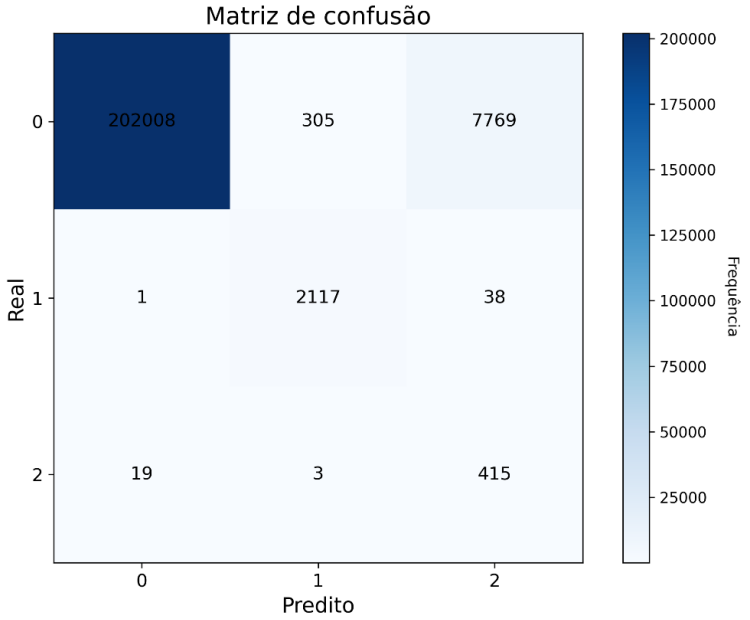



**Figura 5-20 Matriz de confusão do modelo VAE-KAN testado na turbina K03.**

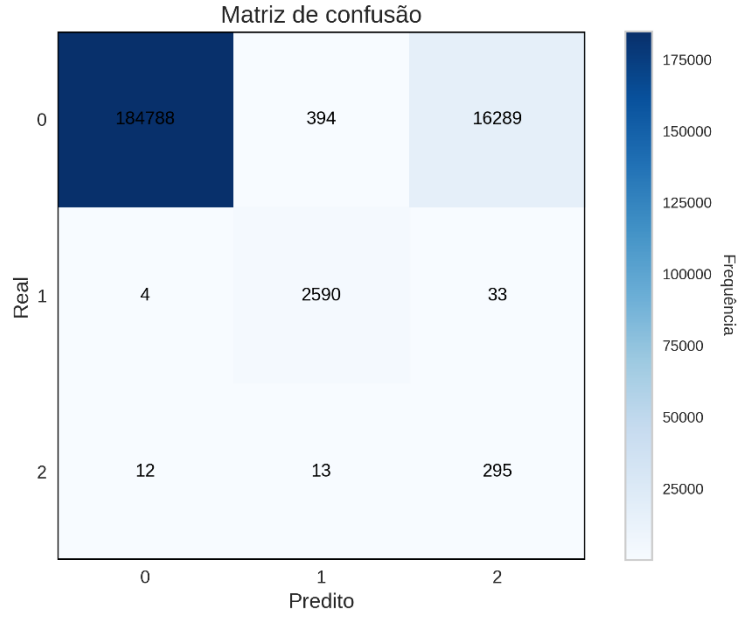
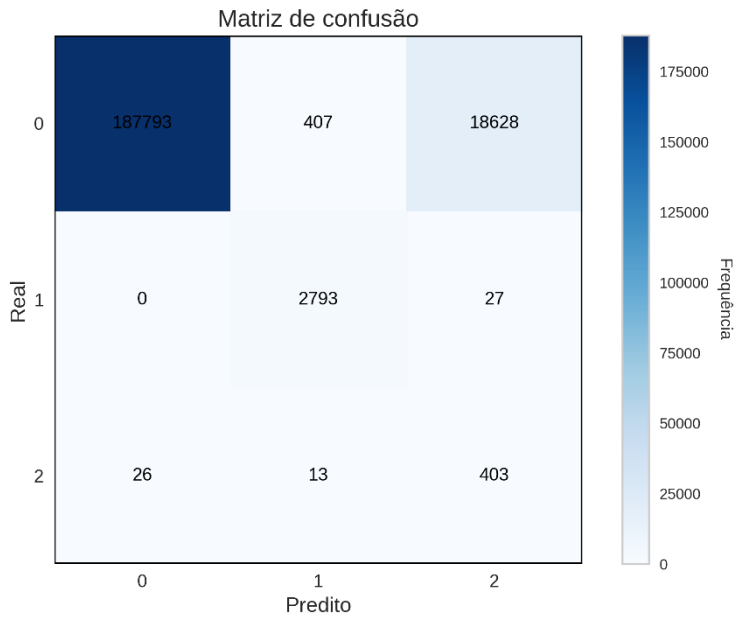


**Figura 5-21 Matriz de confusão do modelo VAE-KAN testado na turbina K04.**




A Figura 5-22, Figura 5-23 e Figura 5-24 apresentam o gráfico da área sob a curva ROC para as três classes, em cada turbina testada.

**Figura 5-22 Área sob a curva ROC, para cada classe - modelo VAE-KAN testado na turbina K02.**

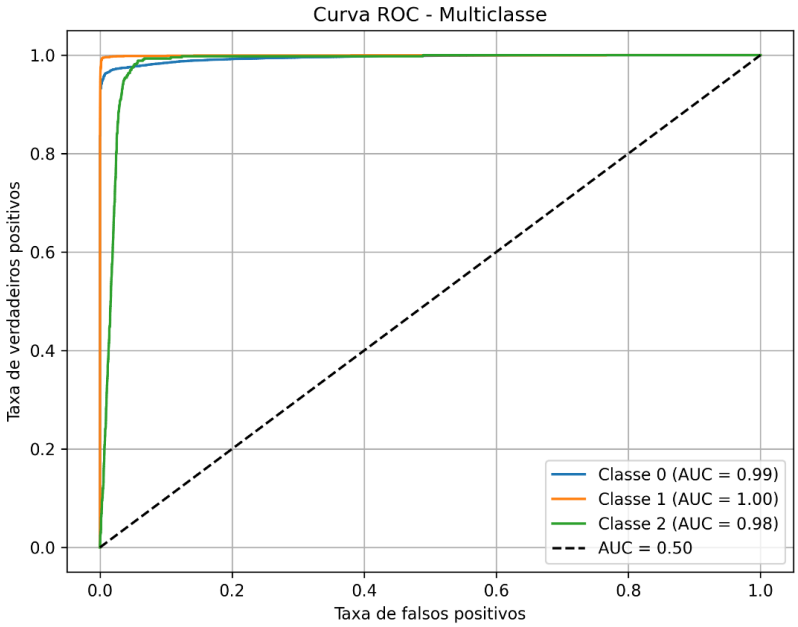


**Figura 5-23 Área sob a curva ROC, para cada classe - modelo VAE-KAN testado na turbina K03.**

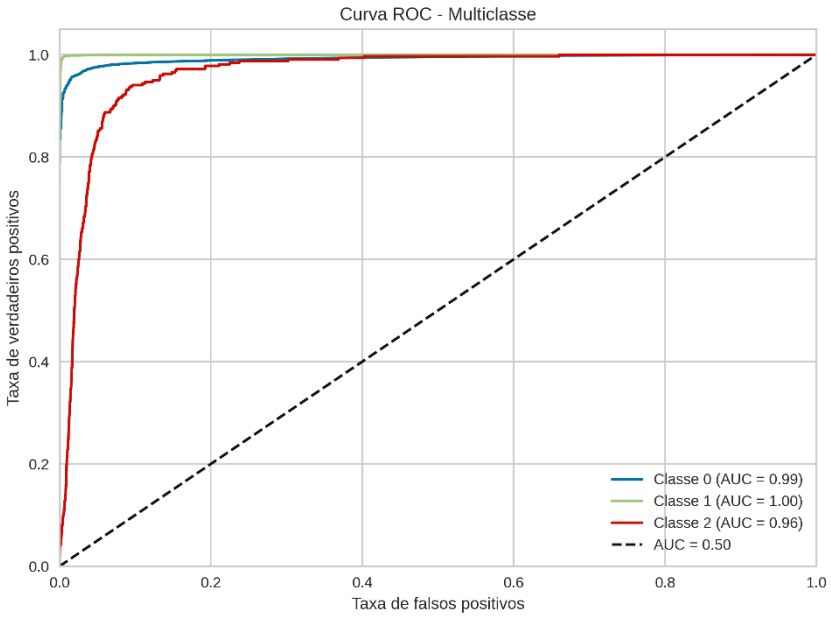
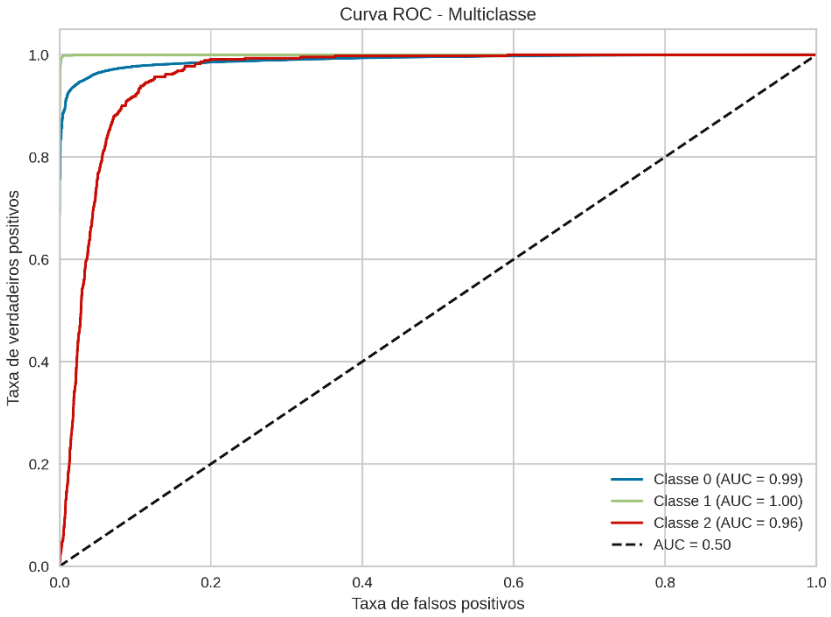



**Figura 5-24 Área sob a curva ROC, para cada classe - modelo VAE-KAN testado na turbina K04.**




A área sob a curva ROC apresentada, para cada turbina, indica que o modelo teve um excelente desempenho na separação das classes, com valores de AUC próximos a 1, reforçando sua eficácia na identificação dos verdadeiros positivos.

A Figura 5-25, Figura 5-26 e Figura 5-27 mostram a curva de potência com a classificação em pontos normais, indisponibilidade e subdesempenho para as turbinas K02, K03 e K04, respectivamente.

**Figura 5-25 Curva de potência normalizada com as respectivas classificações em pontos normais, indisponibilidade e subdesempenho para a turbina K02 – modelo VAE-KAN.**

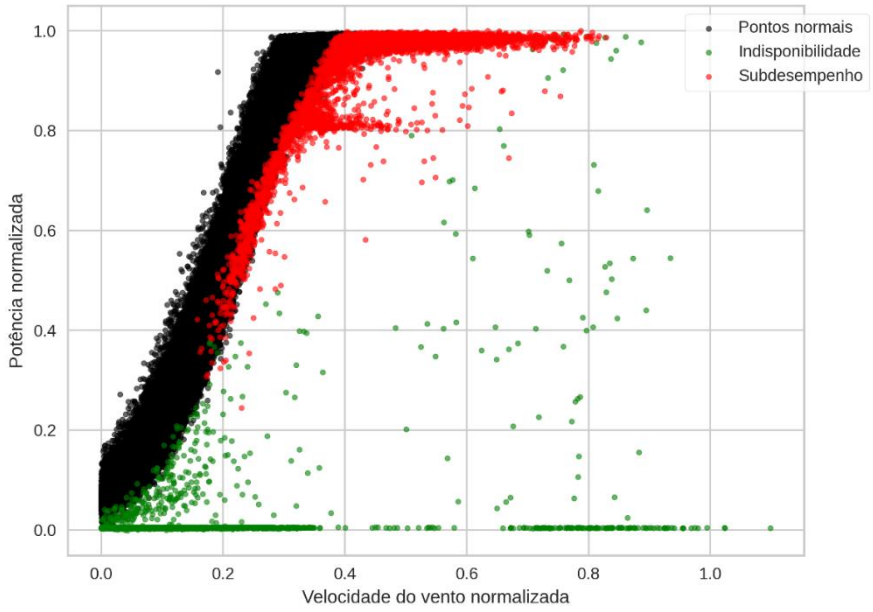



**Figura 5-26 Curva de potência normalizada com as respectivas classificações em pontos normais, indisponibilidade e subdesempenho para a turbina K03 – modelo VAE-KAN.**

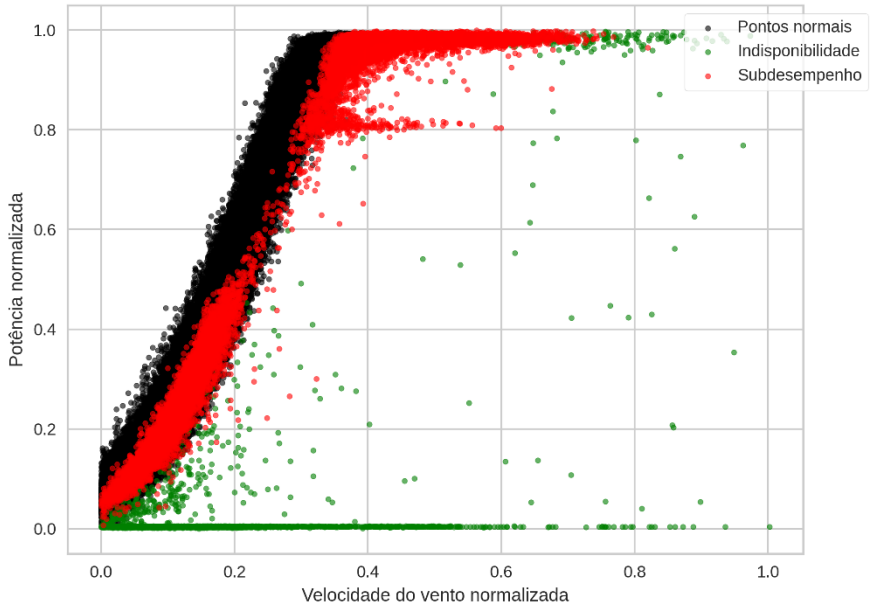
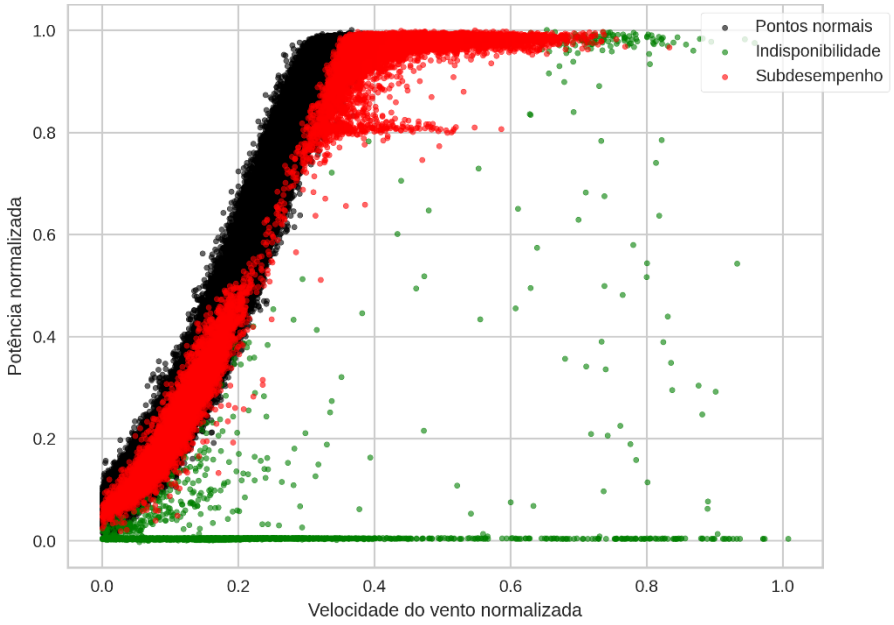


**Figura 5-27 Curva de potência normalizada com as respectivas classificações em pontos normais, indisponibilidade e subdesempenho para a turbina K04 – modelo VAE-KAN.**




A Figura 5-28 apresenta a limpeza da curva realizada por cada um dos métodos desenvolvidos, em comparação com a referência obtida a partir da limpeza feita pelo especialista.

**Figura 5-28 Classificação de pontos nas curvas de potência, pelo AE-KAN, VAE-KAN e especialista da turbina K02.**

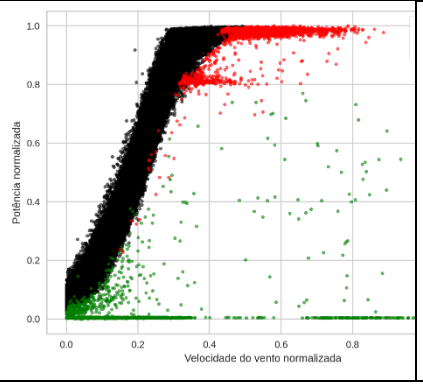
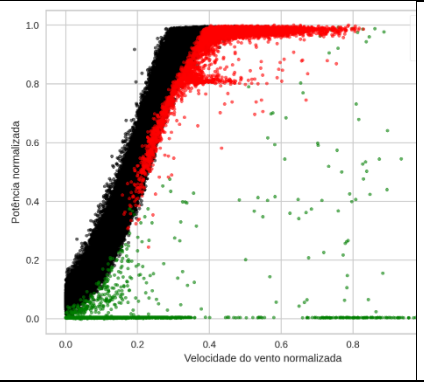
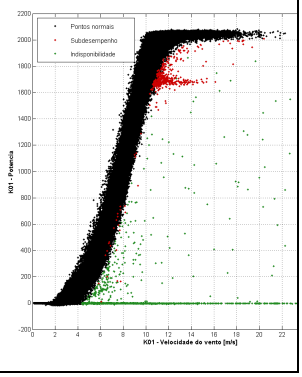
| AE-KAN | VAE-KAN | Especialista |
|---|---|---|
|  |  |  |

**Figura 5-29 Classificação de pontos nas curvas de potência, pelo AE-KAN, VAE-KAN e especialista da turbina K03.**

| AE-KAN | VAE-KAN | Especialista |
|---|---|---|

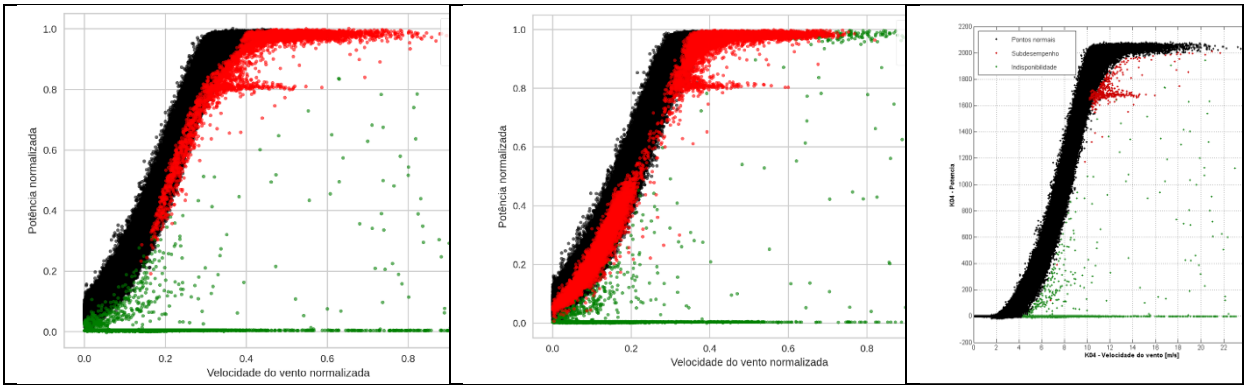
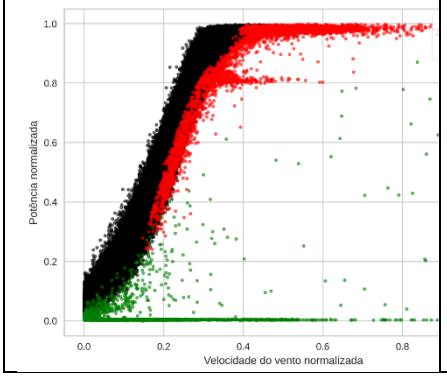
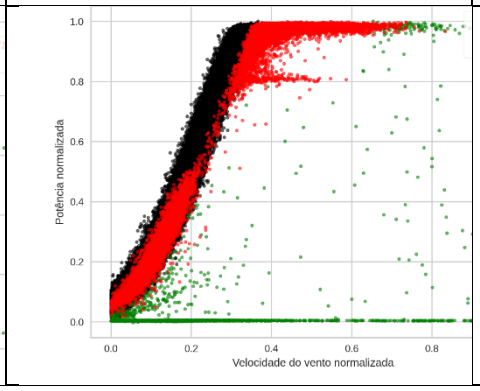
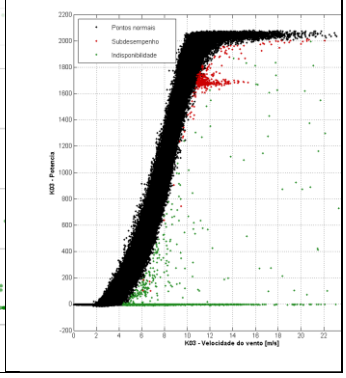



**Figura 5-30 Classificação de pontos nas curvas de potência, pelo AE-KAN, VAE-KAN e especialista da turbina K04.**

| AE-KAN | VAE-KAN | Especialista |
|---|---|---|
|  |  |  |



O desempenho dos modelos na detecção de indisponibilidade (classe 1) foi bastante semelhante, apresentando resultados satisfatórios. No entanto, conforme pode ser observado nas Figuras, o modelo VAE-KAN gerou uma quantidade significativa de falsos positivos, especialmente na região próxima à potência nominal, em todas as turbinas, e na base da curva das turbinas K03 e K04. Para sua aplicação prática, seria necessário incorporar uma etapa de pós-processamento, que limite automaticamente a marcação de pontos de subdesempenho nessas faixas, minimizando assim os impactos desses falsos positivos.

## 5.6 COMPARAÇÃO COM OUTROS ALGORITMOS DE APRENDIZADO DE MÁQUINA

Com o objetivo de avaliar a robustez do modelo desenvolvido neste trabalho e compará-lo a outras abordagens já consolidadas na área de aprendizado de máquina, utilizou-se a biblioteca *PyCaret*, do Python, para a comparação entre os

diferentes algoritmos. Os dados de entrada utilizados na análise correspondem aos utilizados pelos modelos AE-KAN e VAE-KAN, conforme apresentados na Tabela 4-7 e na Tabela 4-8, respectivamente.

### 5.6.1. Modelo AE-KAN

A Tabela 5-14 apresenta as métricas de classificação de cada um dos modelos alternativos testados.

**Tabela 5-14 Métricas para os modelos de classificação testados.**

| Modelo | Acurácia global | AUC ROC | *Recall* | Precisão global | *F1-score* global |
|---|---|---|---|---|---|
| Random Forest | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| XGBoost | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| Extra trees | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| Árvore de decisão | 1,00 | 0,98 | 1,00 | 1,00 | 1,00 |
| KNN | 1,00 | 0,99 | 1,00 | 1,00 | 1,00 |
| Gradient Boosting | 1,00 | 0,00 | 1,00 | 1,00 | 1,00 |
| SVM | 1,00 | 0,00 | 1,00 | 0,99 | 1,00 |
| Regressão logística | 1,00 | 0,00 | 1,00 | 0,99 | 1,00 |
| Ridge Classifier | 1,00 | 0,00 | 1,00 | 0,99 | 0,99 |
| Análise discriminante | 0,99 | 0,00 | 0,99 | 1,00 | 1,00 |
| Light Gradient Boosting Machine | 0,99 | 0,93 | 0,99 | 1,00 | 1,00 |
| Classificador de referência | 0,99 | 0,50 | 0,99 | 0,97 | 0,98 |
| Classificador Ada Boost | 0,94 | 0,00 | 0,94 | 0,99 | 0,97 |
| Análise discriminante quadrática | 0,89 | 0,00 | 0,89 | 1,00 | 0,94 |
| Naive Bayes | 0,82 | 0,96 | 0,82 | 0,99 | 0,90 |

Uma parte dos modelos apresentou bom desempenho geral. Para uma avaliação mais detalhada da capacidade de separação entre as classes, foram selecionados os modelos Random Forest, XGBoost, Extra Trees, Árvore de Decisão e KNN, visando uma análise mais aprofundada. Os resultados de precisão, acurácia e *F1-score*, por classe, são apresentados na Tabela 5-15. Na última linha foram acrescentados os resultados do modelo AE-KAN para fins de comparação.

**Tabela 5-15 Precisão, *recall* e *F1-score*, por classe, para cada modelo alternativo testado.**

| Modelo | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|
| Random Forest | 1,00 | 0,99 | 0,94 | 1,00 | 0,97 | 0,34 | 1,00 | 0,98 | 0,49 |
| XGBoost | 1,00 | 1,00 | 0,96 | 1,00 | 0,83 | 0,36 | 1,00 | 0,91 | 0,52 |
| Extra trees | 1,00 | 0,98 | 0,94 | 1,00 | 0,97 | 0,34 | 1,00 | 0,98 | 0,50 |
| Árvore de decisão | 1,00 | 0,98 | 0,91 | 1,00 | 0,97 | 0,38 | 1,00 | 0,98 | 0,53 |
| KNN | 1,00 | 0,98 | 0,90 | 1,00 | 0,96 | 0,34 | 1,00 | 0,97 | 0,49 |
| AE-KAN | 1,00 | 0,85 | 0,09 | 0,98 | 0,99 | 0,69 | 0,99 | 0,91 | 0,17 |

Em termos de precisão e *F1-score*, os modelos de classificação alternativos testados apresentaram desempenho superior ao AE-KAN. No entanto, o AE-KAN destacou-se na identificação de verdadeiros positivos das classes 1 e 2, o que é relevante, por aumentar a sensibilidade do modelo na detecção de anomalias.

### 5.6.2. Modelo VAE-KAN

A Tabela 5-16 mostra os resultados das métricas de classificação dos modelos avaliados para fins de comparação com o VAE-KAN.

**Tabela 5-16 Métricas para os modelos de classificação testados.**

| Modelo | Acurácia global | AUC ROC | *Recall* | Precisão global | *F1-score* global |
|---|---|---|---|---|---|
| XGBoost | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| Árvore de decisão | 1,00 | 0,98 | 1,00 | 1,00 | 1,00 |
| Random Forest | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| Gradient Boosting | 1,00 | 0,00 | 1,00 | 1,00 | 1,00 |
| Extra Trees | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| SVM | 1,00 | 0,00 | 1,00 | 0,99 | 1,00 |
| Regressão logística | 1,00 | 0,00 | 1,00 | 0,99 | 1,00 |
| Ridge Classifier | 1,00 | 0,00 | 1,00 | 0,99 | 0,99 |
| KNN | 0,99 | 0,88 | 0,99 | 0,99 | 0,99 |
| Dummy Classifier | 0,99 | 0,50 | 0,99 | 0,97 | 0,98 |
| Análise discriminante | 0,98 | 0,00 | 0,98 | 0,99 | 0,99 |
| Light Gradient Boosting Machine | 0,98 | 0,84 | 0,98 | 0,99 | 0,98 |
| Análise discriminante quadrática | 0,96 | 0,00 | 0,96 | 1,00 | 0,98 |
| Classificador Ada Boost | 0,89 | 0,00 | 0,89 | 0,99 | 0,94 |
| Naive Bayes | 0,80 | 0,96 | 0,80 | 1,00 | 0,89 |

Assim como no caso do AE-KAN, uma parte dos modelos é bem-sucedida, com métricas que chegam a 1 ou muito próximo disso. Para uma análise mais detalhada, os mesmos cinco modelos são selecionados e comparados com o VAE-KAN. Os resultados são mostrados na Tabela 5-17.

**Tabela 5-17 Precisão, *recall* e *F1-score*, por classe, para cada modelo alternativo testado.**

| Modelo | P-0 | P-1 | P-2 | R-0 | R-1 | R-2 | F1-0 | F1-1 | F1-2 |
|---|---|---|---|---|---|---|---|---|---|
| XGBoost | 1,00 | 0,99 | 0,93 | 1,00 | 0,99 | 0,71 | 1,00 | 0,99 | 0,80 |
| Árvore de decisão | 1,00 | 0,98 | 0,83 | 1,00 | 0,98 | 0,74 | 1,00 | 0,98 | 0,78 |
| Random Forest | 1,00 | 0,99 | 0,95 | 1,00 | 0,97 | 0,62 | 1,00 | 0,98 | 0,75 |
| Extra trees | 1,00 | 0,95 | 0,96 | 1,00 | 0,97 | 0,38 | 1,00 | 0,96 | 0,54 |
| KNN | 1,00 | 0,92 | 0,00 | 1,00 | 0,74 | 0,00 | 1,00 | 0,82 | 0,00 |
| VAE-KAN | 1,00 | 0,87 | 0,05 | 0,96 | 0,98 | 0,95 | 0,98 | 0,92 | 0,10 |

Análogo ao que acontece com o AE-KAN, os modelos testados também possuem melhor performance na precisão e no *F1-score*, mas desempenho inferior no *recall*, especialmente da classe 2.

# 6 CONCLUSÕES

Este trabalho apresenta uma nova abordagem para a limpeza automática de curvas de potência de turbinas eólicas. Foram desenvolvidas e validadas metodologias híbridas que automatizam a limpeza de curvas de potência, que não apenas identifica e remove anomalias, mas também diferencia entre tipos distintos de eventos anômalos, especificamente indisponibilidade e subdesempenho, com aplicação inédita da rede Kolmogorov-Arnold (KAN) nesse contexto.

Ao longo do desenvolvimento, foram testados algoritmos de agrupamento de dados que, embora eficazes na detecção de anomalias, não apresentaram um desempenho satisfatório na separação entre as classes. Para aprimorar a metodologia, propôs-se a combinação de *autoencoders* com redes neurais Kolmogorov-Arnold, resultando em uma abordagem mais robusta para a limpeza da curva de potência.

Inicialmente, foi testada a combinação entre um *autoencoder* clássico e KAN. O modelo demonstrou excelente desempenho na classificação das classes 0 e 1, apresentando alta precisão e *recall*, além de um desempenho razoável na identificação dos verdadeiros positivos da classe 2. No entanto, a metodologia revelou limitações na precisão da classe 2, uma vez que gerou um número significativo de falsos positivos.

Com o objetivo de aprimorar a separação entre as classes, foi desenvolvido um método que combina um *autoencoder* variacional (VAE) com KAN, aproveitando a saída da camada latente como informação adicional. Comparativamente, o modelo VAE-KAN mostrou-se superior ao AE-KAN na identificação de verdadeiros positivos da classe 2. No entanto, a precisão da classe 2 permaneceu insatisfatória, apresentando valores ainda inferiores aos observados no modelo AE-KAN.

Para fins de comparação com modelos já estabelecidos na literatura, utilizaram-se os mesmos conjuntos de treino e teste para treinar diferentes classificadores. Os modelos tradicionais apresentaram melhores métricas de *F1-score*, indicando maior precisão, mas foram menos eficazes na identificação de verdadeiros positivos, especialmente da classe 2.

Para os resultados de classificação, utilizou-se como referência, uma limpeza conduzida por um especialista no setor, utilizando-se de uma ferramenta validada e usada na indústria.

Pode-se concluir que os algoritmos desenvolvidos apresentaram bom desempenho, cumprindo o objetivo de automatizar a limpeza de curvas de potência. Sua implementação pode ser uma alternativa eficiente para reduzir o esforço manual de engenheiros responsáveis por essa tarefa. Como limitação, destaca-se a alta taxa de falsos positivos, o que torna necessário um pós-processamento. Uma estratégia simples e automatizável seria restringir a marcação de pontos de subdesempenho à região próxima da potência nominal. Além disso, por se tratar de um modelo supervisionado, é necessário que ao menos uma turbina seja previamente limpa manualmente para servir como base de treinamento. Idealmente, essa turbina deve apresentar a maior diversidade possível de falhas, permitindo que o modelo aprenda a reconhecer diferentes padrões de anomalia.

Como recomendações para trabalhos futuros, recomenda-se a exploração de uma gama mais ampla de hiperparâmetros durante o treinamento e a exploração de abordagens para preenchimento de lacunas nos dados SCADA, como o uso dos dados de energia da subestação e o cálculo da eficiência elétrica para estimativa da potência em períodos ausentes. Outra possibilidade é a síntese de dados de velocidade do vento a partir de dados de reanálise, permitindo reconstituir a curva de potência em cenários com falhas prolongadas nos sensores de vento. Por fim, recomenda-se que os treinamentos sejam realizados por faixa (bin) de potência, favorecendo uma segmentação mais precisa e adaptada às diferentes regiões operacionais da turbina.

# 7 REFERÊNCIAS

ABEEÓLICA. **Infovento**. *[S.l.: S.n.]*. Disponível em: <http://www.ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/recordes.aspx>.

ADARAMOLA, Muyiwa. **Wind Turbine Technology: Principles and Design**. *[S.l.]*: Apple Academic Press, 2014.

AGGARWAL, Sakshi. Research on Anomaly Detection in Time Series: Exploring United States Exports and Imports Using Long Short-Term Memory. **Journal of Research, Innovation and Technologies (JoRIT)**, v. 2, n. 16, p. 199, nov. 2023.

**Alba energia**. Disponível em: <https://albaenergia.com.br/cop-28-maior-evento-internacional-sobre-mudancas-climaticas/>. Acesso em: 21 mar. 2024.

ASTOLFI, Davide; DE CARO, Fabrizio; VACCARO, Alfredo. Condition Monitoring of Wind Turbine Systems by Explainable Artificial Intelligence Techniques. **Sensors**, v. 23, n. 12, 1 jun. 2023.

BENNAGI, Aseel *et al.* Comprehensive study of the artificial intelligence applied in renewable energy. **Energy Strategy Reviews**, v. 54, 1 jun. 2024.

BILENDO, Francisco *et al.* Applications and Modeling Techniques of Wind Turbine Power Curve for Wind Farms—A Review. **Energies**, v. 16, n. 1, 24 dez. 2022.

BLEEG, James *et al.* Wind farm blockage and the consequences of neglecting its impact on energy production. **Energies**, v. 11, n. 6, 1 jun. 2018.

BORGES, Bráulio. Estimativas dos impactos dinâmicos do setor eólico sobre a economia brasileira. fev. 2022.

BURTON, Tony *et al.* **Wind energy handbook**. 2ª ed. West Sussex: John Wiley & Sons Ltd, 2011.

CAMBRON, P. *et al.* Power curve monitoring using weighted moving average control charts. **Renewable Energy**, v. 94, p. 126–135, 1 ago. 2016.

CAMPELLO DE SOUZA, Camilla; RIBEIRO, Paulo. ANÁLISE MODAL EM AEROGERADORES OFFSHORE CONSIDERANDO A INTERAÇÃO SOLO-ESTRUTURA. *In*: Florianópolis: nov. 2017.

CARVALHO, P. C. M. **Geração eólica**. Fortaleza: Imprensa universitária, 2003.

CHAN, Kit Yan *et al.* Deep neural networks in the cloud: Review, applications, challenges and research directions. **Neurocomputing**, v. 545, 7 ago. 2023.

CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. **Anomaly Detection : A SurveyACM Computing Surveys**. *[S.l.: S.n.]*.

**Classification: ROC and AUC | Machine Learning | Google for Developers**. Disponível em: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>. Acesso em: 6 jan. 2025.

CRAVO, Edilson. **SCADA: o que é, importância, como funciona e mais | Blog Kalatec**. Disponível em: <https://blog.kalatec.com.br/scada/>. Acesso em: 18 abr. 2025.

DONG, Mi *et al.* Real-time detection of wind power abnormal data based on semi-supervised learning Robust Random Cut Forest. **Energy**, v. 257, 15 out. 2022.

EIXOS. **Entenda os problemas das turbinas eólicas da Siemens Gamesa no Brasil | eixos**. Disponível em: <https://eixos.com.br/empresas/entenda-os-problemas-das-turbinas-eolicas-da-siemens-gamesa-no-brasil/>. Acesso em: 19 jul. 2024.

ENERCON. **Turbina eólica de eixo horizontal - E-70 - Enercon - de três pás / on-shore**. Disponível em: <https://www.archiexpo.com/pt/prod/enercon/product-88093-969034.html>. Acesso em: 24 jun. 2024.

**Energia renovável - hidráulica, biomassa, eólica, solar, oceânica**. . Rio de Janeiro: *[S.n.]*.

ERSOY, Pınar; ERŞAHIN, Mustafa; KILINÇ, Deniz. Evolution of Outlier Algorithms for Anomaly Detection. **Manchester Journal of Artificial Intelligence Applied Sciences**, v. 02, p. 51–57, 2021.

FADIGAS, Eliana. **Energia eólica**. Barueri: Manole, 2011.

GALLI, Sole. **Overcoming Class Imbalance with SMOTE: How to Tackle Imbalanced Datasets in Machine Learning - Train in Data's Blog**. Disponível em: <https://www.blog.trainindata.com/overcoming-class-imbalance-with-smote/>. Acesso em: 21 mar. 2025.

GAO, Yuan *et al.* A revolutionary neural network architecture with interpretability and flexibility based on Kolmogorov–Arnold for solar radiation and temperature forecasting. **Applied Energy**, v. 378, 15 jan. 2025.

GAO, Zibo; KONG, Ming. MP-KAN: An effective magnetic positioning algorithm based on Kolmogorov-Arnold network. **Measurement: Journal of the International Measurement Confederation**, v. 243, 15 fev. 2025.

GASCH, Robert; TWELE, Jochen. **Wind Power Plants**. 2ª ed. Berlin: Springer-Verlag, 2012.

GIROSI, Federico; POGGIO, Tomaso. Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant. **Neural Computation**, v. 1, n. 4, p. 465–469, dez. 1989.

GOMES, Raphael *et al.* **Marco Legal das Eólicas Offshore é sancionado**. Disponível em: <https://lefosse.com/noticias/marco-legal-das-eolicas-offshore-e-sancionado-com-vetos/>. Acesso em: 31 jan. 2025.

GONZALEZ, E. *et al.* On the use of high-frequency SCADA data for improved wind turbine performance monitoring. *In*: Institute of Physics Publishing, 23 nov. 2017.

GRAVES, Alex; MOHAMED, Abdel-Rahman; HINTON, Geoffrey. **Speech recognition with deep recurrent neural networks**. *[S.l.: S.n.]*.

GÜNTHER, Frauke; FRITSCH, Stefan. **neuralnet: Training of Neural Networks**. *[S.l.: S.n.]*.

**GWEC**. . Brussels: *[S.n.]*. Disponível em: <https://26973329.fs1.hubspotusercontent-eu1.net/hubfs/26973329/2.%20Reports/Global%20Wind%20Report/GWR23.pdf?__hstc=45859835.d9c0f6c97d4c6aef5fa2cf38c0c1c2d4.1742599781008.1742599781008.1742599781008.1&__hssc=45859835.8.1742599781008&__hsfp=2797634066>. Acesso em: 19 jun. 2024a.

**GWEC**. . *[S.l.: S.n.]*. Disponível em: <www.gwec.net>.

HAU, Eric. **Wind Turbines: Fundamentals, Technologies, Application, Economics**. 3ª ed. Berlin: Springer-Verlag, 2013.

HAYKIN, Simon. **Neural Networks**. 2ª ed. *[S.l.]*: Pearson Education, 1994.

HEINZELMANN, Barbara. A evolução mecânica das turbinas eólicas comerciais. *In*: CRAVEIRO, Paula; FERREIRA, Silva (Orgs.). **Energia eólica - princípios e operação** . 1ª ed. São Paulo: Erica, 2019. p. 152–168.

HELBING, Georg; RITTER, Matthias. Deep Learning for fault detection in wind turbines. **Renewable and Sustainable Energy Reviews**, v. 98, p. 189–198, 1 dez. 2018.

HINE. **Sistemas hidráulicos integrados, subconjuntos hidráulicos e sistemas de refrigeração para turbinas eólicas**. Disponível em: <https://www.hinegroup.com/pt/energia-eolica/>. Acesso em: 24 jun. 2024.

HORNIK, Kurt; STINCHCOME, Maxwell; WHITE, Halbert. Multilyer Feedforward Networks are Universal Approximators. **Neural Networks**, v. 2, p. 359–366, 1989.

HOSSAIN, A. S. M. Customer Segmentation using Centroid Based and Density Based Clustering Algorithms. *In*: IEEE, dez. 2017.

HUDSON, Mark *et al.* **Neural Network Toolbox™ Reference**. *[S.l.: S.n.]*. Disponível em: <www.mathworks.com>.

**IEC 61400-12-1 - Wind energy generation systems - Part 12-1: Power performance measurements of electricity producing wind turbines**. . *[S.l.: S.n.]*.

JAVADI, Milad *et al.* An algorithm for practical power curve estimation of wind turbines. **CSEE Journal of Power and Energy Systems**, v. 4, n. 1, p. 93–102, 15 mar. 2018.

KARAKASIS, Nektarios *et al.* Active yaw control in a horizontal axis wind system without requiring wind direction measurement. **IET Renewable Power Generation**, v. 10, n. 9, p. 1441–1449, 2016.

KHAN, Prince Waqas; YEUN, Chan Yeob; BYUN, Yung Cheol. Fault detection of wind turbines using SCADA data and genetic algorithm-based ensemble learning. **Engineering Failure Analysis**, v. 148, 1 jun. 2023.

KINGMA, Diederik P.; WELLING, Max. Auto-Encoding Variational Bayes. 20 dez. 2013.

KOLMOGOROV, Andrei. On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables as superpositions of continuous functions of one variale and addition. **Dokl. Akad. Nauk**, p. 953–956, 1957.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. **ImageNet Classification with Deep Convolutional Neural Networks**. *[S.l.: S.n.]*. Disponível em: <http://code.google.com/p/cuda-convnet/>.

KUMAR, Rajesh. **A Guide to the DBSCAN Clustering Algorithm**. Disponível em: <https://www.datacamp.com/tutorial/dbscan-clustering-algorithm>. Acesso em: 22 mar. 2025.

KUSIAK, Andrew. Share data on wind energy. **Nature**, v. 529, p. 19–21, 7 jan. 2016.

KUSIAK, Andrew; VERMA, Anoop. Monitoring wind farms with performance curves. **IEEE Transactions on Sustainable Energy**, v. 4, n. 1, p. 192–199, 2013.

KUSUMA, Yudiawan Fajar *et al.* Navigating challenges on the path to net zero emissions: A comprehensive review of wind turbine technology for implementation in Indonesia. **Results in Engineering**, v. 22, 1 jun. 2024.

KWON, Donghwoon *et al.* A survey of deep learning-based network anomaly detection. **Cluster Computing**, v. 22, p. 949–961, 16 jan. 2019.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, 27 maio 2015.

LETZGUS, Simon; MULLER, Klaus-Robert. Letzgus & Muller 2024. **Energy and AI**, v. 15, 2024.

LI, Pengzhi; PEI, Yan; LI, Jianqiang. A comprehensive survey on design and application of autoencoder in deep learning. **Applied Soft Computing**, v. 138, 1 maio 2023.

LIBERADO, Eduardo. **Tecnologia de Aerogeradores | Passei Direto**. Disponível em: <https://www.passeidireto.com/arquivo/86332547/tecnologia-de-aerogeradores>. Acesso em: 9 nov. 2024.

LIU, Ziming *et al.* KAN: Kolmogorov-Arnold Networks. *In*: 2024. Disponível em: <https://github.com/KindXiaoming/pykan>

LUO, Zhihong *et al.* Method for Cleaning Abnormal Data of Wind Turbine Power Curve Based on Density Clustering and Boundary Extraction. **IEEE Transactions on Sustainable Energy**, v. 13, n. 2, p. 1147–1159, 2021.

LYDIA, M. *et al.* **A comprehensive review on wind turbine power curve modeling techniques**. **Renewable and Sustainable Energy Reviews**, 2014.

MANOBEL, Bartolomé *et al.* Wind turbine power curve modeling based on Gaussian Processes and Artificial Neural Networks. **Renewable Energy**, v. 125, p. 1015–1020, 1 set. 2018.

MARČIUKAITIS, Mantas *et al.* Non-linear regression model for wind turbine power curve. **Renewable Energy**, v. 113, p. 732–741, 2017.

MARINHO, Flavia. **Turbina eólica offshore Haliade-X, do monstro General Electric, agora é oficialmente a mais poderosa do mundo!** Disponível em: <https://clickpetroleoegas.com.br/turbina-eolica-offshore-haliade-x-do-monstro-general-electric-agora-e-oficialmente-a-mais-poderosa-do-mundo/>. Acesso em: 19 maio. 2024.

MARTI-PUIG, Pere *et al.* Wind turbine prognosis models based on scada data and extreme learning machines. **Applied Sciences (Switzerland)**, v. 11, n. 2, p. 1–20, 2 jan. 2021.

MAY, Allan; MCMILLAN, David; THÖNS, Sebastian. Economic analysis of condition monitoring systems for offshore wind turbine sub-systems. **IET Renewable Power Generation**, v. 9, n. 8, p. 900–907, 1 nov. 2015.

MEGAWIND. **Megawind**. Disponível em: <https://megawindproject.com/en-home/>. Acesso em: 22 jun. 2024.

MEHRJOO, Mehrdad; JAFARI JOZANI, Mohammad; PAWLAK, Miroslaw. Wind turbine power curve modeling for reliable power prediction using monotonic regression. **Renewable Energy**, v. 147, p. 214–222, 1 mar. 2020.

MORRISON, Rory; LIU, Xiaolei; LIN, Zi. Anomaly detection in wind turbine SCADA data for power curve cleaning. **Renewable Energy**, v. 184, p. 473–486, 1 jan. 2022.

MUBARAK, Auwalu Saleh *et al.* Quasi-Newton optimised Kolmogorov-Arnold Networks for wind farm power prediction. **Heliyon**, v. 10, n. 23, 15 dez. 2024.

NAJAFABADI, Maryam M. *et al.* Deep learning applications and challenges in big data analytics. **Journal of Big Data**, v. 2, n. 1, 1 dez. 2015.

NERY, Miguel Antônio Cedraz; BOEIRA, Jorge Luís Ferreira; TOSTA, Eduardo Augusto Rodrigues. **Mapeamento da Cadeia Produtiva da Indústria Eólica no Brasil**. *[S.l.: S.n.]*.

OKULOV, Valery; VAN KUIK, G. A. M. The Betz-Joukowsky limit for the maximum power coefficient of wind turbines. 2009.

PAIK, Chunhyun; CHUNG, Yongjoo; KIM, Young Jin. Power Curve Modeling of Wind Turbines through Clustering-Based Outlier Elimination †. **Applied System Innovation**, v. 6, n. 2, 1 abr. 2023.

PEDREGOSA, Fabian *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PIRES, João. **Energia eólica, energia limpa, sustentavel, Brasil melhor**. Disponível em: <https://comprasustentavel.com.br/energia-eolica.html>. Acesso em: 1 jun. 2024.

QIAO, Yanhui *et al.* A multivariable wind turbine power curve modeling method considering segment control differences and short-time self-dependence. **Renewable Energy**, v. 222, 1 fev. 2024.

ROSENBLATT, F. THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN 1. **Psychological Review**, v. 65, n. 6, p. 19–27, 1958.

SANDER, Jorg *et al.* Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. **Data Mining and Knowledge Discovery**, v. 2, p. 169–194, 1998.

SCHMIDHUBER, Jürgen. **Deep Learning in neural networks: An overview**. **Neural Networks** Elsevier Ltd, 1 jan. 2015.

SEFIDIAN, Amir. **How to determine epsilon and MinPts parameters of DBSCAN clustering**. Disponível em: <https://www.sefidian.com/2022/12/18/how-to-determine-epsilon-and-minpts-parameters-of-dbscan-clustering/>. Acesso em: 4 dez. 2024.

SHEN, Xiaojun; FU, Xuejiao; ZHOU, Chongcheng. A Combined Algorithm for Cleaning Abnormal Data of Wind Turbine Power Curve Based on Change Point

Grouping Algorithm and Quartile Algorithm. **IEEE Transactions on Sustainable Energy**, v. 10, n. 1, p. 46–54, 1 jan. 2019.

STETCO, Adrian *et al.* Machine learning methods for wind turbine condition monitoring: A review. **Renewable Energy**, v. 133, p. 620–635, 1 abr. 2019.

SULAIMAN, Mohd Herwan *et al.* Utilizing the Kolmogorov-Arnold Networks for chiller energy consumption prediction in commercial building. **Journal of Building Engineering**, v. 96, 1 nov. 2024.

TASLIMI-RENANI, Ehsan *et al.* Development of an enhanced parametric model for wind turbine power curve. **Applied Energy**, v. 177, p. 544–552, 1 set. 2016.

TAUTZ-WEINERT, Jannis; WATSON, Simon. Using SCADA data for wind turbine condition monitoring - a review. **IET Renewable Power Generation**, 2016.

TOSHNIWAL, Durga *et al.* Application of clustering algorithms for spatio-temporal analysis of urban traffic data. *In*: Elsevier B.V., 2020.

UC, Berkely. **What Is Machine Learning (ML)?**

UDO, Wisdom; MUHAMMAD, Yar. Data-Driven Predictive Maintenance of Wind Turbine Based on SCADA Data. **IEEE Access**, v. 9, p. 162370–162388, 2021.

**UN Climate Change Conference - United Arab Emirates | UNFCCC**. Disponível em: <https://unfccc.int/cop28>. Acesso em: 29 abr. 2024.

VEERS, Paul *et al.* Grand challenges in the design, manufacture, and operation of future wind turbine systems. **Wind Energy Science**, v. 8, n. 7, p. 1071–1131, 11 jul. 2023.

VILLANUEVA, Daniel; FEIJÓO, Andrés. Comparison of logistic functions for modeling wind turbine power curves. **Electric Power Systems Research**, v. 155, p. 281–288, 1 fev. 2018.

WANG, Yun *et al.* Wind Power Curve Modeling and Wind Power Forecasting With Inconsistent Data. **IEEE Transactions on Sustainable Energy**, v. 10, n. 1, p. 16–25, 1 jan. 2016.

WANG, Yun *et al.* Wind Power Curve Modeling and Wind Power Forecasting With Inconsistent Data. **IEEE Transactions on Sustainable Energy**, v. 10, n. 1, p. 16–25, 1 jan. 2018.

WANG, Yun *et al.* Approaches to wind power curve modeling: A review and discussion. **Renewable and Sustainable Energy Reviews**, v. 116, 1 dez. 2019.

**WeDoWind**. Disponível em: <https://www.wedowind.ch/blog/interim-report-ode-space-aug2023>. Acesso em: 7 ago. 2024.

WORLD BANK GROUP. **The energy to drive Brazil's future comes from the windy seas**. Disponível em: <https://www.worldbank.org/en/news/feature/2020/05/27/energia-eolica-offshore-brasil-esmap>. Acesso em: 22 mar. 2024.

YANG, Wenxian; COURT, Richard; JIANG, Jiesheng. Wind turbine condition monitoring by the approach of SCADA data analysis. **Renewable Energy**, v. 53, p. 365–376, maio 2013.

YAO, Qi *et al.* Power Curve Modeling for Wind Turbine Using Hybrid-driven Outlier Detection Method. **Journal of Modern Power Systems and Clean Energy**, v. 11, n. 4, p. 1115–1125, 1 jul. 2023.

YESILBUDAK, Mehmet. Partitional Clustering-Based Outlier Detection for Power Curve Optimization of Wind Turbines. *In*: IEEE, nov. 2016.

YIN, Wenliang *et al.* Advanced power curve modeling for wind turbines: A multivariable approach with SGBRT and grey wolf optimization. **Energy Conversion and Management**, v. 332, 15 maio 2025.

ZHANG, Chen; HU, Di; YANG, Tao. Anomaly detection and diagnosis for wind turbines using long short-term memory-based stacked denoising autoencoders and XGBoost. **Reliability Engineering and System Safety**, v. 222, 1 jun. 2022.

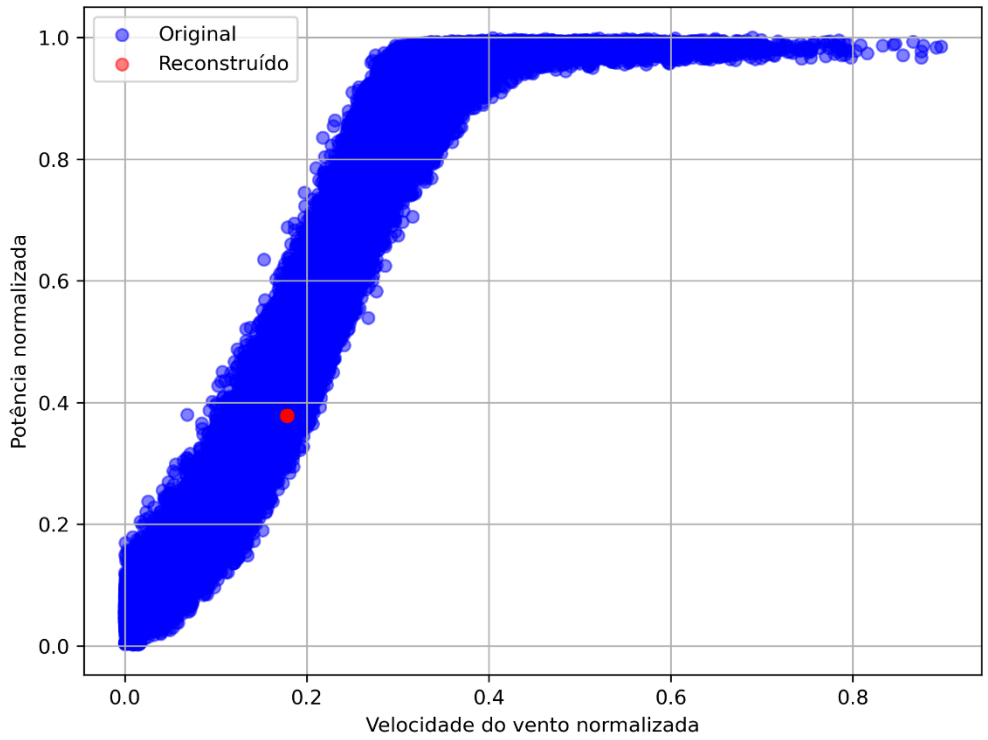
ZHAO, Yongning *et al.* Data-driven correction approach to refine power curve of wind farm under wind curtailment. **IEEE Transactions on Sustainable Energy**, v. 9, n. 1, p. 95–105, 1 jan. 2018.

ZHENG, Le; HU, Wei; MIN, Yong. Raw wind data preprocessing: A data-mining approach. **IEEE Transactions on Sustainable Energy**, v. 6, n. 1, p. 11–19, 1 jan. 2015.

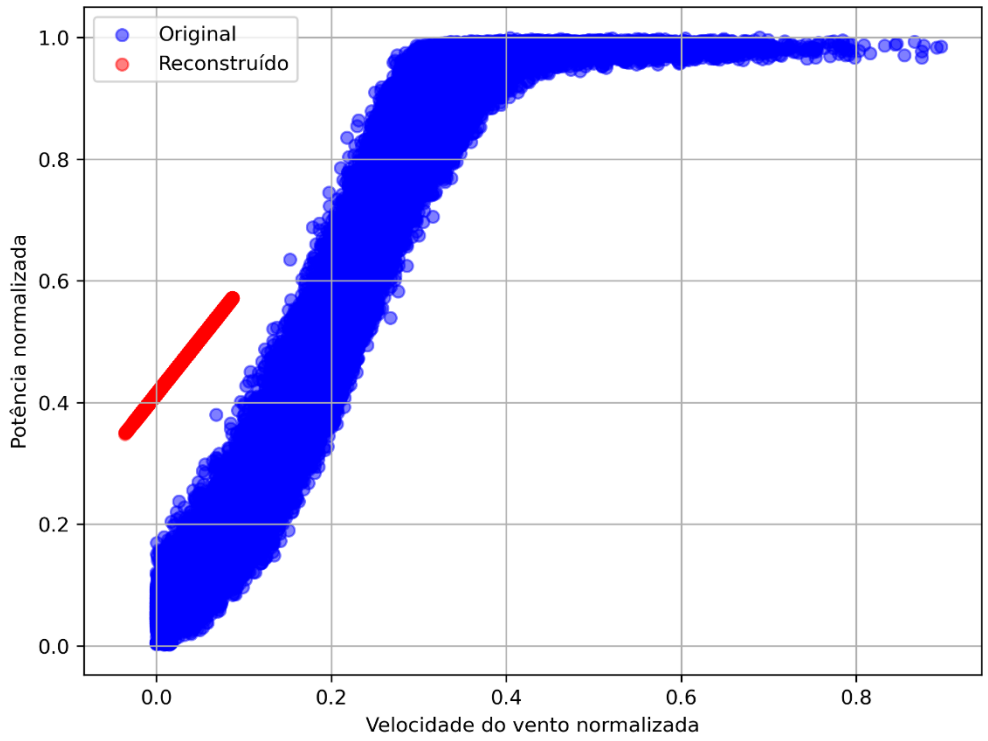
ZOU, Mingzhe; DJOKIC, Sasa Z. **A review of approaches for the detection and treatment of outliers in processing wind turbine and wind farm measurements**. **Energies**MDPI AG, , 1 ago. 2020.

## APÊNDICE A – RECONSTRUÇÃO DA CURVA DE POTÊNCIA COM O *AUTOENCODER* CLÁSSICO

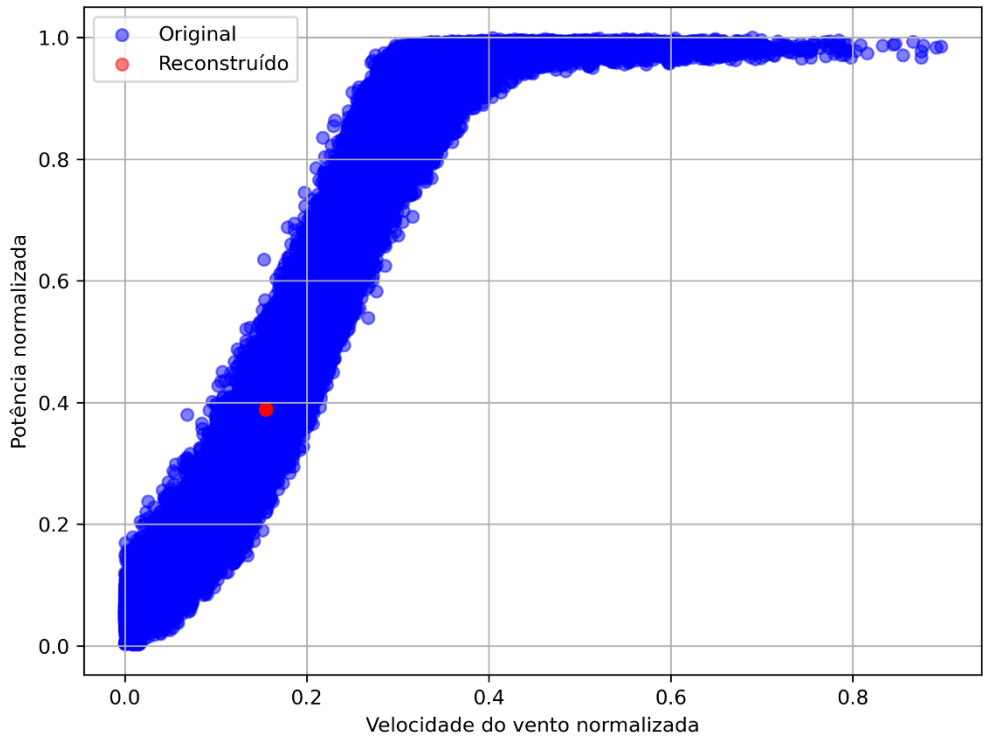
| **Rodada 1** |
|---|



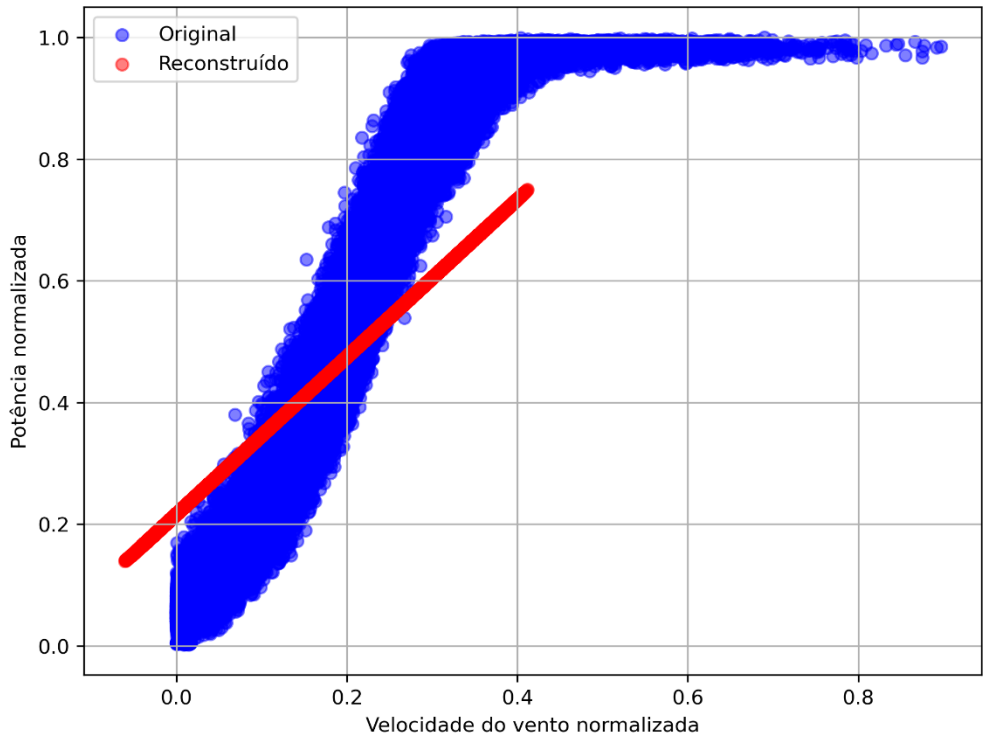

**Rodada 2**



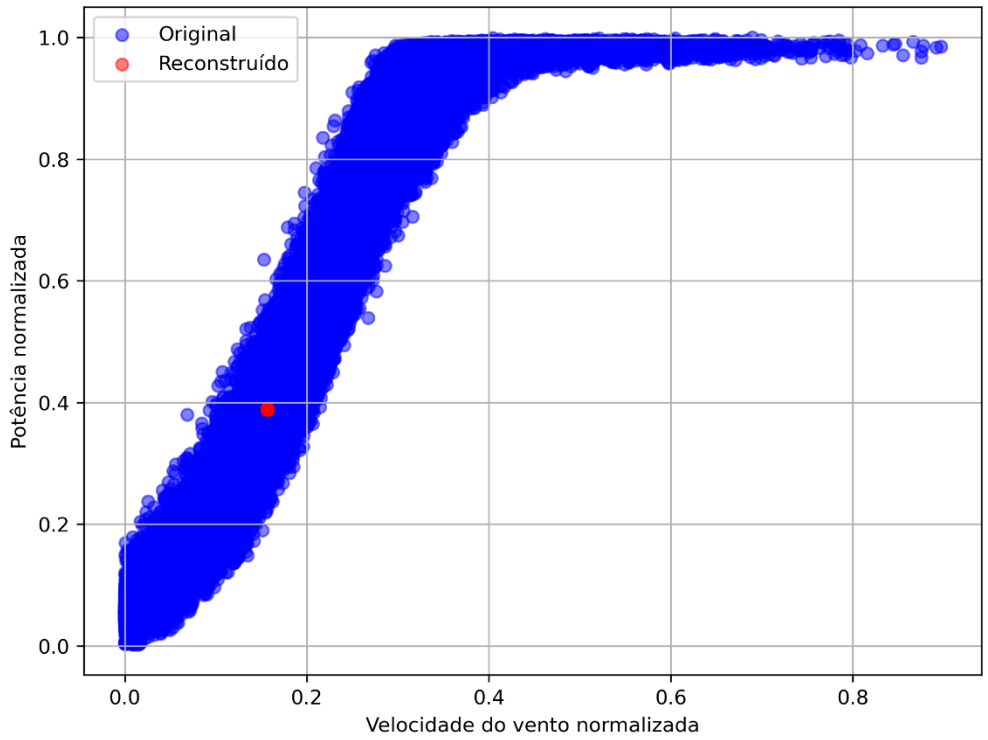

**Rodada 3**



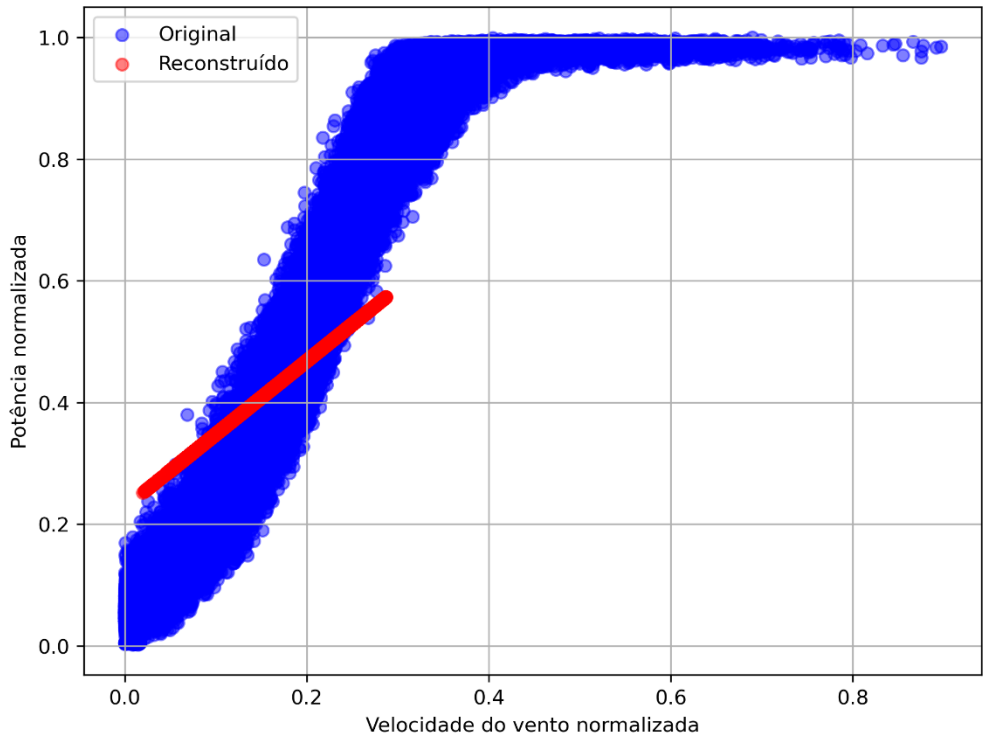

**Rodada 4**



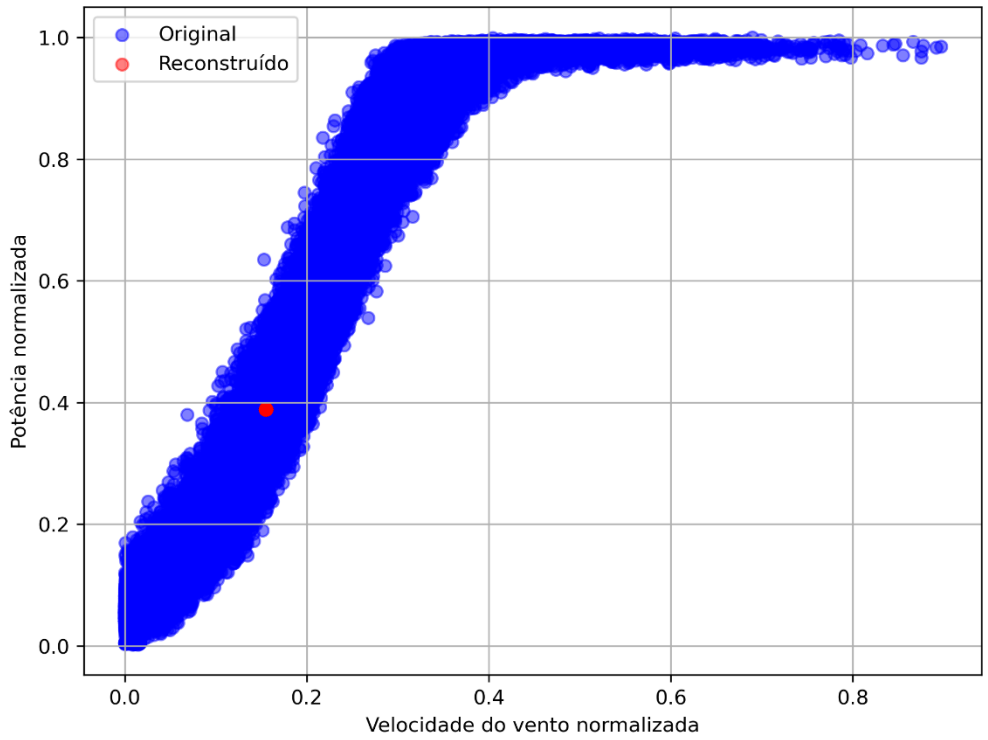

**Rodada 5**



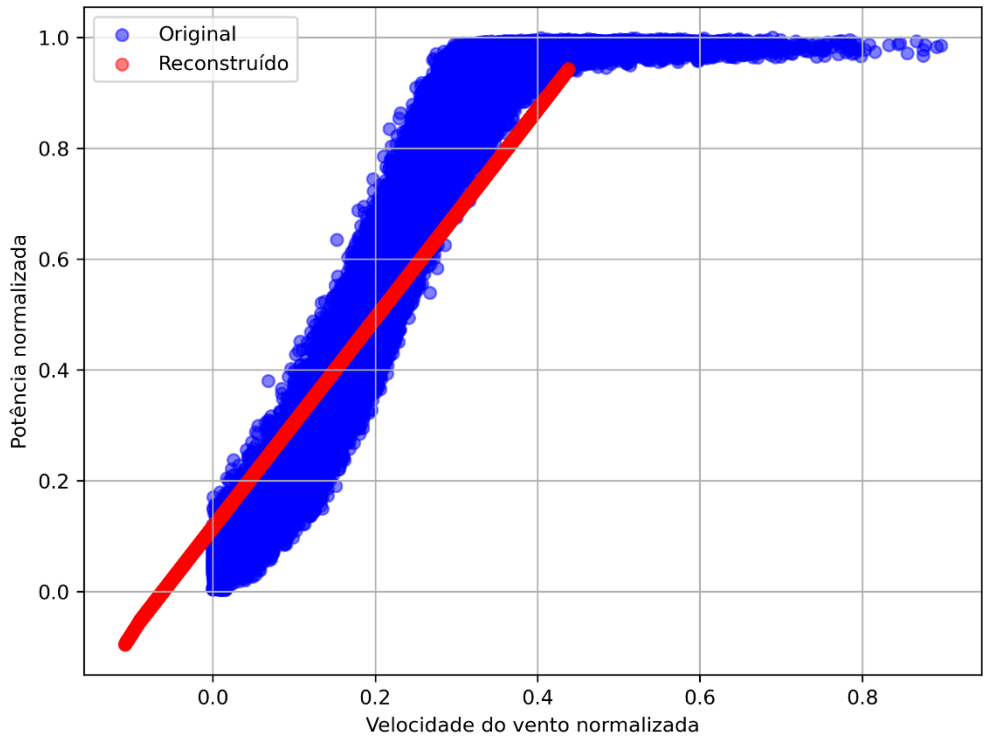

**Rodada 6**



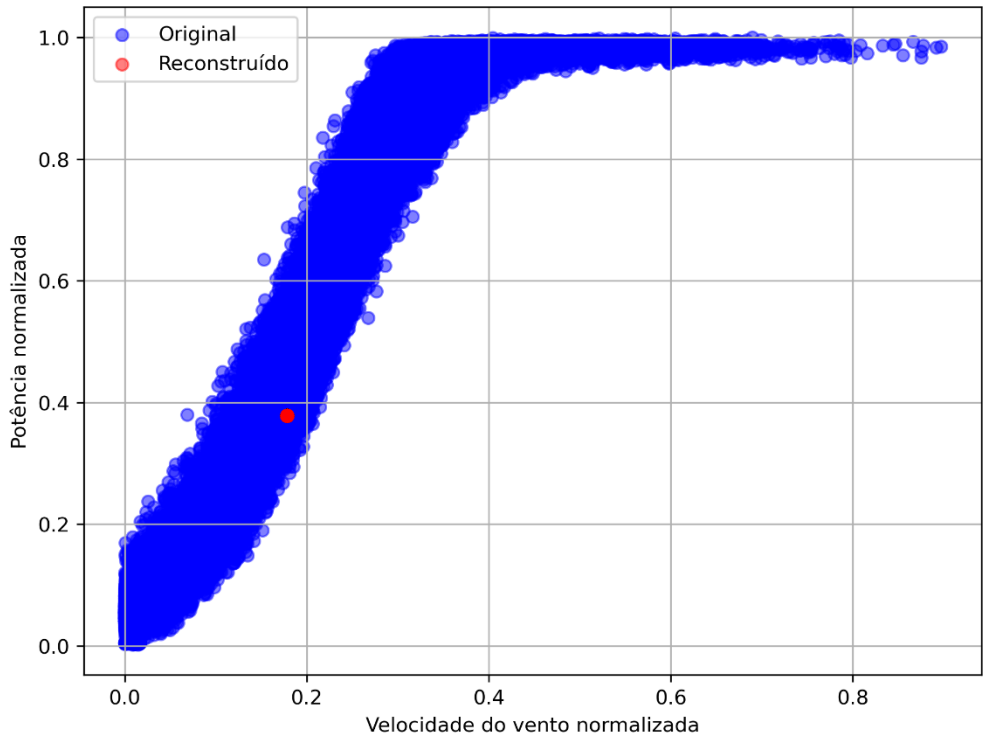

**Rodada 7**



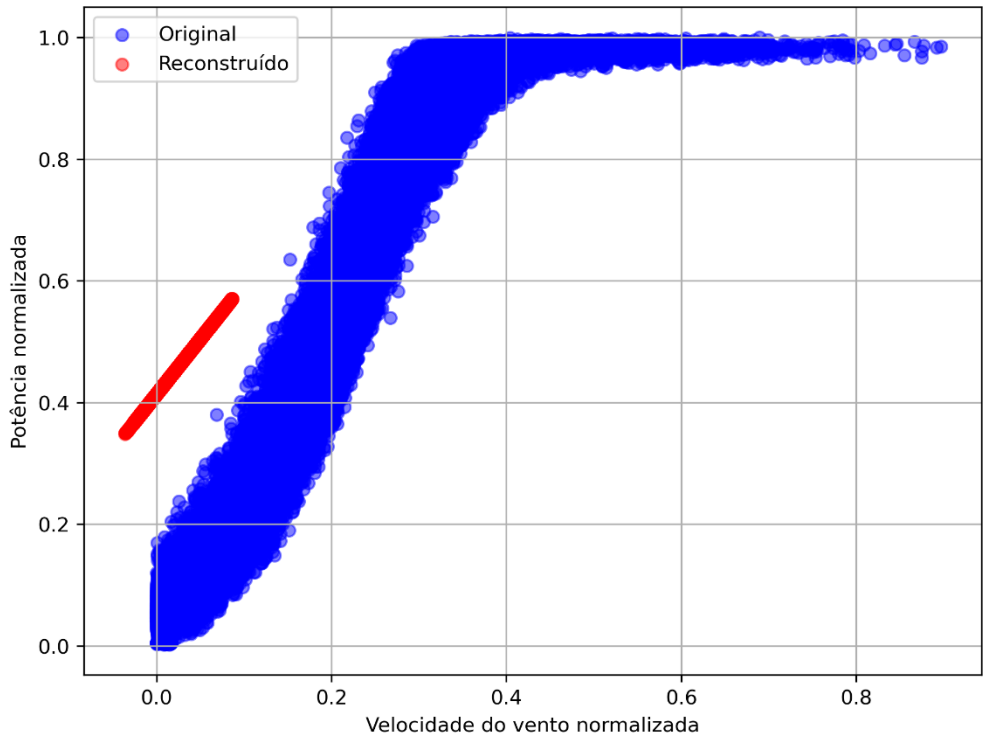

**Rodada 8**



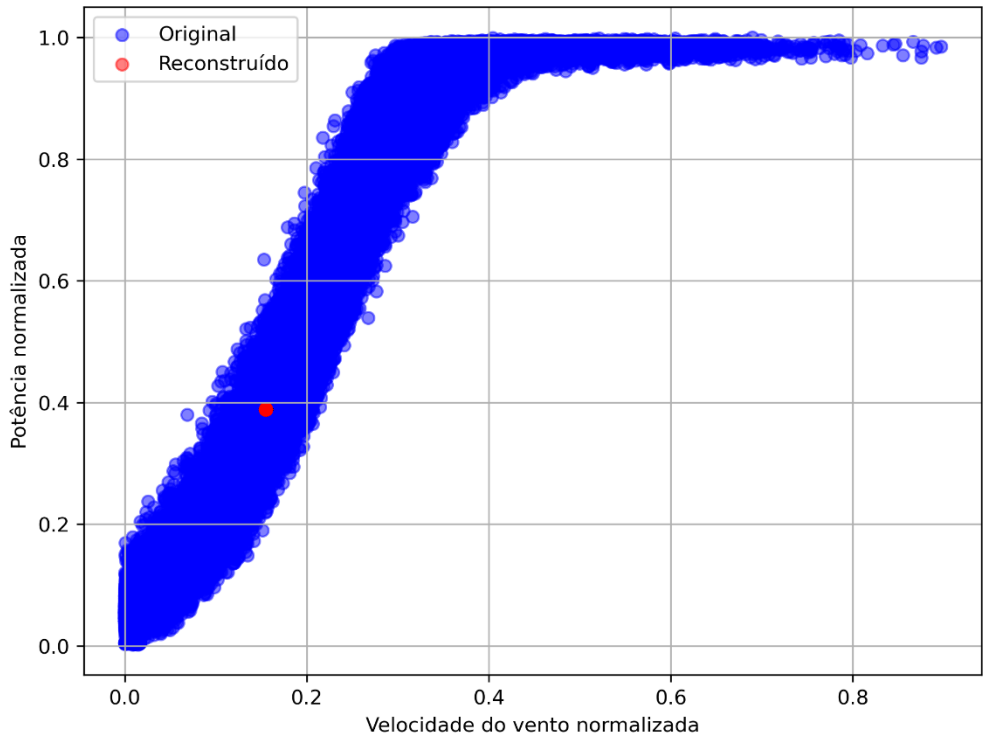

**Rodada 9**



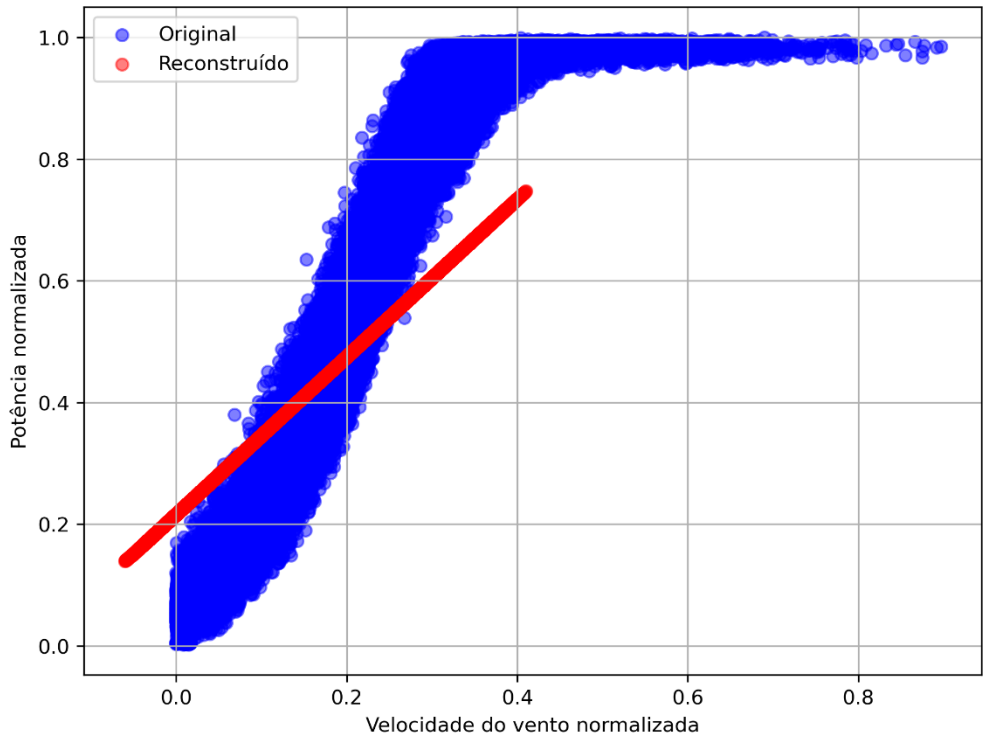

**Rodada 10**



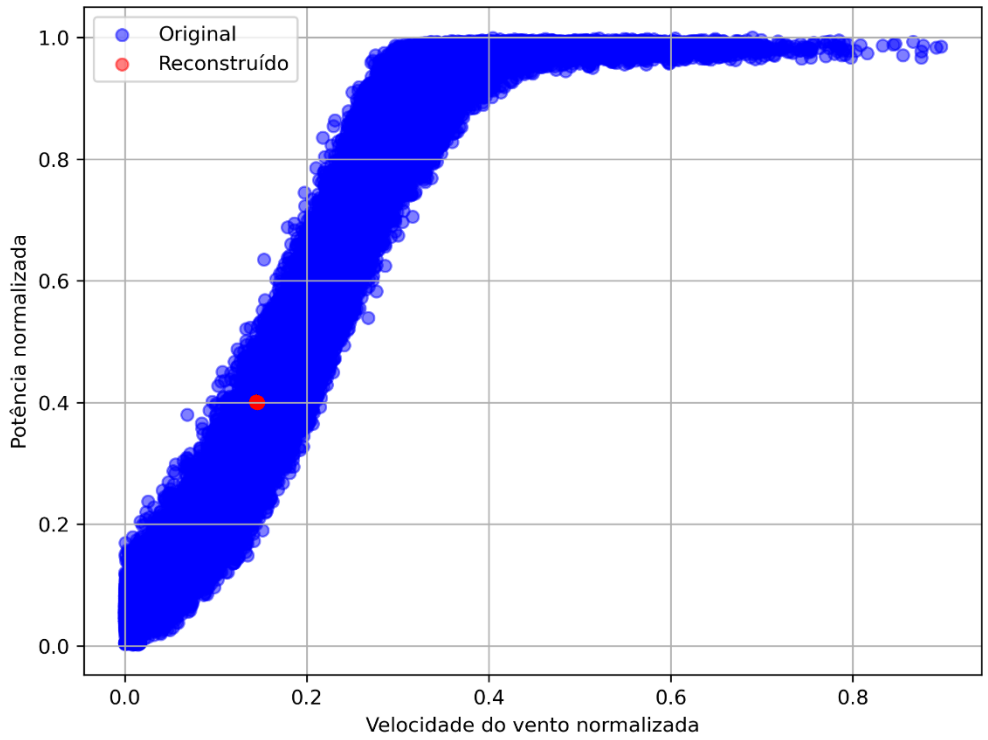

**Rodada 11**



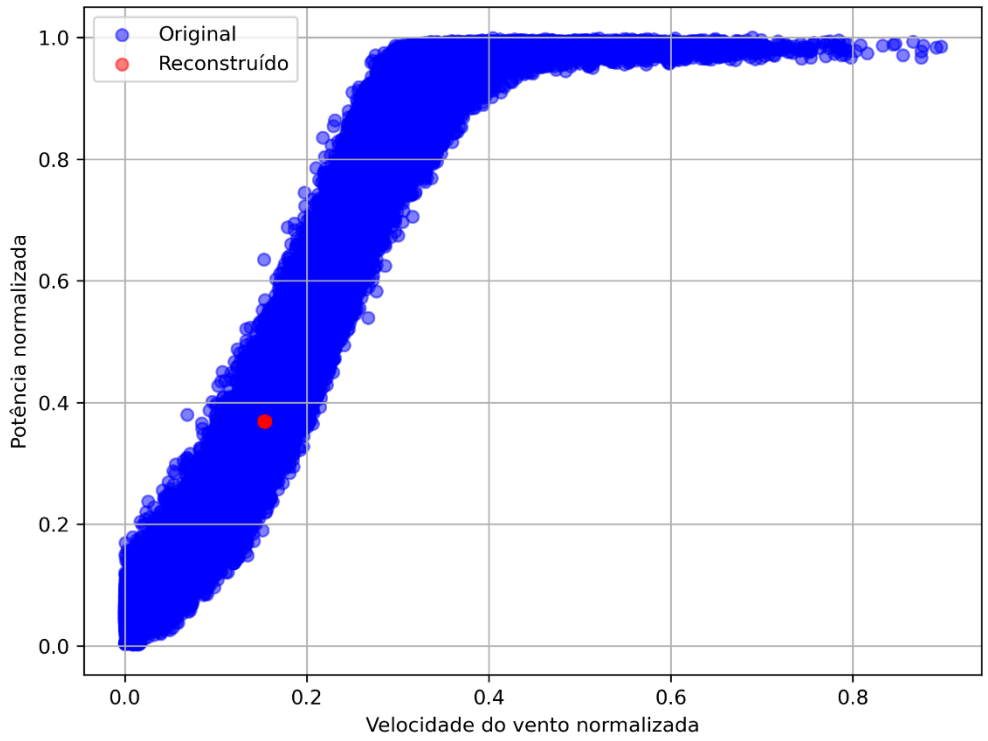

**Rodada 12**



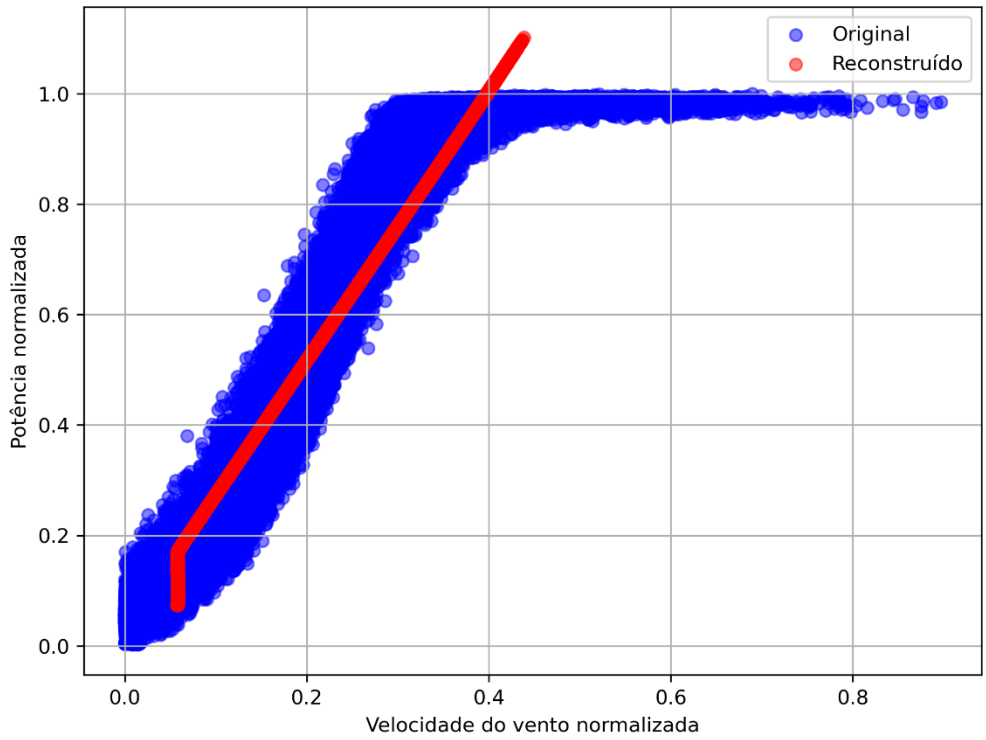

**Rodada 13**



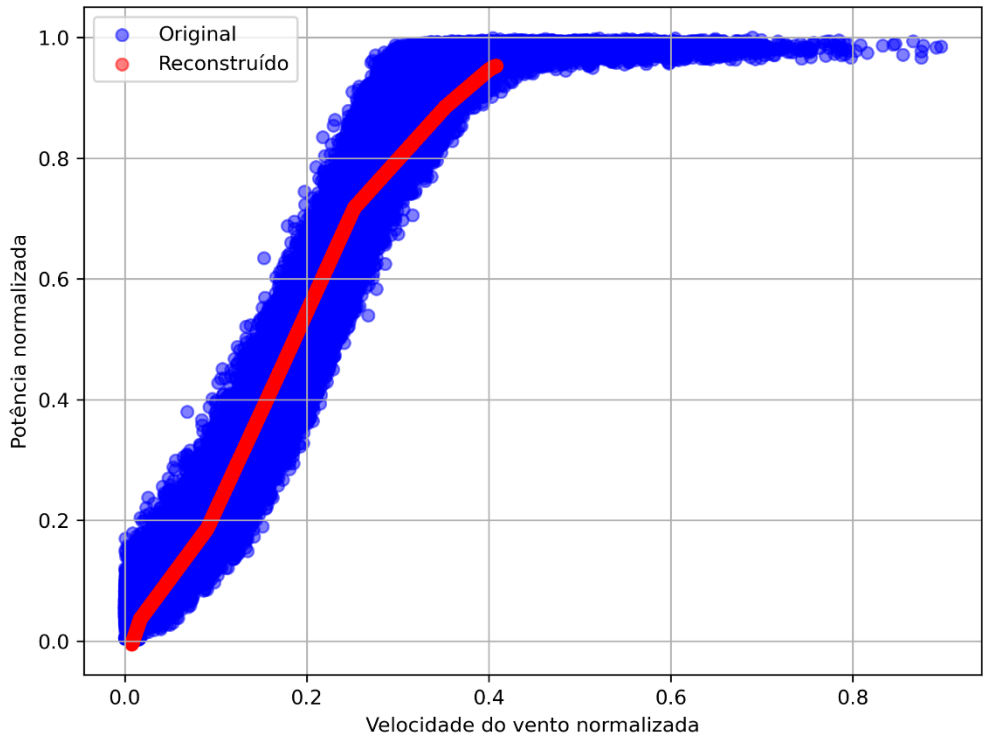

**Rodada 14**



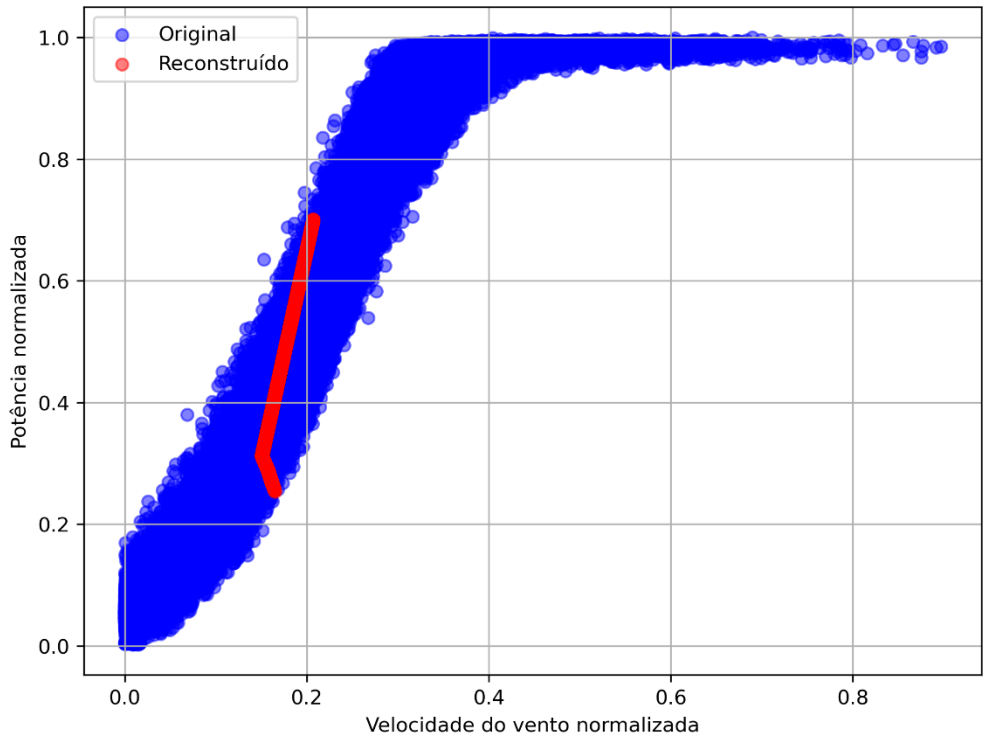

**Rodada 15**



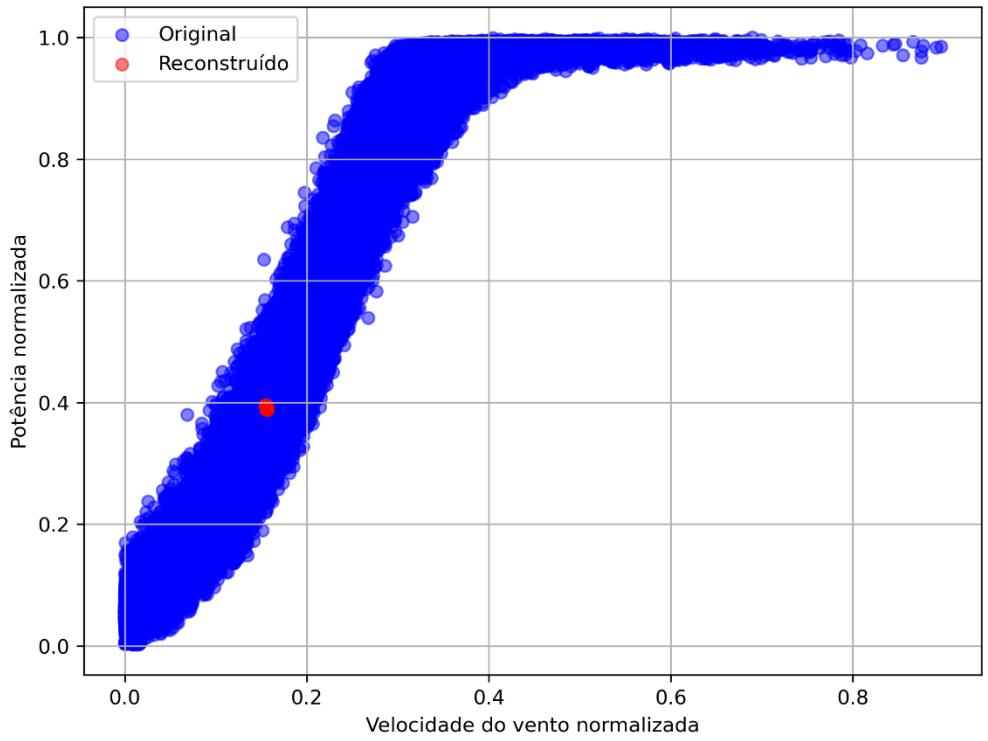

**Rodada 16**



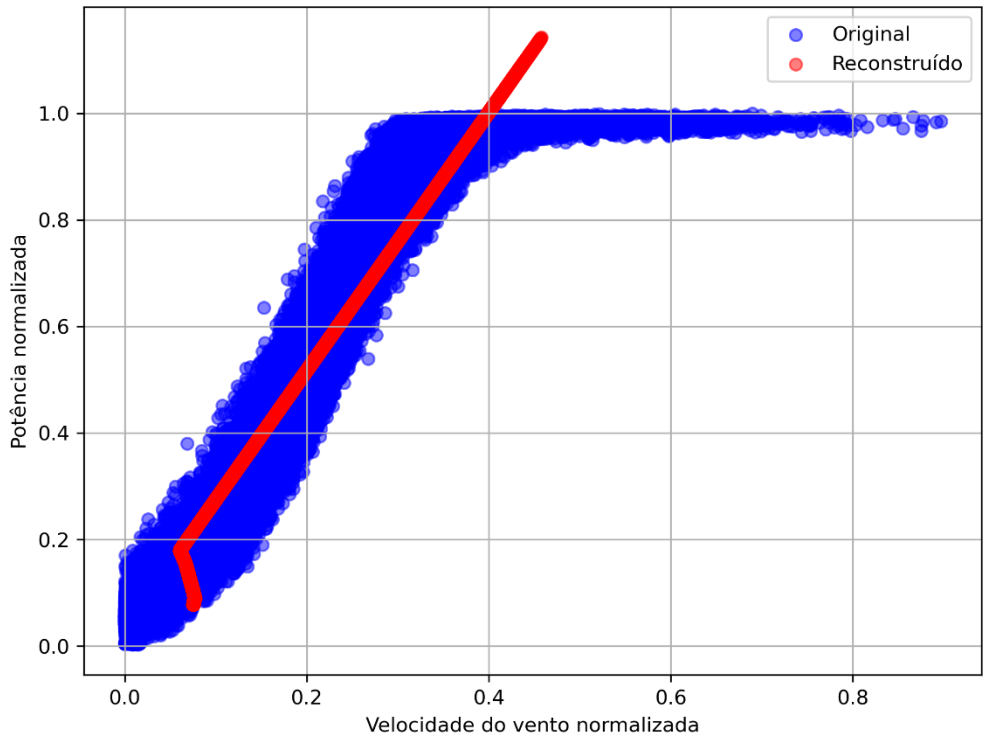

**Rodada 17**



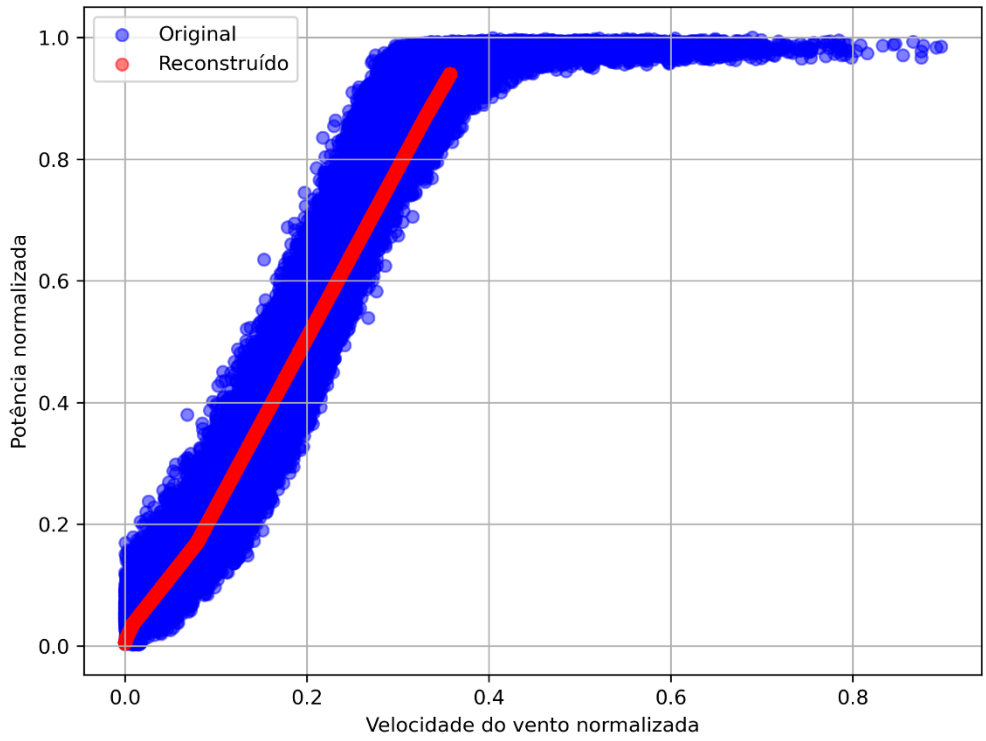

**Rodada 18**



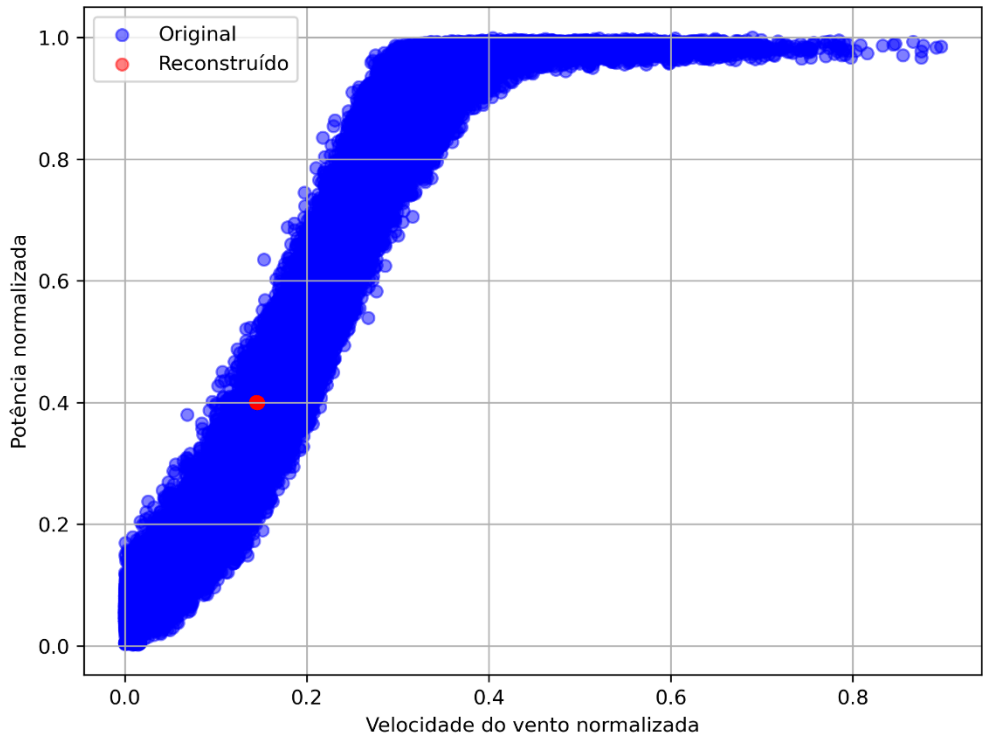

**Rodada 19**



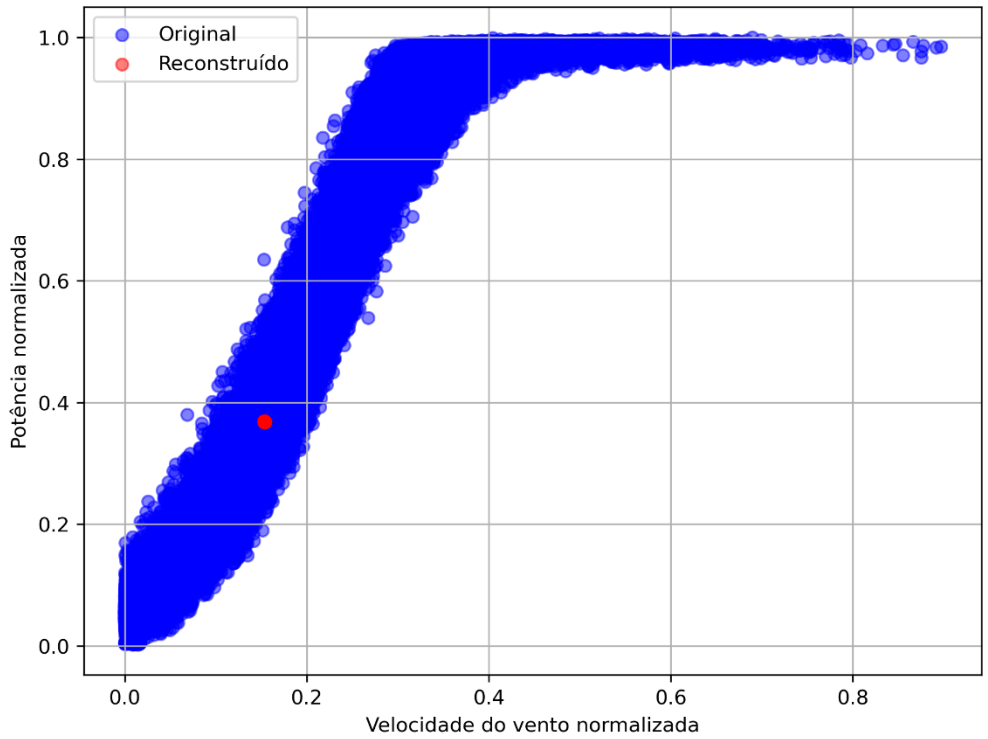

**Rodada 20**



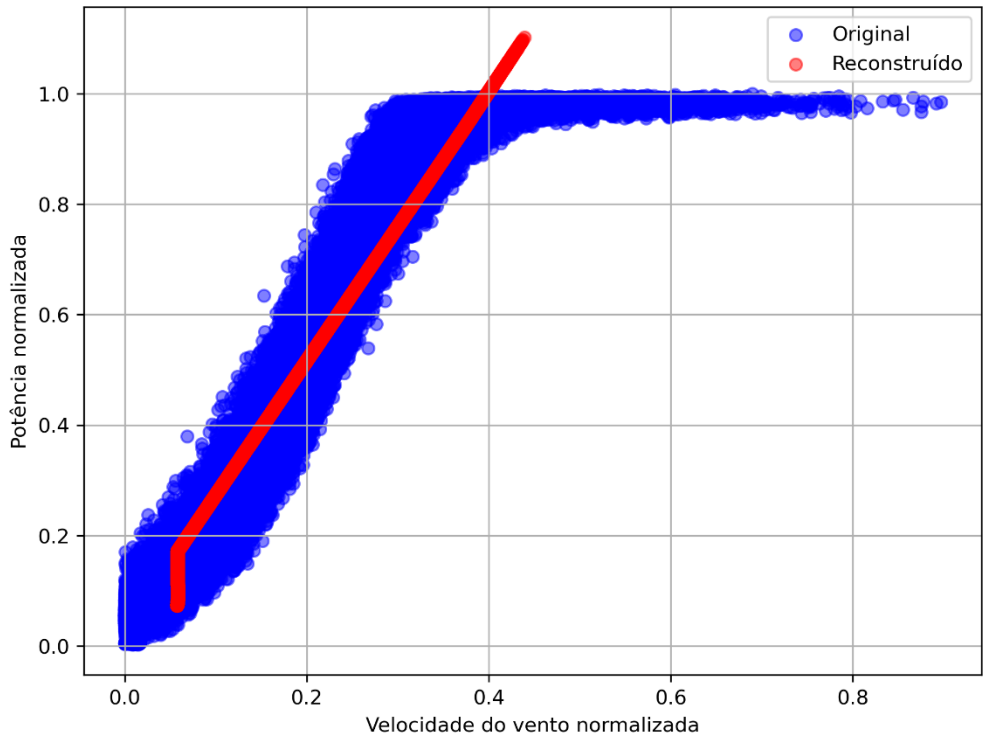

**Rodada 21**



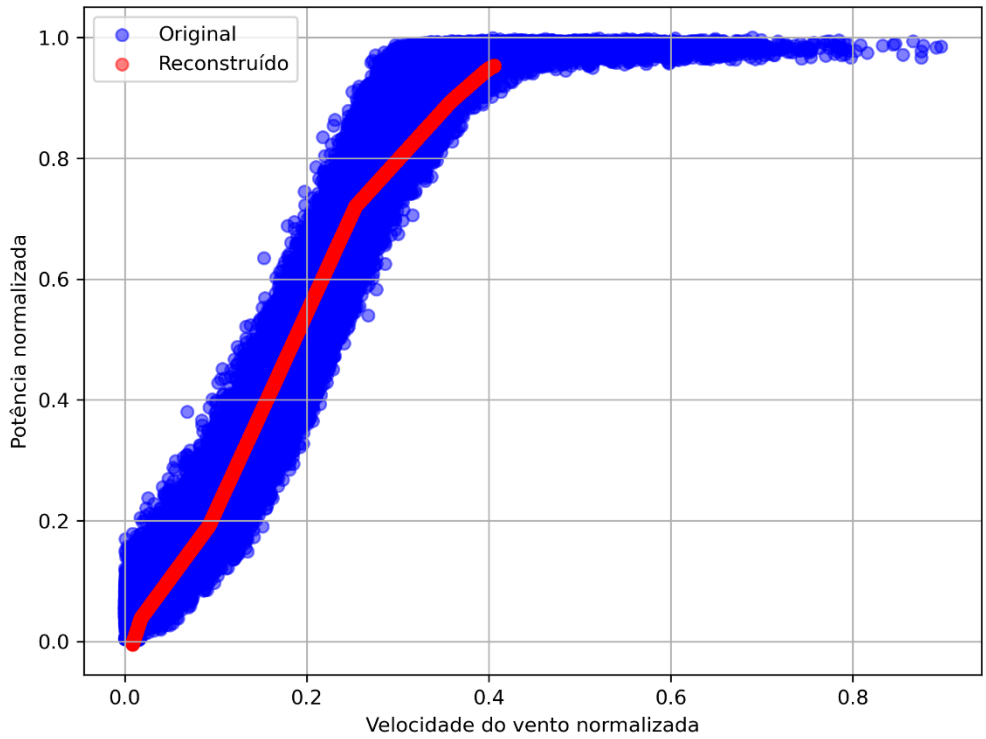

**Rodada 22**




**Rodada 23**

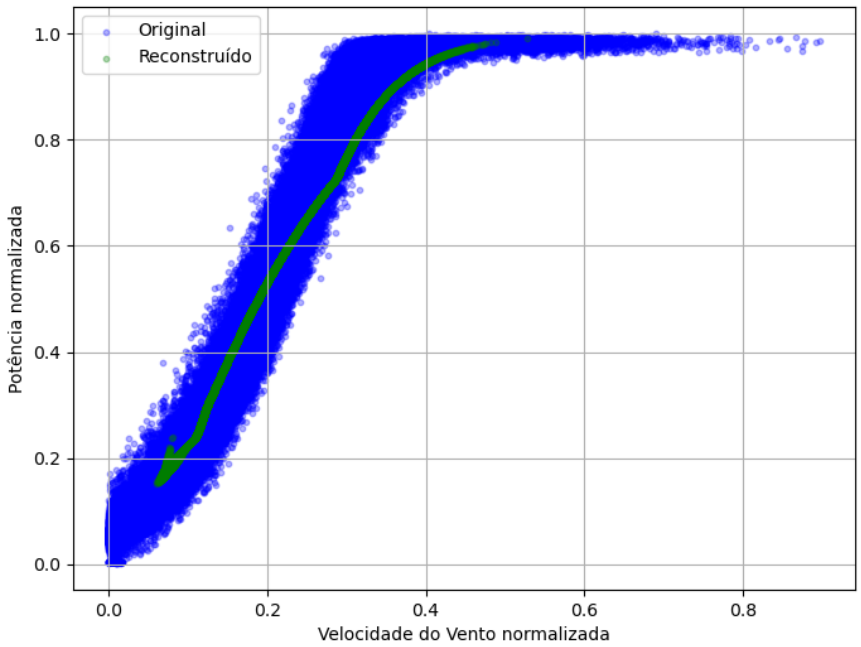

Original
Reconstruído
Potência normalizada
Velocidade do vento normalizada
0.0
0.2
0.4
0.6
0.8
1.0

**Rodada 24**

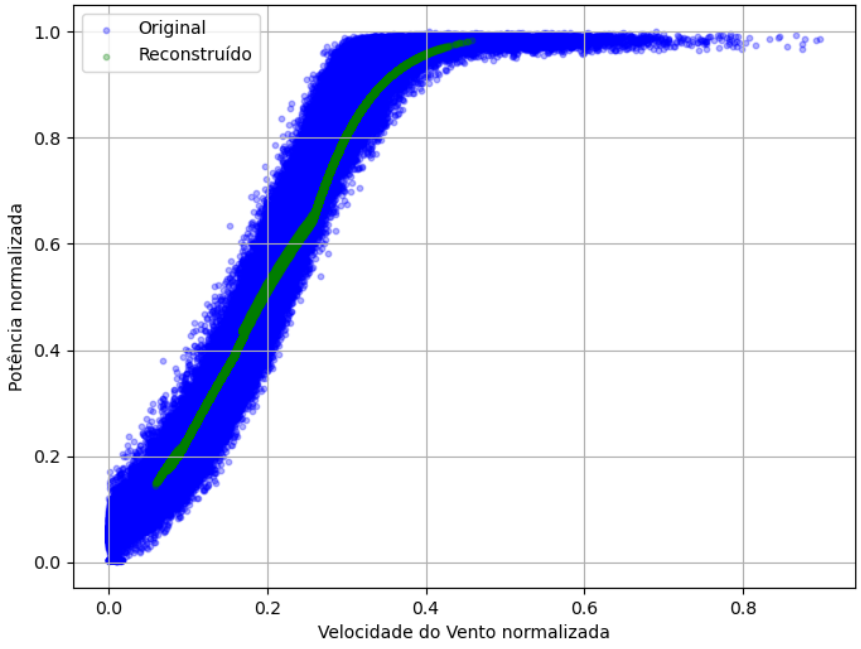

Original
Reconstruído
Potência normalizada
Velocidade do vento normalizada
0.0
0.2
0.4
0.6
0.8
1.0

# APÊNDICE B – RECONSTRUÇÃO DA CURVA DE POTÊNCIA COM O *AUTOENCODER* VARIACIONAL

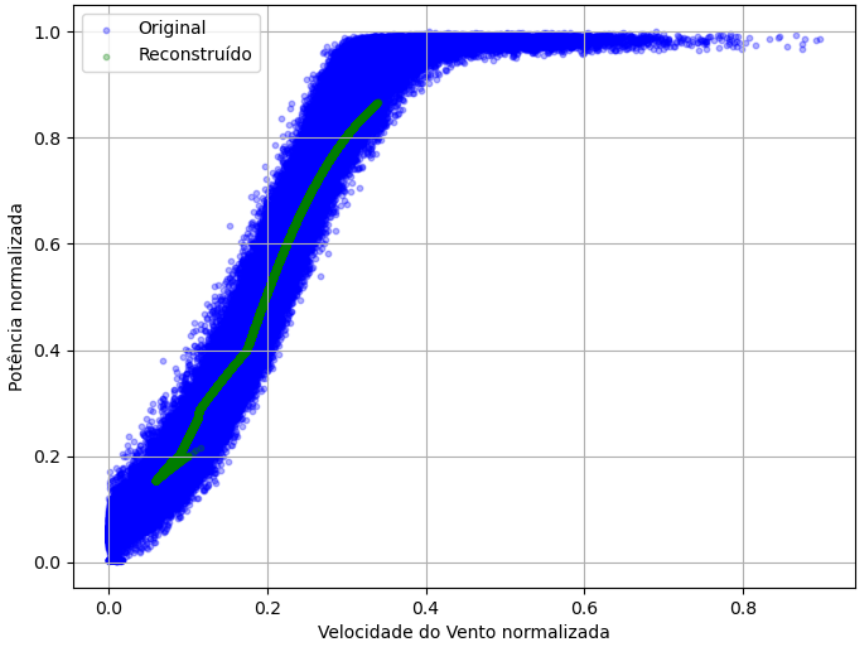
## Rodada 1



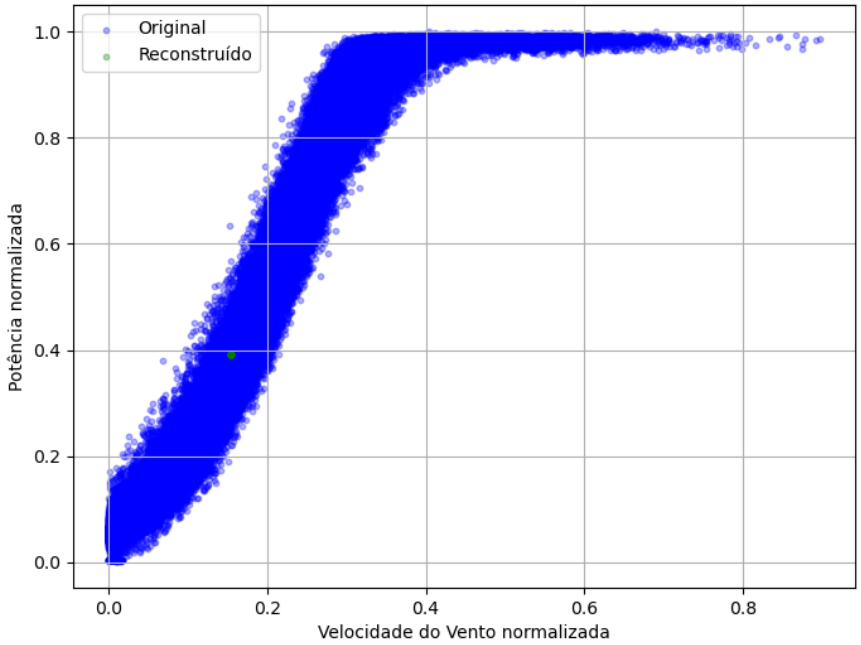

## Rodada 2



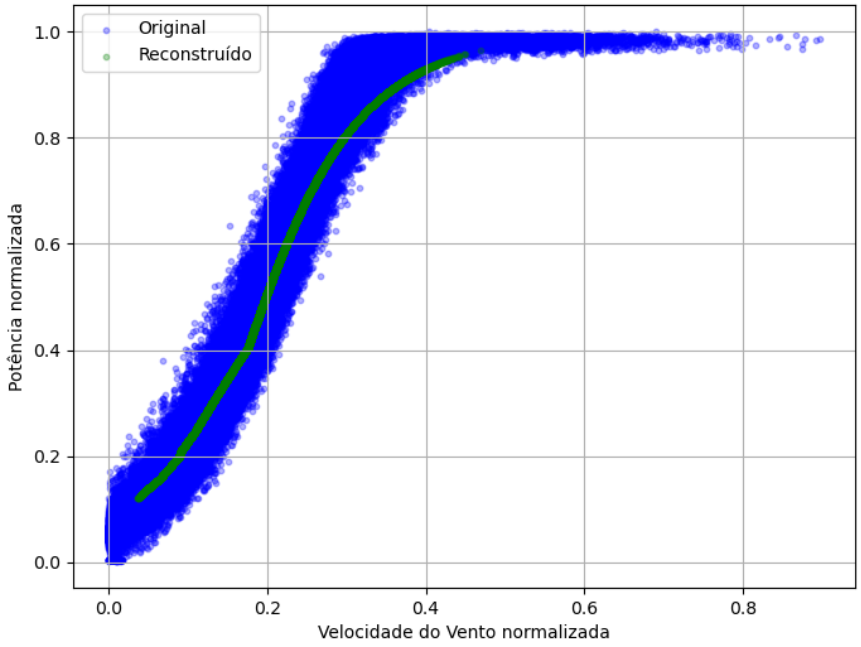
**Rodada 3**



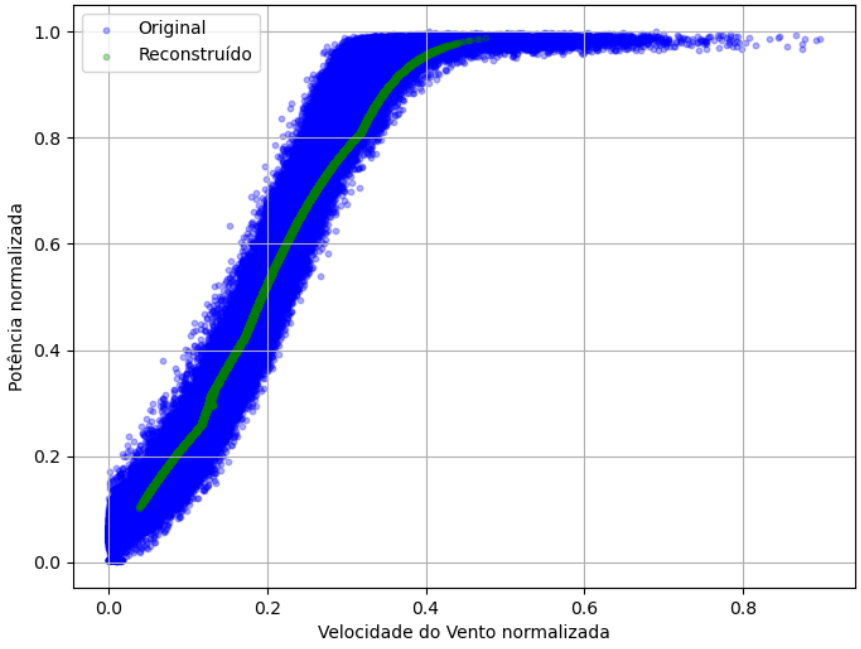

**Rodada 4**



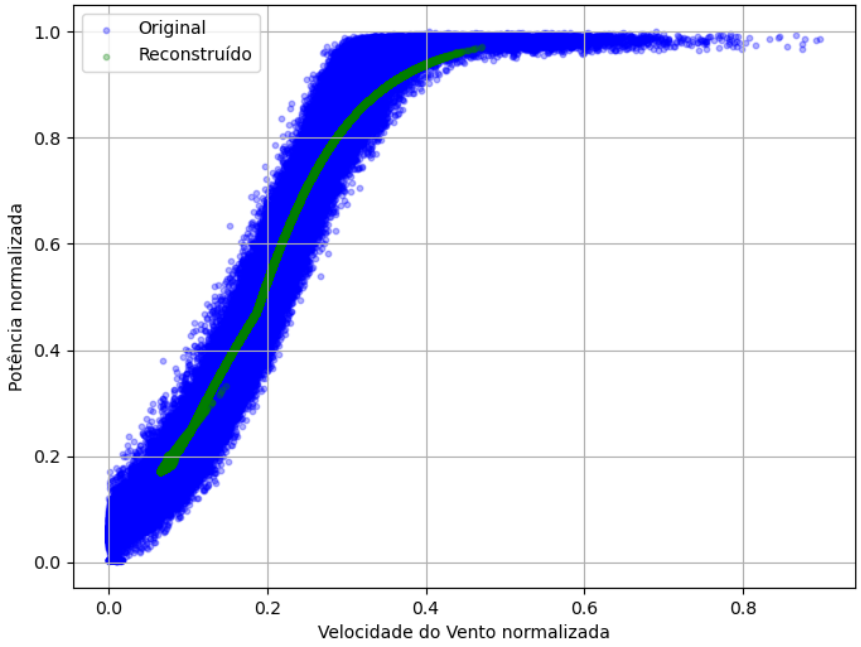

**Rodada 5**



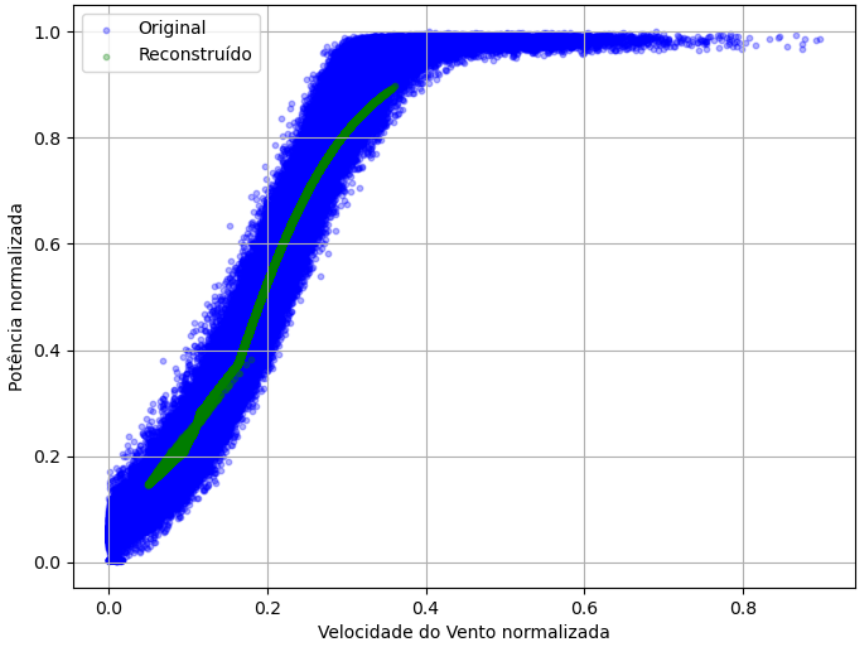

**Rodada 6**



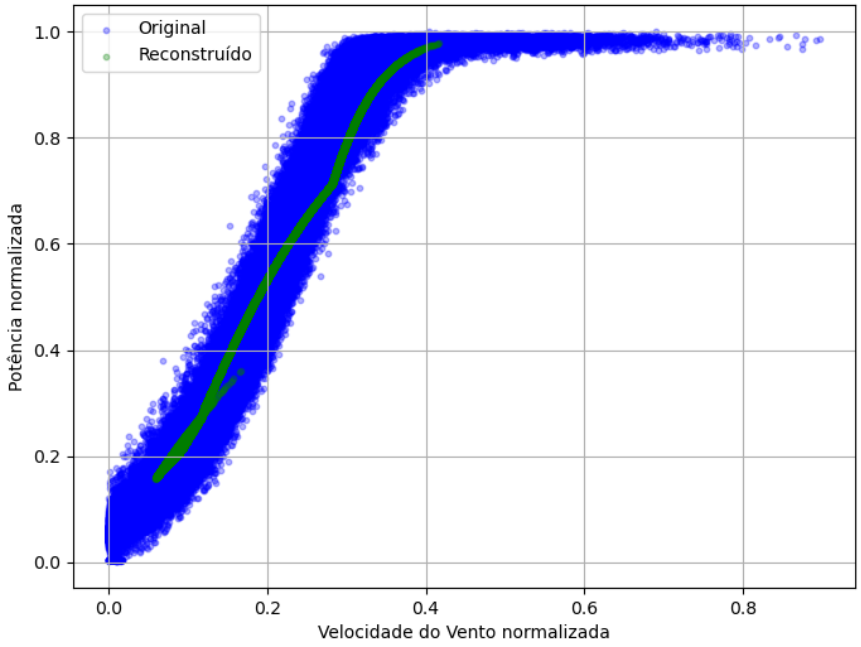

**Rodada 7**



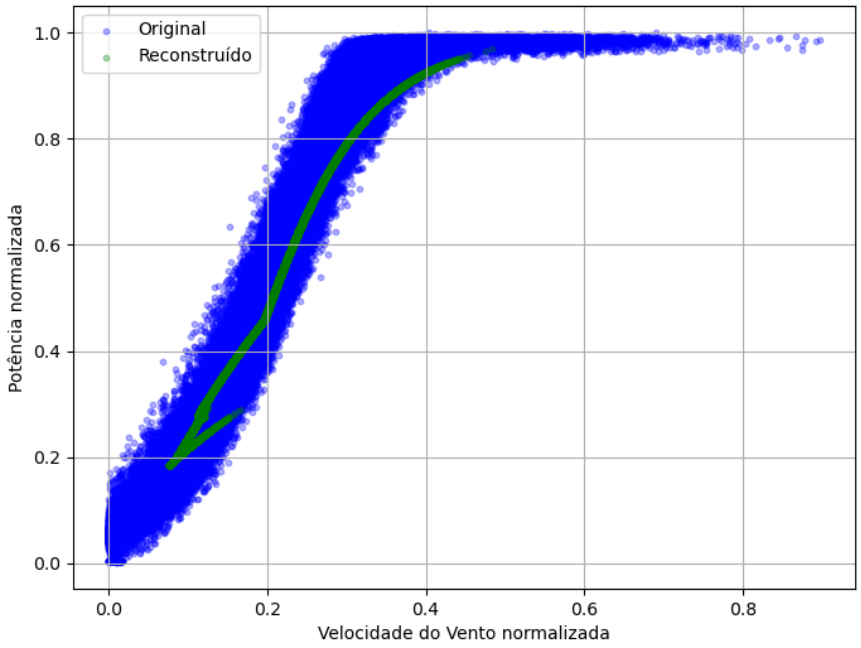

**Rodada 8**



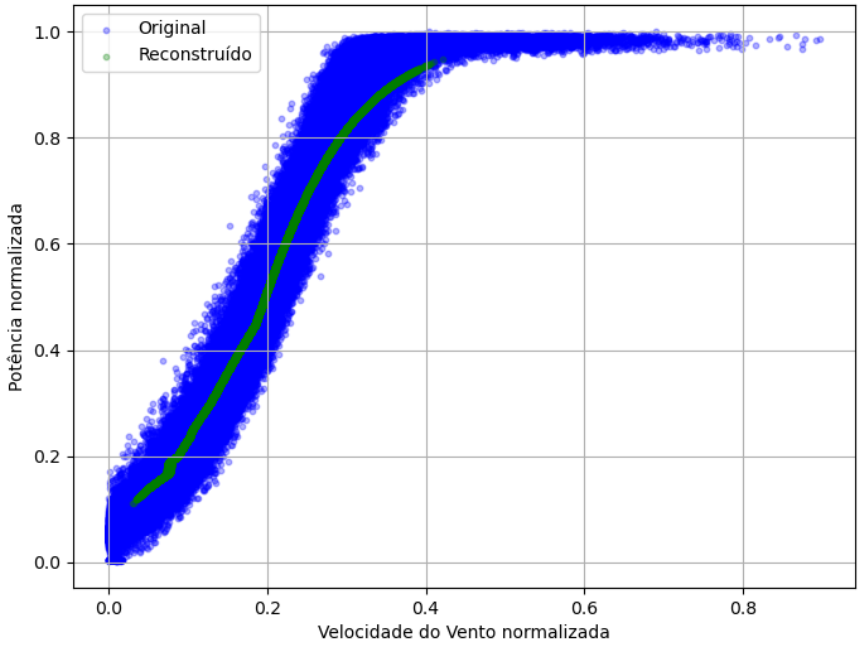

**Rodada 9**



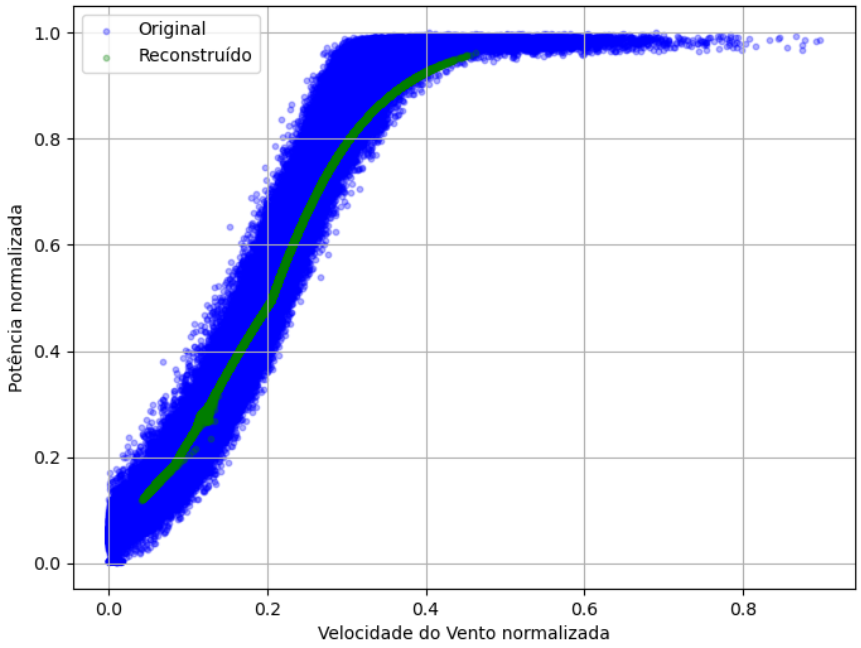

**Rodada 10**



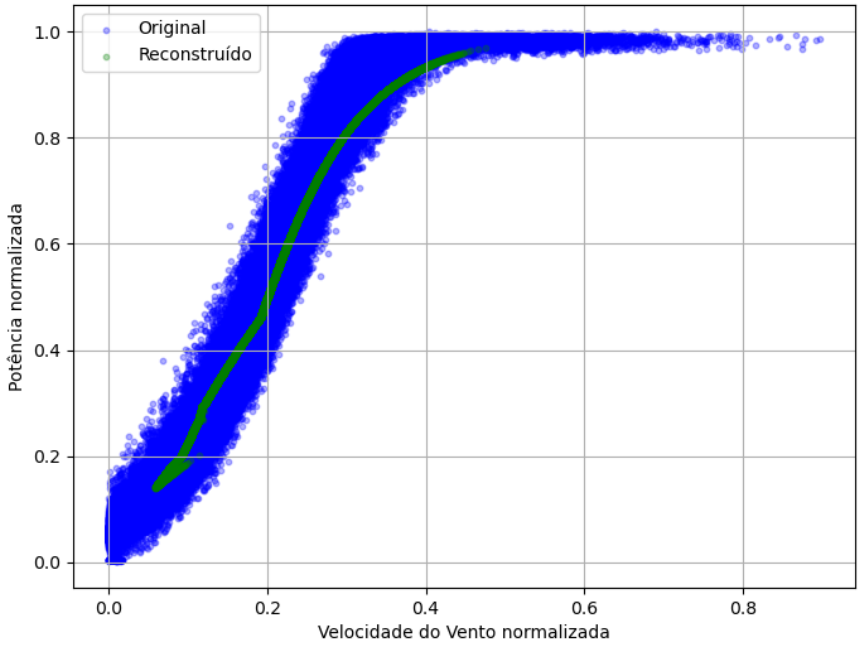

**Rodada 11**



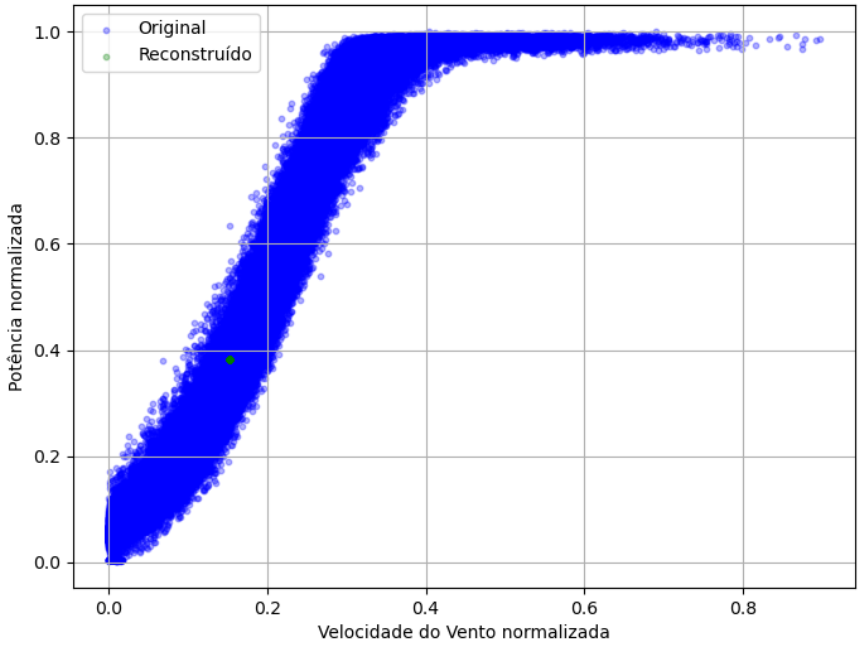

**Rodada 12**



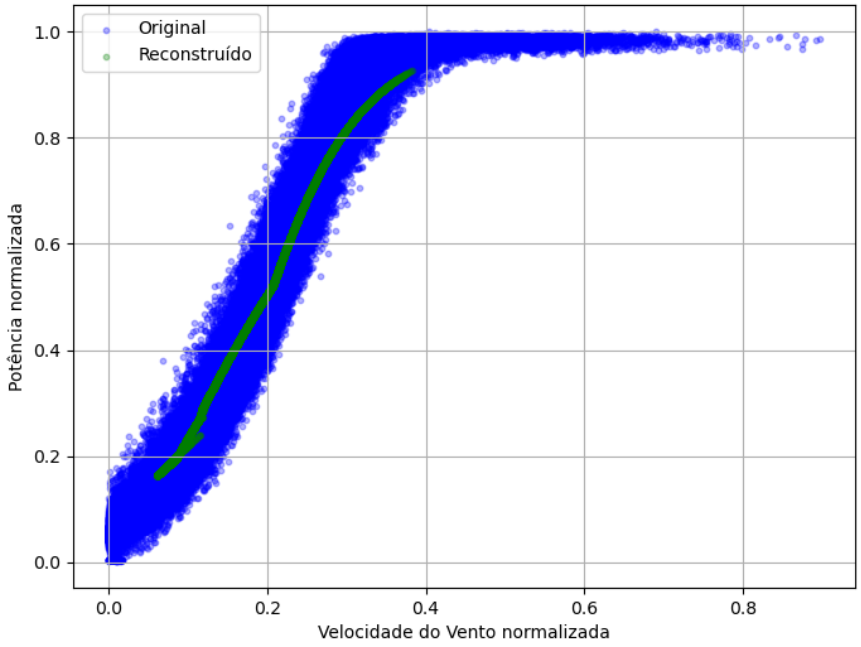

**Rodada 13**



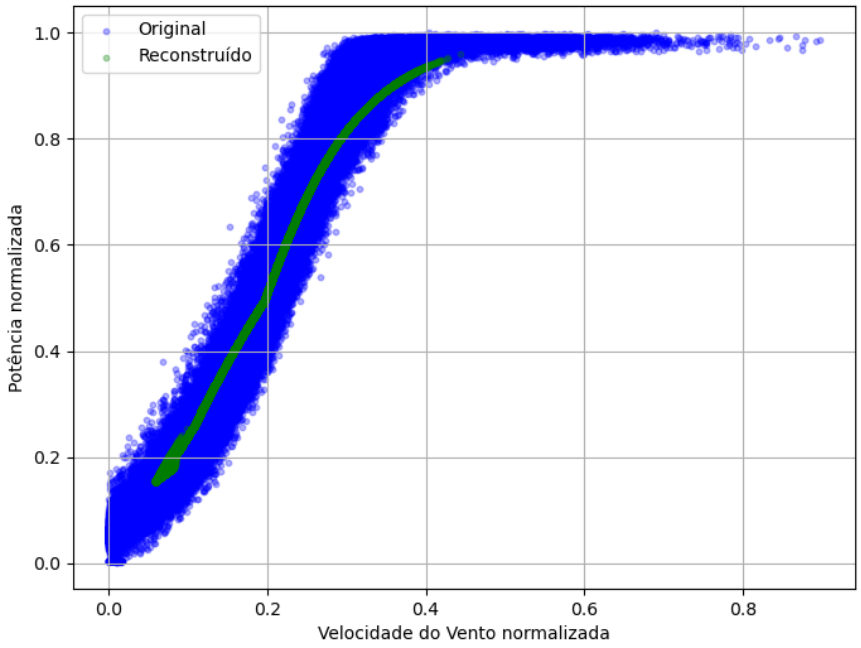

**Rodada 14**




**Rodada 15**


Original
Reconstruído
Potência normalizada
Velocidade do Vento normalizada
0.0
0.2
0.4
0.6
0.8
1.0

**Rodada 16**


Original
Reconstruído
Potência normalizada
Velocidade do Vento normalizada
0.0
0.2
0.4
0.6
0.8
1.0