



**CENTRO FEDERAL DE EDUCAÇÃO
TECNOLÓGICA DE MINAS GERAIS**
Campus Leopoldina - Programa de Pós-Graduação
em Automação e Sistemas

William Rodrigues Silva

ANÁLISE DE RUÍDO AUDÍVEL PARA AEROGERADORES UTILIZANDO APRENDIZADO DE MÁQUINAS

Leopoldina

2025

William Rodrigues Silva

ANÁLISE DE RUÍDO AUDÍVEL PARA AEROGERADORES UTILIZANDO APRENDIZADO DE MÁQUINAS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Automação e Sistemas do Centro Federal de Educação Tecnológica de Minas Gerais, Campus Leopoldina, como parte dos requisitos para obtenção do título de Mestre em Automação e Sistemas.

Orientador: Prof. D. Eng. Murillo Ferreira dos Santos

Coorientador Prof. D. Eng. Mathaus Ferreira da Silva

Leopoldina

2025

Silva, William Rodrigues.

S586a Análise de ruído audível para aerogeradores utilizando aprendizado de máquinas. / William Rodrigues Silva. – 2025.
106 f. : il.

Dissertação apresentada ao Programa de Pós-Graduação em Automação e Sistemas (PPGAS), 2025.

Orientador: Murillo Ferreira dos Santos.

Coorientador: Mathaus Ferreira da Silva.

Bibliografia.

Dissertação (mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais. *Campus* Leopoldina.

1. Energia eólica. 2. Inteligência artificial. 3. Ruído. I. Santos, Murillo Ferreira dos. II. Silva, Mathaus Ferreira da. III. Título.

CDU:681.5:62-533.4

Elaboração da ficha catalográfica pelo bibliotecário Gabriel Jorge Rodrigues Oliveira, CRB 6º 4392 / Cefet/MG *campus* Leopoldina.



ATA DE DEFESA DE DISSERTAÇÃO Nº 9 / 2025 - PPGAS (11.52.20)

Nº do Protocolo: 23062.042975/2025-42

Leopoldina-MG, 14 de agosto de 2025.

DISSERTAÇÃO DE MESTRADO

**ANÁLISE DE RUÍDO AUDÍVEL PARA AEROGERADORES UTILIZANDO
APRENDIZADO DE MÁQUINA**

AUTOR: WILLIAM RODRIGUES SILVA

ORIENTADOR: Professor D. Eng. MURILLO FERREIRA DOS SANTOS

COORIENTADOR EXTERNO: D. Eng. MATHAUS FERREIRA DA SILVA

A Banca Examinadora composta pelos membros abaixo aprovou em 14 de agosto de 2025 esta dissertação:

**Professor Doutor Murillo Ferreira dos Santos (ORIENTADOR)
Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG**

**Doutor Mathaus Ferreira da Silva (COORIENTADOR)
Empresa Robotictech**

**Professor Doutor Ângelo Rocha de Oliveira (MEMBRO INTERNO)
Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG**

**Professor Doutor Bruno Henrique Groenner Barbosa (MEMBRO EXTERNO)
Universidade Federal de Lavras**

(Assinado digitalmente em 15/08/2025 10:03)
ANGELO ROCHA DE OLIVEIRA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DEELP (11.61.04)
Matrícula: 2561441

(Assinado digitalmente em 15/08/2025 15:50)
MURILLO FERREIRA DOS SANTOS
PROFESSOR ENS BASICO TECN TECNOLOGICO
DEELP (11.61.04)
Matrícula: 2919636

(Assinado digitalmente em 15/08/2025 10:22)
BRUNO HENRIQUE GROENNER BARBOSA
ASSINANTE EXTERNO
CPF: ###.###.386-##

(Assinado digitalmente em 15/08/2025 07:50)
MATHAUS FERREIRA DA SILVA
ASSINANTE EXTERNO
CPF: ###.###.456-##

Visualize o documento original em <https://sig.cefetmg.br/public/documentos/index.jsp>
informando seu número: **9**, ano: **2025**, tipo: **ATA DE DEFESA DE DISSERTAÇÃO**, data de
emissão: **14/08/2025** e o código de verificação: **b06abc35f4**

William Rodrigues Silva

ANÁLISE DE RUÍDO AUDÍVEL PARA AEROGERADORES UTILIZANDO APRENDIZADO DE MÁQUINAS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Automação e Sistemas do Centro Federal de Educação Tecnológica de Minas Gerais, Campus Leopoldina, como parte dos requisitos para obtenção do título de Mestre em Automação e Sistemas.

Banca Examinadora:

Prof. D. Eng. Murillo Ferreira dos Santos
CEFET-MG - Brasil
Orientador

Prof. D. Eng. Mathaus Ferreira da Silva
Robotictech - Brasil
Coorientador

Prof. D. Eng. Bruno Henrique Groenner Barbosa
Universidade Federal de Lavras - Brasil
Convidado Externo

Prof. D. Eng. Ângelo Rocha de Oliveira
CEFET-MG - Brasil
Convidado Interno

Leopoldina, 14 de agosto de 2025

Dedicatória

Agradeço a Deus pela oportunidade de concluir mais um curso profissionalizante nesta instituição que tanto marcou minha trajetória. Dedico este trabalho de pesquisa a toda a minha família, em especial à minha amada esposa, que sempre me apoiou ao longo de todo o período deste mestrado, e aos meus pais e ao meu irmão, que, com incentivo e encorajamento constantes, foram essenciais nesta jornada acadêmica.

Agradecimentos

Agradeço à toda a equipe da Robotictech Serviço de Tecnologia pela dedicação e empenho em fazer esse projeto caminhar, em especial ao meu coorientador e CEO da empresa, D. Eng. Mathaus Ferreira da Silva juntamente com o CTO, M. Eng. Juliano Emir Nunes Masson pelo direcionamento e estruturação deste projeto inovador e de grande importância para o setor elétrico que, sem sombra de dúvidas, tem capacidade de evolução para diversas outras áreas.

Agradeço aos meus colegas de trabalho e amigos Iuri Vladimir Molina Junior e Gabriel Miguel Castro Martins, cuja dedicação e empenho foram fundamentais para a concretização deste projeto.

Meus agradecimentos também à toda Equipe da Elera Renováveis por estarem conosco nesta parceria através da troca de conhecimentos entre as equipes e o uso de suas instalações no município de Trairi, no Ceará. Juntos caminhamos para uma jornada de inovação tecnológica para o setor de energia elétrica, em especial, eólica. Ressalto também que esta dissertação se encontra desenvolvida no âmbito do projeto PD-02393-0223/2023 (Sistema para Monitoramento de Ruídos Audíveis de Aerogeradores com IA).

Agradeço também ao meu orientador, Prof. D. Eng. Murillo Ferreira dos Santos, por todo o apoio e empenho na instrução e revisão dos artigos relacionados a este projeto, além desta dissertação.

Por fim, estendo meus agradecimentos a todos que, indiretamente, contribuíram para a realização deste projeto, seja por meio de palavras de incentivo, apoio moral ou compartilhamento de conhecimentos ao longo da jornada.

“Em tudo dai graças”
(Bíblia Sagrada, 1 Tessalonicenses 15, 8)

Resumo

A energia eólica, uma fonte sustentável e crescente na matriz energética brasileira, tem ganhado destaque nos últimos anos. Esse crescimento gera uma demanda crescente por soluções de manutenção preventiva e corretiva para os equipamentos responsáveis pela geração dessa energia. As turbinas eólicas, que consistem na sua grande maioria em sistemas mecânicos rotativos, podem gerar sons na faixa audível (frequentemente fora da faixa audível humana) que podem ser analisados para a detecção de falhas em seus componentes. Esta dissertação visa investigar uma técnica inovadora para a detecção de falhas em turbinas eólicas, utilizando a análise acústica dos sons produzidos por seus mecanismos. A abordagem proposta envolve o uso de redes neurais não supervisionadas para identificar padrões sonoros anômalos e a aplicação de redes neurais especialistas para catalogar e classificar essas falhas. Ao final, são apresentados os resultados experimentais que demonstram a eficácia da técnica proposta, suas limitações e o potencial de aplicação em outros setores industriais. Através do monitoramento de cinco aerogeradores, foram separadas duas falhas com precisão entre 98 e 99% de acurácia na sua classificação.

Palavras-chave: Ruído Audível, Energia Eólica, Inteligência Artificial.

Abstract

Wind energy, a sustainable and increasingly prominent source in Brazil's energy matrix, has gained significant attention in recent years. This growth has led to a rising demand for preventive and corrective maintenance solutions for the equipment involved in its generation. Wind turbines, composed of rotating mechanical systems, can produce audible sounds (often atypical) that may be analyzed for fault detection in their components. This dissertation explores an innovative technique for fault detection in wind turbines based on the acoustic analysis of sounds emitted by their mechanical assemblies. The proposed approach employs unsupervised neural networks to identify anomalous sound patterns and expert neural networks to catalog and classify the detected faults. Finally, experimental results are presented to demonstrate the effectiveness of the proposed method, its limitations, and its potential for application in other industrial sectors. Through the monitoring of five wind turbines, two faults were accurately classified with an accuracy ranging from 98 to 99%.

Keywords: Audible Noise, Wind Power, Artificial Intelligence.

Lista de ilustrações

Figura 1.1 – Capacidade total instalada <i>onshore</i> . Adaptado de GWEC 2024.	2
Figura 1.2 – Oferta interna de eletricidade do Brasil. Adaptado de MME 2024.	3
Figura 1.3 – Estrutura de um aerogerador e suas partes constituintes (Windbox, 2020).	4
Figura 3.1 – Fluxo geral da solução.	17
Figura 3.2 – Fluxograma do algoritmo de detecção.	18
Figura 3.3 – Esquemático de montagem do hardware de captura de áudio.	19
Figura 3.4 – Vista superior do primeiro protótipo do dispositivo gravador de áudio: (a) com os componentes à mostra. (b) mostrando os cabos e conectores.	21
Figura 3.5 – Localização do dispositivo gravador de áudios no aerogerador.	22
Figura 3.6 – Diagrama simplificado do controle de acesso ao servidor.	23
Figura 3.7 – Diagrama de entidades do banco de dados.	28
Figura 3.8 – Diagrama simplificado do controle de acesso ao servidor.	33
Figura 3.9 – Diagrama de endpoints.	34
Figura 3.10 – Representação do Denoising Autoencoder (DAE).	38
Figura 3.11 – Arquitetura DAE utilizada no projeto.	38
Figura 3.12 – Funções de ativação sigmóide e ReLU.	40
Figura 3.13 – Arquitetura Variational Autoencoder (VAE) utilizada.	41
Figura 3.14 – Função de ativação LeakyReLU para diferentes parâmetros α	42
Figura 3.15 – Função de ativação SELU para diferentes valores de α	43
Figura 3.16 – Processamento de dados para adaptação às redes autoencoder.	43
Figura 3.17 – Áudio de um gerador em vento (a) fraco e (b) forte.	48
Figura 3.18 – Desempenho do DAE treinado em 150 épocas.	49
Figura 3.19 – Desempenho do VAE treinado em 200 épocas.	50
Figura 3.20 – Diagrama de cálculo do SSIM.	51
Figura 3.21 – Demonstração das Bandas de Bollinger aplicadas a um conjunto de dados.	58
Figura 3.22 – Separação linear de dados pelo algoritmo SVM.	59
Figura 4.1 – Áudios dos ruídos capturados inicialmente como possíveis falhas.	62
Figura 4.2 – Amostras Normais x Ruídos de 10% do áudio original.	63
Figura 4.3 – Outro comparativo adicionando ruído aos dados a partir da amostra duzentos.	63
Figura 4.4 – Comparativo adicionando ruído aos dados a partir da amostra duzentos com a métrica SSIM.	64
Figura 4.5 – Comparativo de áudio (a) de uma turbina normal (FAI1-05) e (b) da turbina apresentando falha no rolamento (FAI2-07).	65

Figura 4.6 – Inferência do VAE por meio da análise do SSIM e PSNR.	66
Figura 4.7 – Inferência do SSIM analisada no domínio de tempo.	66
Figura 4.8 – Inferência do PSNR analisada no domínio de tempo.	67
Figura 4.9 – Distribuição normal do SSIM e PSNR da análise de (a) rolamento e (b) limiar.	68
Figura 4.10–Inferência da análise de falha no componente gerador de indução. . . .	69
Figura 4.11–Distribuição normal do SSIM e PSNR da análise de (a) falha no Doubly Fed Induction Generator (DFIG) e (b) limiar.	70
Figura 4.12–Detector de anomalias utilizando Filtro de Kalman trazendo (a) a plo- tagem das amostras juntamente com as médias e Bandas de Bollinger; (b) (b) retirada das amostras; e (c) acionamento do detector de anomalias pelos parâmetros.	71
Figura 4.13–Detector de anomalias utilizando Filtro de Kalman, apresentando: (a) a plotagem das amostras com as médias e as Bandas de Bollinger; (b) a retirada das amostras; (c) o acionamento contínuo do detector de anomalias com base nos parâmetros; e (d) o acionamento discreto do detector de anomalias com base nos mesmos parâmetros.	73
Figura 4.14–Espaço latente gerado pelo Conditional Denoising Autoencoder (CDAE).	74
Figura 4.15–Espaço latente gerado pelo Conditional Denoising Autoencoder (CVAE). .	74
Figura 4.16–Aplicação do Support Vector Machine (SVM) no espaço latente do CDAE para detecção de falha no anel estrela.	75
Figura 4.17–Aplicação do SVM no espaço latente do CVAE para detecção de falha no mancal.	76

Lista de tabelas

Tabela 1.1 – Fator de disponibilidade médio das plantas por fonte de energia e por regime regulatório (Elera Renováveis, 2024).	4
Tabela 3.1 – Pipeline de processamento de áudio e transformação de dados.	19
Tabela 3.2 – Divisão dos dados com base na velocidade do vento.	48
Tabela 3.3 – Limiares obtidos empiricamente para o detector de anomalias.	58
Tabela 3.4 – Divisão dos dados segmentados em treino e teste para as redes especialistas.	60
Tabela 4.1 – Resultado da análise de falha no rolamento.	67
Tabela 4.2 – Análise da inferência para detecção de falha no gerador de indução. . .	69
Tabela 4.3 – Acurácia para dados rotulados e não rotulados nas classificações com <i>CVAE</i> e <i>CDAE</i>	77

Lista de abreviaturas e siglas

API	Interface de Programação de Aplicações, do inglês: <i>Application Programming Interface</i>
AUC	Área abaixo da curva, do inglês: <i>Area Under the Curve</i>
CDAE	Autoencoder Denoising Condicional, do inglês: <i>Conditional Denoising Auto-encoder</i>
CNN	Redes Neurais Convolucionais, do inglês: <i>Convolutional Neural Network</i>
CVAE	Autoencoder Variacional Condicional, do inglês: <i>Conditional Variational Auto-encoder</i>
DAE	Autoencoder Denoising, do inglês: <i>Denoising Auto-encoder</i>
DFIG	Gerador de Indução Duplamente Alimentado, do inglês: <i>Doubly Fed Induction Generator</i>
DT	Árvore de decisão, do inglês: <i>Decision Tree</i>
FFT	Transformada rápida de Fourier, do inglês: <i>Fast Fourier Transform</i>
IA	Inteligência Artificial
JSON	Notação de Objeto JavaScript, do inglês: <i>JavaScript Object Notation</i>
MMA	Média Móvel Aritmética
MSE	Erro Quadrático Médio, do inglês: <i>Mean Squared Error</i>
OCSVM	Máquina de Vetores de Suporte de Uma Classe, do inglês: <i>One-Class Support Vector Machine</i>
OSH	Hiperplano de Separação Ideal, do inglês: <i>Optimal Separating Hyperplane</i>
PCA	Análise de Componentes Principais, do inglês: <i>Principal Component Analysis</i>
PSNR	Relação Pico-Sinal-Ruído, do inglês: <i>Peak-Signal Noise Ratio</i>
ReLU	Unidade Linear Rectificada, do inglês: <i>Rectified Linear Unit</i>
RF	Floresta aleatória, do inglês: <i>Random Forest</i>

RNA	Rede Neural Artificial
SCADA	Sistema de Supervisão e Aquisição de Dados, do inglês: <i>Denoising Auto-Encoder</i>
SED	Detecção de Eventos Sonoros, do inglês: <i>Sound Event Detection</i>
SELU	Unidades Lineares Exponenciais em Escala, do inglês: <i>Scaled Exponential Linear Units</i>
SPL	Nível de Pressão Sonora, do inglês: <i>Sound Pressure Level</i>
SIN	Sistema Interligado Nacional
SSIM	Medida de Similaridade Estrutural, do inglês: <i>Structural Similarity Index Measure</i>
STFT	Transformada de Fourier de Curto Prazo, do inglês: <i>Short-Time Fourier Transform</i>
SVM	Máquina de Vetores de Suporte, do inglês: <i>Support Vector Machine</i>
t-SNE	Incorporação Estocástica de Vizinhos Distribuídos em t, do inglês: <i>t-distributed Stochastic Neighbor Embedding</i>
VAE	Autoencoder Variacional, do inglês: <i>Variational Auto-Encoder</i>

Lista de símbolos

ϕ, θ	Parâmetros do autoencoder utilizados na função de perda.
N	Número total de amostras no conjunto de dados.
$x^{(i)}$	i -ésima amostra de entrada original.
f_ϕ	Função de codificação parametrizada por ϕ .
g_θ	Função de decodificação parametrizada por θ .
$\hat{x}$	Versão ruidosa da amostra de entrada x .
z	Vetor latente resultante da codificação da entrada ruidosa.
y	Representação intermediária nas camadas ocultas do autoencoder.
W	Matrizes de pesos da camada de codificação.
W'	Matrizes de pesos da camada de decodificação.
b	Vetores de bias da camada de codificação.
b'	Vetores de bias da camada de decodificação.
q_D	Distribuição de ruído aplicada à entrada x para gerar $\hat{x}$.
L	Função de perda.
L_H	Função de perda entre baseada em entropia cruzada ou divergência KL.
H	Entropia cruzada.
B_x	Distribuição binária do dado original.
B_z	Distribuição binária do dado reconstruído.
$x^{(k)}$	Valor da k -ésima dimensão da amostra de entrada atual.
$x_i^{(k)}$	Valor da k -ésima dimensão da i -ésima amostra no batch.
m	Tamanho do batch (número de amostras no lote).
μ_{batch}	Média do batch na dimensão k
σ_{batch}^2	Variância do batch na dimensão k .

ϵ	Pequeno valor positivo adicionado para evitar divisão por zero.
$\hat{x}^{(k)}$	Valor normalizado da k -ésima dimensão, com média aproximadamente 0 e variância aproximadamente 1.
γ	Parâmetro de escala aprendível aplicado após a normalização.
β	Parâmetro de viés aprendível aplicado após a normalização.
$BN_{\gamma,\beta}$	Resultado da operação de Batch Normalization.
$y^{(k)}$	Valor final da normalização com escala e viés aplicados na dimensão k .
$\mathcal{L}$	Função de perda do VAE
θ	Parâmetros da rede neural do decodificador.
ϕ	Parâmetros da rede neural do codificador.
$\mathbb{E}$	Valor esperado.
$p_{\theta}(x z)$	Distribuição condicional dos dados x dados os vetores latentes z , definida pelo decodificador.
D_{KL}	Divergência de Kullback-Leibler.
$p(z)$	Distribuição a priori do espaço latente z , geralmente uma normal padrão.
$\mathcal{N}$	Distribuição normal.
$p_{\theta}(z)$	Distribuição do espaço latente parametrizada por θ .
$q_{\phi}(z)$	Distribuição aproximada do espaço latente segundo o codificador.
α	Coefficiente angular utilizado na função de ativação LeakyReLU.
$f(x)$	Função de ativação, utilizada para modelar a saída de uma unidade da rede.
SELU	Função de ativação, do inglês “ <i>Scaled Exponential Linear Unit</i> ”.
λ	Coefficiente de escala da função SELU.
μ_x	Média dos valores de intensidade luminosa da imagem x .
μ_y	Média dos valores de intensidade luminosa da imagem y .
σ_x	Desvio padrão dos valores de intensidade luminosa da imagem x .

σ_y	Desvio padrão dos valores de intensidade luminosa da imagem y .
x_i	i -ésimo valor de intensidade luminosa da imagem x .
$\bar{x}$	Valor de intensidade luminosa normalizado da imagem x .
N	Número total de amostras (pixels ou sinais) considerados.
M	Número total de elementos (sinais ou pixels) utilizados no cálculo da média ou variância.
$l(x,y)$	Função de luminância entre as imagens x e y .
$c(x,y)$	Função de contraste entre as imagens x e y .
$s(x,y)$	Função de estrutura entre as imagens x e y .
$SSIM(x,y)$	Índice de Similaridade Estrutural entre as imagens x e y .
C_1, C_2, C_3	Constantes de estabilização utilizadas para evitar divisão por zero.
k_1, k_2	Constantes escalares utilizadas no cálculo das constantes C_1 e C_2 .
L	Valor dinâmico máximo possível para os pixels da imagem (por exemplo, 255 para imagens em 8 bits).
α, β, γ	Pesos aplicados às funções de luminância, contraste e estrutura, respectivamente, na fórmula do SSIM.
PSNR	Razão de pico sinal-ruído (<i>Peak Signal-to-Noise Ratio</i>), em decibéis.
MSE	Erro quadrático médio (<i>Mean Squared Error</i>) entre duas imagens.
Max	Valor máximo possível de intensidade de pixel (255 para imagens em 8 bits).
$I(i,j)$	Valor de intensidade do pixel na posição (i,j) da imagem original (entrada).
$K(i,j)$	Valor de intensidade do pixel na posição (i,j) da imagem reconstruída (saída).
B_{superior}	Valor da banda superior de Bollinger.
B_{inferior}	Valor da banda inferior de Bollinger.
MMA	Média móvel aritmética dos dados da série temporal.
σ	Desvio padrão dos dados da série temporal.

k	Fator de sensibilidade que multiplica o desvio padrão.
n	Tamanho da janela deslizante utilizada para o cálculo da média e do desvio padrão.
B_{superior}	Valor da banda superior de Bollinger.
B_{inferior}	Valor da banda inferior de Bollinger.
MMA	Média móvel aritmética dos dados da série temporal.
Z_{score}	Valor padronizado obtido a partir da média e do desvio padrão da série, usado para identificar <i>outliers</i> .
$g(x)$	Função de decisão do SVM que realiza a classificação binária com base na combinação dos vetores de suporte, pesos, rótulos e função kernel.
a_i	Coefficiente de Lagrange associado à i -ésima amostra de treinamento no SVM.
y_i	Rótulo binário da i -ésima amostra de treinamento, podendo assumir valores -1 ou 1 .
$K(x_i, x)$	Função kernel que mede a similaridade entre o vetor de entrada x e a i -ésima amostra x_i no espaço transformado.
$\text{sign}(\cdot)$	Função sinal que retorna 1 se a entrada for positiva, -1 se for negativa, e 0 se for zero — usada para definir a classe de saída no SVM.

Sumário

1	INTRODUÇÃO	1
1.1	Motivação e Definição do Problema	4
1.2	Objetivos	6
1.2.1	Objetivo Geral	6
1.2.2	Objetivos Específicos	6
1.3	Contribuições e Publicações	7
1.4	Estrutura do Trabalho	7
2	FUNDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA	9
2.1	Fundamentação Teórica	9
2.1.1	Falhas em Máquinas	9
2.1.2	Aprendizado de Máquinas	9
2.1.3	Reconhecimento de Padrões em Sons	11
2.2	Revisão da Literatura	12
3	SOLUÇÃO PROPOSTA	16
3.1	Formulação do Problema	17
3.2	Hardware para Captura de Dados	19
3.3	Acesso aos Dados	22
3.3.1	Design do software	22
3.3.1.1	Componentes Gerais	23
3.3.1.2	Ingestão de Áudio	24
3.3.1.3	Indexação de Dados e Informações Sobre Aerogeradores	24
3.3.1.4	Agendamento de Processamento	25
3.3.1.5	Processamento do Audio	25
3.3.2	Banco de Dados	26
3.3.2.1	Levantamento das Entidades do Banco	26
3.3.2.2	Implementação do Banco	29
3.3.2.3	Disponibilização Online	31
3.3.3	Interface de Programação de Aplicações (API)	32
3.3.3.1	Implementação do Controle de Usuários e Permissão	32
3.3.3.2	Implementação dos Pontos de Acesso	34
3.3.3.3	Disponibilização Online	36
3.4	Redes Generalistas	37
3.4.1	Autoencoder Denoising	37

3.4.2	Autoencoder Variacional	40
3.5	Processamento de Dados	43
3.6	Treinamento das Redes Generalistas	47
3.7	Inferência e Validação	50
3.8	Suavização das Métricas	55
3.9	Classificação	59
4	APLICAÇÕES E RESULTADOS EXPERIMENTAIS	61
4.1	Testes com Ruídos Sintéticos	61
4.1.1	Caso 1: Máquina Normal x Ruídos	62
4.1.2	Caso 2: Máquina Normal x Ruídos PSNR	63
4.1.3	Caso 3: Máquina Normal x Ruídos SSIM	64
4.2	Separação de Máquina Defeituosa	64
4.2.1	Detecção de Falha no Rolamento	65
4.2.2	Detecção de Falha no DFIG	69
4.3	Implementação do Detector de Anomalias	71
4.4	Aplicação das Redes Especialistas	73
5	CONSIDERAÇÕES FINAIS	78
5.1	Trabalhos Futuros	79
	REFERÊNCIAS	80

1 Introdução

A energia eólica tem se consolidado como uma tendência crescente em todo o mundo, impulsionada principalmente por ser uma fonte limpa e sustentável. Um dos seus principais atrativos é a baixa emissão de poluentes, já que a geração de eletricidade por meio dos ventos não envolve a queima de combustíveis fósseis, evitando assim a liberação de gases nocivos à atmosfera, como o dióxido de carbono (CO_2).

Além disso, trata-se de uma fonte de energia considerada inesgotável, ao depender de um recurso natural amplamente disponível: o vento. Essa disponibilidade contínua torna a energia eólica uma alternativa viável e estratégica para diversificar a matriz energética e reduzir a dependência de fontes não renováveis, como o petróleo e o carvão mineral.

Dessa forma, a adoção da energia eólica representa um passo importante no combate às mudanças climáticas, podendo contribuir significativamente para a redução dos gases de efeito estufa na atmosfera. Ao investir nesse tipo de tecnologia, os países não somente promovem a preservação ambiental, mas também incentivam o desenvolvimento de soluções energéticas mais eficientes e de longo prazo.

Dados recentes do Global Wind Energy Council (GWEC) de 2025 reforçam essa tendência, indicando a instalação de 109 GW de nova capacidade onshore e 8 GW offshore apenas no último ano. Com isso, a capacidade acumulada mundial alcançou 1.136 GW, distribuída por todos os continentes, com a participação de 55 países na expansão do setor. Esse avanço é reflexo de um movimento global em direção a fontes de energia mais limpas e sustentáveis, alinhado aos compromissos internacionais de redução das emissões de carbono e à transição energética.

Um dos principais impulsionadores desse crescimento é a China, que se destaca como líder mundial na expansão da energia eólica. O país asiático tem a projeção de adicionar aproximadamente 458 gigawatts (GW) de nova capacidade instalada no setor eólico entre os anos de 2021 e 2030 (Wood Mackenzie, 2021). Esse número expressivo representa quase metade da expansão global prevista para o período, assim como evidencia o papel estratégico da China na consolidação da energia eólica como pilar fundamental da matriz energética mundial.

Esse cenário de crescimento reforça não apenas a viabilidade técnica e econômica da energia eólica, mas também sua relevância como instrumento de política ambiental e de desenvolvimento sustentável, promovendo segurança energética e contribuindo diretamente para os objetivos globais de combate às mudanças climáticas.

Em dados mais recentes do ano de 2024, o Brasil alcançou a quinta posição no Ranking de Capacidade Total Instalada de Energia Eólica *Onshore* chegando a uma capacidade total instalada de 33,7 GW atrás de China, EUA, Alemanha e Índia, respectivamente (ABEEólica, 2025). Estes dados refletem o crescimento e importância do país no cenário global de energias renováveis e, especificamente, eólica, o que demanda cada vez mais soluções focadas na inovação em supervisão e manutenção de seus componentes:

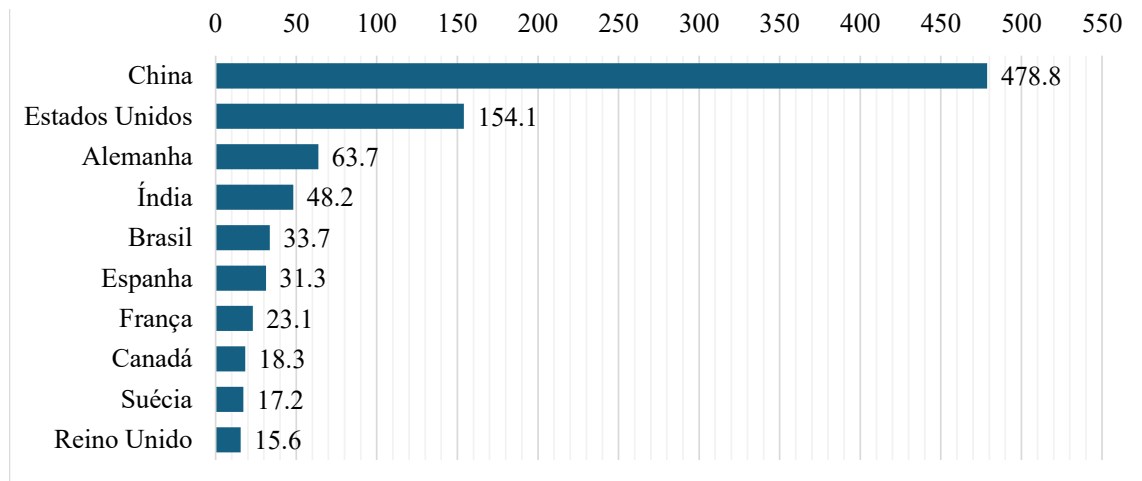


Figura 1.1 – Capacidade total instalada *onshore*. Adaptado de GWEC 2024.

Esta forma de geração de energia ocupa a segunda posição na matriz energética nacional com cerca de 13,2% da oferta interna, atrás somente da geração hidrelétrica (58,9%), como apresentado em dados na Figura 1.2 referente ao ano de 2023 (Ministério de Minas e Energia (MME), 2024). Recentemente, no mesmo ano de 2024, o Brasil se tornou o terceiro país no mundo que mais atraiu investimentos em energias renováveis (incluindo a energia eólica) segundo dados do relatório Energy Transition Investment Trends 2024, publicado pela BloombergNEF (BNEF), somando-se mais de US\$25 bilhões (Valor Econômico, 2024):

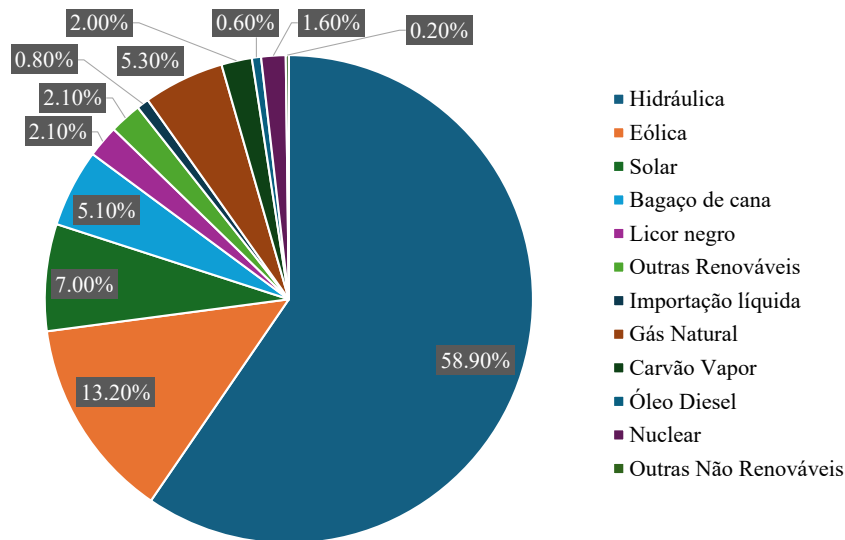


Figura 1.2 – Oferta interna de eletricidade do Brasil. Adaptado de MME 2024.

O investimento em energia eólica vai além da construção de parques eólicos e da infraestrutura necessária para a geração e distribuição de energia. Ele também engloba aportes significativos em tecnologia e inovação, com o objetivo de aprimorar continuamente os sistemas utilizados, aumentar a eficiência dos aerogeradores e garantir a durabilidade e a preservação das estruturas envolvidas. Dentre esses aportes, destacam-se os sistemas de supervisão remota, aos quais o presente trabalho se dedica.

O sistema eólico de geração de energia é composto por grandes estruturas conhecidas por aerogeradores que, através da energia cinética gerada pela corrente de ar, rotaciona as gigantes pás acopladas a um rotor atrelado a um mecanismo multiplicador de velocidade até chegar no gerador que converte a rotação entregue em energia elétrica. Estes equipamentos ficam alocados em uma estrutura conhecida como *nacelle*, a qual é a estrutura mais pesada de um aerogerador e funciona como uma estação de manutenção e operação do aerogerador onde se realizam as devidas correções, quando necessário (Windbox, 2020). A Figura 1.3 apresenta as partes constituintes do aerogerador:

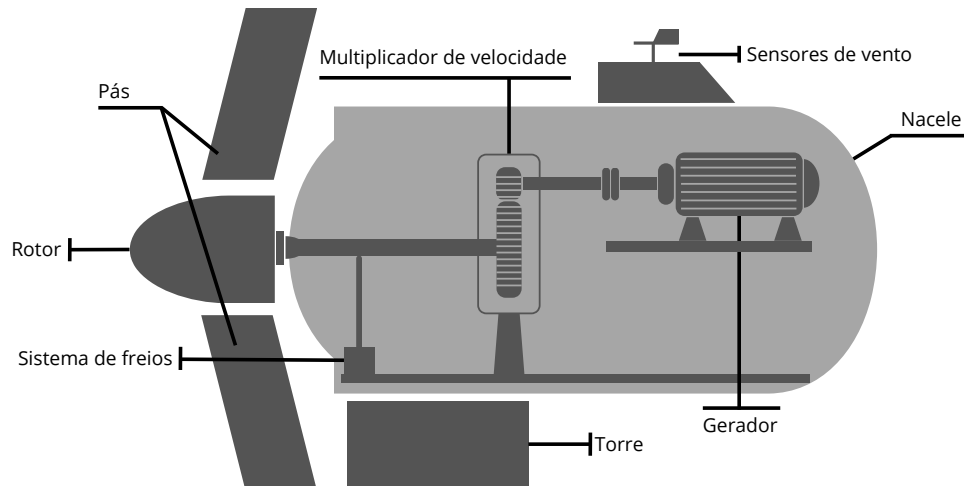


Figura 1.3 – Estrutura de um aerogerador e suas partes constituintes (Windbox, 2020).

1.1 Motivação e Definição do Problema

A ênfase em manutenção preditiva surge do fato de que, se uma tendência de anomalia não é detectada previamente, esta pode encadear em uma sequência de outras falhas que podem prejudicar outros componentes do mecanismo. Estes, por sua vez, podem até mesmo levar a uma pane no equipamento, ocasionando um maior tempo de parada e, consequentemente, em custos operacionais e desabastecimento de energia.

Segundo relatório emitido pela empresa Elera Renováveis (2024), observou-se uma tendência de queda no fator de disponibilidade das plantas eólicas ao longo dos últimos três anos. Essa redução, evidenciada na Tabela 1.1, pode estar associada a diversos fatores, como o aumento na frequência de falhas operacionais, manutenções corretivas não programadas ou até mesmo condições climáticas adversas. Tal cenário acende um alerta importante para a necessidade de estratégias mais eficazes de manutenção preditiva e monitoramento contínuo, a fim de garantir maior confiabilidade e eficiência na geração de energia eólica:

Tabela 1.1 – Fator de disponibilidade médio das plantas por fonte de energia e por regime regulatório (Elera Renováveis, 2024).

Ano	Hidrelétrica	Solar	Eólica	Total
2022	99,15%	98,66%	97,58%	98,65%
2023	99,27%	93,50%	97,35%	97,11%
2024	98,63%	93,36%	96,71%	96,49%

O tempo de parada pode acarretar prejuízos significativos para as empresas administradoras dos parques eólicos. De acordo com dados apurados do ABBEólica por

(Armando de Oliveira Lima, 2024), as eólicas cearenses perderam R\$ 104,5 milhões devido à interrupção na geração de energia durante o ano de 2024. Em termos de capacidade de energia, este prejuízo representa 867,98 mil MWh a menos no Sistema Interligado Nacional (SIN). Para fins comparativos, o consumo de energia no Brasil em novembro de 2024 somou 47,79 mil MWh.

O acesso restrito a estes componentes também é uma tarefa árdua na qual envolve o deslocamento de trabalhadores para elevadas alturas, requerendo uma preparação e equipamentos especializados para a execução das tarefas com segurança e precisão (BU *et al.*, 2023). Desta forma, nota-se uma importância de reduzir as regulares inspeções ao topo dos aerogeradores e, conseqüentemente, a otimização dos processos de manutenção com um diagnóstico prévio preciso e, especialmente, à distância.

Além dos desafios operacionais e de manutenção, é importante considerar também os impactos ambientais e sociais associados ao funcionamento dos aerogeradores. Eventualmente, estas máquinas podem emitir sons na faixa audível humana, o que pode ser perceptível por moradores em torno do parque eólico e, em alguns casos, prejudicial à sua saúde com sintomas de insônia, por exemplo (MOORHOUSE *et al.*, 2007; PEDERSEN; WAYE, 2007).

O aumento do Nível de Pressão Sonora (SPL) das turbinas eólicas eleva as chances de percepção em 30% e de incômodo em 10% pelo ruído (PEDERSEN; WAYE, 2007). Esses efeitos são mais intensos em áreas rurais, especialmente em terrenos complexos como colinas ou regiões rochosas, em comparação com áreas suburbanas e terrenos planos. Além disso, o incômodo está relacionado à visibilidade das turbinas (tanto objetiva quanto subjetiva) e se associa a piora na qualidade do sono e a emoções negativas.

Ainda neste contexto, sabe-se que sons incomuns também são perceptíveis pelos próprios operadores que estão presentes localmente. Alguns destes sons não são interpretados pelos sensores de vibração, frequentemente utilizados na supervisão desses equipamentos (WU *et al.*, 2019).

A automação industrial exerce um papel essencial na supervisão de parques eólicos, seja por meio de sistemas tradicionais, como o Supervisory Control and Data Acquisition (SCADA), seja por meio de técnicas baseadas em Inteligência Artificial (IA), foco do presente trabalho. No contexto da IA, destaca-se o aprendizado de máquina (*Machine Learning*), um ramo da computação em que o desempenho de um programa é aprimorado a partir do seu próprio processo de aprendizagem. Nesse contexto, os algoritmos de aprendizado de máquina são capazes de identificar padrões nos dados e, a partir deles, realizar previsões sobre a geração de energia e as condições operacionais dos equipamentos, sendo este último o foco do presente trabalho (ALI, 2023).

1.2 Objetivos

Os ruídos incomuns gerados pelos componentes são frequentemente correlacionados com falhas e/ou potenciais falhas. Baseado nisto, tem-se um horizonte a ser explorado, que é a detecção de falhas através da captura de sons oriundos das turbinas eólicas e alertar aos operadores quando uma anomalia ou sua iminência ocorra.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um produto capaz de detectar anomalias em aerogeradores utilizando eletrônica embarcada, periféricos acessíveis e arquiteturas de redes neurais artificiais. Dessa forma, a captura dos dados é realizada por meio do som (com microfones) e, utilizando processamento de imagem, interpretam-se os espectrogramas para a rede reconhecer padrões e, por meio de métricas, saiba distinguir uma condição desejável de uma anomalia.

Essa abordagem baseada em áudio se justifica por relatos de operadores sobre a presença de ruídos incomuns durante a operação dos aerogeradores. A partir dessa percepção prática, direcionou-se o foco do sistema para abranger a faixa audível ao ser humano, entre 0 e 20 kHz, que também é compatível com os microfones disponíveis no mercado.

1.2.2 Objetivos Específicos

Seguindo a definição do objetivo principal, este trabalho abordada os seguintes objetivos específicos:

1. Integração com *hardware* dedicado para captura de sons;
2. Desenvolver um algoritmo de detecção de anomalias em aerogeradores por meio do som;
3. Definição da anomalia via redes neurais artificiais especialistas e algoritmos de classificação;
4. Estudo do comportamento do espectrograma para diferentes condições.

1.3 Contribuições e Publicações

Esta dissertação de mestrado, desenvolvida com rigor acadêmico e embasada em pesquisa original, culminou na elaboração de um artigo que, após passar por um processo de revisão por pares, foi publicado na revista MDPI Sensors no dia 28 de fevereiro de 2025:

- FERREIRA DA SILVA, M.; NUNES MASSON, J.E.; SANTOS, M.F.d.; **RODRIGUES SILVA, W.**; WLADIMIR MOLINA, I.; MARTINS, G.M.C. *Audible Noise Evaluation in Wind Turbines Through Artificial Intelligence Techniques*. MDPI - Sensors, 2025, n° 5, p. 1492¹.

Um segundo artigo em periódico foi alcançado, tendo como base o desenvolvimento do hardware:

- MARTINS, G.M.C.; SANTOS, M.F.d.; SILVA, M.F.; NUNES MASSON, J.E.; SCHETTINO, V.B.; WLADIMIR MOLINA, I.; **RODRIGUES SILVA, W.** *Audible Noise-Based Hardware System for Acoustic Monitoring in Wind Turbines*. MDPI - Inventions, 2025, n° 58².

O terceiro artigo, por sua vez, foi elaborado na fase inicial do projeto e apresentado no dia 20 de maio de 2025 na conferência *26th International Carpathian Control Conference* (ICCC) que ocorreu em Starý Smokovec, na Eslováquia:

- **RODRIGUES SILVA, W.**; FERREIRA DA SILVA, M.; SANTOS, M.F.d.; NUNES MASSON, J.E.; WLADIMIR MOLINA, I.; MARTINS, G.M.C. *Preliminary Tests on a System for Monitoring Audible Noise from Wind Turbines with AI*. 26th International Carpathian Control Conference (ICCC), Starý Smokovec, High Tatras, Slovak Republic, 2025³.

1.4 Estrutura do Trabalho

Esta dissertação está organizada em cinco capítulos, conforme segue:

- **Capítulo 2:** Abordar o estado da arte e trabalhos relacionados com as tecnologias utilizadas, assim como cada uma destas se relacionam com o presente trabalho;

¹<<https://www.mdpi.com/1424-8220/25/5/1492>>

²<<https://www.mdpi.com/2411-5134/10/4/58>>

³<<https://ieeexplore.ieee.org/document/11022923>>

- **Capítulo 3:** Descreve a solução proposta para a detecção de anomalias nos aerogeradores, citando o dispositivo utilizado para a detecção de sons e detalhando algoritmos utilizados e o passo-a-passo para a estruturação;
- **Capítulo 4:** Apresenta os resultados dos testes funcionais aplicados em ruídos simulados, falhas conhecidas, construção do detector de anomalias e rotulação do tipo de falha;
- **Capítulo 5:** Apresenta a consolidação dos resultados da dissertação resumindo as principais contribuições, assim como sugestões de trabalhos futuros.

2 Fundamentação Teórica e Revisão Bibliográfica

O presente capítulo aborda as tecnologias pertinentes ao tema, bem como os trabalhos correlatos. A Seção 2.1 apresenta a fundamentação teórica trazendo os elementos presentes nos atuais sistemas de gerenciamento e detecção de anomalias nos sistemas em geral. A Seção 2.2 exibe trabalhos relacionados com o tema, bem como sua aplicabilidade para o presente trabalho.

2.1 Fundamentação Teórica

A presente seção apresenta o estado da arte na detecção de falhas em aerogeradores, abordando inicialmente, na Seção 2.1.1, as falhas mais recorrentes nessas máquinas. Em seguida, na Seção 2.1.2 são introduzidos os métodos tradicionais de detecção e as técnicas de aprendizado de máquina comumente aplicadas à detecção de anomalias. Por fim, na Seção 2.1.3, o contexto e os desafios no reconhecimento de padrões através do som.

2.1.1 Falhas em Máquinas

A caixa de engrenagens ou multiplicador de velocidades, além de ser o componente mais importante na performance e eficiência na geração de energia, é o componente considerado o mais suscetível a falhas, sendo responsável por aproximadamente 20% do tempo de parada dos aerogeradores, de acordo com estudos focados na geração eólica sueca (RIBRANT; BERTLING, 2007). Segundo JOOSSE *et al.*, falhas neste componente são responsáveis por 70% em pequena escala e 50% em grande escala do tempo de inatividade das turbinas eólicas. Essa alta taxa de falhas se dá devido à mesma estar sujeita às condições de cargas de ventos variáveis e impactos externos (HU; WANG; MA, 2023). Os atuais sistemas de supervisão de aerogeradores e máquinas rotativas, no geral, contam com sensores de vibração e de emissão acústica para o monitoramento de falhas nesse componente.

2.1.2 Aprendizado de Máquinas

O monitoramento *online* vem sendo cada vez mais implementado para a supervisão de várias máquinas, especialmente nos parques eólicos que possuem uma distância

considerável entre os aerogeradores. A detecção em tempo real aliada à capacidade de integração dos sensores via SCADA possibilita o monitoramento remoto de todas as unidades, eliminando a necessidade do operador estar presente na planta. Dessa forma, o SCADA possibilita uma interface e um sistema de alerta para os operadores para cada condição específica capturada em campo (PANDIT *et al.*, 2023).

Aliada aos sistemas supervisórios modernos, a inteligência artificial (IA) surge como uma ferramenta que pode se tornar indispensável para a predição de falhas, oferecendo respostas rápidas, alta precisão e tempos de antecedência confiáveis antes que as falhas ocorram (GRANADOS *et al.*, 2023; CHEN *et al.*, 2021; WANG; XIE; WANG, 2023). Modelos de IA, especialmente de aprendizado de máquina, são capazes de aprender a detectar condições anormais com base em padrões estatísticos presentes nos dados de treinamento e melhorar continuamente com a experiência (XIAO; CHENG; LI, 2021).

Nesse contexto, destaca-se uma área de grande relevância para a detecção preventiva de falhas: a análise e o processamento de dados por meio do aprendizado de máquina. Aplicada aos processos industriais, essa abordagem divide-se principalmente em métodos de aprendizado não supervisionado, voltados à identificação de padrões e anomalias sem rótulos prévios, e métodos supervisionados, que utilizam dados rotulados para treinar modelos preditivos.

A seguir, são descritos alguns dos principais métodos de aprendizado de máquina amplamente empregados em tarefas de classificação de falhas, destacando suas características fundamentais, formas de aplicação e relevância no contexto do monitoramento de máquinas e sistemas industriais.

Decision Tree (DT): Como o próprio nome sugere, trata-se de um algoritmo que adota uma estrutura em forma de árvore para descrever os relacionamentos entre diferentes variáveis e apoiar o processo de tomada de decisão. Sua principal vantagem reside na simplicidade de interpretação e na velocidade de execução, características que o tornam bastante atrativo em aplicações práticas. No contexto de classificação, o algoritmo funciona prevendo a categoria na qual um dado deve ser enquadrado. De acordo com WITTEN E. FRANK, cada nó interno da árvore representa um atributo específico do conjunto de dados, enquanto as folhas correspondem às classes finais de decisão, sendo que o nó localizado no topo da árvore, denominado raiz, estabelece o ponto inicial da análise e a sequência de decisões subsequentes.

Principal Component Analysis (PCA): O PCA consiste em um algoritmo que usa de uma transformação ortogonal para converter um conjunto de variáveis correlacionadas para variáveis linearmente não-correlacionadas. O objetivo principal é extrair o máximo de variações possíveis em um conjunto de dados semelhantes (JOLLIFFE, 2006). Essa técnica apresenta desafios devido à sua elevada sensibilidade, que pode tornar complexa

a distinção entre condições de falha e condições normais, sobretudo diante da diversidade e variabilidade dos dados em ambas as classificações.

SVM: O SVM é um método de aprendizado de máquina supervisionado, desenvolvido originalmente por VAPNIK, cuja ideia central consiste na construção de hiperplanos capazes de separar os dados em diferentes regiões. Dessa forma, no contexto de classificação, o método atribui rótulos aos dados de acordo com o agrupamento definido. Em contrapartida ao PCA, este lida tanto com dados linearmente separados quanto não lineares. O algoritmo identifica vetores de suporte e os utiliza para estabelecer fronteiras e separar os dados em grupos conhecidos como *clusters*. Existe uma variação desse método, denominada One-Class Support Vector Machine (OCSVM), que se destina especificamente à diferenciação entre condições normais e anômalas, além de possibilitar a detecção de dados desconhecidos, caso esses não tenham sido previamente aprendidos pelo algoritmo, o qual é treinado apenas com uma classe (conjunto de dados) correspondente à condição normal. Nesse contexto, por ser treinado em uma única classe, o OCSVM é considerado uma extensão não supervisionada do SVM (SCHÖLKOPF *et al.*, 2001).

Rede Neural Artificial (RNA): As redes neurais artificiais constituem uma classe de modelos inspirados no funcionamento das redes neurais biológicas, nas quais neurônios artificiais trocam informações entre si por meio de conexões interligadas. Cada conexão possui um peso associado, que é ajustado durante o processo de treinamento, permitindo o refinamento progressivo da rede e a melhoria de seu desempenho em tarefas específicas (BISHOP, 1995). Dentre elas, têm-se os autoencoders, um tipo de arquitetura de rede neural artificial que visa a reconstrução de dados por meio da compressão através da redução da dimensionalidade. Dentre suas utilizações principais, destaca-se a remoção de ruídos e detecção de falhas por meio do aprendizado não supervisionado (HE; JIN, 2021).

2.1.3 Reconhecimento de Padrões em Sons

A técnica de detecção de eventos sonoros, conhecida pela nomenclatura Sound Event Detection (SED) consiste em examinar um áudio e detectar a presença de eventos através de algoritmos de redes neurais artificiais. Esta técnica é amplamente utilizada tanto na etapa de pré-processamento de áudio, como é o caso de reconhecimento de fala como também visto em algumas aplicações como vigilância (RADHAKRISHNAN; DIVAKARAN; SMARAGDIS, 2005), monitoramento bioacústico (SALAMON *et al.*, 2016) e veículos autônomos (CHAVDAR *et al.*, 2020).

Trata-se de uma técnica cujo objetivo central é replicar, de forma artificial, a sensibilidade humana na percepção auditiva, permitindo a diferenciação entre distintos sons em um ambiente acusticamente complexo, no qual coexistem diversas fontes sonoras em operação simultânea (MESAROS *et al.*, 2021).

Esta técnica possui uma série de desafios a serem enfrentados devido à sobreposição de eventos sonoros e à dificuldade de se estruturar um conjunto de dados robusto para o aprendizado destes eventos, pois não há um conjunto de dados universal (YAN *et al.*, 2019).

2.2 Revisão da Literatura

Nesta seção, apresentam-se artigos relacionados ao embasamento teórico discutido anteriormente, com o objetivo de reforçar a relevância dos temas tratados e evidenciar sua importância para o presente trabalho. Para isso, é realizada uma breve exposição do conteúdo dos estudos, de seus resultados e de sua aplicabilidade no presente trabalho.

Sobre diagnóstico de falhas utilizando IA, o trabalho Carratù *et al.* (2023) utiliza uma abordagem não supervisionada que combina a técnica de clusterização, um método de agrupamento de dados com base em similaridades, com a análise de espectrogramas para identificar padrões incomuns que podem indicar anomalias.

Os resultados desta pesquisa demonstraram a separação de dados normais e anormais mesmo em datasets desbalanceados, o que enfatiza a eficiência da clusterização dos dados. O método de aprendizado não supervisionado utilizando autoencoders não apresentou falsos positivos e menos de 4% de *outliers* não detectados, demonstrando alta eficácia. Além disso, tem-se uma diminuição significativa dos falsos positivos e falsos negativos em comparação com outros métodos tradicionais, como o limiar adaptativo e o uso de autoencoder profundo.

Ainda neste sentido, a metodologia de aprendizado não supervisionado é aplicada na detecção precoce de falhas em aerogeradores do presente, onde a identificação inicial da presença de anomalia seria feita por redes autoencoders treinadas para reconhecer padrões. Na identificação de falhas, utilizam-se métodos de clusterização para agrupar os dados e auxiliar na definição das falhas, em conjunto com outras técnicas de agrupamento. Essa abordagem combinada de metodologias torna o sistema mais robusto e confiável.

O trabalho de Natili *et al.* (2021) aborda a metodologia SVM na predição de dados do comportamento do rolamento dos aerogeradores utilizando parâmetros do SCADA de temperatura e pressão do óleo. Com essas métricas, foi possível obter um aviso prévio para alertar o operador de uma anomalia. Uma análise de vibração também foi realizada com o auxílio de acelerômetros presos na carcaça da *gearbox* (caixa de transmissão), que é a parte do aerogerador responsável por multiplicar a velocidade de rotação das pás entregue ao aerogerador em baixo torque. Utilizando a técnica de separação por PCA, é possível agrupar os dados normais e anômalos.

Através da variável de pressão do óleo, o presente trabalho demonstrou um enorme

potencial de diagnóstico, podendo ser sinalizado por volta de 5 meses de antecedência da parada do aerogerador. Quanto à análise de temperatura, a falha pôde ser isolada com 4 meses antes da parada. Pela análise da técnica de PCA, foi constatado que mais de 70% dos pontos foram identificados corretamente como condições anômalas.

Essa metodologia pode contribuir para a classificação das anomalias tendo uma abordagem similar à técnica PCA. Neste contexto, ao invés da utilização dos parâmetros do sistema SCADA, o modelo proposto neste trabalho priorizou a utilização do som como a fonte primordial e completa de informações a serem extraídas. Além disso, os parâmetros do sistema SCADA contribuirão para a preparação dos dados de treinamento.

O trabalho de An e Cho (2015) apresentou o uso do VAE para detecção de anomalias, em que o processo de treinamento foi feito com dados que representam as condições normais. O princípio era avaliar a qualidade de reconstrução de imagens: se a imagem de saída (reconstrução) fosse mais próxima da de entrada, significava que os dados de entrada se assemelhavam ao conjunto de dados que a rede foi treinada. Logo, era considerado um dado representativo de condição normal. Caso houvesse uma má reconstrução do dado, refletia em uma possível falha.

O uso de IA em predições em aerogeradores é abordado no trabalho (RASHID; HAIDER; BATUNLU, 2020), para prever a potência de saída produzida pelo equipamento. A coleta de dados é realizada com o auxílio de três parâmetros do SCADA: direção e velocidade do vento e temperatura. Ao efetuar um treinamento com dados de dois anos, o modelo pôde reconhecer os padrões das métricas e prever a potência de saída do aerogerador.

O erro absoluto médio estimado para o modelo proposto para os fatores de capacidade 0,4 e 0,2 é de 3,6 e 7,3%, respectivamente, refletindo um resultado satisfatório com baixo erro na predição de potência.

Embora esta medição específica não seja o foco central deste trabalho, é relevante destacar a aplicabilidade do aprendizado de máquina na coleta de dados do aerogerador. Neste contexto, foi utilizado o sistema SCADA como referência para a predição de dados futuros de geração. Por outro lado, a proposta deste projeto é utilizar o SCADA como um recurso auxiliar na análise do comportamento das variáveis, com a perspectiva de evolução para um sistema autônomo, capaz de aprender padrões de ruídos anômalos e sinalizá-los automaticamente à operação, conforme os princípios do *machine learning*.

Logo, nota-se que as imagens podem ser consideradas elementos de maior representação de dados utilizadas em modelos de aprendizado de máquina. O processamento de imagem é aplicado em diferentes áreas como a medicina e até em processos industriais. A partir da criação de uma base de dados para ensinar o modelo, o pré-processamento para eliminação de ruídos e segmentação para análise temporal, é possível realizar a extração

de características relevantes (BI, 2024; PUTTAGUNTA; RAVI, 2021; KHAGI; KWON; LAMA, 2019).

Semelhantemente, o trabalho realizado por Zhang *et al.* (2022) aborda um método de supervisão baseado no comparativo entre os dados de reconstrução e de entrada de uma rede autoencoder para a detecção de falhas de folhas de papel alumínio em uma linha de produção. Métricas como Structural Similarity Index Measure (SSIM) e Peak-Signal Noise Ratio (PSNR) são utilizadas para avaliar a qualidade da reconstrução e, desta forma, balizar uma condição desejável e indesejável do produto final sem a necessidade de rotular os elementos para o modelo.

Os resultados desta aplicação apresentam uma acurácia 4% maior do que os demais métodos tradicionais comparados, sinalizando um potencial desta forma de aprendizado capturar detalhes a partir de imagens do objeto de interesse a ser analisado.

Similar ao trabalho anterior, esta abordagem também contribui ativamente aplicada no presente trabalho no que tange ao processamento de espectrogramas dos aerogeradores através do treinamento da rede para reconhecimento de padrões normais e utilização das métricas em questão para quantificar a qualidade de reconstrução e, consequentemente, obter-se um diagnóstico.

A respeito da classificação de áudio, espectrogramas são elementos-chave na análise deste tipo de dado, ao oferecer uma análise em tempo, frequência e amplitude em uma única imagem bidimensional. Desta forma, a classificação de áudio passa pelas etapas de processamento de imagem, comumente vistas em algoritmos de reconhecimento de padrões, transferindo a tarefa de análise do domínio do áudio para o domínio visual (BANSAL; GARG, 2022).

O recurso da imagem do espectrograma é gerado a partir da Transformada Rápida de Fourier (FFT) aplicada em sucessivas janelas de tempo de um sinal de amplitude do áudio. Esse processo, conhecido como Short-Time Fourier Transform (STFT), permite capturar a variação das componentes de frequência ao longo do tempo, produzindo uma matriz bidimensional onde o eixo horizontal representa o tempo, o eixo vertical representa a frequência, e a intensidade de cada ponto corresponde à magnitude do espectro em determinado instante. Essa representação facilita a análise visual e computacional de padrões acústicos, sendo amplamente utilizada em tarefas como classificação de áudio, reconhecimento de fala e detecção de eventos sonoros.

Neste contexto, o método desenvolvido por Zhang, McLoughlin e Song (2015) consiste em extrair parâmetros de espectrogramas utilizando Convolutional Neural Network (CNN) para reconhecimento de eventos. Ao longo do trabalho, é demonstrada a robustez deste método que pode alcançar por volta de 94% de acurácia média para diferentes níveis de ruído (limpo, 20 dB, 10 dB e 0dB).

Os resultados demonstram a capacidade de reconhecimento da rede em termos de resolução da frequência do espectrograma, bem como sua robustez frente à presença de ruídos, que atuam como fator de penalização no desempenho.

Para o processamento dos espectrogramas, utilizam-se autoencoders convolucionais que incorporam os princípios das CNN por meio de camadas convolucionais (CHOI; FAZEKAS; SANDLER, 2017; MASCI *et al.*, 2011). Na classificação das falhas, é fundamental contar com a robustez dessas redes aliada a algoritmos de clusterização, permitindo a identificação e categorização dos diversos tipos de falha que o aerogerador pode apresentar com base em seu comportamento sonoro.

Aliado às técnicas de clusterização dos padrões de funcionamento de aerogeradores, emprega-se o algoritmo preditivo do Filtro de Kalman (conforme discutido por Kim *et al.* (2023) na detecção de falhas em motores de indução) como uma estratégia para aumentar a confiabilidade do sistema, ao permitir a identificação precoce de desvios anômalos no comportamento operacional.

Neste trabalho, o método proposto identifica desvios de padrão nas métricas de reconstrução por meio de parâmetros estatísticos. A partir da concordância entre esses desvios, um detector de anomalias é acionado, notificando o sistema sobre possíveis irregularidades no comportamento analisado.

Sheibat-Othman *et al.* (2013) aborda os conceitos do Filtro de Kalman em conjunto com o algoritmo SVM, aplicando-os especificamente para predição de dados e detecção de falhas em aerogeradores onde se capturam dados relativos à vibração da máquina no tempo referente ao fator de amortecimento e frequência natural.

Para falhas que ocorrem especificamente em curtos períodos (25 segundos, conforme o caso estudado) ocasionadas por vibrações anômalas instantâneas, o Filtro de Kalman mostrou-se mais eficiente, enquanto o SVM, neste contexto, requer uma maior quantidade de dados para um desempenho adequado.

No presente trabalho, foram utilizadas médias móveis de curto e longo prazo para capturar os diferentes tipos de falhas a fim de se obter um prognóstico mais robusto ao decorrer do tempo das métricas de similaridade de reconstrução dos espectrogramas, ao invés do monitoramento de vários sensores acoplados ao aerogerador.

3 Solução Proposta

Neste capítulo são descritas, de forma organizada, todas as etapas que compõem a construção do algoritmo de detecção de falhas em aerogeradores. Na Seção 3.1, apresenta-se a formulação do problema, acompanhada de uma visão geral da solução proposta. A Seção 3.2 dedica-se à caracterização do dispositivo utilizado para a captura dos áudios, com breve descrição e referenciamento. Em seguida, a Seção 3.3 aborda o design de software, a estrutura do banco de dados e a API implementada. Na Seção 3.4, apresentam-se as redes generalistas utilizadas. Já a Seção 3.5 descreve as etapas de processamento de dados, que são seguidas, na Seção 3.6, pelo treinamento das redes. A Seção 3.7 contempla os procedimentos de inferência e validação, enquanto a Seção 3.8 trata da suavização da análise temporal. Por fim, a Seção 3.9 apresenta o processo de classificação de falhas.

Foram monitorados cinco aerogeradores do Complexo Eólico Faísa, localizado na Fazenda Faísa, no município de Trairi, Estado do Ceará, onde, por meio do dispositivo para captura de áudio (tratado na Seção 3.2) individualmente instalado em cada equipamento. Os aerogeradores são identificados pelos códigos: FAI2-07, FAI2-12, FAI1-05, FAI1-07 e FAI4-07.

A Figura 3.1 apresenta uma visão macro do projeto, que se inicia com a captura dos áudios, passa pelas etapas de processamento e culmina no servidor, onde estão alocadas as redes neurais artificiais responsáveis pela detecção de eventos. Esses eventos geram notificações, que podem ser enviadas por e-mail, SMS, *push* e também visualizadas por meio de uma aplicação web:

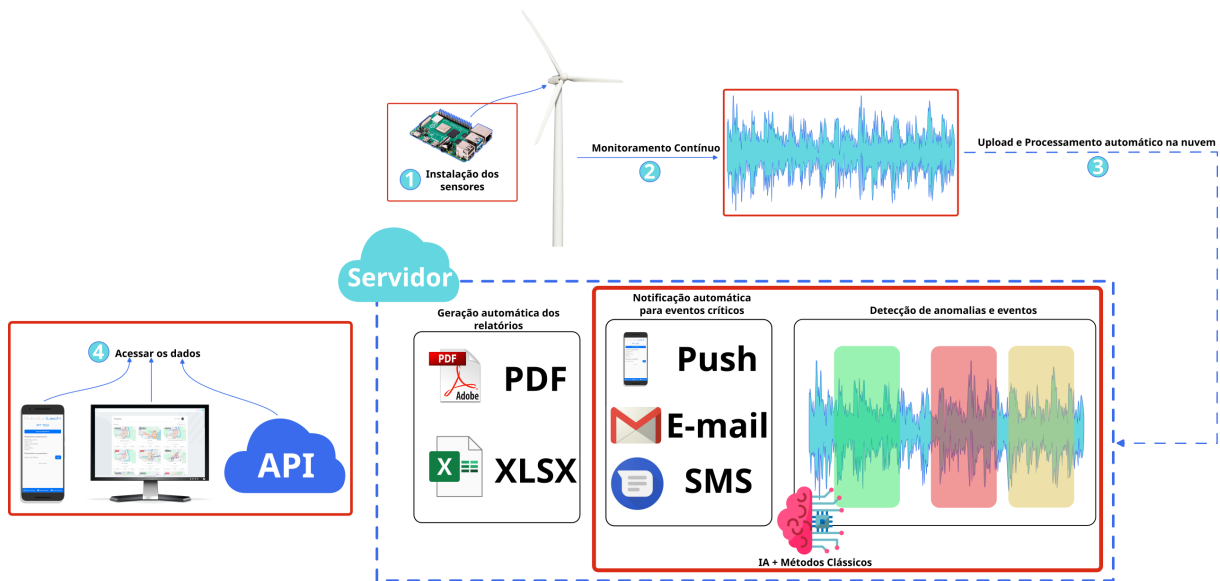


Figura 3.1 – Fluxo geral da solução.

O escopo do presente trabalho está centrado na detecção de anomalias e eventos em sistemas de geração eólica, com foco na análise e interpretação de dados acústicos por meio de técnicas de inteligência artificial. Os detalhes dessa abordagem serão explorados adiante, abrangendo desde a coleta dos dados até o processamento e a classificação dos eventos.

3.1 Formulação do Problema

Inicialmente, para a obtenção de áudios de aerogeradores, foi elaborado um *hardware* para tal propósito de forma que o mesmo cumpra requisitos de robustez e conectividade, buscando o envio constante de áudios para análise e inferência dos modelos de IA.

Para avaliar se determinado dado se trata de uma condição normal ou anormal, é necessário conhecer o espectrograma para cada cenário. Sabendo que, na maior parte do tempo, as turbinas operam normalmente, tem-se o desafio de construir um conjunto de dados rotulado de falhas para uma rede supervisionada devido à baixa quantidade de dados para treinar o modelo.

Por este motivo, é abordado o aprendizado não supervisionado baseado em redes autoencoder, que possui uma arquitetura capaz de reconstruir com considerável precisão os padrões de dados para os quais foi treinada. Dados fora deste escopo terão uma reconstrução de baixa similaridade com o dado de entrada. Além disto, autoencoders uti-

lizam camadas de compressão extraindo características importantes, retornando na saída decodificada uma predição do dado original.

A fim de orientar o passo-a-passo da detecção, a Figura 3.2 demonstra por meio de um fluxograma o caminho para a elaboração do algoritmo de captura de falhas de aerogeradores utilizando áudios gravados que começa desde a captura de áudios armazenados no servidor até a definição da falha:

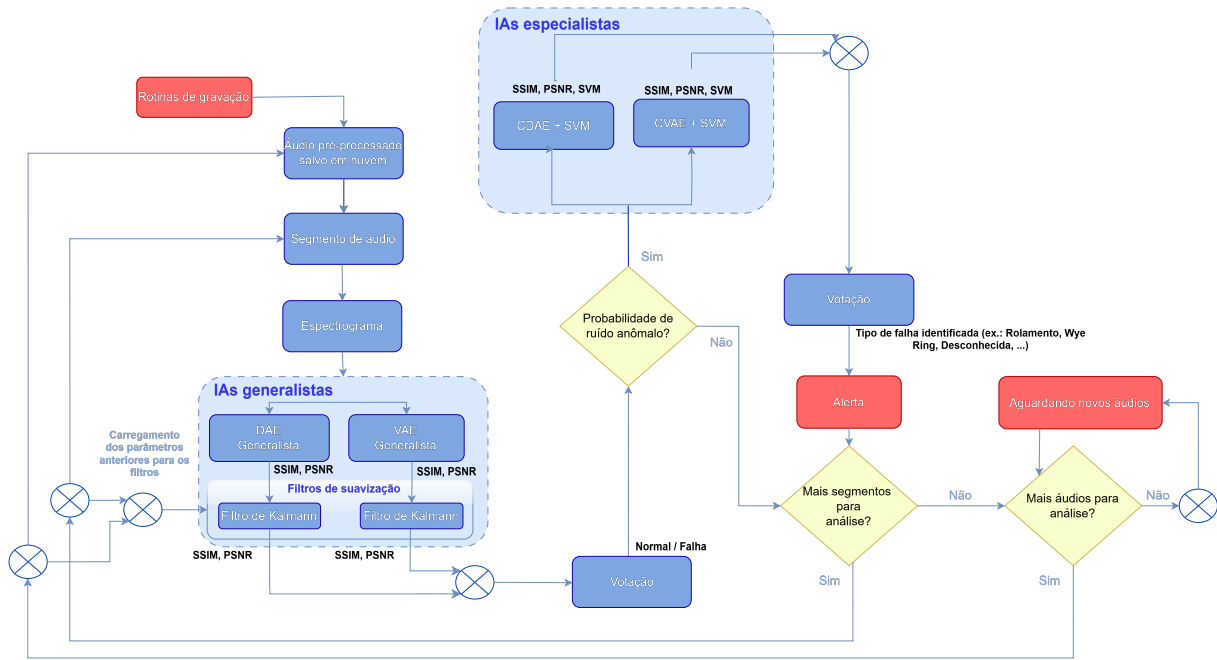


Figura 3.2 – Fluxograma do algoritmo de detecção.

O processamento tem início com a captura dos áudios provenientes do *hardware* dedicado para a gravação (brevemente abordado na Seção 3.2), seguida pela segmentação dos sinais e posterior geração dos espectrogramas. O processo de manipulação de áudio está detalhado na Seção 3.5 .

Em seguida, esses espectrogramas são analisados por redes generalistas, responsáveis por identificar e classificar as condições como normais ou anômalas. Estas IAs são constituídas pelas redes autoencoder abordadas na Seção 2.1.2.

Após o retorno das métricas pelas redes, aplica-se o Filtro de Kalman, conforme descrito na Seção 3.8. para suavizá-las. Esse filtro, por meio de componentes estatísticos, contribui para a definição da condição operacional da máquina, constituindo o detector de anomalias.

Após distinguir e separar as falhas, os segmentos dos espectrogramas são enviados para as redes especialistas tomarem a decisão e retornarem a falha rotulada na saída, tratada na Seção 3.9.

As etapas de votação realizadas após cada conjunto de IAs serão ponderadas segundo o peso individual atribuído a cada rede autoencoder. A definição desses pesos, no entanto, não é abordada neste trabalho.

A Tabela 3.1 apresenta uma visão geral das etapas do pipeline, destacando a saída gerada em cada fase e o respectivo tipo de dado produzido ao longo do processo de transformação e análise do áudio:

Tabela 3.1 – Pipeline de processamento de áudio e transformação de dados.

Etapa	Saída	Tipo de Dado
Aquisição pré-processado	Áudio pré-processado	.mp3 ou .wav
Segmentação	Segmentos de áudio	.mp3 ou .wav
Espectrograma	Imagem de espectrograma	.png
Normalização	Matriz normalizada	Array NumPy
DAE/VAE	Métricas de similaridade (SSIM, PSNR)	Numérico
Filtro de Kalman	Métricas suavizadas (SSIM, PSNR)	Numérico
Votação	Condição da máquina (Normal / Falha)	Booleano
CDAE/CVAE + SVM	Tipo de falha identificada	Texto

3.2 Hardware para Captura de Dados

O *hardware* foi construído a partir de componentes de mercado visando o custo-benefício, robustez contra sujidade e, principalmente, a conectividade (inicialmente) para enviar em regime constante de áudios via API para armazenamento em nuvem.

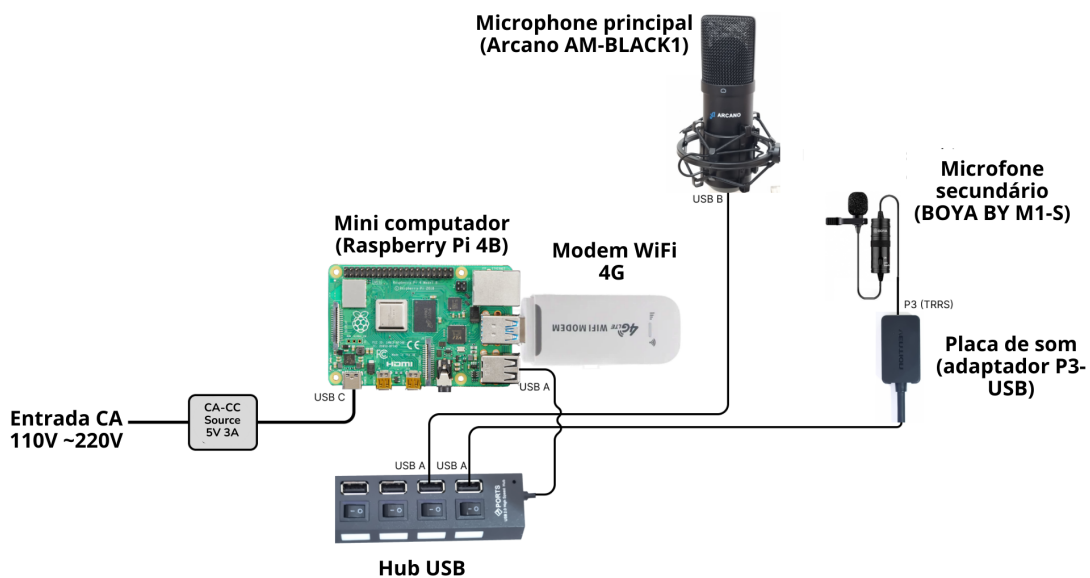


Figura 3.3 – Esquemático de montagem do hardware de captura de áudio.

Segue abaixo os componentes utilizado em sua montagem e sua respectiva funcionalidade:

- **Fonte conversora CC-CA:** converte a tensão alternada da distribuição para alimentar o mini computador com a tensão adequada;
- **Mini computador:** o microprocessador Raspberry Pi é responsável pela execução das rotinas de gravação. Através de seus periféricos, é possível conectá-lo à internet. Trata-se de um dispositivo portátil, de baixo custo, com desempenho adequado à tarefa e com uma grande comunidade de suporte.
- **Modem WiFi 4G:** permite que o *Raspberry* se conecte à internet para a execução de tarefa de envio dos áudios para a nuvem;
- **Hub USB:** dispositivo capaz de multiplicar as entradas de conexão USB do Raspberry Pi permitindo a conexão de mais de um microfone e eliminar interferências na gravação;
- **Microfone principal:** microfone externo de cabo extenso para a captura do som oriundo dos componentes do aerogerador. Sua flexibilidade, dado o tamanho do cabo, permite posicioná-lo próximo aos componentes. O modelo escolhido foi o ARCANO AM-BLACK-1 que se destacou no quesito sensibilidade dentre outros microfones pesquisados.
- **Microfone secundário:** oferece redundância para a gravação de áudio em caso de falha no microfone principal, permitindo a captura de áudio em condições adversas, já que se trata de um microfone de lapela que se encontra instalado no interior do compartimento. O modelo BOYA BY M1-S escolhido contém uma saída P3 adaptada para a conexão USB
- **Placa de som:** converte a saída P3 do microfone secundário para uma saída USB, permitindo a conexão com o HUB USB.

Estes componentes foram encapsulados em uma caixa plástica hermética que se divide em dois níveis: o primeiro com os componentes mencionados para o acesso rápido; o segundo para abrigar os cabos e conexões para manter o protótipo esteticamente organizado. Nisto, os componentes são protegidos contra ações do tempo como poeira. A Figura 3.4 apresenta os componentes devidamente organizados.

As rotinas relacionadas ao funcionamento do hardware, como gravação e envio de áudios, são abordadas por Martins (2025), que explica detalhadamente o aspecto construtivo e operacional do dispositivo. Desta maneira, o presente trabalho faz uso do dispositivo

para auxiliar na utilização dos áudios reais de aerogeradores na parte de treinamento e inferência das IAs.

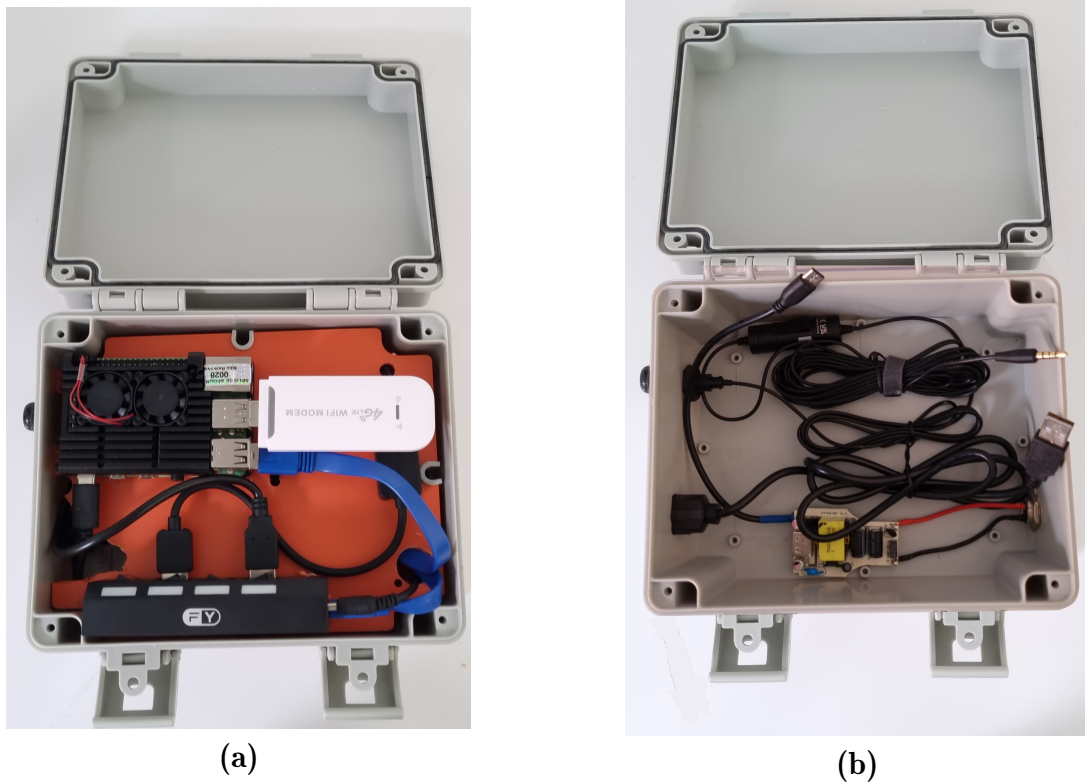


Figura 3.4 – Vista superior do primeiro protótipo do dispositivo gravador de áudio: **(a)** com os componentes à mostra. **(b)** mostrando os cabos e conectores.

Vale ressaltar que um dos atributos do dispositivo, incluso em sua rotina, é possuir um limiar de gravação implementado, o que interrompe ou segmenta automaticamente os períodos de silêncio do aerogerador (em baixa rotação ou parado) que podem ser interpretados equivocadamente pelas redes neurais como uma condição anômala.

Na Figura 3.5 é apresentada uma ilustração detalhada da localização do dispositivo no interior da nacela do aerogerador:

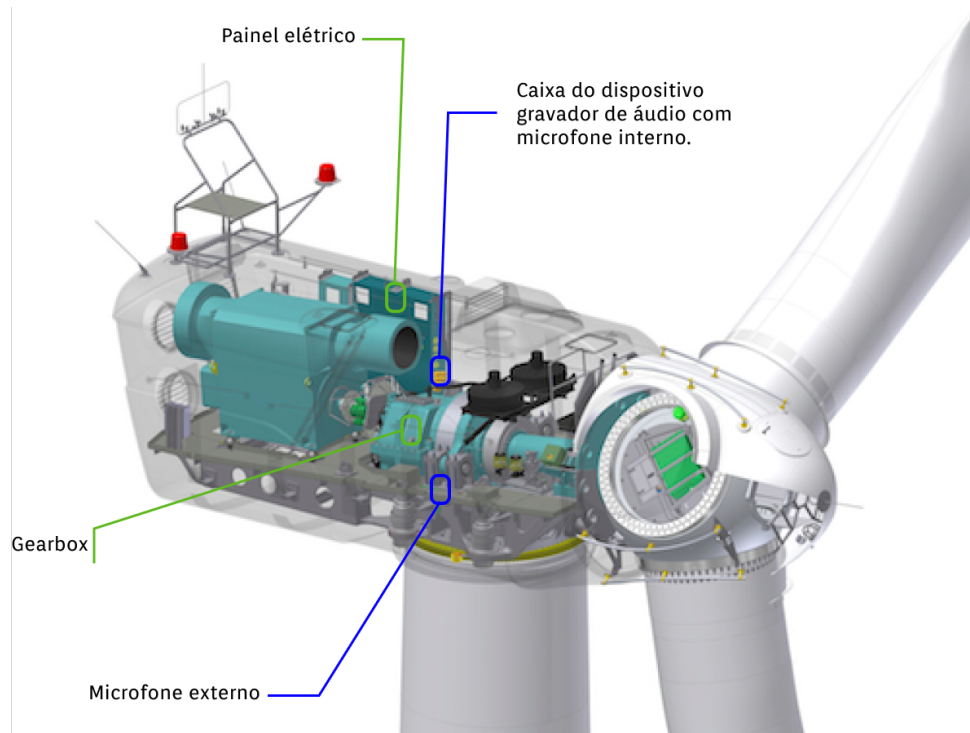


Figura 3.5 – Localização do dispositivo gravador de áudios no aerogerador.

3.3 Acesso aos Dados

Esta etapa consiste em definir uma arquitetura que organize o fluxo de dados do *hardware* para o sistema, mapeando as funções executadas pelo mesmo e garantindo que as informações armazenadas no servidor atendam às necessidades de acesso do usuário.

Primeiramente é definido um *design* de *software* que abrange a parte de banco de dados e os pontos de acesso, que são tratados posteriormente em subseções dedicadas.

3.3.1 Design do software

Para a organização e como uma prática recomendada no desenvolvimento de *software*, é essencial possuir um panorama claro da arquitetura que delineia o fluxo de dados da aplicação. A arquitetura descrita nesta seção será responsável pela ingestão, indexação e processamento dos áudios gravados pelos *hardwares* instalados nos aerogeradores. A Figura 3.6 representa a arquitetura completa e as dependências entre todas as suas partes.

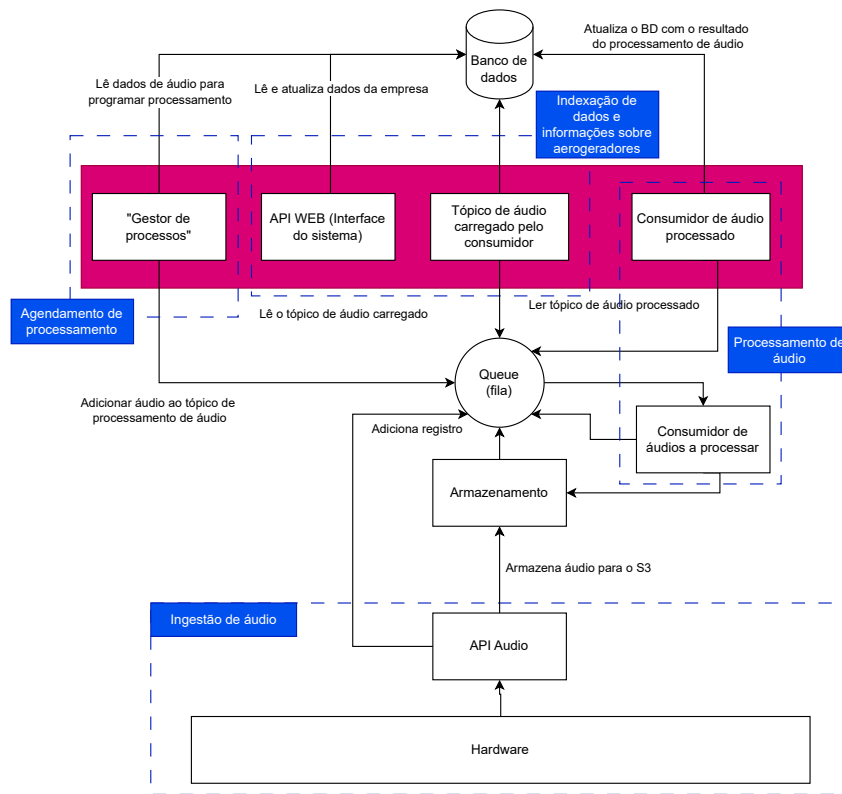


Figura 3.6 – Diagrama simplificado do controle de acesso ao servidor.

A arquitetura em questão inclui todos os componentes necessários para ingestão de áudio ao processamento e visualização de resultados. A comunicação entre as diferentes partes é dada pelo conceito de *“event-driven”* onde serviços não se comunicam diretamente, mas por filas, onde eventos representando o resultado de uma execução são adicionados e processados por outras partes da aplicação à medida do possível. Esta abordagem permite que todos os componentes trabalhem individualmente e possam ser escalados a depender da demanda. Além disso, esta arquitetura permite também a evolução e manutenção individual das diferentes partes.

3.3.1.1 Componentes Gerais

Várias partes da arquitetura utilizam soluções disponibilizadas pela Amazon Web Services (AWS). Estas tomam vantagem de vasta disponibilidade e fácil escalabilidade, e representam as seguintes partes/soluções:

- **Queue:** Representa uma fila. Mensagens são adicionadas e mantidas nesta fila até que uma outra aplicação possa consumir essa mensagem. Este padrão remove a dependência entre aplicações, colocando a carga sob a fila. Filas são facilmente esca-

láveis e altamente disponíveis. Para esta arquitetura, o Amazon SQS será utilizado como tecnologia de fila.

- **Armazenamento:** O S3 é um servidor de arquivos altamente disponível de baixo custo. Com S3 os arquivos de áudio serão armazenados para futuro processamento.
- **Execução das aplicações:** O Fargate será utilizado para execução das aplicações. Fargate permite a execução de aplicações em infraestrutura provisionada dependendo das necessidades do usuário, facilitando o processo de implantação das aplicações..
- **Banco de dados:** Para o banco de dados, o Amazon RDS será utilizado. O Amazon RDS permite a execução de bancos de dados com armazenamento e recursos escaláveis. O banco utilizado para esta aplicação era Postgresql SQL pela sua facilidade de uso e grande adoção pela comunidade de desenvolvimento. Este tópico é abordado com detalhes na Seção 3.3.2.2.

3.3.1.2 Ingestão de Áudio

Esta parte representa a parte direita inferior do diagrama, agrupada pelo retângulo tracejado em azul com a mesma descrição. Ela é responsável pela extração e armazenamento dos áudios.

Os componentes desta parte são os seguintes:

- **Hardware:** *Hardware* responsável pela gravação e envio de áudios para API Audio;
- **API Audio:** Simples API Rest para recebimento de áudios. Esta API armazena os dados num servidor de arquivos utilizando a data de gravação e o usuário (*hardware*) enviando o áudio para organização dos arquivos. Ao fazer o *upload* do arquivo para o S3, uma notificação é enviada automaticamente para a fila, informando que o arquivo está disponível no servidor de arquivos. Esta parte é executada no Fargate.

3.3.1.3 Indexação de Dados e Informações Sobre Aerogeradores

Esta parte representa a parte superior do diagrama da Figura 3.6, agrupada pelo retângulo tracejado em azul com a mesma descrição da subseção. Esta parte do diagrama é responsável pela indexação dos dados para futuro processamento e também recebe e armazena as informações sobre os aerogeradores.

Os componentes desta parte são seguintes:

- **Tópico de áudio carregado pelo consumidor:** Aplicação responsável por consumir as informações de dados carregados ao servidor de arquivos e adicionados à fila. Esta

aplicação armazenará metadados dos dados e o respectivo caminho do servidor de arquivos para busca e execução no banco de dados. Esta aplicação será executada no Fargate;

- **API Web:** Esta aplicação será uma API Rest responsável pela leitura e armazenamento de informações sobre os aerogeradores no banco de dados. Por meio desta API será também possível o download dos áudios e seus metadados. Esta API está unificada em um mesmo endereço juntamente com a API de áudio na qual é tratada com mais detalhes na subseção 3.3.3 sendo também executada no Fargate.

3.3.1.4 Agendamento de Processamento

Esta parte representa a parte superior direita do diagrama. Ela é responsável pelo agrupamento das informações necessárias para o processamento e agendamento.

O *Scheduler/Trigger Processing* é uma simples aplicação que fará a leitura das informações do banco de dados e as agrupará para o agendamento do processamento de dados. O agendamento é adicionado à fila e é processado eventualmente, quando a parte responsável pelo processamento possuir capacidade. Esta aplicação será executada no Fargate.

3.3.1.5 Processamento do Audio

Esta parte representa a parte inferior direita do diagrama. Ela é responsável pelo processamento de dados em busca de anomalias e pelo armazenamento das possíveis anomalias no banco de dados.

Os componentes desta parte são seguintes:

- **Consumidor de áudios a processar:** Aplicação responsável por executar os processos previamente adicionados à fila. Esta aplicação utiliza os metadados, combinados com as informações adicionais dos aerogeradores, faz o download dos arquivos do servidor de arquivo e os analisa em uma rede neural. Os resultados são então adicionados novamente à fila. Esta aplicação será executada no Fargate;
- **Consumidor de áudio processado:** Aplicação responsável por indexar os resultados do processamento de áudio no banco de dados para leitura e busca. Esta aplicação será executada no Fargate.

3.3.2 Banco de Dados

O banco de dados desempenha um papel fundamental na organização e manipulação eficiente das informações coletadas pelos sensores e processadas nas rotinas do *hardware*, permitindo análises precisas e a geração de relatórios relevantes para o monitoramento do desempenho e manutenção dos equipamentos. Como etapa inicial da construção do banco de dados, estruturaram-se, por meio do Diagrama de Entidade-Relacionamento (DER), as tabelas, campos e relacionamentos necessários para armazenar e gerenciar os dados do sistema de monitoramento de aerogeradores.

O DER servirá como uma representação visual da estrutura do banco de dados, destacando as entidades principais do sistema, seus atributos e os relacionamentos entre elas. A partir desse diagrama, será possível definir e criar as tabelas necessárias no banco de dados, identificando as chaves primárias e estrangeiras que garantirão a integridade dos dados e a eficiência das consultas.

3.3.2.1 Levantamento das Entidades do Banco

Para realizar o levantamento das entidades do banco de dados, uma estratégia comumente utilizada é partir dos casos de uso do sistema. Isto permite mapear grande parte das ações que o usuário deverá realizar e, portanto, quais dados devem ser armazenados no banco de dados para permitir que estas ações ocorram. Sabendo disso, esta foi a primeira atividade realizada para esta tarefa: levantar as ações que poderiam ser realizadas no sistema. O resultado foi a lista descrita abaixo.

- Funções do hardware (API Áudio):
 - Envia áudio para servidor;
 - Faz cache do áudio caso não tenha conexão;
 - Possui N microfones;
 - Conectar automaticamente ao servidor de SSH.
- Funções do usuário (API WEB):
 - Listar todos os hardwares;
 - Listar os hardwares com base no aerogerador;
 - Listar os hardwares com base na regional;
 - Listar os hardwares sem aerogerador;
 - Listar os hardwares com base na hora do último envio;

- Ver qual foi a última vez que o hardware enviou um dado;
- Listar todos os eventos que ocorreram em uma regional;
- Listar todos os eventos que ocorreram em um aerogerador;
- Listar os eventos com base em criticidade;
- Listar os eventos com base na data/hora da ocorrência;
- Procurar por eventos com descrição específica;
- Procurar por áudios de uma data/horário específicos;
- Procurar por áudios com base no aerogerador;
- Criar uma regional;
- Criar um aerogerador;
- Relacionar hardware com gerador específico;
- Deletar evento;
- Criar um evento, selecionando um período com início e fim em um áudio específico, adicionar criticidade e descrição;
- Modificar um evento;
- Marcar um evento como errado;
- Gerar relatórios com os eventos, com links para as faixas específicas;
- Criar/Excluir relatórios, selecionando o período em que o relatório será gerado;
- Criar relatório por equipamento, aerogerador ou regional.

Partindo desta lista, foram elencadas as entidades necessárias, as suas propriedades e o relacionamento entre elas que resulta no diagrama ilustrado na Figura 3.7:

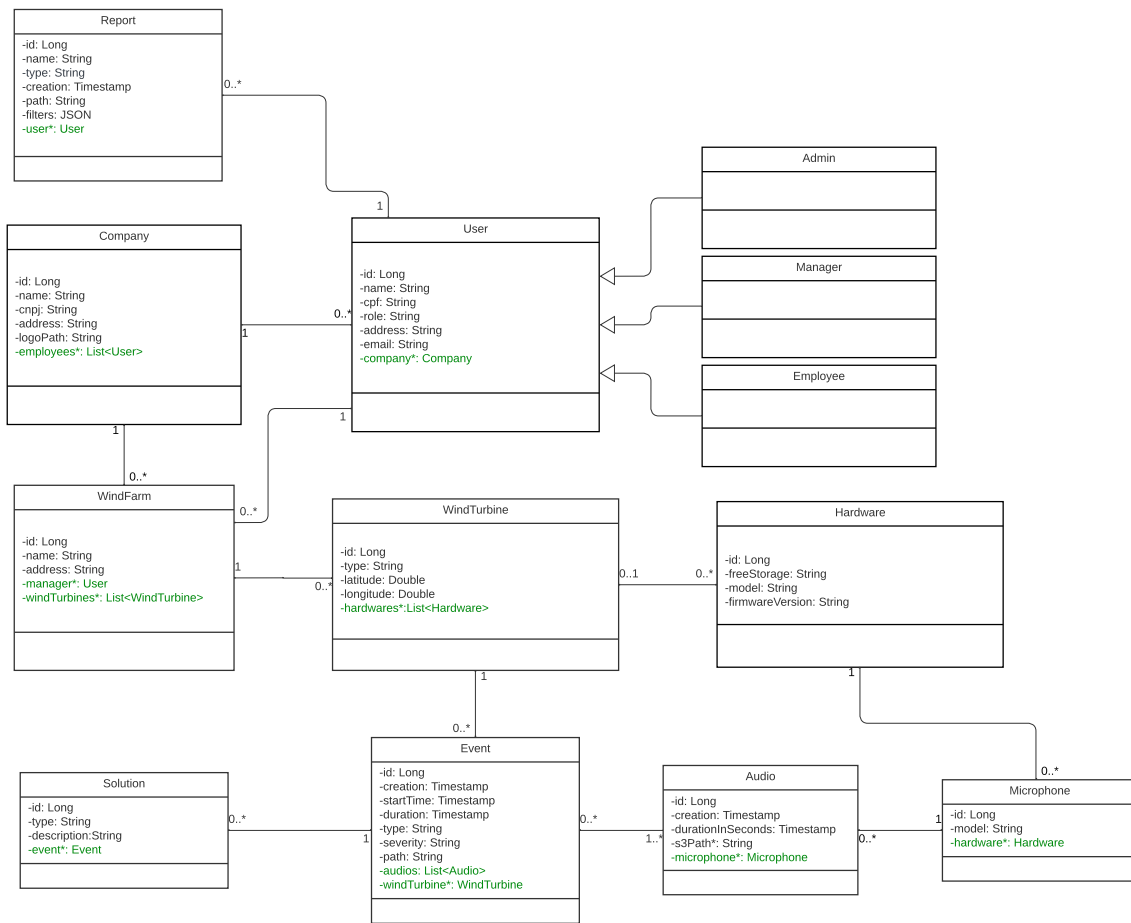


Figura 3.7 – Diagrama de entidades do banco de dados.

Através do diagrama, pode-se perceber os principais relacionamentos criados para satisfazer as necessidades do projeto. Tem-se uma entidade para as regionais (*WindFarm*), que pode ter 0 ou N aerogeradores (*WindTurbine*), os quais podem possuir 0 ou N *Hardwares*. Como a flexibilidade é um ponto crucial do sistema, um *Hardware* pode possuir 0 ou N microfones (*Microphone*), isto permite que o usuário liste os áudios por tipo de microfone. Cada microfone, por sua vez, estará ligado a 0 ou N áudios, e cada áudio pode gerar 0 ou N eventos. Entende-se que o áudio pode estar relacionado a mais que um evento pois sua duração é indefinida, podemos ter tanto áudios de 1 quanto de 30 minutos, deste modo, a quantidade de eventos dentro de um único áudio é desconhecida. Apesar disto, todo o evento estará ligado a apenas um aerogerador (*WindTurbine*) e poderá ter 0 ou N soluções (*Solution*). A entidade *Solution* foi considerada para permitir a interação do usuário com os eventos de maneira direta, indicando eventos errados, por exemplo.

Além das entidades já citadas, têm-se também as entidades mais básicas, que são encontradas na maioria dos sistemas, como usuários (*User*), empresa (*Company*) e

relatório (*Report*). No diagrama, também foram adicionadas algumas generalizações da entidade usuário, o administrador (*Admin*), gerente (*Manager*) e colaborador (*Employee*). Como pode-se perceber, o diagrama foi muito além da lista descrita anteriormente, mas ela serve como um bom ponto de partida para este tipo de desenvolvimento.

3.3.2.2 Implementação do Banco

Como visto anteriormente, tanto as propriedades das entidades quanto o relacionamento entre elas são bem definidos; deste modo, é natural a implementação do banco de dados seguindo o paradigma de banco de dados relacional. Deste modo, a tarefa foi focada em gerar o *script* na linguagem SQL para gerar todas as tabelas necessárias do banco de dados. Este *script* pode ser visto abaixo.

```
1 CREATE TABLE User (
2     id INT PRIMARY KEY,
3     name VARCHAR(255),
4     cpf VARCHAR(11),
5     address VARCHAR(255),
6     role VARCHAR(20),
7     email VARCHAR(100),
8     company_id INT,
9     FOREIGN KEY (company_id) REFERENCES Company(id)
10 );
11
12 CREATE TABLE Company (
13     id INT PRIMARY KEY,
14     name VARCHAR(255),
15     cnpj VARCHAR(50),
16     address VARCHAR(255),
17     logoPath VARCHAR(255)
18 );
19
20 CREATE TABLE WindFarm (
21     id INT PRIMARY KEY,
22     name VARCHAR(255),
23     address VARCHAR(255),
24     manager_id INT,
25     FOREIGN KEY (manager_id) REFERENCES User(id)
26 );
27
```

```
28 CREATE TABLE WindTurbine (  
29     id INT PRIMARY KEY,  
30     type VARCHAR(50),  
31     latitude DOUBLE,  
32     longitude DOUBLE,  
33     windFarm_id INT,  
34     FOREIGN KEY (windFarm_id) REFERENCES WindFarm(id)  
35 );  
36  
37 CREATE TABLE Hardware (  
38     id INT PRIMARY KEY,  
39     freeStorage INT,  
40     model VARCHAR(50),  
41     firmwareVersion VARCHAR(50),  
42     windTurbine_id INT,  
43     FOREIGN KEY (windTurbine_id) REFERENCES WindTurbine(id)  
44 );  
45  
46 CREATE TABLE Microphone (  
47     id INT PRIMARY KEY,  
48     model VARCHAR(50),  
49     hardware_id INT,  
50     FOREIGN KEY (hardware_id) REFERENCES Hardware(id)  
51 );  
52  
53 CREATE TABLE Audio (  
54     id INT PRIMARY KEY,  
55     creation TIMESTAMP,  
56     durationInSeconds TIMESTAMP,  
57     s3Path VARCHAR(255),  
58     microphone_id INT,  
59     FOREIGN KEY (microphone_id) REFERENCES Microphone(id)  
60 );  
61  
62 CREATE TABLE Event (  
63     id INT PRIMARY KEY,  
64     creation TIMESTAMP,  
65     startTime TIMESTAMP  
66     durationInSeconds TIMESTAMP,
```

```
67     type VARCHAR(50),
68     severity VARCHAR(50),
69     s3Path VARCHAR(255),
70     windTurbine_id INT,
71     FOREIGN KEY (windTurbine_id) REFERENCES WindTurbine(id)
72 );
73
74 CREATE TABLE Audio_Event (
75     audio_id INT,
76     event_id INT,
77     PRIMARY KEY (audio_id, event_id),
78     FOREIGN KEY (audio_id) REFERENCES Audio(id),
79     FOREIGN KEY (event_id) REFERENCES Event(id)
80 );
81
82 CREATE TABLE Solution (
83     id INT PRIMARY KEY,
84     type VARCHAR(50),
85     description VARCHAR(255),
86     event_id INT,
87     FOREIGN KEY (event_id) REFERENCES Event(id)
88 );
89
90 CREATE TABLE Report (
91     id INT PRIMARY KEY,
92     name VARCHAR(255),
93     type VARCHAR(50),
94     creation TIMESTAMP,
95     s3Path VARCHAR(255),
96     filters JSON,
97     user_id INT,
98     FOREIGN KEY (user_id) REFERENCES User(id)
99 );
```

3.3.2.3 Disponibilização Online

Com a codificação do banco de dados feita, pode-se fazer a disponibilização do mesmo em ambiente *online*. Considerando a etapa de arquitetura de *software*, descrita na seção anterior, todos os serviços disponibilizados online estão considerando as estruturas

disponíveis no AWS. Sendo assim, o banco de dados também será disponibilizado no AWS, através do serviço RDS, que oferece bancos de dados relacionais na nuvem. Manter todos os serviços em um só provedor não é obrigatório, principalmente pela estrutura flexível escolhida, mas reduz os custos de implementação, transmissão de dados e diminui a latência nas operações de busca/gravação no banco de dados.

Para realizar o *deploy* do banco de dados, é utilizado o Terraform, que permite a codificação da infraestrutura na nuvem; deste modo, não é necessário fazer o *deploy* de cada elemento de maneira manual. O código para subir o banco de dados está descrito a seguir:

```
1 resource "aws_db_instance" "ruído_audível" {
2     identifier          = "ruídoaudível"
3     db_name             = "ruídoaudível"
4     allocated_storage  = 50
5     engine              = "postgres"
6     instance_class     = "db.t3.micro"
7     username           = "XXXXXXXX"
8     password           = "XXXXXXXX"
9     final_snapshot_identifier = true
10    publicly_accessible = true
11 }
```

Os dados sensíveis foram mascarados para garantir a segurança do banco. O endereço do banco de dados fica disponível na interface do AWS RDS e, como ele é imutável, esta operação manual não é um problema, pois será feita apenas uma vez. O banco de dados foi devidamente protegido por senha, garantindo maior segurança e mitigando riscos de vazamento de informações.

3.3.3 API

Esta etapa tem por objetivo implementar toda a parte de segurança da aplicação, com controle de usuários e permissões, além do desenvolvimento e disponibilização *online* de todos os pontos de acesso necessários para a execução do projeto. Para isso, utiliza-se da API, que consiste em um conjunto de regras e protocolos comumente utilizado no desenvolvimento de *software* para a finalidade descrita.

3.3.3.1 Implementação do Controle de Usuários e Permissão

Considerando a criticidade do ambiente operacional onde o *hardware* estará instalado e o nível de segurança esperado para este tipo de aplicação, decidiu-se pela utilização

de um serviço dedicado ao controle de usuários e permissões. O serviço escolhido foi o Keycloak, que é um serviço de gerenciamento de identidade e acesso de código aberto, desenvolvido pela Red Hat. Ele fornece recursos de autenticação, autorização e gerenciamento de identidade para aplicativos e serviços. Com ele, pode-se adicionar facilmente recursos de autenticação baseados em padrões como OAuth, OpenID Connect e SAML em todas as camadas do sistema. Ele também oferece recursos avançados, como controle fino de acesso, autenticação multifator, entre outros.

Um dos principais pontos para a escolha do Keycloak foi sua reputação, com constantes atualizações para resolução de quaisquer brechas de segurança, autenticação seguindo padrões de segurança globais e principalmente sua integração com as *frameworks* utilizadas para o desenvolvimento das APIs. Esta integração permite que toda a configuração de acesso e permissões possa ser implementada de maneira totalmente separada da aplicação principal, mantendo cada sistema focado em seu objetivo principal, além de acelerar o desenvolvimento da equipe, evitando a criação de pontes entre serviços de autenticação e as aplicações, e consequentemente, possíveis falhas de segurança durante este processo. Na Figura 3.8 está um exemplo simplificado da rota para acesso aos dados do servidor.

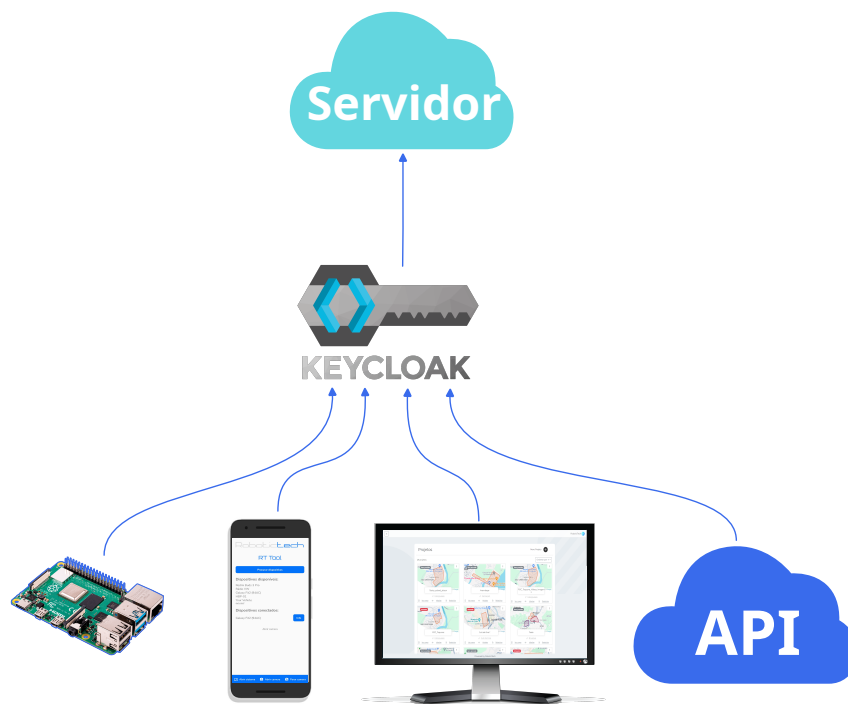


Figura 3.8 – Diagrama simplificado do controle de acesso ao servidor.

Pela Figura 3.8 pode-se perceber que quaisquer tipos de acesso aos dados do projeto devem passar primeiramente pelo servidor de autenticação (Keycloak), seja um *Hardware* em campo enviando os áudios, alguém acessando o aplicativo WEB pelo celular/computador, ou uma requisição direta da API. Isto garante que as rotas pré-definidas

mantêm a segurança dos dados. Pelo fato do Keycloak ter o código aberto, e permitir o uso para aplicações comerciais, tem-se um servidor próprio rodando uma instância do Keycloak, garantindo a segurança do sistema.

3.3.3.2 Implementação dos Pontos de Acesso

O objetivo desta API é suprir as necessidades da interface, garantindo que todas as ações do sistema tenham um *endpoint* correspondente. Do mesmo modo, a interface WEB segue a estrutura definida pelo banco de dados, estabelecendo que a API deve possuir um conjunto de *endpoints* para cada tabela.

Caso não exista uma regra que dite o contrário, este conjunto de *endpoints* segue o padrão definido como “CRUD”, que representa as quatro operações básicas, *Create*, *Read*, *Update* e *Delete*. Seguindo este padrão e as tabelas já definidas no banco de dados, tem-se na Figura 3.9 o diagrama de *endpoints* para a API WEB:

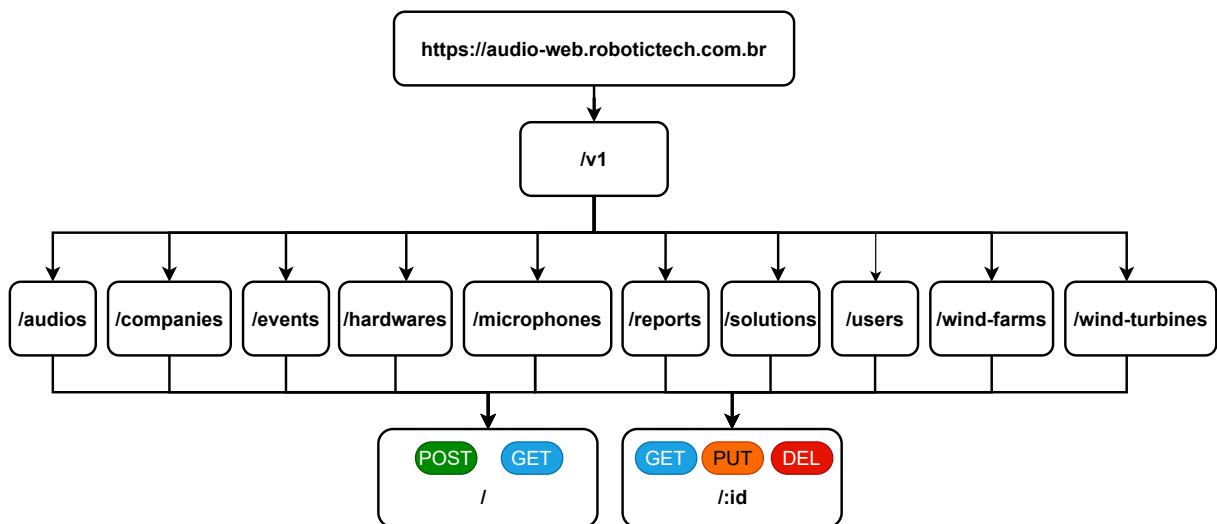


Figura 3.9 – Diagrama de endpoints.

A Figura 3.9 ilustra os pontos de acesso que serão utilizados pelo *frontend* (interface WEB). Cada ponto corresponde a uma tabela do banco de dados, representando os controladores da aplicação. Em cada ponto, existem 5 rotas (caminhos que um pedido HTTP segue para chegar a um ponto de acesso específico), cada uma sendo mapeada para diferentes pontos de acesso. Cada rota possui o seu método HTTP adequado predefinido, estando disponíveis os métodos HTTP padronizados para requisitar os dados: POST para criar novos registros, GET para ler os registros (incluindo GET com ID para um elemento específico), PUT para atualizar registros e DELETE para remover registros.

Na sequência, tem-se como exemplo um conjunto de requisições POST, GET, PUT e DELETE para o ponto de acesso dos áudios.

1. *Upload* de um arquivo de áudio:

```
bash curl -X 'POST' 'https://audio-web.robotictech.com.br/v1/audios' -H 'accept: */*' -H 'Content-Type: application/json' -d ' "id": 0, "creationDate": "2024-05-06T22:08:34.530Z", "durationInSeconds": 0, "s3Path": "string", "s3Bucket": "string", "microphoneId": 0, "eventId": 0 '
```

Nessa requisição, você pode fazer o upload de um arquivo de áudio para o servidor.

2. Obter todos os áudios:

```
bash curl -X 'GET' 'https://audio-web.robotictech.com.br/v1/audios' -H 'accept: */*'
```

Essa requisição permite obter todos os áudios armazenados no servidor. O servidor retornará uma lista de áudios disponíveis.

3. Acessar um áudio específico:

```
bash curl -X 'GET' 'https://audio-web.robotictech.com.br/v1/audios/0' -H 'accept: */*'
```

Essa requisição permite obter um áudio específico informando o ID do mesmo.

4. Atualizar um áudio específico:

```
bash curl -X 'PUT' 'https://audio-web.robotictech.com.br/v1/audios/0' -H 'accept: */*' -H 'Content-Type: application/json' -d ' "id": 0, "creationDate": "2024-05-06T22:14:33.854Z", "durationInSeconds": 0, "s3Path": "string", "s3Bucket": "string", "microphoneId": 0, "eventId": 0 '
```

Com essa requisição, pode-se atualizar os dados de um áudio específico. Deve-se lembrar de fornecer o ID correto do áudio que deseja atualizar.

5. Excluir um áudio específico:

```
bash curl -X 'DELETE' 'https://audio-web.robotictech.com.br/v1/audios/0' -H 'accept: */*'
```

Essa requisição permite excluir um áudio específico com base no ID fornecido. Certifique-se de que o ID corresponda ao áudio que deseja remover, visto que esta ação não pode ser desfeita sem um backup dos dados.

Com estas requisições, é possível realizar as quatro operações básicas, *Create*, *Read*, *Update* e *Delete*. Caso qualquer dado enviado esteja errado, será retornado um

valor diferente de 200 (Código que indica que a operação ocorreu de maneira correta). Este exemplo é uma demonstração de como utilizar um dos pontos de acesso da API, para o restante das funcionalidades, pode-se acessar a documentação completa da API através do seguinte endereço: documentação.

Esta API, assim como a anterior, mantém as características de segurança e escalabilidade esperadas para uma aplicação com dados sensíveis e críticos.

Para a segurança, se mantém a barreira do servidor de autenticação executando o Keycloak. Qualquer usuário que tentar acessar a API, seja por requisição direta ou pelo *frontend*, deverá informar um *token* de segurança, que será validado pela API para garantir a autenticidade do usuário.

Para a escalabilidade, considerando que múltiplos usuários podem fazer requisições para a API, foi feita a adição de balanceadores de carga. Estes balanceadores têm como função primária orquestrar a carga recebida entre múltiplas instâncias de um mesmo serviço, neste caso, a API WEB. Considerando a disponibilidade de CPU de cada instância, o balanceador redireciona as requisições para a instância menos carregada. Como função secundária, este balanceador também agrega segurança, pois consegue detectar ataques, evitando que requisições maliciosas cheguem até a API e sobrecarreguem a mesma.

3.3.3.3 Disponibilização Online

Assim como feito para a API de ingestão de dados, para realizar o *deploy* da API WEB foi utilizado o *Terraform*. O código para criar o serviço que executará a API está descrito a seguir:

```
1 module "AudioWebAPI" {
2     source           = "USSBA/easy-fargate-service/aws"
3     version          = "9.0"
4     family           = "audiowebapi"
5     container_port    = 8080
6     certificate_arn    = "XXXXXXXXXX"
7     hosted_zone_id    = "XXXXXXXXXX"
8     service_fqdn      = "audio-web.robotictech.com.br"
9     route53_allow_overwrite = true
10    cluster_name      = aws_ecs_cluster.RoboticTechCluster.name
11    desired_capacity   = 1
12    max_capacity       = 2
13    scaling_metric     = "cpu"
14    scaling_threshold  = 50
```

```
15     container_definitions    = [  
16         {  
17             name    = "audiowebapi"  
18             image    = "${aws_ecr_repository.robotic_tech_audio_web_api.repo  
19         }  
20     ]  
21     task_cpu              = 256  
22     task_memory           = 1024  
23     health_check_path    = "/actuator/health"  
24 }
```

Os dados sensíveis foram mascarados para garantir a segurança da aplicação. Assim como na API de ingestão de dados, também foram definidos os parâmetros de escalabilidade automática. Neste caso, para o ambiente de testes, limitou-se em 2 o número de instâncias executando a API (“max_capacity”), e a referência para aumentar a capacidade continuou sendo 50% do uso de CPU (“scaling_threshold” e “scaling_metric”). Como resultado final deste código, tem-se a API disponível no endereço: API WEB.

3.4 Redes Generalistas

Nesta subseção, discute-se a estruturação das redes generalistas aplicadas à sinalização de condições de falha por meio do método não supervisionado tipo one-class, no qual apenas dados referentes às condições normais são utilizados para o treinamento, enquanto os desvios em relação a esse padrão são considerados anômalos. Visando maior robustez, foram empregados dois modelos de autoencoders convolucionais, um tipo de rede neural artificial já apresentado na Seção 2.1.2: o Autoencoder Denoising e o Autoencoder Variacional, discutidos nas Seções 3.4.1 e 3.4.2, respectivamente.

Para a construção dos dois modelos de autoencoder, foi utilizada a biblioteca de código-aberto do TensorFlow, desenvolvida pela Google. Esta biblioteca possui um aparato completo para lidar com aplicações que envolvem aprendizado de máquina e redes neurais artificiais, em geral.

3.4.1 Autoencoder Denoising

DAE é uma versão estocástica dos autoencoders tradicionais na qual é treinado a reconstruir um conjunto de dados ruidosos na entrada para retornar uma versão limpa destes dados aproximados do original (VINCENT *et al.*, 2008).

A Figura 3.10 demonstra este princípio no qual o dado x é corrompido para uma versão ruidosa $\tilde{x}$, a ser mapeada pelas camadas ocultas $y = f_\theta(\tilde{x}) = s(W\tilde{x} + b)$ e reconstruída para a forma $z = g'_\theta(y) = s(W'y + b')$, onde W e b são peso e bias da matriz, respectivamente. Estes parâmetros são treinados para minimizar o erro médio de reconstrução dado por $L_H(x, z) = H(Bx \parallel Bz)$, expresso pelo MSE, que se aproxima z da entrada original x :

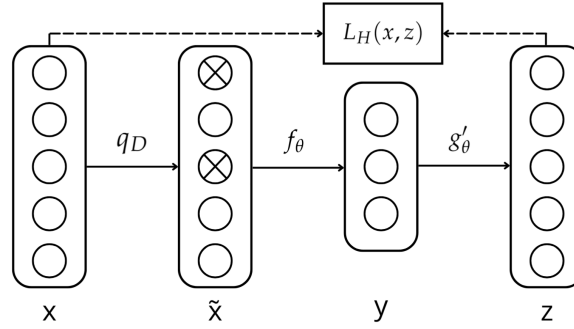


Figura 3.10 – Representação do DAE.

Dessa forma, foi elaborado um modelo de autoencoder convolucional para o processamento das imagens do espectrograma representado na Figura 3.11. Esse modelo tem como base a arquitetura proposta por GONDARA, originalmente aplicada na área medicinal, sendo aqui adaptada para o contexto deste trabalho, resultando na versão apresentada:

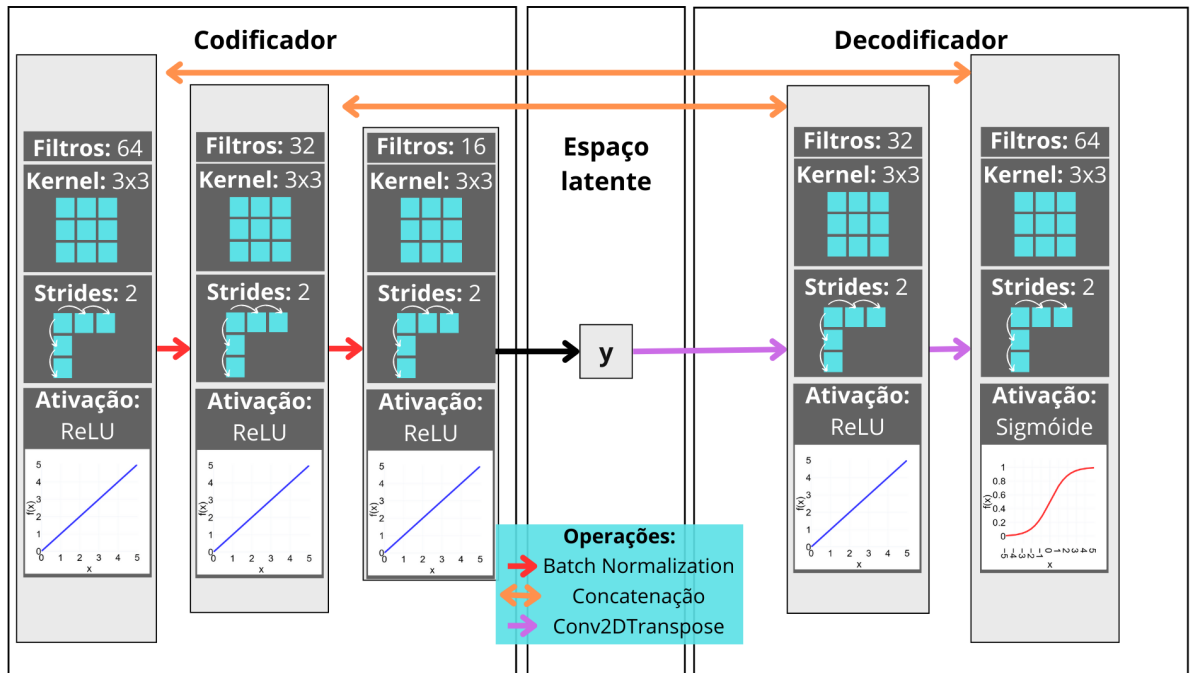


Figura 3.11 – Arquitetura DAE utilizada no projeto.

Na camada de codificação, o dado é comprimido e passado para o processo de Batch Normalization nas camadas ocultas até que se atinja a máxima compressão no espaço latente. Este processo é responsável por acelerar o treinamento de redes neurais profundas, tornando-as mais estáveis, prevenindo contra *outliers*, alcançando altas taxas de aprendizado e reduzindo o *over-fitting* (IOFFE; SZEGEDY, 2015).

Como demonstrada pela Equação 3.1, a normalização de um dado de entrada x na k -ésima posição é a diferença do dado de entrada e a média dos dados do presente batch (μ_{batch}), que consiste nas m amostras (*batch-size*) divididas pelo desvio padrão do conjunto de dados (*batch*) $\sqrt{\sigma_{\text{batch}}^2}$:

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mu_{\text{batch}}}{\sqrt{\sigma_{\text{batch}}^2 + \epsilon}} = \frac{x^{(k)} - \frac{1}{m} \sum_{i=1}^m x_i^{(k)}}{\sqrt{\frac{1}{m} \sum_{i=1}^m (x_i^{(k)} - \mu_{\text{batch}})^2 + \epsilon}}, \quad (3.1)$$

onde o termo tendendo a zero ϵ é inserido no denominador para evitar a divisão por zero.

Aplicando um parâmetro de escala γ e bias β para normalização do valor de $\hat{x}^{(k)}$, tem-se a equação do Batch Normalization $BN_{\gamma, \beta(x_i)}$:

$$BN_{\gamma, \beta(x_i)} = y^{(k)} = \gamma \hat{x}^{(k)} + \beta. \quad (3.2)$$

Os 64 filtros aplicados na primeira camada (cada um com um tamanho de kernel de 3x3 que se traduz em uma sub-matriz que percorre toda a matriz da imagem) extraem parâmetros-chave como texturas, cantos e bordas. A função de ativação ReLU zera os neurônios negativos, como demonstrado pela Equação 3.3. Esta função aplicada em um neurônio x permite uma melhoria na velocidade de aprendizado nas camadas ocultas, reduzindo o custo computacional (GLOROT; BORDES; BENGIO, 2011). Os *strides*, por sua vez, definem a quantidade de neurônios ignorados que, ao mesmo tempo que se perde informação, tem-se um ganho na aceleração do treinamento (GUO; LIU; JIAO, 2019):

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}. \quad (3.3)$$

O espaço latente é a representação compacta dos dados na qual é aperfeiçoado a cada época de passo de aprendizado da rede e visa capturar os aspectos mais relevantes do dado original de forma determinística.

Na camada de decodificação, o dado é reconstruído a partir da representação comprimida da última camada oculta. Desta camada, o dado passa por uma transposição para aumentar sua resolução, trazendo de volta a resolução original do dado. A função de ativação ReLU também é aplicada na camada oculta. Na camada de saída foi aplicada a função sigmóide, dada por:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (3.4)$$

Esta função é apropriada para a representação de qualquer valor real em um intervalo de 0 a 1, onde se tem o intervalo comumente utilizado nos processos de normalização de imagens.

Graficamente, as funções ReLU e sigmóide estão representadas no gráfico da Figura 3.12:

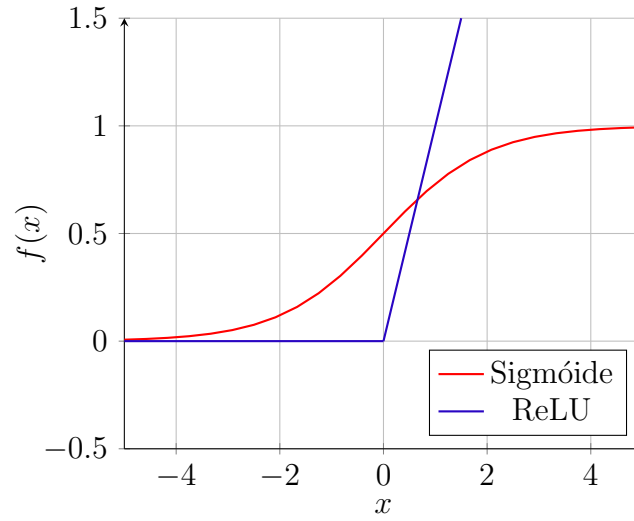


Figura 3.12 – Funções de ativação sigmóide e ReLU.

O processo de concatenação, comumente aplicado em DenseNet (HUANG *et al.*, 2018), permite que o modelo receba a saída das camadas anteriores de mesma dimensão, melhorando o fluxo de informação que, por outro lado, incrementa o custo computacional. Com isso, a camada L^{th} carrega as características das camadas anteriores $x_0, x_1, \dots, x_n$, como suas entradas:

$$x'_L = H'_L([x_0, x_1, \dots, x_{\ell-1}]). \quad (3.5)$$

3.4.2 Autoencoder Variacional

O VAE é uma versão probabilística dos autoencoders tradicionais. Sua função de perda descrita na Equação 3.6 combina dois termos primordiais: o erro de reconstrução (consistência dos dados) e o termo de regularização que força a distribuição latente a se aproximar da distribuição anterior (regularização):

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x) \| p(z)). \quad (3.6)$$

É importante destacar que o VAE possui a capacidade de aprendizado da representação dos dados no espaço latente z por meio da probabilidade de distribuição pela Gaussiana $p_\theta(z) = \mathcal{N}(z; 0, I)$ e probabilidade condicional parametrizada pelo codificador $p_\theta(x|z)$ e pelo decodificador $q_\phi(z|x)$ ao invés de pontos fixos. Essa aproximação permite a geração de novas amostras similares às originais obtidas a partir da distribuição latente (KINGMA; WELLING, 2022).

A variável $\mathbb{E}_{q_\phi(z|x)}$ trata-se do valor esperado, dado pela integral da função de probabilidade de densidade ($\mathbb{E}[X] = \int_{-\infty}^{\infty} xp(x) dx$). A divergência KL do termo $D_{KL}(q_\phi(z|x) \parallel p_\theta(z))$ expressa a distância entre as duas distribuições de probabilidade $p_\theta(z)$ e $q_\phi(z|x)$, mostrada na Equação 3.7:

$$D_{KL}(q_\phi(z|x) \parallel p_\theta(z)) = \int q_\phi(z|x) \log \frac{q_\phi(z|x)(z|x)}{p_\theta(z)} dz. \quad (3.7)$$

A arquitetura do VAE utilizada na presente aplicação é relatada na Figura 3.13. O modelo em questão foi baseado no abordado por SHE; ZHANG; WANG, especificamente para diagnóstico de falhas e que, da mesma forma, passou por adaptações até se chegar ao presente modelo:

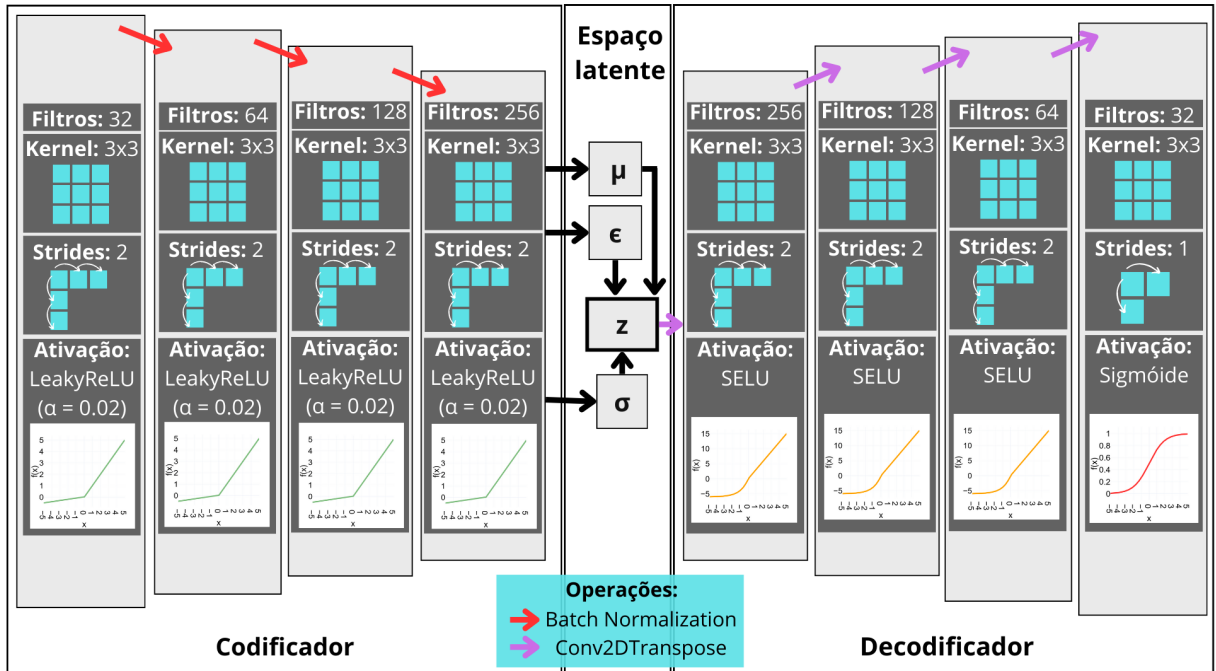


Figura 3.13 – Arquitetura VAE utilizada.

A camada de codificação da arquitetura em questão incorpora camadas adicionais e utiliza a função de ativação LeakyReLU como a principal para manipulação dos neurônios. Esta função é uma variação da ReLU, onde a diferença se baseia em que a

LeakyReLU permite valores negativos, mitigando o problema dos “neurônios mortos”, encontrado na função ReLU por meio de um coeficiente angular α .

Posteriormente, a ligeira inclinação negativa garante que alguns gradientes ainda estejam presentes na ativação negativa, onde pode ser crucial para a manutenção da estabilidade de treinamento e obtenção de uma melhor convergência em redes neurais profundas.

A Equação 3.8 representa matematicamente a função LeakyReLU, enquanto a Figura 3.8 ilustra graficamente sua representação para diferentes valores de α (MAAS, 2013). A função sigmóide também é aplicada na saída do decodificador (assim como feito no DAE):

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}. \quad (3.8)$$

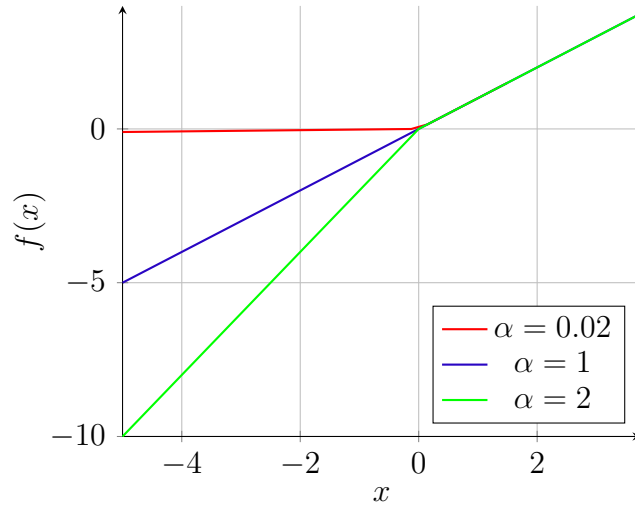


Figura 3.14 – Função de ativação LeakyReLU para diferentes parâmetros α .

Quando $\alpha = 0$, a função *LeakyReLU* se reduz à função *ReLU*. Para $\alpha \neq 0$, ela se comporta como uma função linear com coeficiente angular α para os valores de $x < 0$, permitindo assim pequenas ativações negativas. Um valor pequeno de $\alpha = 0,02$ é aplicado na camada de codificação do VAE, como observado na Figura 3.13.

Na camada de decodificação, foi utilizada a função de ativação SELU proposta por (KLAMBAUER *et al.*, 2017). A Equação 3.9 descreve a sua operação, evitando também que o gradiente desapareça durante o processo de treinamento:

$$f(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}, \quad (3.9)$$

onde λ é um fator de escala (por padrão, configurado com o valor aproximado de 1,0507); α é o coeficiente angular (por padrão, é de 1,67326 (RASAMOELINA; ADJAILIA; SINČÁK, 2020)).

A Figura 3.15 representa o comportamento desta função de ativação:

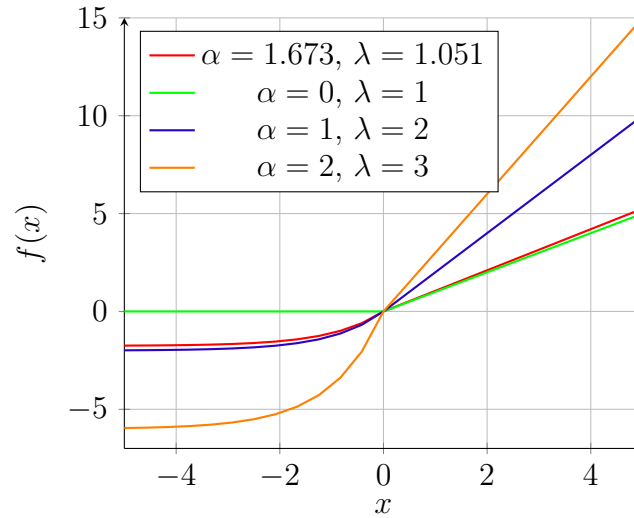


Figura 3.15 – Função de ativação SELU para diferentes valores de α .

3.5 Processamento de Dados

A presente seção descreve como os dados são recebidos e interpretados pelos modelos autoencoder, bem como o fluxo de informações ao longo do servidor *online* descrito ao final.

O processamento de conversão do arquivo de áudio para a interpretação das redes autoencoders segue a rotina descrita na Figura 3.16

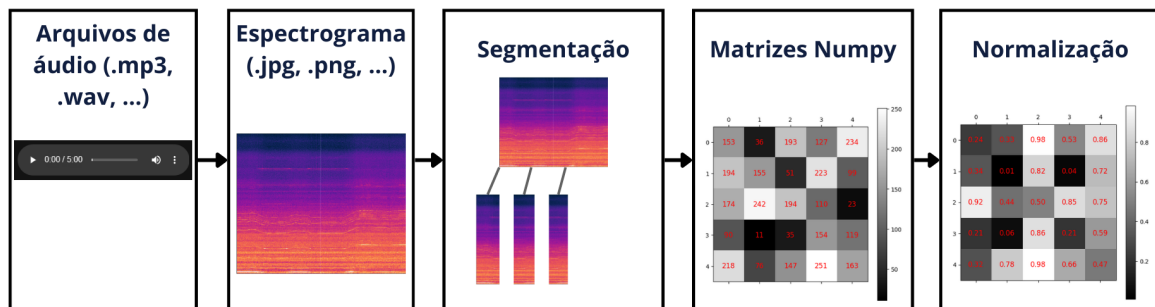


Figura 3.16 – Processamento de dados para adaptação às redes autoencoder.

Esta figura demonstra o fluxo da preparação do áudio para a posterior interpretação pelas redes autoencoder. Seu início se dá com o arquivo de áudio, passando pela transformação em espectrograma até o alcance de uma matriz na entrada das redes autoencoders.

Primeiramente, os áudios dos aerogeradores são armazenados no servidor dedicado da AWS, que podem ser capturados nos formatos MP3 e WAV para o processamento no algoritmo.

Em seguida, com o auxílio da biblioteca de manipulação de arquivos de áudio Sox, converte-se o áudio extraído do repositório em uma representação gráfica em termos de tempo, frequência e amplitude conhecida como espectrograma. A partir daí, trabalha-se com técnicas de processamento de imagem.

Para se obter uma análise mais restrita por um tempo, segmenta-se o espectrograma referente ao arquivo de áudio completo em segmentos de 5 segundos de duração.

Em processamento de imagens, as mesmas são manipuladas como matrizes representativas nas quais suas dimensões são dadas pela largura e altura da imagem (número de canais de cores da imagem). Para o esquema de cores RGB, por exemplo, tem-se três canais. Para otimização de processamento, optou-se pelo uso da escala em cinza, que possui somente um canal e é suficiente para representar a variação de amplitude do espectrograma através da intensidade da cor branca até o preto.

Em imagens digitais, a escala se dá geralmente entre os valores de 0 a 255, onde o valor 0 representa um pixel da imagem de intensidade nula (cor branca) e 255 de forte intensidade (cor preta). No caso da biblioteca Tensorflow, estas são designadas para operar com valores normalizados entre 0 e 1. Esse processo de normalização envolve dimensionar os valores de intensidade de pixel para ficarem dentro dessa faixa menor, o que não somente melhora a estabilidade computacional, mas também garante que o modelo possa interpretar e processar corretamente os dados de entrada. A normalização adequada é uma etapa crítica de pré-processamento, ao alinhar a representação dos dados com as expectativas dos algoritmos subjacentes, permitindo treinamento e inferência mais eficazes.

Do ponto de vista do servidor, inicialmente, são lidos dois arquivos no formato JavaScript Object Notation (JSON), um formato simples e de fácil leitura para a troca de dados. Um desses arquivos contém os parâmetros iniciais da rede, conforme ilustrado no exemplo abaixo. Nele, são especificados, respectivamente: o caminho do modelo salvo no servidor AWS, o tamanho do batch, a largura e a altura das imagens utilizadas no treinamento, e uma variável booleana que indica se as imagens reconstruídas devem ser salvas:

```
1 {  
2   "model_path": "/models/vae_grayscale.h5",
```

```
3   "batch_size": 32,  
4   "input_image_width": 128,  
5   "input_image_height": 64,  
6   "save_reconstructed_images": false  
7 }
```

Outro modelo de JSON, apresentado abaixo, refere-se aos segmentos de áudio e contém, de forma detalhada, as informações de entrada referentes a cada segmento analisado para cada arquivo de áudio.

```
1  {  
2    "execution_data": "2025-03-18T13:12:22.883951",  
3    "algorithm": "Segmentation",  
4    "algorithm_version": "2.0",  
5    "arguments": "Namespace(  
6      input= '.../DownloadAudiosByID.json',  
7      output= '.../segmentation.json',  
8      time= 5.0,  
9      log= 'INFO'  
10   )",  
11   "output": [  
12     {  
13       "id": 354656,  
14       "creationDate": "2025-03-18T12:38:56.159119",  
15       "durationInSeconds": 300,  
16       "status": "PROCESSED",  
17       "s3Path": ".../1742301536159.mp3",  
18       "s3Bucket": "robotic-tech-audios",  
19       "microphoneId": "8",  
20       "hasEvent": false,  
21       "url": "https://...",  
22       "path": ".../354656.mp3",  
23       "segments": [  
24         {  
25           "path": ".../354656_001.wav",  
26           "spectrogram": ".../354656_001.png",  
27           "startTimeInMilliseconds": 0.0,  
28           "endTimeInMilliseconds": 5000.0  
29         },  
30         {
```

```
31         "path": ".../354656_002.wav",
32         "spectrogram": ".../354656_002.png",
33         "startTimeInMilliseconds": 5000.0,
34         "endTimeInMilliseconds": 10000.0
35     },
36     {
37         "path": ".../354656_003.wav",
38         "spectrogram": ".../354656_003.png",
39         "startTimeInMilliseconds": 10000.0,
40         "endTimeInMilliseconds": 15000.0
41     }
42 ]
43 }
44 ]
45 }
```

O JSON apresentado contém os resultados do processamento de arquivos de áudio, com os seguintes atributos:

- **execution_data**: Data e horário de execução do algoritmo.
- **algorithm**: Nome do algoritmo empregado.
- **algorithm_version**: Versão do algoritmo utilizada.
- **arguments**: Parâmetros de entrada fornecidos ao algoritmo, incluindo:
 - **input**: Caminho do arquivo de entrada.
 - **output**: Caminho do arquivo de saída.
 - **time**: Tempo de processamento.
 - **log**: Nível de *log*.
- **output**: Lista com informações sobre cada arquivo de áudio processado.
 - **id**: Identificador único do arquivo de áudio.
 - **creationDate**: Data de criação do arquivo.
 - **durationInSeconds**: Duração total do áudio em segundos.
 - **status**: Estado do processamento (ex.: “PROCESSED”).
 - **s3Path**: Caminho do arquivo armazenado no bucket S3.
 - **s3Bucket**: Nome do bucket S3.

- **microphoneId**: Identificador do microfone utilizado.
- **hasEvent**: Indica se houve detecção de evento relevante.
- **url**: URL de acesso remoto ao arquivo de áudio.
- **path**: Caminho local do arquivo de áudio processado.
- **segments**: Lista de segmentos do áudio, onde cada segmento contém:
 - * **path**: Caminho local do arquivo de áudio segmentado.
 - * **spectrogram**: Caminho da imagem do espectrograma correspondente.
 - * **startTimeInMilliseconds**: Tempo inicial do segmento, em milissegundos.
 - * **endTimeInMilliseconds**: Tempo final do segmento, em milissegundos.

3.6 Treinamento das Redes Generalistas

Para treinar os autoencoders convolucionais, é crucial construir um conjunto de dados (*dataset*) de imagens robusto, com espectrogramas de áudios das turbinas eólicas em condições normais (condições desejáveis de operação) para serem reconstruídos com alta confiabilidade.

O processo de treinamento envolve cerca de 20000 espectrogramas balanceados em termos de velocidade do vento para as cinco diferentes turbinas eólicas. Para isso, foram separados segmentos de imagens de espectrograma de três máquinas em boas condições conhecidas: FAI2-12, FAI1-05, FAI1-07.

O principal desafio desta etapa é coletar as diversas variações de comportamento dos espectrogramas que representam uma turbina eólica em boas condições. Esta variedade de espectrogramas se deve a condições como, por exemplo, a velocidade do vento, mostrada na Figura 3.17 e condições climáticas que podem destacar algumas frequências que possam aparentar algum ruído indesejável:

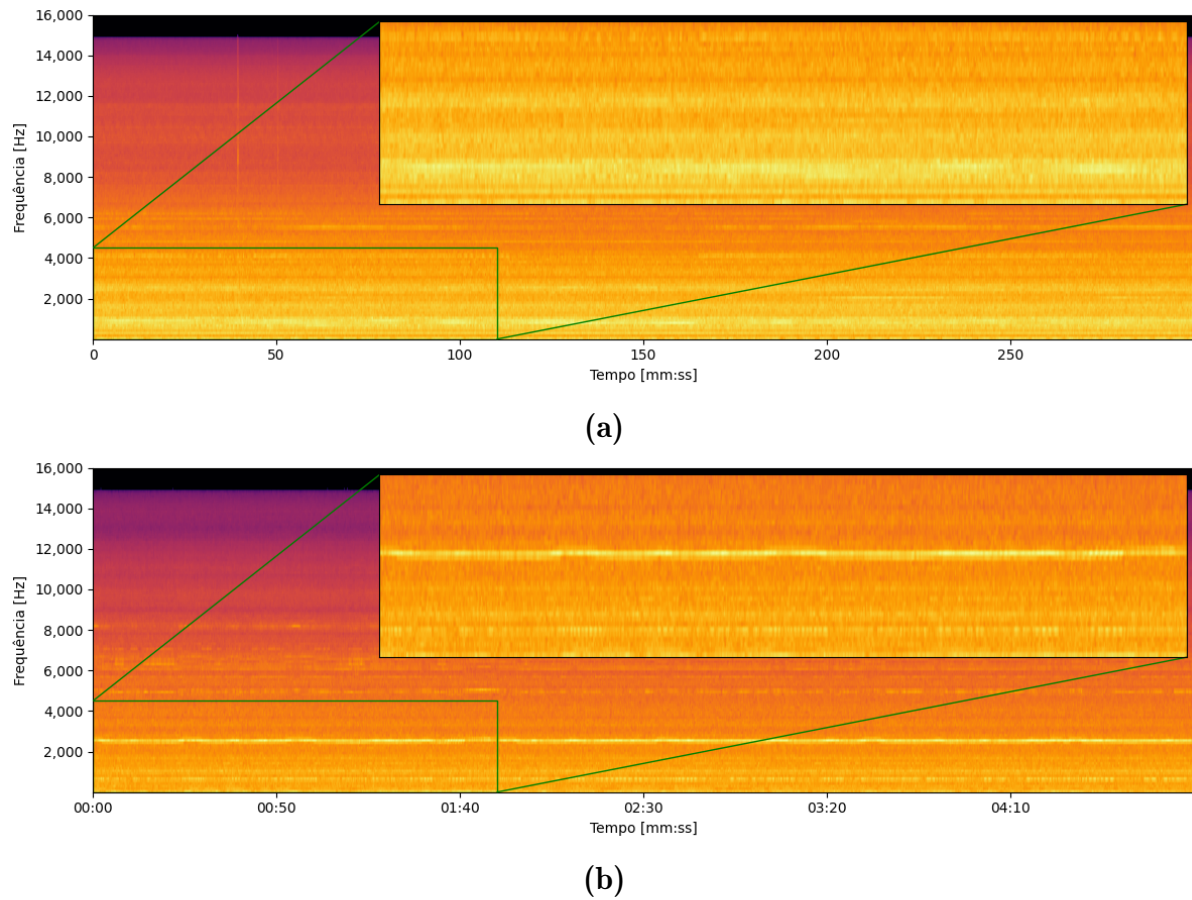


Figura 3.17 – Áudio de um gerador em vento (a) fraco e (b) forte.

Desta forma, para a montagem do *dataset*, foram utilizados áudios coletados durante o período em que os *hardwares* foram instalados, enviando posteriormente seus dados para o servidor em nuvem. Logo após, através do cruzamento das informações dos dados SCADA (vigente na empresa gerenciadora com os áudios), os arquivos foram separados conforme diferentes níveis de velocidade do vento (na operação normal) dos aerogeradores, conforme a Tabela 3.2:

Tabela 3.2 – Divisão dos dados com base na velocidade do vento.

Condição	Velocidade do vento (m/s)
Vento fraco	$v < 5$
Vento médio	$5 \leq v \leq 8$
Vento forte	$v > 8$

Foram coletados os arquivos de áudio do servidor relativo às máquinas previamente identificadas como condições normais de funcionamento (treinamento das redes autoencoder). Estes áudios passam pelas etapas de processamento previamente relatadas na Figura 3.16 para serem corretamente interpretados pela rede. Uma parcela destes da-

dos será utilizada como validação para o modelo poder extrair as métricas relativas às perdas que irão relatar o desempenho da rede treinada.

Para efetuar o treinamento das redes, os dados foram divididos entre treinamento e validação em uma proporção 75/25 somente com amostras de dados normais de aerogeradores, com base no SCADA fornecido em modo de classificação *one-class*, ou seja, o modelo utiliza somente dados de treinamentos normais para, no processo de inferência, diferenciar os modelos anômalos que não se encaixam no padrão treinado. Os dados de treinamento irão servir de base para o ajuste de pesos do modelo, enquanto os dados de validação serão destinados a avaliar o desempenho, assim como realizar ajustes de hiperparâmetros. O objetivo aqui é garantir que o modelo generalize os novos dados, evitando *overfitting* e *underfitting*. As Figuras 3.18 e 3.19 demonstram, respectivamente, os desempenhos dos modelos DAE e VAE desenvolvidos:

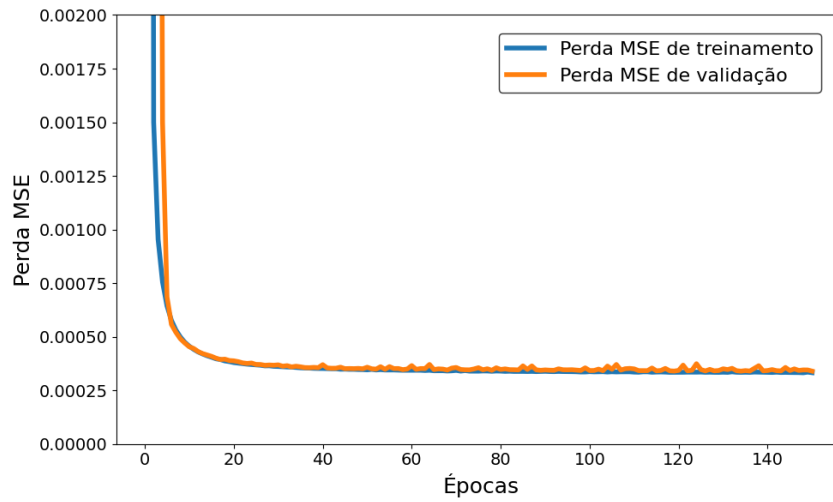


Figura 3.18 – Desempenho do DAE treinado em 150 épocas.

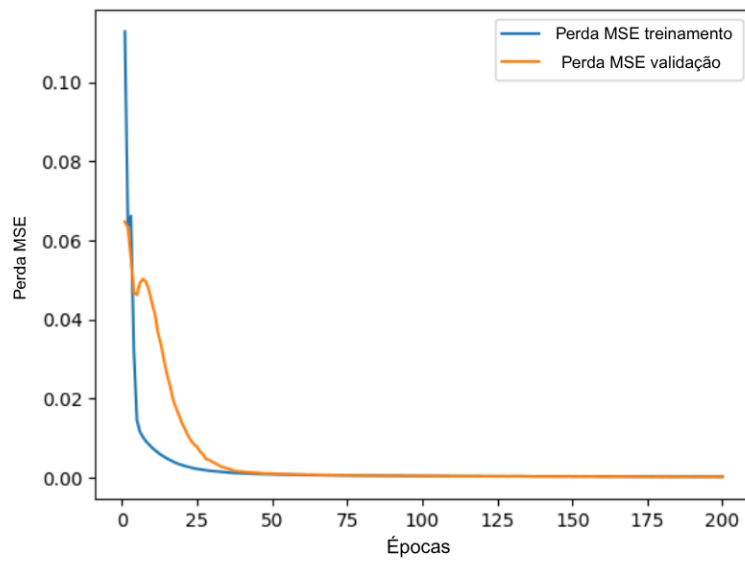


Figura 3.19 – Desempenho do VAE treinado em 200 épocas.

Como demonstrado, o desempenho do modelo durante o processo de treinamento apresenta baixas perdas de MSE para os grupos de treinamento e validação ao longo dos períodos. Isso garante a capacidade de reconstrução dos dados normais de ambos os autoencoders.

3.7 Inferência e Validação

Para avaliar uma condição normal ou anômala, é realizado um comparativo de cada segmento de espectrograma reconstruído pelas redes autoencoders DAE e VAE com o respectivo segmento de entrada da rede. Para isso, têm-se duas principais medidas comparativas que avaliam esta qualidade de reconstrução e, por consequência, se o segmento se encaixa no conjunto de dados de treinamento que a rede foi ensinada a reproduzir, logo, se está no escopo de dados normais.

A primeira métrica abordada é o SSIM. Esta métrica analisa características de luminância, estrutura e contraste para o cálculo de similaridade entre duas imagens (WANG *et al.*, 2004). A Figura 3.20 traz um diagrama representativo de como é feito este cálculo de similaridade:

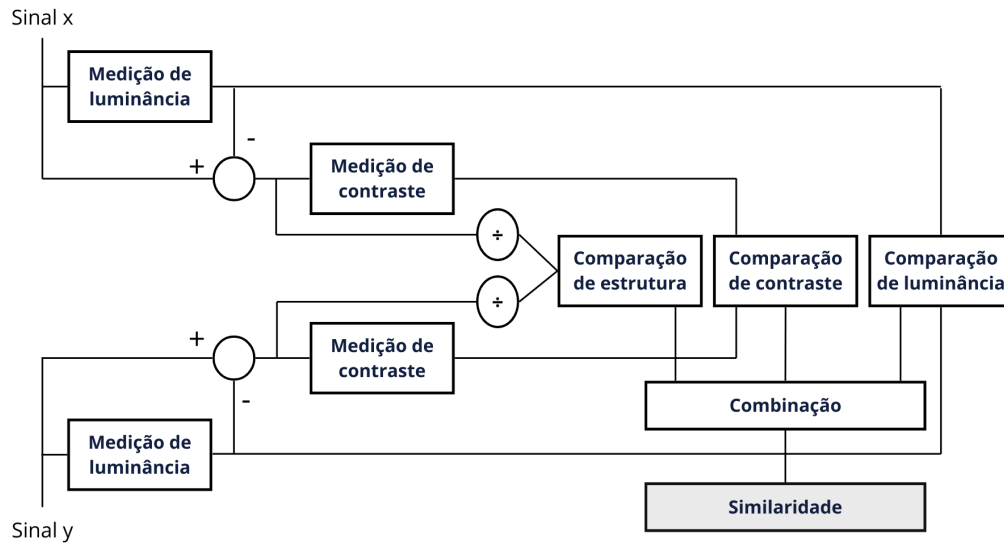


Figura 3.20 – Diagrama de cálculo do SSIM.

A luminância é determinada como a combinação da intensidade luminosa com propriedades de reflexão. A Equação 3.10 descreve este parâmetro em termos da média de N sinais para cada sinal de imagem não-negativo x . Por fim, a função de comparação entre as luminâncias das duas imagens $l(x,y)$ é definida como uma função de μ_x e μ_y :

$$\mu_x = \frac{1}{N} \sum_{i=1}^M x_i. \quad (3.10)$$

O contraste de uma imagem é definido pela diferença perceptível entre a parte mais escura e a mais clara de uma imagem. Matematicamente, este parâmetro pode ser aproximado para um desvio padrão σ dos sinais da imagem descrito na Equação 3.11. O comparativo de contraste de duas imagens $c(x,y)$ é uma função de σ_x e σ_y :

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^M (x_i - \mu_x)^2}. \quad (3.11)$$

Em seguida, é realizado um processo de normalização através da comparação de estrutura chegando à Equação 3.12. Quanto maior este valor, mais a aproximação da média e menor sua variância:

$$s_x = \frac{x - \mu_x}{\sigma_x}. \quad (3.12)$$

Por fim, o bloco de combinação expresso pela Equação 3.13, combinando os três atributos mencionados:

$$S_x = f(l(x,y), c(x,y), s(x,y)). \quad (3.13)$$

onde a comparação da luminância, contraste e estrutura entre dois sinais são representadas pelas Equações 3.14, 3.15, e 3.16:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (3.14)$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (3.15)$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \quad (3.16)$$

com $C_1 = C_2 = (k_2L)^2$ e $C_3 = C_2/2$.

Logo, a fórmula do SSIM é definida como:

$$SSIM(x,y) = [l(x,y)]^\alpha [c(x,y)]^\beta [s(x,y)]^\gamma, \quad (3.17)$$

$$= \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (3.18)$$

onde os parâmetros α , β e γ são úteis para enfatizar a importância de cada parâmetro, que pode ser simplificado para $\alpha = \beta = \gamma = 1$.

Outra métrica utilizada como retorno das redes autoencoders é o PSNR. Este é comumente associado a uma medida quantitativa da qualidade de reconstrução em termos de compressão de imagem. Esta métrica tem correlação com o MSE, como demonstrado na Equação 3.19. O PSNR é calculado sendo a função logarítmica da razão entre o máximo valor de pixel ao quadrado e o MSE:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{Max}^2}{\text{MSE}} \right), \quad (3.19)$$

onde:

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [I(i,j) - K(i,j)]^2. \quad (3.20)$$

O cálculo do MSE envolve a altura e a largura de ambas as imagens (que devem ser iguais) e os valores de intensidade dos pixels em cada posição da imagem ($I(i,j)$ para a reconstrução $K(i,j)$ para a entrada) e, desta forma, calcula a média entre cada uma destas posições para as duas imagens (MARTINI, 2023).

Em resumo, a inclusão da métrica PSNR incrementa a avaliação dos segmentos por não possuir uma correlação linear com o SSIM (HORé; ZIOU, 2013). Assim, tem-se uma métrica adicional como suporte para a avaliação das reconstruções das redes autoencoders que analisam diferentes aspectos: uma referente a aspectos de imagem e outra de sinais.

No ponto de vista do servidor, tem-se também um arquivo em JSON retornado pelas redes generalistas que expõe os dados de métricas dos áudios da forma descrita abaixo.

```
1  {
2      "output": [
3          {
4              "id": 354656,
5              "creationDate": "2025-03-18T12:38:56.159119",
6              "durationInSeconds": 300,
7              "status": "PROCESSED",
8              "s3Path": "RuidoAudivel/.../1742301536159.mp3",
9              "s3Bucket": "exemplo-bucket-audios",
10             "microphoneId": "8",
11             "hasEvent": false,
12             "url": "https://...1742301536159.mp3?...signature",
13             "path": "/processing/audios/354656.mp3",
14             "segments": [
15                 {
16                     "path": "/processing/segments
17                         /354656/354656_001.wav",
18                     "spectrogram": "/processing/segments
19                         /354656/354656_001.png",
20                     "startTimeInMilliseconds": 0.0,
21                     "endTimeInMilliseconds": 5000.0,
22                     "metrics": {
23                         "DAE_SSIM": {
24                             "value": 0.76
25                         },
26                         "DAE_PSNR": {
27                             "value": 19.93
28                         }
29                     }
30                 }
31             ]
32         }
33     ]
34 }
```

```
27         }
28     },
29     {
30         // ... proximos segmentos
31     }
32 ]
33 }
34 ]
35 }
```

- **output**: Lista que contém os dados processados de cada arquivo de áudio. Cada elemento representa um áudio individual.

Para cada arquivo de áudio dentro de **output**:

- **id**: Identificador único do arquivo de áudio processado.
- **creationDate**: Data e hora de criação ou registro do arquivo de áudio.
- **durationInSeconds**: Duração total do áudio em segundos.
- **status**: Estado do processamento, indicando se o áudio foi processado corretamente (ex.: “PROCESSED”).
- **s3Path**: Caminho completo do arquivo de áudio armazenado no bucket S3.
- **s3Bucket**: Nome do bucket S3 onde o arquivo está armazenado.
- **microphoneId**: Identificador do microfone utilizado para capturar o áudio.
- **hasEvent**: Valor booleano que indica se algum evento relevante foi detectado no áudio.
- **url**: URL de acesso remoto ao arquivo de áudio, geralmente assinada para download seguro.
- **path**: Caminho local do arquivo de áudio processado no servidor.

Dentro do atributo **segments**, cada elemento descreve um segmento do áudio:

- **path**: Caminho local do arquivo de áudio referente ao segmento específico.
- **spectrogram**: Caminho do espectrograma do segmento, geralmente em formato de imagem PNG.

- **startTimeInMilliseconds**: Tempo inicial do segmento, em milissegundos, relativo ao início do áudio.
- **endTimeInMilliseconds**: Tempo final do segmento, em milissegundos, relativo ao início do áudio.
- **metrics**: Métricas calculadas sobre o segmento de áudio.
 - **DAE_SSIM**: Índice de similaridade estrutural entre a entrada e a reconstrução pelo autoencoder, com valor em “value”.
 - **DAE_PSNR**: Valor de PSNR (Peak Signal-to-Noise Ratio) do segmento reconstruído pelo autoencoder, indicado em “value”.

3.8 Suavização das Métricas

Para lidar com as oscilações abruptas das reconstruções dos segmentos de espectrogramas que podem ter diferentes razões (como, por exemplo, a queda de velocidade do aerogerador, que diminui a intensidade do som), utiliza-se a técnica do Filtro de Kalman (KALMAN, 1960) para a suavização das medições. Esta filtragem opera de forma similar com uma média móvel, tendo a vantagem de ser mais dinâmica e efetiva em suas previsões.

O cálculo do filtro de Kalman segue as etapas a seguir:

1. **Previsão**: O filtro de Kalman realiza uma estimativa do estado do sistema para o próximo instante de tempo, com base no modelo dinâmico previamente definido. Utiliza-se a equação de transição de estado para projetar o valor esperado (estado estimado) a partir do estado atual, incorporando o conhecimento sobre a evolução temporal do sistema. Simultaneamente, é feita a projeção da incerteza (covariância do erro de estimação), considerando a incerteza intrínseca do modelo, representada pelo ruído de processo. Essa previsão fornece a base sobre a qual será realizada a etapa de correção, quando uma nova observação estiver disponível.
2. **Cálculo do Ganho de Kalman**: O ganho de Kalman, denotado por K , é um fator de ponderação que quantifica a confiança relativa entre a estimativa prevista (*a priori*) e a nova medição. Ele é calculado com base na covariância do erro de previsão e na incerteza da medição (ruído do sensor). Um valor de K próximo de 1 indica alta confiança na medição, enquanto valores próximos de 0 indicam maior confiança na previsão. O ganho busca minimizar a variância do erro da estimativa *a posteriori*, tornando a fusão entre modelo e dados observados estatisticamente ótima.
3. **Correção**: Nesta etapa, a estimativa do estado é ajustada utilizando a diferença entre a medição real e a medição prevista (denominada *inovação* ou *resíduo*). Essa

diferença é multiplicada pelo ganho de Kalman, e o resultado é somado à previsão feita anteriormente. Esse processo gera uma nova estimativa do estado, mais precisa, ao combinar a previsão com a informação observada, ponderadas segundo suas incertezas relativas.

4. **Atualização da Incerteza:** Após a correção, o filtro atualiza a matriz de covariância do erro de estimativa, reduzindo-a para refletir a nova confiança adquirida com a incorporação da medição. A atualização é feita com base no ganho de Kalman e na covariância da previsão. Como resultado, a incerteza diminui (ou se adapta), representando uma estimativa mais confiável do estado atual do sistema.

Matematicamente, tem-se abaixo o algoritmo que realiza o cálculo:

Algoritmo 1 Filtro de Kalman Simples

```

1: função KALMAN_FILTER(new_data, Q, R, P, old_mean)
2:    $P_{pred} \leftarrow P + Q$                                 ▷ Previsão da variância da estimativa
3:    $K \leftarrow \frac{P_{pred}}{P_{pred} + R}$                         ▷ Cálculo do ganho de Kalman
4:    $new\_mean \leftarrow old\_mean + K \cdot (new\_data - old\_mean)$   ▷ Atualização da média
   estimada
5:    $new\_P \leftarrow (1 - K) \cdot P_{pred}$                     ▷ Atualização da incerteza
6:   Retorna new_mean, new_P
7: Fim função
  
```

onde:

- **new_data:** nova medição;
- **Q:** variância do ruído do processo (incerteza na previsão);
- **R:** variância do ruído da medição (incerteza na medição);
- **P:** variância da estimativa anterior (confiança na última estimativa);
- **old_mean:** média anterior estimada.

Assim, foram desenvolvidas duas janelas de captura para o filtro de Kalman, cada uma voltada para um intervalo específico de oscilação: um focado em variações de curto prazo e outro voltado para oscilações de longo prazo.

A janela de curto prazo é ajustada para responder rapidamente às variações e flutuações que ocorrem em um período curto de aproximadamente 2 horas. Com parâmetros configurados para refletir um maior nível de sensibilidade, sendo designado para detectar mudanças transitórias e anomalias pontuais.

Por outro lado, a janela de longo prazo foi elaborada para capturar oscilações e flutuações em um período de 1 semana, o que é útil para observar a degradação do aerogerador ao longo do tempo e prever falhas mais severas que podem decorrer do acúmulo de eventos de curto prazo. Assim, a atualização de seus parâmetros é feita de forma mais suavizada.

Para alcançar um resultado aproximado e determinar a janela de tempo adequada, foi necessário estabelecer os valores das variâncias Q e R , fundamentando-se na média móvel de 2 horas. Essa escolha é essencial, pois as variâncias Q e R influenciam diretamente a sensibilidade e a precisão do Filtro de Kalman. A variância Q , que representa a incerteza do modelo, e a variância R , que reflete a incerteza das medições, foram ajustadas para otimizar a performance do filtro em relação aos dados analisados. Portanto, para cada nova janela de áudio desejada, é necessário encontrar os melhores valores de Q e R .

Em conjunto com as médias das métricas, são utilizadas Bandas de Bollinger (BOLLINGER, 2002), um indicador de volatilidade que auxiliará no cálculo da equação do filtro de Kalman. As bandas de Bollinger consistem em três componentes principais: a média móvel, a banda superior e a banda inferior. A Média Móvel Aritmética (MMA) representa a tendência central dos dados, geralmente calculada sobre uma janela deslizante de n observações. A banda superior é definida como a soma da média móvel com um múltiplo k do desvio padrão σ da série, dado por:

$$B_{\text{superior}} = \text{MMA} + k\sigma, \quad (3.21)$$

enquanto a banda inferior é obtida pela subtração desse mesmo múltiplo do desvio padrão da média móvel:

$$B_{\text{inferior}} = \text{MMA} - k\sigma, \quad (3.22)$$

onde o parâmetro k é ajustável e determina o nível de sensibilidade das bandas à volatilidade da série.

Vale ressaltar que valores típicos de k variam entre 1 e 2, sendo $k = 2$ uma escolha comum para capturar aproximadamente 95% das variações dos dados, assumindo distribuição normal.

A Figura 3.21 a seguir ilustra de forma visual os elementos das bandas de Bollinger aplicados a um conjunto de dados fictício para melhor compreensão.

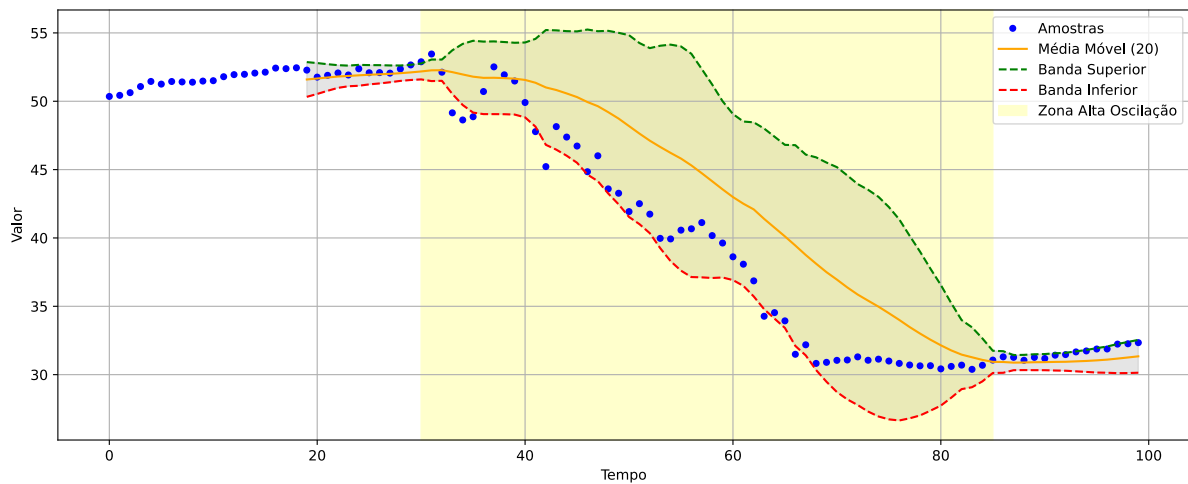


Figura 3.21 – Demonstração das Bandas de Bollinger aplicadas a um conjunto de dados.

Observa-se que, quando os dados apresentam maior variabilidade, sinalizada pela área em amarelo, as Bandas de Bollinger se expandem como reflexo do aumento do desvio padrão (σ).

Ao suavizar as métricas de similaridade no domínio do tempo, são aplicadas métricas estatísticas para acionar o detector de anomalias que enviará para a aplicação o instante em que houver alguma baixa ou oscilação significativa dos valores. Essas métricas envolvem a diferença entre a média curta e longa, a variabilidade e o z-score.

A variabilidade é calculada consoante a dispersão dos dados retornada pelas Bandas de Bollinger, que podem ser quantificadas pelo desvio padrão das bandas.

O z-score, é definido pela Equação 3.23, sendo a diferença entre o valor atual e a média dividido pelo desvio padrão que indica o quão distante o valor está da média em desvios-padrões:

$$Z_{score} = \frac{x - \mu}{\sigma}. \quad (3.23)$$

Na elaboração do filtro, foi determinada a condição de falhas seguindo os limiares da Tabela 3.3 obtido através de testes empíricos para que, em caso de valores superiores aos determinados, cada parâmetro sinalize a condição de anomalia:

Tabela 3.3 – Limiares obtidos empiricamente para o detector de anomalias.

Métrica	Limiar
Z-score	3,55
Variabilidade	1.5
Diferença entre as médias longa e curta	1,8

O comportamento do detector de anomalias pode ser observado na Seção 4.3 que se apresenta como uma metodologia complementar aos limiares definidos pelas redes autoencoders.

3.9 Classificação

Após detectada uma falha, o próximo passo é rotular a mesma, almejando um diagnóstico completo das condições da máquina. Para isso, foram desenvolvidas versões dedicadas para tal propósito das redes autoencoder: DAE e VAE, conhecidas como CDAE e CVAE. Estas redes são treinadas com agrupamentos de espectrogramas rotulados com os defeitos catalogados.

Após projetar o segmento no espaço latente do CDAE e CVAE, utiliza-se o algoritmo SVM para a classificação da falha (VAPNIK, 1999). Este algoritmo é comumente utilizado em regressão e classificação, nas quais os dados são agrupados e delimitados por um hiperplano. Este algoritmo classifica dados desconhecidos baseados em um conjunto de dados previamente rotulado no processo de treinamento. A Figura 3.22 demonstra um exemplo de como são representados os dados no plano bidimensional e como o algoritmo SVM opera:

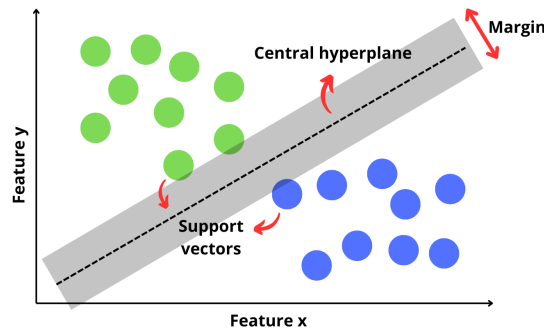


Figura 3.22 – Separação linear de dados pelo algoritmo SVM.

A ideia central do SVM é transformar o dado de entrada em um espaço de dimensão superior utilizando uma função kernel e criar um OSH para distinguir entre duas classes no espaço transformado. Os vetores de dados próximos da margem irão conter informações sobre o OSH e, ao invés do modelo depender de todos os dados para balizar a fronteira dos agrupamentos, os vetores de suporte irão definir este limite; A equação a seguir traz a classificação binária do SVM:

$$g(x) = \text{sign} \left[\sum_{i=1}^n a_i y_i K(x_i, x) + b \right], \quad (3.24)$$

onde é descrita a função de decisão $g(x)$ para n amostras depende do coeficiente de treinamento a_i do rótulo binário y_i (-1 ou 1) e da função Kernel $K(X_i, i)$ que mensura a similaridade entre os pontos adicionados com o *bias* b do neurônio.

A saída desta função retornará o valor -1 ou 1 que determinará a região onde os dados serão rotulados.

Para o presente modelo, os dados projetados no espaço latente pelo CDAE e CVAE apresentam alta dimensionalidade, podendo apresentar uma forma não linear no agrupamento de dados.

O algoritmo SVM classifica a condição de falha com base em um limite predefinido, determinado pela acurácia da respectiva classificação. Isso garante que a detecção de falhas seja precisa e confiável, alinhando-se às métricas de desempenho do modelo para minimizar falsos positivos e negativos. Se a acurácia estiver abaixo do limite predefinido (mesmo que o classificador inicialmente tente atribuir um segmento a um grupo específico), ele será classificado como desconhecido, com a possibilidade de ser analisado e incorporado a um conjunto de dados futuro.

As redes especialistas foram treinadas exclusivamente com segmentos de áudio representativos de duas falhas principais: anel estrela e rolamento. Os conjuntos de dados foram obtidos a partir de períodos imediatamente anteriores à ocorrência dessas falhas, assegurando que os modelos aprendessem as características acústicas específicas associadas a cada tipo de degradação.

- **Anel estrela:** segmentos extraídos no dia 4 de julho, das 08:42 às 11:59.
- **Rolamento:** segmentos extraídos no dia 30 de agosto, das 00:00 às 06:00.

Assim, o objetivo das redes especialistas é distinguir entre as classes Anel estrela e Rolamento, classificando corretamente o tipo de falha presente nos segmentos avaliados.

Tabela 3.4 – Divisão dos dados segmentados em treino e teste para as redes especialistas.

Conjunto	Wyering	Rolamento	Total
Treinamento	400	700	1.100
Teste	200	300	500
Total	600	1.000	1.600

4 Aplicações e Resultados Experimentais

Nesta seção, serão tratadas como as redes autoencoders vistas na seção anterior são aplicadas no processo de detecção de anomalias dos aerogeradores, bem como o processo de manipulação dos dados para detecção. Ao final, têm-se os resultados experimentais em testes laboratoriais.

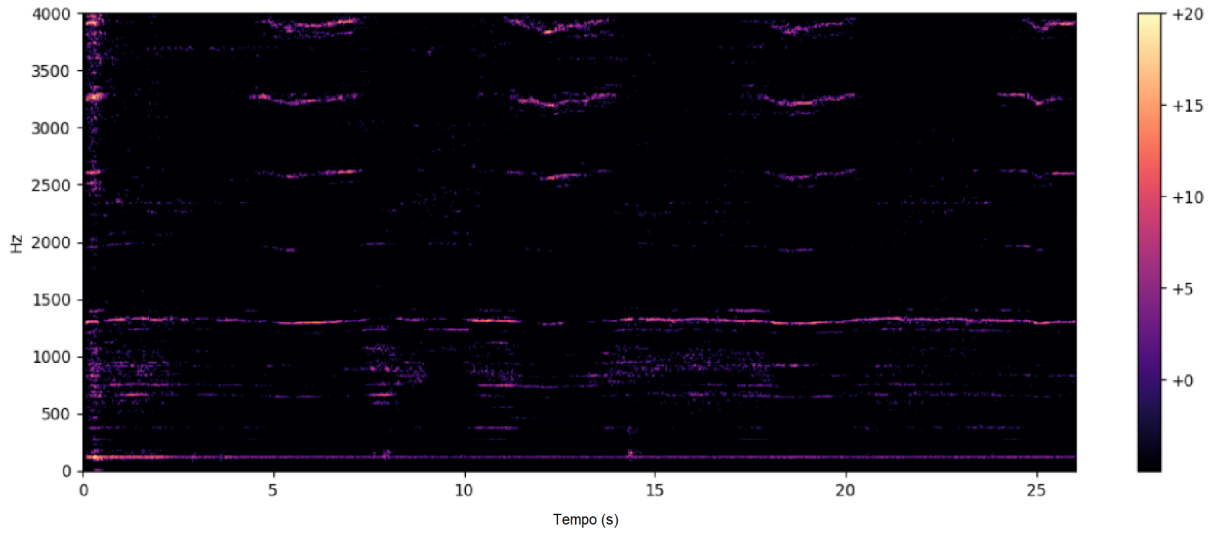
Para garantir a confiabilidade dos autoencoders, são demonstrados, conforme segue nas subseções, dois processos de inferência para detectar duas condições de falta conhecidas em turbinas eólicas específicas.

Após a distinção das máquinas boas das máquinas anômalas, parte-se para a rotulação de falhas, abordando as redes autoencoder CDAE e CVAE em conjunto com o algoritmo SVM para definir algumas das falhas conhecidas pela equipe.

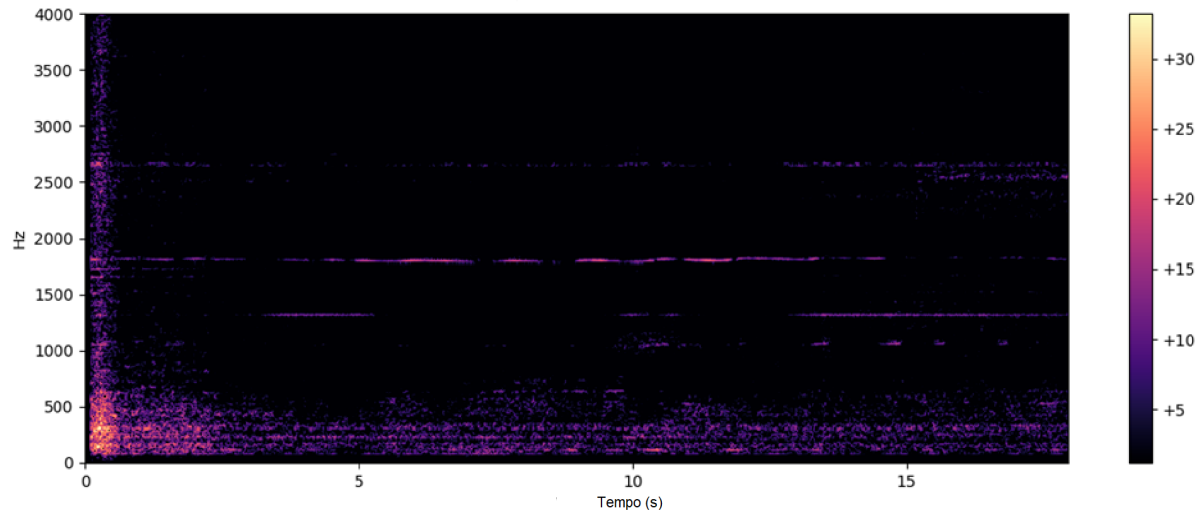
4.1 Testes com Ruídos Sintéticos

Inicialmente, a maioria das máquinas se encontra em funcionamento pleno, o que se traduz em um obstáculo para a obtenção de ruídos reais para análise e validação das redes. Para isso, foram aplicados inicialmente ruídos sintéticos que, ao somados aos áudios dos aerogeradores, simulam um espectrograma de uma máquina defeituosa, que apresente ruídos incomuns.

Os ruídos adicionados foram previamente capturados e isolados pela equipe de campo, sendo classificados como anomalias. Os espectrogramas desses ruídos somados a segmentos de espectrogramas em condições normais estão apresentados na Figura 4.1.



(a)



(b)

Figura 4.1 – Áudios dos ruídos capturados inicialmente como possíveis falhas.

Os testes a seguir da Figura 4.2 até a Figura 4.4 ilustram um típico padrão de operação anômalo das máquinas interpretadas pelas redes VAE/DAE, onde se observa uma diferenciação no comportamento entre os espectrogramas puros e os que contêm ruído adicionado.

4.1.1 Caso 1: Máquina Normal x Ruídos

No primeiro caso, adicionam-se ruídos de anomalias equivalentes a 10% do sinal de áudio original, com o objetivo de reduzir propositalmente os valores de PSNR entre o

dados reconstruído e a entrada da rede autoencoder a fim de observar o seu comportamento frente a ruídos sintéticos adicionados ao áudio.

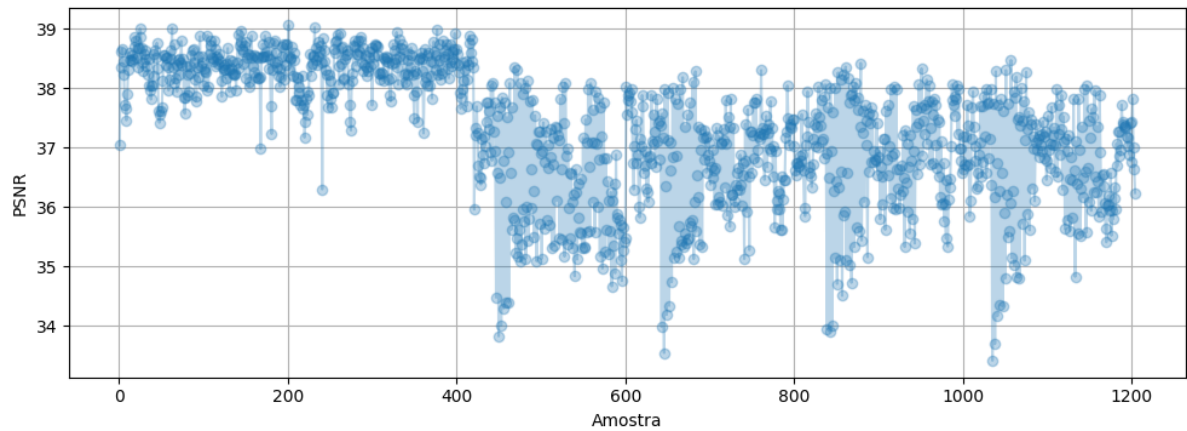


Figura 4.2 – Amostras Normais x Ruídos de 10% do áudio original.

A partir da amostra 410, observa-se uma redução contínua nos valores da métrica PSNR, ressaltando sua capacidade de identificar e mensurar deteriorações no sinal acústico.

4.1.2 Caso 2: Máquina Normal x Ruídos PSNR

A seguir, faz-se outro teste de aplicação do ruído sintético para um diferente conjunto de dados:

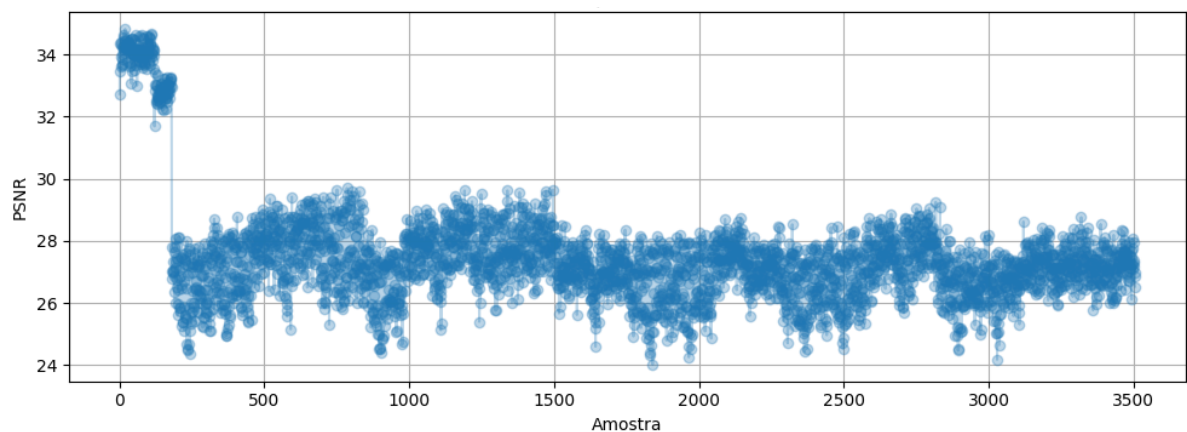


Figura 4.3 – Outro comparativo adicionando ruído aos dados a partir da amostra duzentos.

A Figura 4.3 destaca essa queda significativa da métrica PSNR após a introdução do ruído por volta da amostra 200, evidenciando a sensibilidade da métrica na identificação de degradações sonoras.

4.1.3 Caso 3: Máquina Normal x Ruídos SSIM

Nesta etapa, é testada a sensibilidade da métrica SSIM frente à adição de ruídos sintéticos, com o mesmo objetivo de avaliar sua capacidade de detectar alterações estruturais no conteúdo do sinal reconstruído em relação ao original.

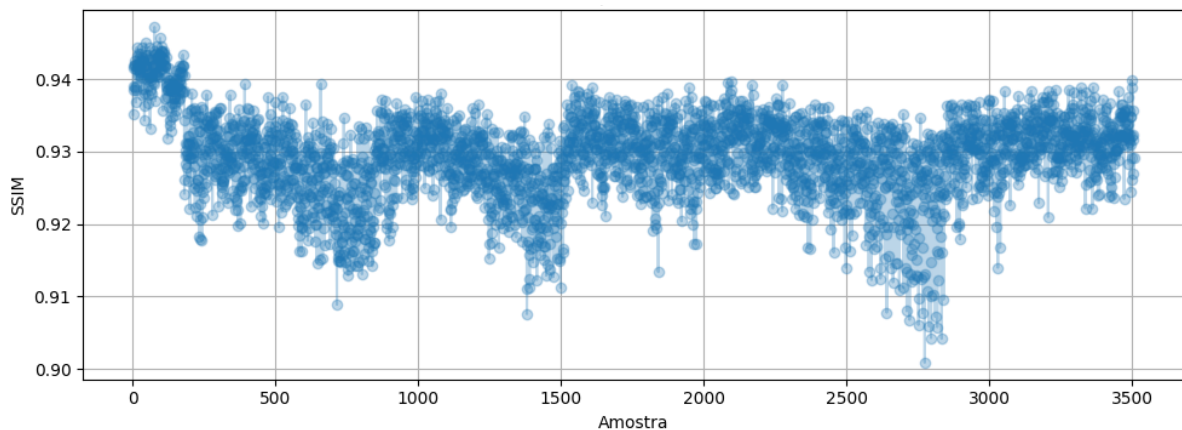


Figura 4.4 – Comparativo adicionando ruído aos dados a partir da amostra duzentos com a métrica SSIM.

A partir da amostra 200 dos dados da Figura 4.4, observa-se que a métrica SSIM apresenta variação menos significativa que o PSNR, mantendo valores mais estáveis frente às interferências, mas ainda menores que a normalidade:

Assim, tem-se um indicativo inicial de distinção de ruídos simulados no que diz respeito à faixa de PSNR e SSIM onde ficam concentrados os dados.

4.2 Separação de Máquina Defeituosa

Nesta etapa, são relatados testes a respeito de duas falhas conhecidas pela equipe de campo previamente observadas pela equipe de laboratório referentes ao comportamento dos espectrogramas das máquinas defeituosas frente às máquinas de operação normal. São relatadas por meio de gráficos e tabelas as métricas de comparação entre o dado reconstruído e o dado de entrada das redes autoencoder.

Esta etapa visa avaliar inicialmente a capacidade dos autoencoders em diferenciar os áudios de máquinas em boas condições para as máquinas em condições adversas reais, relacionando os relatos da equipe de campo e do SCADA que monitora os aerogeradores com a similaridade de reconstrução das IAs. Com isso, foram identificados dois tipos de falhas nos aerogeradores monitorados: falha no rolamento e falha no gerador de indução.

4.2.1 Detecção de Falha no Rolamento

A falha no rolamento foi inicialmente identificada por meio da inspeção visual do espectrograma da máquina FAI2-07, que apresentava um padrão distinto em comparação às demais máquinas que mostra os espectrogramas no mesmo instante de captura:

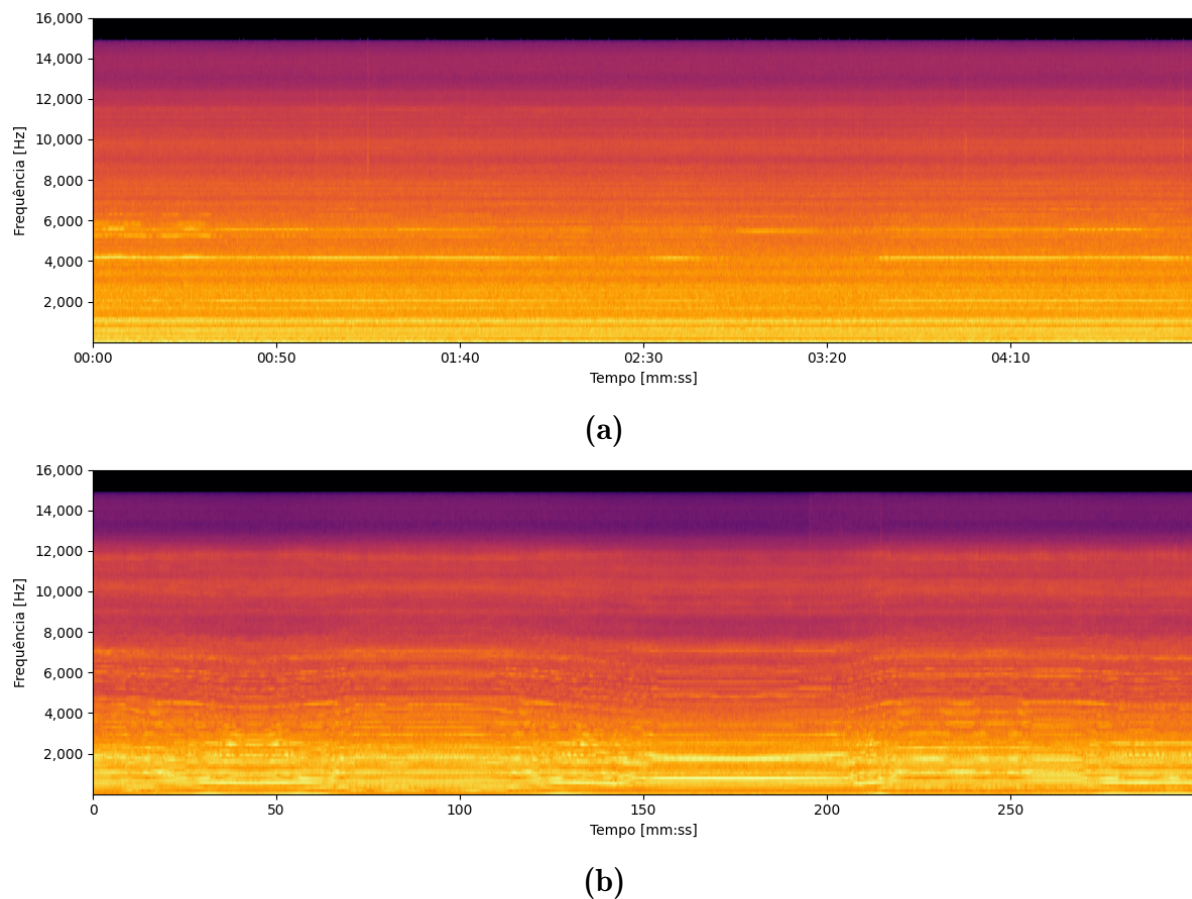


Figura 4.5 – Comparativo de áudio **(a)** de uma turbina normal (FAI1-05) e **(b)** da turbina apresentando falha no rolamento (FAI2-07).

Observa-se que o primeiro espectrograma da Figura 4.5 apresenta um padrão mais homogêneo, característico do funcionamento de uma máquina em condições normais. Já o espectrograma inferior destaca-se pela presença de componentes significativos em frequências abaixo de 4 kHz.

Para a realização deste primeiro teste de inferência, segmentos de espectrogramas de condições normais de operação foram capturados das quatro turbinas normais, enquanto a turbina FAI2-07, que apresenta uma falha conhecida no seu rolamento, também foi capturada para análises:

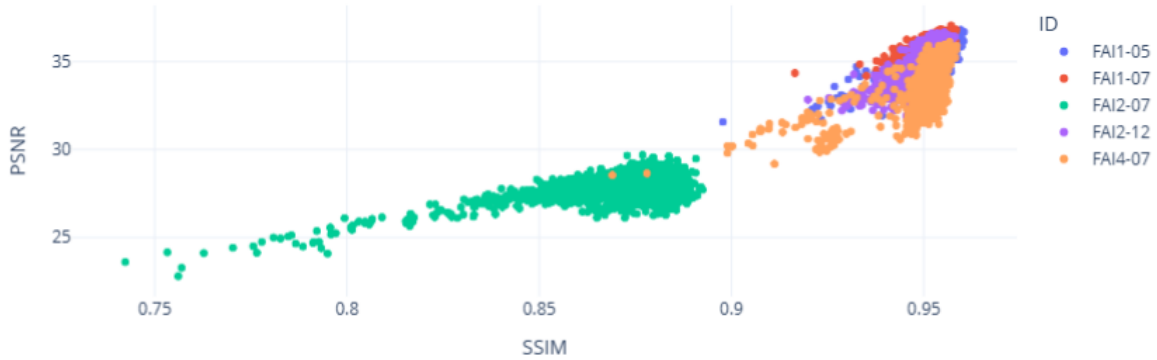


Figura 4.6 – Inferência do VAE por meio da análise do SSIM e PSNR.

Cada ponto no gráfico da Figura 4.6 representa o resultado da reconstrução de um segmento de 5 segundos de espectrograma dos áudios gravados em cada um dos cinco aerogeradores analisados. Em cada um dos eixos dos gráficos estão representadas as métricas de similaridade abordadas na Seção 3.7. Com isso, é perceptível que o aerogerador defeituoso em questão (FAI2-07) apresenta baixa similaridade tanto por meio do SSIM quanto por PSNR comparado aos outros aerogeradores analisados, haja vista que seus segmentos estão concentrados abaixo dos segmentos das demais máquinas. As Figuras 4.7 e 4.8 trazem outra perspectiva da evolução das métricas no eixo do tempo:

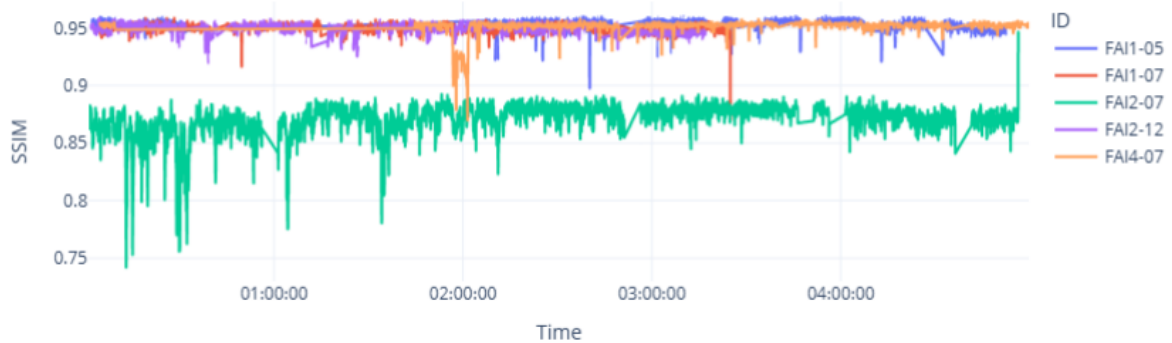


Figura 4.7 – Inferência do SSIM analisada no domínio de tempo.

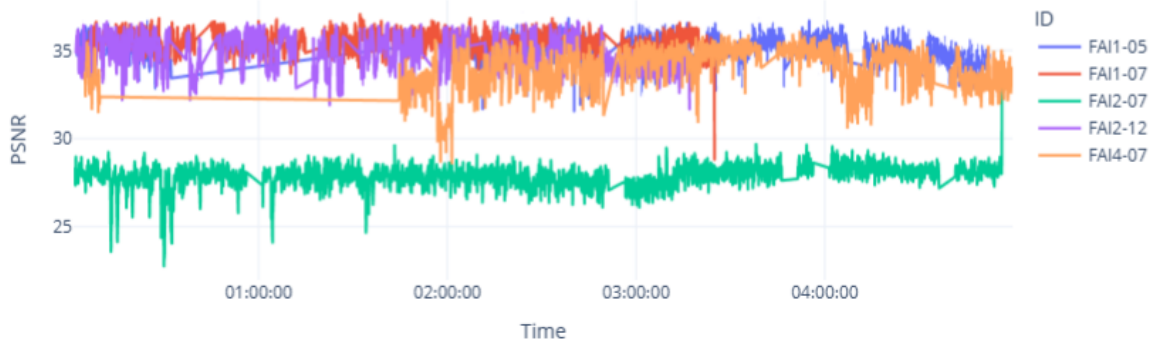


Figura 4.8 – Inferência do PSNR analisada no domínio de tempo.

É notável, novamente, que através da linha verde abaixo das outras que o aerogerador FAI2-07 apresenta baixa similaridade entre os dados de reconstrução e os dados de entrada para o período selecionado. Da mesma forma, é retratado este comportamento pelo comportamento do PSNR ao longo do tempo abaixo na Figura 4.8.

Verifica-se a ausência de valores de métricas, sobretudo na máquina FAI4-07, entre 0h00 e aproximadamente 2h00, em razão do corte provocado pelo limiar de gravação descrito anteriormente.

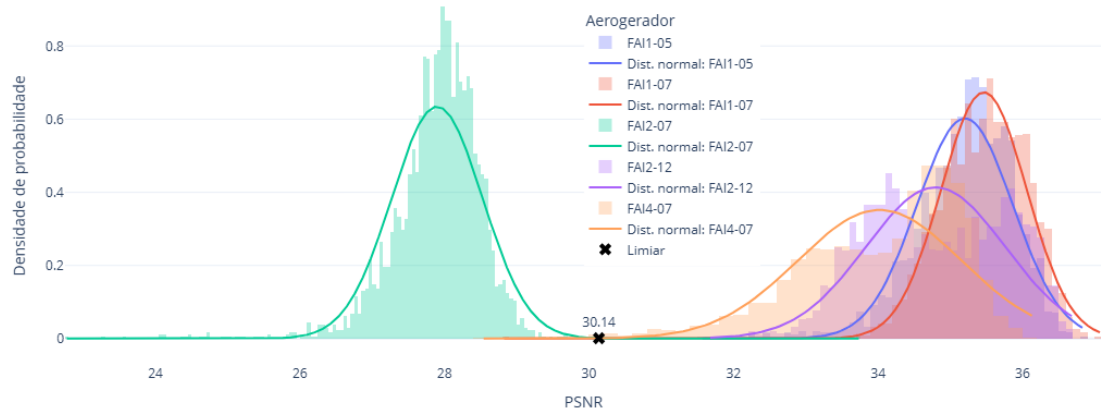
A Tabela 4.1 a seguir traz consigo o resultado da inferência em termos numéricos:

Tabela 4.1 – Resultado da análise de falha no rolamento.

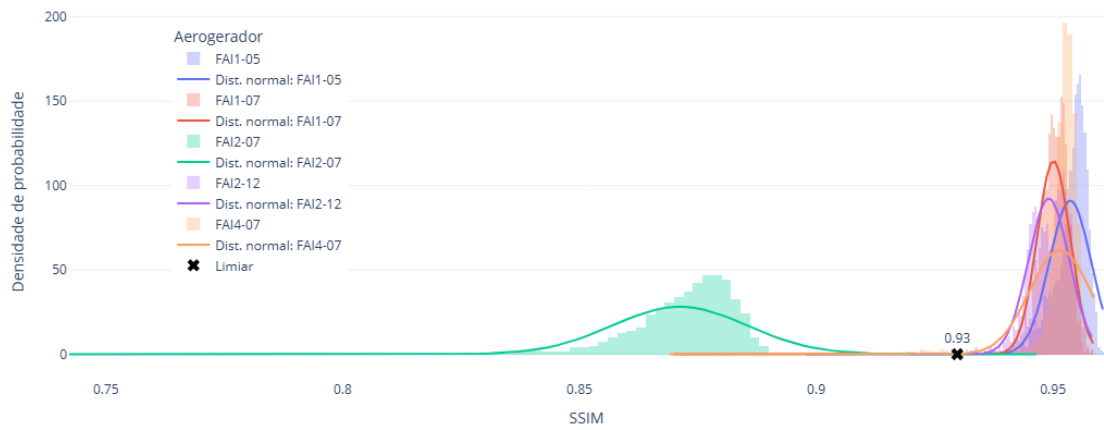
Id aerogerador	SSIM		PSNR	
	Média	Desvio padrão	Média	Desvio padrão
FAI1-05	0.954	0.004	35,20	0,66
FAI1-07	0.950	0.003	35,47	0,59
FAI2-07	0.871	0.014	27,90	0,63
FAI2-12	0.949	0.004	34,80	0,96
FAI4-07	0.952	0.006	34,02	1,13

Nisto, o aerogerador FAI2-07 se destaca pela baixa média das métricas de PSNR e SSIM frente às demais máquinas em condições normais, o que possibilita diferenciá-lo e classificar a condição como defeituosa.

A Figura 4.9 apresenta a distribuição normal dos dados detalhadamente, permitindo a definição de um limiar mais preciso. Esse limiar, obtido a partir do ponto de maior interseção entre as curvas dos aerogeradores em condição anômala e em bom estado (com menor dispersão), contribui para a redução de falsos positivos e falsos negativos na detecção de falhas:



(a)



(b)

Figura 4.9 – Distribuição normal do SSIM e PSNR da análise de (a) rolamento e (b) limiar.

Com base na análise desta amostragem, pode-se adotar provisoriamente o valor de $\text{PSNR} = 30,14$ e $\text{SSIM} = 0,93$ como limiares de separação entre aerogeradores em condição normal e anômala. Esse valor representa o ponto de interseção entre a distribuição de uma máquina cujo padrão se desvia significativamente dos demais, servindo como referência para discriminar comportamentos atípicos.

4.2.2 Detecção de Falha no DFIG

Adicionalmente, foi estudada uma falha detectada pelo SCADA da empresa gerenciadora; foi a falha no DFIG, um importante dispositivo que otimiza a geração de energia para ventos variáveis. A Figura 4.10 demonstra a captura dos segmentos relativos a esse evento de falha relatado na turbina FAI4-07 em um diferente período de tempo no mês de novembro de 2024, cuja manifestação visual não se distingue claramente em relação à falha anterior.

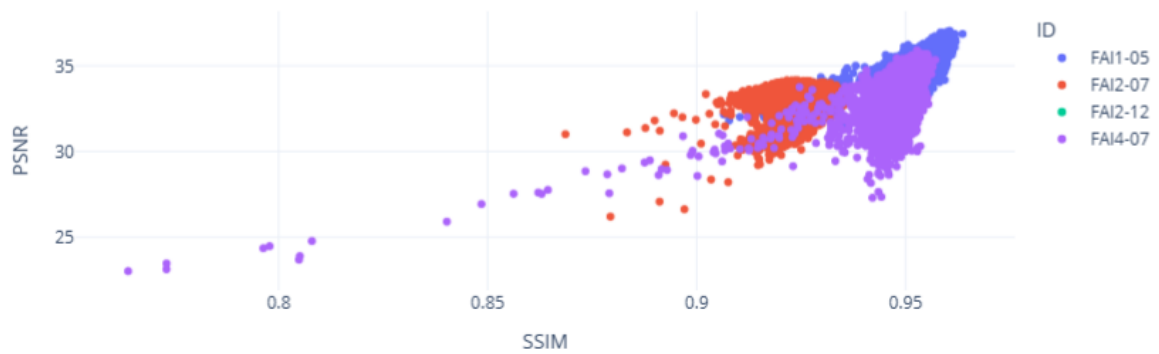


Figura 4.10 – Inferência da análise de falha no componente gerador de indução.

Como pode ser visto na Figura 4.10, os segmentos de espectrograma relativos à turbina FAI4-07 vão se desagrupando das máquinas consideradas em condições normais e vão em direção à máquina FAI2-07, que ainda se apresentava defeituosa. Vale ressaltar que a ausência de dados da máquina FAI1-07 é devida à mesma não possuir áudios gravados no período analisado por motivos de conectividade.

A Tabela 4.2 traz consigo os dados estatísticos relativa à janela de tempo analisada em termos de desvio padrão e média dos dados:

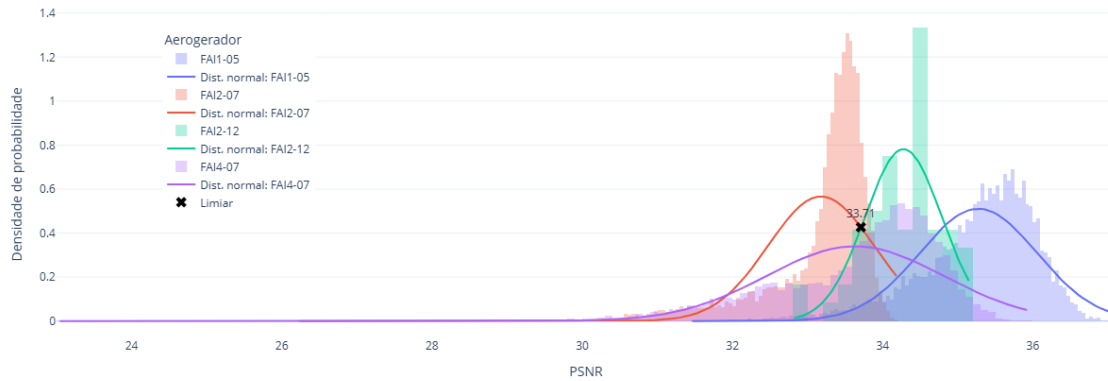
Tabela 4.2 – Análise da inferência para detecção de falha no gerador de indução.

Id aerogerador	SSIM		PSNR	
	Média	Desvio padrão	Média	Desvio padrão
FAI1-05	0.954	0.005	35.274	0.781
FAI2-07	0.923	0.005	33.180	0.704
FAI2-12	0.951	0.004	34.277	0.510
FAI4-07	0.949	0.007	33.632	1.173

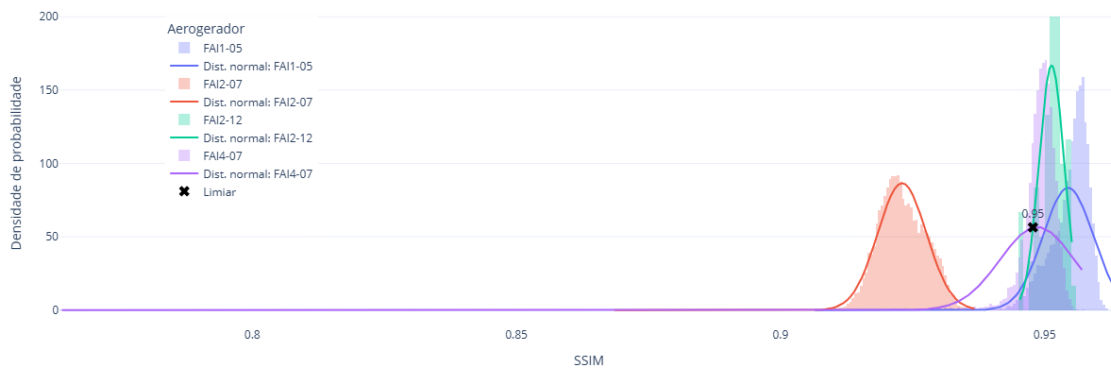
A Tabela 4.2 demonstra em números a baixa média das métricas de SSIM e

PSNR, assim como um alto desvio padrão das métricas sinalizando o deslocamento dos segmentos dos áudios da turbina FAI4-07 em relação às demais.

Retomando a mesma análise de distribuição feita no teste anterior, tem-se na Figura 4.11 a distribuição normal das métricas dos segmentos de espectrograma analisados para o caso desta falha.



(a)



(b)

Figura 4.11 – Distribuição normal do SSIM e PSNR da análise de (a) falha no DFIG e (b) limiar.

Considerando que a máquina FAI2-07 ainda apresentava falha no rolamento segundo a análise dos operadores, os limiares foram definidos com base na separação dos dois aerogeradores (FAI2-07 E FAI4-07) do restante.

Observa-se que, o limiar do SSIM subiu para o valor de 0,95 que classificaria aproximadamente metade dos dados do aerogerador FAI4-07 como anômalos. O valor de PSNR de 33,79 também produz um efeito semelhante além de capturar falso positivos da

máquina FAI2-12.

A análise realizada revela que, embora a máquina FAI4-07 apresente um desvio padrão elevado, sua curva de distribuição permanece alinhada às demais máquinas em condições operacionais adequadas. Diante disso, propõe-se uma análise complementar, fundamentada na variação temporal do desvio padrão das métricas. Essa abordagem, aliada ao limiar previamente estabelecido, contribui para um diagnóstico mais robusto. Nesse contexto, destaca-se a aplicação sequencial do filtro de Kalman como ferramenta de refinamento da detecção de anomalias.

4.3 Implementação do Detector de Anomalias

Nesta etapa de teste, tem-se o princípio de funcionamento do detector de anomalias que utiliza o filtro de Kalman em duas janelas temporais abordadas na Seção 3.8 para a suavização das métricas e cálculo de parâmetros estatísticos que auxiliam no envio de notificação para o sistema.

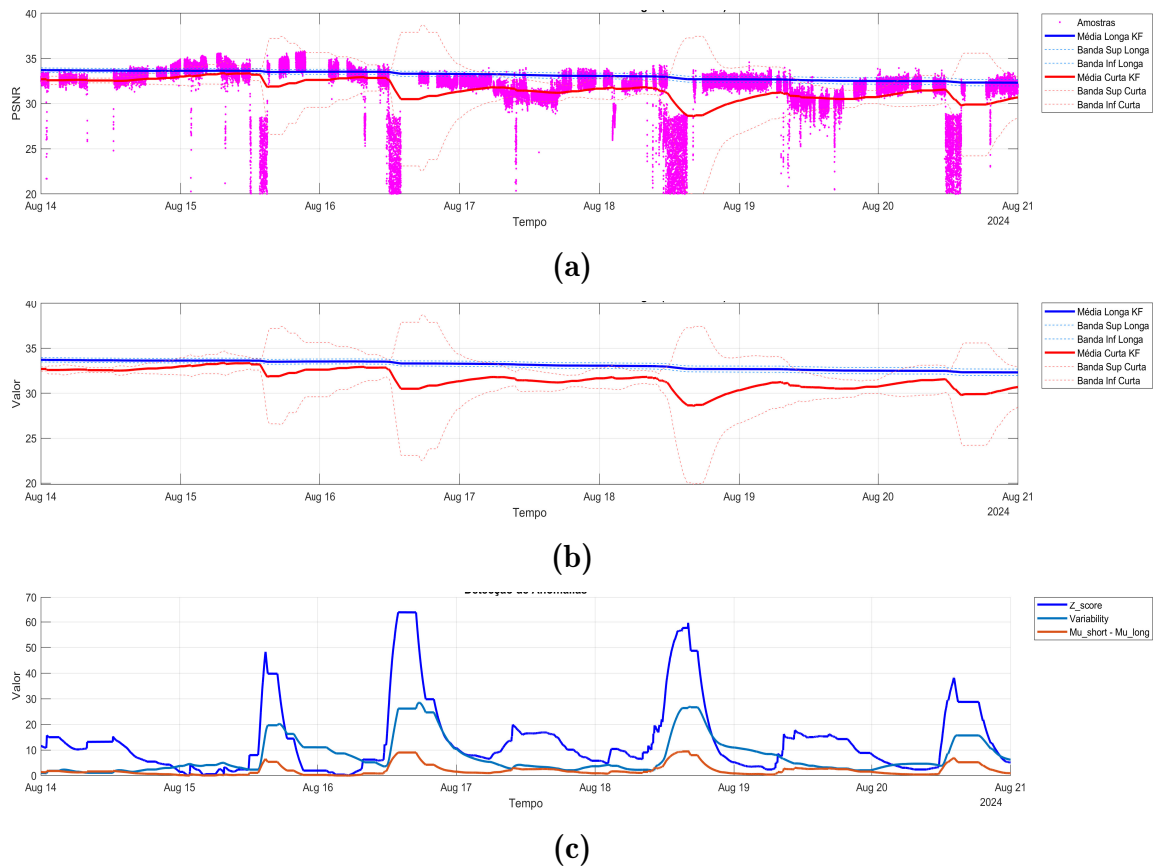


Figura 4.12 – Detector de anomalias utilizando Filtro de Kalman trazendo **(a)** a plotagem das amostras juntamente com as médias e Bandas de Bollinger; **(b)** retirada das amostras; e **(c)** acionamento do detector de anomalias pelos parâmetros.

A Figura 4.12 apresenta alguns resultados para o detector de anomalias para uma máquina. No primeiro gráfico, têm-se as amostras dos segmentos de 5 segundos de espectrograma plotadas (em rosa) juntamente com as médias curtas e longas e a demarcação das bandas superiores e inferiores de cada média.

Logo abaixo, no segundo gráfico, é retirada a plotagem das amostras, com o destaque para as médias e suas respectivas bandas de Bollinger, traçando a faixa de variabilidade do sinal.

No terceiro gráfico, tem-se o detector de anomalias para os rigorosos critérios: diferença entre médias curta e longa (em laranja), a variabilidade pela curva (em ciano) e z-score (em azul) entre as métricas de similaridade.

Observa-se a partir do dia 15 que todas as métricas apresentam picos simultâneos, sinalizando claramente uma anomalia detectada no sistema. A concordância entre os três indicadores mencionados reduz significativamente falsos positivos e confere maior confiabilidade ao mecanismo de alerta.

Na Figura 4.13 tem-se uma evolução no detector de anomalias ao capturar um novo conjunto de segmentos de um mês inteiro:

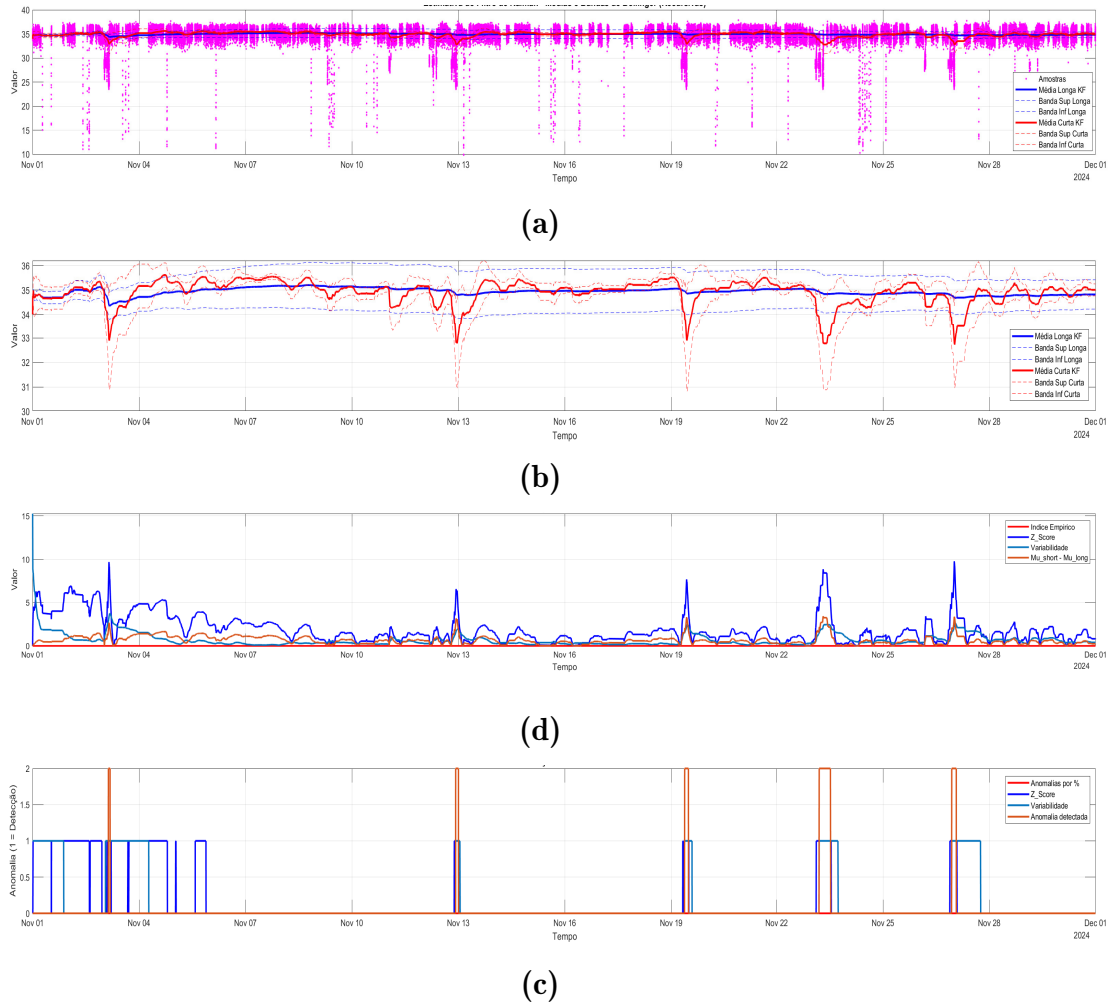


Figura 4.13 – Detector de anomalias utilizando Filtro de Kalman, apresentando: **(a)** a plotagem das amostras com as médias e as Bandas de Bollinger; **(b)** a retirada das amostras; **(c)** o acionamento contínuo do detector de anomalias com base nos parâmetros; e **(d)** o acionamento discreto do detector de anomalias com base nos mesmos parâmetros.

Este gráfico traz os momentos em que o detector de anomalias é acionado de forma mais clara, representado em forma binária e por meio da convergência dos três critérios vistos anteriormente.

4.4 Aplicação das Redes Especialistas

Após inferências sucessivas para testar o modelo e definir o limite da métrica para distinguir normais e anômalas, os segmentos são enviados para as IAs especialistas.

Estas IAs são previamente treinadas com espectrogramas de falhas em mancais, recebem esses dados desconhecidos com baixo nível de similaridade e os classificam adequadamente, como demonstrado nas Figuras 4.14 e 4.15, permitindo o agrupamento dos

dados de treinamento para as diferentes condições capturadas. Esse agrupamento utiliza o algoritmo t-distributed Stochastic Neighbor Embedding (t-SNE) para agrupar dados semelhantes:

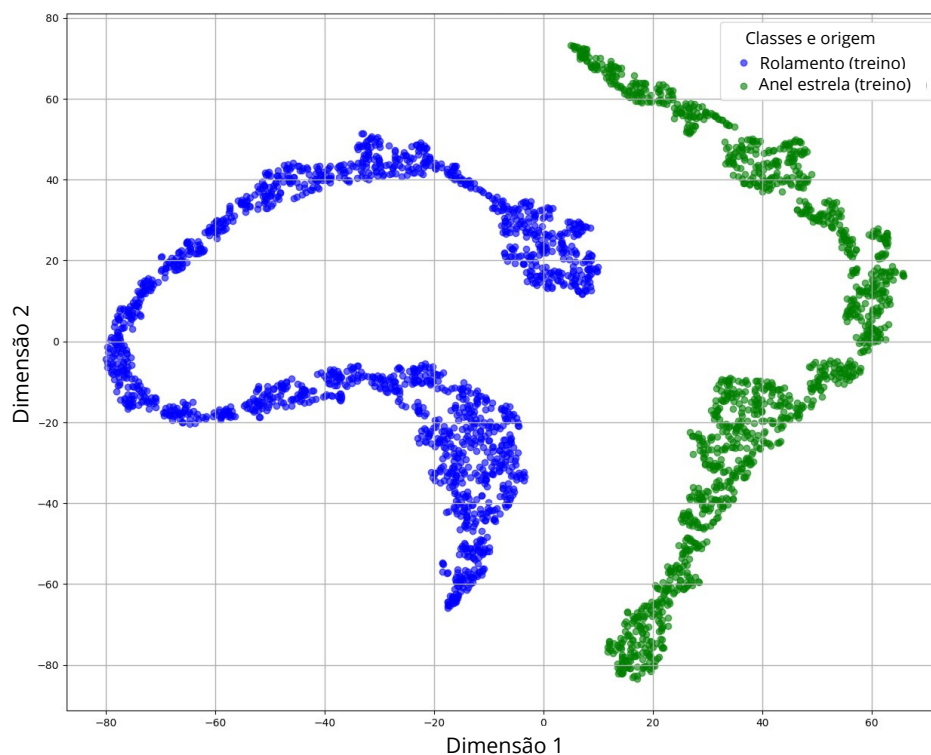


Figura 4.14 – Espaço latente gerado pelo CDAE.

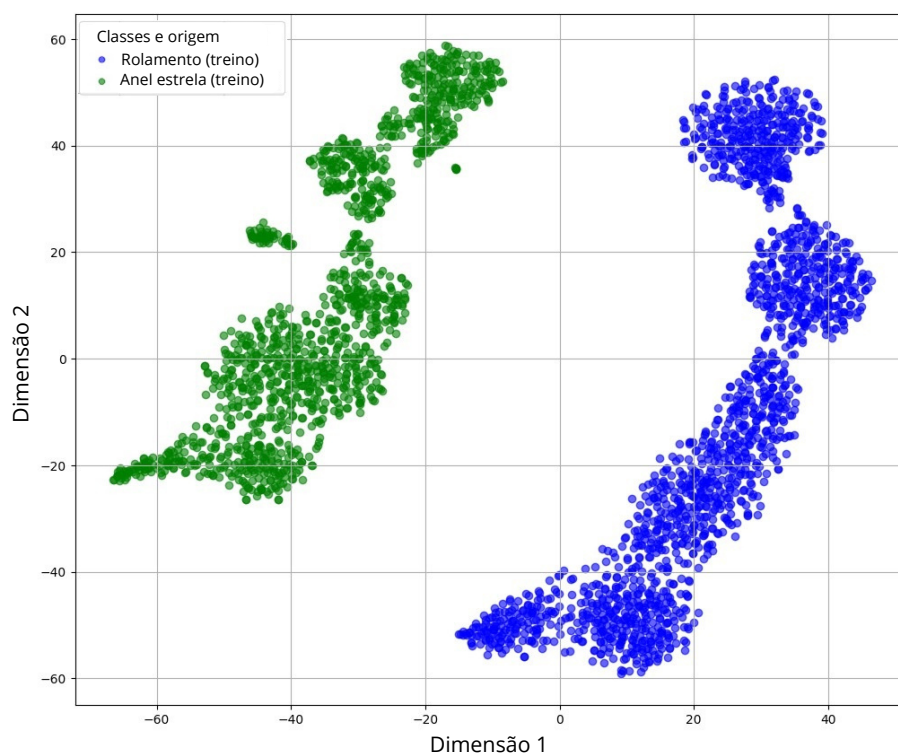


Figura 4.15 – Espaço latente gerado pelo CVAE.

A abordagem CVAE cria um espaço latente no qual os dados de entrada são codificados. Esse espaço latente captura as características essenciais dos espectrogramas, possibilitando uma representação compacta e significativa dos dados. Semelhantemente, o CDAE cria um espaço latente e emprega técnicas de remoção de ruído para melhorar a reconstrução dos espectrogramas, focando na remoção de ruídos que poderiam mascarar sinais iniciais de falhas.

O espaço latente mostra que as falhas conhecidas no mancal e no anel estrela (wey ring), que também são falhas identificadas pelos operadores, foram agrupadas em dois grupos distintos. Essa característica garante que o CDAE e o CVAE possam agrupar e classificar diferentes dados representados pelos segmentos do espectrograma em um espaço bidimensional rotulado como “Dimensão 1” e “Dimensão 2”.

Após essa projeção no espaço latente, o algoritmo SVM é aplicado para classificar as diferentes condições de falha. As Figuras 4.16 e 4.17 demonstram a combinação da classificação utilizando o algoritmo SVM e o espaço latente gerado pelos autoencoders. Essas figuras comparam os dados de treinamento e inferência, demonstrando a aproximação entre eles:

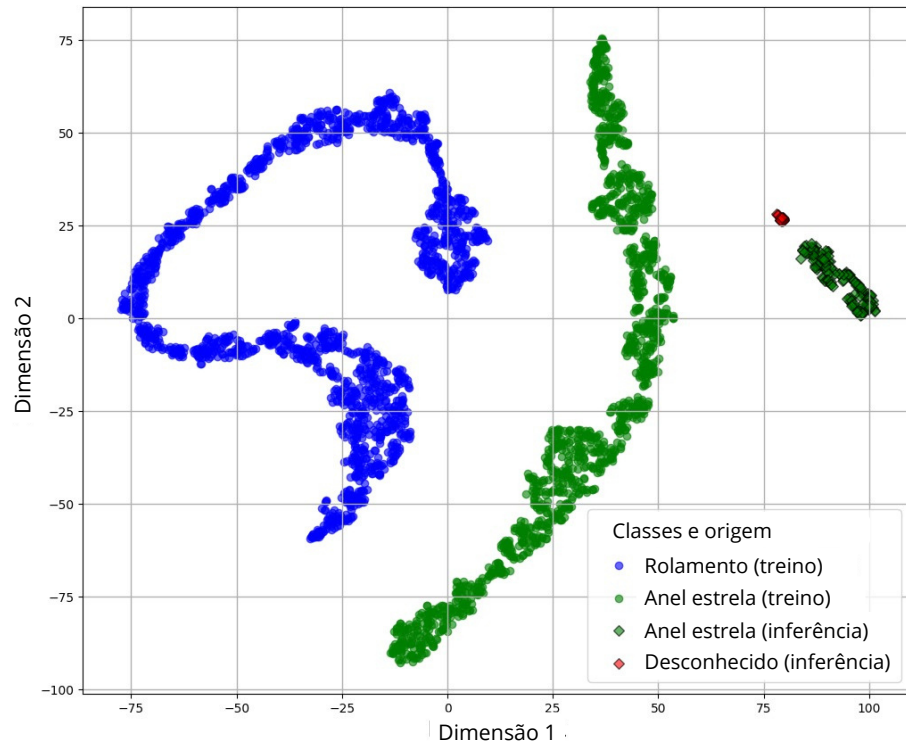


Figura 4.16 – Aplicação do SVM no espaço latente do CDAE para detecção de falha no anel estrela.

A Figura 4.16 apresenta a atuação da rede especialista CDAE para a detecção de falha no anel estrela, correspondendo a uma condição de falha adicional, analisada e aprendida pela rede autoencoder durante o treinamento.

Nota-se que a maioria dos dados de inferência ficou concentrada à direita da ilha dos dados de treinamento da mesma falha, indicando que ambos os clusters podem convergir em mesmo tipo de dado. O que evidencia a medida correta dos dados.

Alguns poucos segmentos foram classificados como desconhecidos pela rede, seja por pertencerem a um cluster diferente ou por não alcançarem os valores de acurácia estipulados:

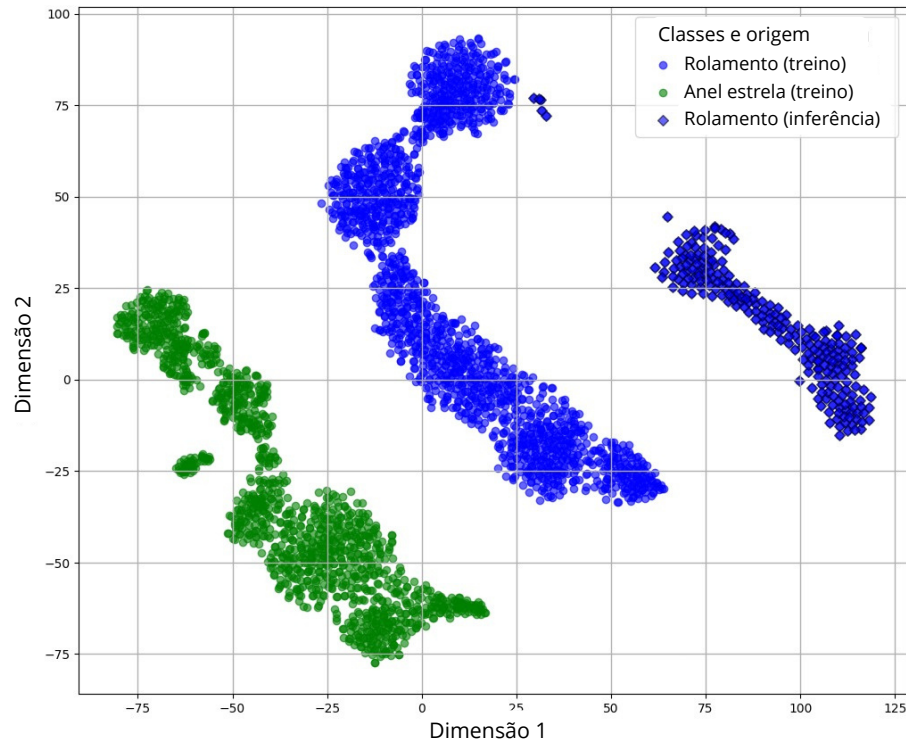


Figura 4.17 – Aplicação do SVM no espaço latente do CVAE para detecção de falha no mancal.

A Figura 4.17, por sua vez, apresenta o teste de inferência para a falha no rolamento do gerador, cujos resultados demonstraram consistentes, principalmente devido ao espectrograma dessa condição apresentar características visivelmente distintas em relação às demais falhas.

Em resumo, a presente análise comparou os dados de treinamento (indicados por círculos) com os dados de inferência (indicados por losangos de cores escuras) para determinar seu nível de similaridade. Os dados desconhecidos foram agrupados próximos aos dados conhecidos, validando a precisão da técnica SVM.

A Tabela 4.3 apresenta os resultados do CVAE e do CDAE para 200 amostras (segmentos de espectrogramas) de falhas rotuladas, bem como de condições ruidosas e normais. Para o áudio com ruído, distribuições gaussianas de 0,1, 0,2 e 0,3 foram aplicadas. Neste exemplo, foi adotado um limite de acurácia de 90%:

Tabela 4.3 – Acurácia para dados rotulados e não rotulados nas classificações com *CVAE* e *CDAE*.

Tipo	Acurácia (%)		Classificação da Falha
	CVAE	CDAE	
Rolamento (Bearing)	99	99	Rolamento
Anel Estrela (Wye Ring)	98	98	Anel Estrela
Áudio com Ruído 1 — 0,1	35	30	Falha Desconhecida
Áudio com Ruído 1 — 0,2	28	25	Falha Desconhecida
Áudio com Ruído 1 — 0,3	20	18	Falha Desconhecida
Áudio com Ruído 2 — 0,1	33	29	Falha Desconhecida
Áudio com Ruído 2 — 0,2	27	24	Falha Desconhecida
Áudio com Ruído 2 — 0,3	19	16	Falha Desconhecida
Áudio Normal 1	22	18	Falha Desconhecida
Áudio Normal 2	17	14	Falha Desconhecida
Áudio Normal 3	8	7	Falha Desconhecida

Ambos os autoencoders especialistas foram treinados com falhas nos mancais e no anel estrela, reconhecendo-os com alta precisão. Para outras condições, o modelo tentou agrupar os dados nos clusters treinados, mas os classificou como falhas desconhecidas com baixa precisão.

5 Considerações Finais

O trabalho proposto teve como objetivo desenvolver um algoritmo de detecção de falhas em aerogeradores, tendo como fonte primária de detecção o som emitido pelas suas partes mecânicas constituintes. Os testes realizados demonstram a capacidade do algoritmo de diferenciação de aerogeradores defeituosos e possibilitam um alerta antes que o problema se agrave. Os testes foram divididos em três estágios principais:

- Testes iniciais com ruídos sintéticos: Esta etapa consiste em simular ruídos anômalos e somá-los ao espectrograma para avaliar a capacidade de distinção da rede devido à ausência de condições anômalas pontuais nos primeiros dias de gravação de áudio dos aerogeradores. Observa-se uma queda nas métricas ao adicionar ruídos simulados para o áudio;
- Testes de clusterização: Esta etapa teve como objetivo a demonstração da capacidade das redes elaboradas em agrupar espectrogramas de aerogeradores em boas condições separados de um aerogerador defeituoso. Utilizando as métricas de similaridade para a comparação do dado reconstruído pela rede com o dado original de entrada, percebe-se uma separação notória e diferenciação pela média das métricas e desvio padrão;
- Elaboração do detector de anomalias: Após testar sucessivas falhas, foi elaborado o algoritmo de detecção que captura a média longa e curta das métricas e, por meio de indicadores estatísticos, chega-se numa convergência de resposta para o acionamento de alarme de falha;
- Aplicação de redes especialistas: Tendo uma rede que distingue aerogeradores defeituosos de normais, fizeram-se testes para a designação da falha, baseados nas falhas conhecidas pela equipe, obtendo resultados de alta precisão com o algoritmo chegando entre 98 e 99% de precisão para as falhas catalogadas.

Assim, o acionamento da equipe de manutenção é feito a curto prazo, poupando prejuízos para as empresas do setor elétrico e minimizando o incômodo para moradores próximos a parques eólicos.

5.1 Trabalhos Futuros

Este projeto abre caminho para diversas linhas de pesquisa e desenvolvimento futuro:

- Extração de características do sinal: Para obter um diagnóstico mais robusto oferecendo maior redundância, pretende-se extrair características baseadas em dados estatísticos dos áudios. Desta forma, um algoritmo de votação para decidir em qual condição final o aerogerador se encontra;
- Aplicação em outros setores industriais: A metodologia utilizada para a detecção de falhas em aerogeradores utilizando como fonte primária o som, permite expandir a aplicação para diferentes ambientes que utilizem de máquinas rotativas possibilitando o treinamento de redes não-supervisionadas com os dados de condições normais destas máquinas e o uso destas redes para a distinção das condições normais e anômalas;
- Expansão do Sistema: O projeto encontra-se em fase de ampliação, com a inclusão de novos dispositivos no processo de captura de falhas, além dos cinco aerogeradores inicialmente testados. Esse aumento da base de dados exigirá constantes ajustes nos modelos, a fim de adaptá-los às possíveis novas características acústicas apresentadas pelos diferentes equipamentos. Existe, inclusive, a possibilidade de que padrões sonoros inéditos — ainda não identificados pela rede — representem condições de falha. A expansão do sistema prevê o monitoramento de aproximadamente 60 aerogeradores, impondo um desafio significativo à capacidade de generalização dos modelos, especialmente no que diz respeito à otimização dos espectrogramas sem comprometer a qualidade de reconstrução dos sinais provenientes de aerogeradores em boas condições de operação.

Referências

- ABEEólica. *Brasil sobe uma posição e fica em 5º lugar no Ranking Mundial de Energia Eólica*. 2025. <<https://abeeolica.org.br/brasil-sobre-uma-posicao-e-fica-em-5o-lugar-no-ranking-mundial-de-energia-eolica/>>. Publicado em 23 de abril de 2025.
- ALI, B. M. Wind energy prediction: Artificial intelligence perspective. In: *2023 6th International Conference on Engineering Technology and its Applications (IICETA)*. [S.l.: s.n.], 2023. p. 885–891.
- AN, J.; CHO, S. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, v. 2, n. 1, p. 1–18, 2015.
- Armando de Oliveira Lima. *Eólicas cearenses perderam R\$ 104,5 mi com interrupção de geração em 2024*. 2024. <<https://mais.opovo.com.br/colunistas/armando-de-oliveira-lima/2024/12/03/eolicas-cearenses-perderam-rs-1045-mi-com-interruptcao-de-geracao-em-2024.html>>. Publicado em 03 de dezembro de 2024.
- BANSAL, A.; GARG, N. K. Environmental sound classification: A descriptive review of the literature. *Intelligent Systems with Applications*, v. 16, p. 200115, 2022. ISSN 2667-3053. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2667305322000539>>.
- BI, S. Research on image processing technology based on artificial intelligence and visual algorithms. In: *2024 Second International Conference on Data Science and Information System (ICDSIS)*. [S.l.: s.n.], 2024. p. 1–4.
- BISHOP, C. *Neural Networks for Pattern Recognition*. [S.l.]: Oxford, U.K.: Oxford Univ. Press, 1995.
- BOLLINGER, J. *Bollinger on Bollinger Bands*. New York, NY, USA: McGraw-Hill Education, 2002. (Professional Finance & Investment).
- BU, J. *et al.* Operation and maintenance strategy of wind turbine based on maintenance time window. In: *2023 2nd Asian Conference on Frontiers of Power and Energy (ACFPE)*. [S.l.: s.n.], 2023. p. 711–715.
- CARRATÙ, M. *et al.* A novel methodology for unsupervised anomaly detection in industrial electrical systems. *IEEE Transactions on Instrumentation and Measurement*, v. 72, p. 1–12, 2023.
- CHAVDAR, M. *et al.* Towards a system for automatic traffic sound event detection. In: *2020 28th Telecommunications Forum (TELFOR)*. [S.l.: s.n.], 2020. p. 1–4.
- CHEN, H. *et al.* Predictive maintenance of abnormal wind turbine events by using machine learning based on condition monitoring for anomaly detection. *Journal of Mechanical Science and Technology*, v. 35, n. 12, p. 5323–5333, 2021. ISSN 1976-3824. Disponível em: <<https://doi.org/10.1007/s12206-021-1105-z>>.

CHOI, K.; FAZEKAS, G.; SANDLER, M. Convolutional recurrent neural networks for music classification. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2017. p. 2392–2396.

Elera Renováveis. *Relatório de Sustentabilidade - 2024*. 2024.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: GORDON, G.; DUNSON, D.; DUDÍK, M. (Ed.). *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale, FL, USA: PMLR, 2011. (Proceedings of Machine Learning Research, v. 15), p. 315–323. Disponível em: <<https://proceedings.mlr.press/v15/glorot11a.html>>.

GONDARA, L. Medical image denoising using convolutional denoising autoencoders. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. [S.l.: s.n.], 2016. p. 241–246.

GRANADOS, D. P. *et al.* A wind turbine vibration monitoring system for predictive maintenance based on machine learning methods developed under safely controlled laboratory conditions. *Energies*, v. 16, n. 5, 2023. ISSN 1996-1073. Disponível em: <<https://www.mdpi.com/1996-1073/16/5/2290>>.

GUO, C.; LIU, Y. lan; JIAO, X. Study on the influence of variable stride scale change on image recognition in cnn. *Multimedia Tools and Applications*, v. 78, n. 21, p. 30027–30037, 2019. ISSN 1573-7721. Disponível em: <<https://doi.org/10.1007/s11042-018-6861-0>>.

HE, A.; JIN, X. Deep variational autoencoder classifier for intelligent fault diagnosis adaptive to unseen fault categories. *IEEE Transactions on Reliability*, v. 70, n. 4, p. 1581–1595, 2021.

HORÉ, A.; ZIOU, D. Is there a relationship between peak-signal-to-noise ratio and structural similarity index measure? *IET Image Processing*, v. 7, n. 1, p. 12–24, 2013. Disponível em: <<https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-ipr.2012.0489>>.

HU, M.; WANG, G.; MA, K. Identification of wind turbine gearbox weak compound fault based on optimal empirical wavelet transform. *Measurement Science and Technology*, IOP Publishing, v. 34, n. 4, p. 045003, jan 2023. Disponível em: <<https://dx.doi.org/10.1088/1361-6501/acacb9>>.

HUANG, G. *et al.* *Densely Connected Convolutional Networks*. 2018. Disponível em: <<https://arxiv.org/abs/1608.06993>>.

IOFFE, S.; SZEGEDY, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. Disponível em: <<https://arxiv.org/abs/1502.03167>>.

JOLLIFFE, I. T. *Principal Component Analysis*. 2. ed. [S.l.]: Springer New York, NY, 2006. 488 p. ISBN 978-1-4419-2999-0.

JOOSSE, P. A. *et al.* Acoustic Emission Monitoring of Small Wind Turbine Blades . *Journal of Solar Energy Engineering*, v. 124, n. 4, p. 446–454, 11 2002. ISSN 0199-6231. Disponível em: <<https://doi.org/10.1115/1.1509769>>.

- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, v. 82, n. 1, p. 35–45, 03 1960. ISSN 0021-9223. Disponível em: <<https://doi.org/10.1115/1.3662552>>.
- KHAGI, B.; KWON, G.-R.; LAMA, R. Comparative analysis of alzheimer’s disease classification by cdr level using cnn, feature selection, and machine-learning techniques. *International Journal of Imaging Systems and Technology*, v. 29, 03 2019.
- KIM, J. *et al.* An adaptive kalman filter-based condition-monitoring technique for induction motors. *IEEE Access*, v. 11, p. 46373–46381, 2023.
- KINGMA, D. P.; WELLING, M. *Auto-Encoding Variational Bayes*. 2022. Disponível em: <<https://arxiv.org/abs/1312.6114>>.
- KLAMBAUER, G. *et al.* *Self-Normalizing Neural Networks*. 2017. Disponível em: <<https://arxiv.org/abs/1706.02515>>.
- MAAS, A. L. Rectifier nonlinearities improve neural network acoustic models. In: . [s.n.], 2013. Disponível em: <<https://api.semanticscholar.org/CorpusID:16489696>>.
- MARTINI, M. A simple relationship between ssim and psnr for dct-based compressed images and video: Ssim as content-aware psnr. In: *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*. [S.l.: s.n.], 2023. p. 1–5.
- MARTINS, G. M. C. *Dispositivo gravador de áudio para análise de ruídos audíveis*. 2025. Trabalho de Conclusão de Curso (Graduação) – Centro Federal de Educação Tecnológica de Minas Gerais.Campus Leopoldina. Engenharia de Controle e Automação, Departamento de Eletroeletrônica, 2025. Publicado em 13 de março de 2024.
- MASCI, J. *et al.* Stacked convolutional auto-encoders for hierarchical feature extraction. In: *International Conference on Artificial Neural Networks (ICANN)*. [S.l.]: Springer, 2011. p. 52–59.
- MESAROS, A. *et al.* Sound event detection: A tutorial. *IEEE Signal Processing Magazine*, v. 38, n. 5, p. 67–83, 2021.
- Ministério de Minas e Energia (MME). *Balanço Energético Brasileiro - Ano 2023*. 2024.
- MOORHOUSE, A. *et al.* *Research into aerodynamic modulation of wind turbine noise: final report*. [S.l.], 2007. Depositing User : Davies, MDJ Department : School of Computing, Science and Engineering Themes : QC221-246 Acoustics - Sound; Subjects outside of the University Themes.
- NATILI, F. *et al.* Multi-scale wind turbine bearings supervision techniques using industrial scada and vibration data. *Applied Sciences*, v. 11, n. 15, 2021. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/11/15/6785>>.
- PANDIT, R. *et al.* Scada data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future trends. *Wind Engineering*, v. 47, n. 2, p. 422–441, 2023. Disponível em: <<https://doi.org/10.1177/0309524X221124031>>.

PEDERSEN, E.; WAYE, K. P. Wind turbine noise, annoyance and self-reported health and well-being in different living environments. *Occupational and Environmental Medicine*, BMJ Publishing Group Ltd, v. 64, n. 7, p. 480–486, 2007. ISSN 1351-0711. Disponível em: <<https://oem.bmj.com/content/64/7/480>>.

PUTTAGUNTA, M.; RAVI, S. Medical image analysis based on deep learning approach. *Multimedia Tools and Applications*, v. 80, n. 16, p. 24365–24398, 2021. ISSN 1573-7721. Disponível em: <<https://doi.org/10.1007/s11042-021-10707-4>>.

RADHAKRISHNAN, R.; DIVAKARAN, A.; SMARAGDIS, A. Audio analysis for surveillance applications. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005*. [S.l.: s.n.], 2005. p. 158–161.

RASAMOELINA, A. D.; ADJAILIA, F.; SINČÁK, P. A review of activation function for artificial neural network. In: *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMi)*. [S.l.: s.n.], 2020. p. 281–286.

RASHID, H.; HAIDER, W.; BATUNLU, C. Forecasting of wind turbine output power using machine learning. In: *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*. [S.l.: s.n.], 2020. p. 396–399.

RIBRANT, J.; BERTLING, L. M. Survey of failures in wind power systems with focus on swedish wind power plants during 1997–2005. *IEEE Transactions on Energy Conversion*, v. 22, n. 1, p. 167–173, 2007.

SALAMON, J. *et al.* Towards the automatic classification of avian flight calls for bioacoustic monitoring. *PLOS ONE*, Public Library of Science, v. 11, n. 11, p. 1–26, 11 2016. Disponível em: <<https://doi.org/10.1371/journal.pone.0166866>>.

SCHÖLKOPF, B. *et al.* Estimating the support of a high-dimensional distribution. *Neural Computation*, v. 13, n. 7, p. 1443–1471, 2001.

SHE, B.; ZHANG, H.; WANG, J. Deep convolution variational autoencoder network based transfer learning strategy for fault diagnosis. In: *2021 International Conference on Digital Society and Intelligent Systems (DSIS)*. [S.l.: s.n.], 2021. p. 378–384.

SHEIBAT-OTHMAN, N. *et al.* Fault detection and isolation in wind turbines using support vector machines and observers. In: *2013 American Control Conference*. [S.l.: s.n.], 2013. p. 4459–4464.

Valor Econômico. *Brasil foi 3º país no mundo que mais atraiu investimentos em energias renováveis em 2023*. 2024. <<https://valor.globo.com/brasil/noticia/2024/03/13/brasil-foi-3-pas-no-mundo-que-mais-atraiu-investimentos-em-energias-renovveis-em-2023.ghml>>. Publicado em 13 de março de 2024.

VAPNIK, V. *The Nature of Statistical Learning Theory*. Springer New York, 1999. (Information Science and Statistics). ISBN 9780387987804. Disponível em: <<https://books.google.com.br/books?id=sna9BaxVbj8C>>.

VINCENT, P. *et al.* Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2008. (ICML '08), p.

1096–1103. ISBN 9781605582054. Disponível em: <<https://doi.org/10.1145/1390156.1390294>>.

WANG, G.; XIE, J.; WANG, S. Application of artificial intelligence in power system monitoring and fault diagnosis. *Energies*, v. 16, n. 14, 2023. ISSN 1996-1073. Disponível em: <<https://www.mdpi.com/1996-1073/16/14/5477>>.

WANG, Z. *et al.* Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, v. 13, n. 4, p. 600–612, 2004.

Windbox. *Componentes dos aerogeradores: conheça toda a sua estrutura*. 2020. <<https://windbox.com.br/blog/componentes-dos-aerogeradores/>>. Publicado em 3 de janeiro de 2020.

WITTEN E. FRANK, M. A. H. C. P. I. H. *Data Mining: Practical Machine Learning Tools and Techniques*. [S.l.]: Morgan Kaufmann, 2016.

Wood Mackenzie. *Global installed wind power capacity set to grow by 9% to 2030*. 2021. <https://www.woodmac.com/press-releases/global-installed-wind-power-capacity-set-to-grow-by-9-to-2030/>.

WU, X. *et al.* A novel active power dispatch method for onshore wind farms to reduce wind turbine noise. In: *2019 IEEE Power & Energy Society General Meeting (PESGM)*. [S.l.: s.n.], 2019. p. 1–5.

XIAO, Z.; CHENG, Z.; LI, Y. A review of fault diagnosis methods based on machine learning patterns. In: *2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing)*. [S.l.: s.n.], 2021. p. 1–4.

YAN, J. *et al.* A region based attention method for weakly supervised sound event detection and classification. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2019. p. 755–759.

ZHANG, F. *et al.* Ae5-ssim: A novel unsupervised tinfoli defect detection model with deep autoencoder. In: *2022 16th IEEE International Conference on Signal Processing (ICSP)*. [S.l.: s.n.], 2022. v. 1, p. 212–217.

ZHANG, H.; MCLOUGHLIN, I.; SONG, Y. Robust sound event recognition using convolutional neural networks. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2015. p. 559–563.