

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

**DETECÇÃO DE ANOMALIAS NAS RESPOSTAS DE VIBRAÇÃO DE
ROTORES USANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL**

Victor Gonçalves Marques

Uberlândia - MG

2026

Victor Gonçalves Marques

**DETECÇÃO DE ANOMALIAS NAS RESPOSTAS DE VIBRAÇÃO DE
ROTORES USANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL**

Dissertação aprovada pelo Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do grau de **Mestre em Engenharia Mecânica**.

Área de Concentração: Mecânica dos Sólidos e Vibrações.
Linha de pesquisa: Dinâmica de Sistemas Mecânicos.

Banca Examinadora:

Prof. Dr. Aldemir Aparecido Cavallini Jr - Orientador (UFU)

Prof. Dr. Leonardo Rosa Ribeiro da Silva (UFU)

Prof. Dr. Júlio Apolinário Cordioli (UFSC)

Uberlândia, 8 de agosto de 2026

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

M357
2026 Marques, Victor Gonçalves, 1988-
DETECÇÃO DE ANOMALIAS NAS RESPOSTAS DE VIBRAÇÃO DE
ROTORES USANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL
[recurso eletrônico] / Victor Gonçalves Marques. - 2026.

Orientador: Aldemir Aparecido Cavallini Jr.
Dissertação (Mestrado) - Universidade Federal de Uberlândia,
Pós-graduação em Engenharia Mecânica.
Modo de acesso: Internet.
DOI <http://doi.org/10.14393/ufu.di.2026.642>
Inclui bibliografia.

1. Engenharia mecânica. I. Jr, Aldemir Aparecido Cavallini, 1983-
, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação
em Engenharia Mecânica. III. Título.

CDU: 621

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Engenharia
Mecânica

Av. João Naves de Ávila, nº 2121, Bloco 1M, Sala 212 - Bairro Santa Mônica, Uberlândia-
MG, CEP 38400-902

Telefone: (34) 3239-4282 - www.posmecanicaufu.com.br - secposmec@mecanica.ufu.br



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Mecânica				
Defesa de:	Dissertação de Mestrado Acadêmico, número 687, PPGEM				
Data:	8 de agosto de 2026	Hora de início:	08:00	Hora de encerramento:	11:00
Matrícula do Discente:	12522EMC005				
Nome do Discente:	Victor Gonçalves Marques				
Título do Trabalho:	DETECÇÃO DE ANOMALIAS NAS RESPOSTAS DE VIBRAÇÃO DE ROTORES USANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL				
Área de concentração:	Mecânica dos Sólidos e Vibrações				
Linha de pesquisa:	Dinâmica de Sistemas Mecânicos				
Projeto de Pesquisa de vinculação:	não se aplica				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Mecânica, assim composta: Professores Doutores: Leonardo Rosa Ribeiro da Silva - FEMEC/UFU; Júlio Apolinário Cordioli - UFSC e Aldemir Aparecido Cavallini Jr. - FEMEC/UFU, orientador do candidato.

Iniciando os trabalhos o presidente da mesa, Dr. Aldemir Aparecido Cavallini Jr., apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Aldemir Aparecido Cavallini Junior, Professor(a) do Magistério Superior**, em 10/08/2026, às 13:58, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Leonardo Rosa Ribeiro da Silva, Professor(a) do Magistério Superior**, em 14/08/2026, às 18:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **JULIO APOLINARIO CORDIOLI, Usuário Externo**, em 18/08/2026, às 12:46, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **7479361** e o código CRC **C679E0D3**.

*Dedico este trabalho à minha amada esposa, Rebecca,
e à minha super mãe, Nilsa.*

"Não existe uma estrada real para a ciência, e somente aqueles que não temem a fadiga de galgar suas trilhas escarpadas têm chance de atingir seus cumes luminosos"
(Karl Marx)

Agradecimentos

À minha esposa, Rebecca, por todo incentivo, amor e compreensão durante a realização deste trabalho.

À minha mãe, Nilsa, por todo amor e apoio incondicional durante todas as etapas da minha vida, sempre fornecendo o respaldo para que eu pudesse me dedicar aos meus estudos.

Ao Prof. e Orientador, Dr. Aldemir, por toda inspiração, orientação e apoio durante a jornada do mestrado, compartilhando seu conhecimento sempre de forma tão generosa e paciente.

Aos trabalhadores e trabalhadoras das Universidades públicas brasileiras, em especial da UFU, responsáveis pela excelência do Ensino, Pesquisa e Extensão, mesmo diante de tantas adversidades.

Aos meus colegas da Petrobras, Thiago Emrich, Cristófer Sousa e Raphael Timbó, pelo incentivo, colaboração e companheirismo na concretização deste trabalho.

Aos colegas do LMEST, Murillo Álvaro, Daniel Ferreira e Eduardo Gouveia, pelo suporte e compartilhamento de conhecimento durante a realização deste trabalho.

Marques, V. G., **DETECÇÃO DE ANOMALIAS NAS RESPOSTAS DE VIBRAÇÃO DE ROTORES USANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL**. 2026. Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.

Resumo

Máquinas rotativas desempenham papel essencial em setores industriais como óleo e gás, geração de energia e mineração, e suas falhas podem resultar em paradas não planejadas, perdas econômicas significativas e riscos à segurança. A análise de vibração destaca-se como uma das principais técnicas de monitoramento da condição, permitindo a identificação precoce de problemas mecânicos e apoiando estratégias de manutenção preditiva. A crescente adoção de sistemas de monitoramento contínuo tem gerado grandes volumes de dados, impondo limitações às abordagens tradicionais baseadas exclusivamente em análises manuais.

Esta dissertação propõe e avalia metodologias de Aprendizado de Máquina para detecção de anomalias em respostas de vibração de rotores, com ênfase em estratégias não supervisionadas, alinhadas à realidade industrial de escassez de dados rotulados de falha. O trabalho integra e expande o *framework Edge Analytics*, desenvolvido pelo Laboratório de Mecânica de Estruturas da Universidade Federal de Uberlândia (LMEst - UFU), que combina técnicas estatísticas simples e modelos de aprendizado profundo para detecção de anomalias.

Além dos modelos originais da arquitetura *Edge Analytics* — *Gates*, *Autoencoder*, *DeepAnT* e *Histogram* — são desenvolvidos e avaliados novos modelos baseados em *Multilayer Perceptron* (MLP) e *Long Short-Term Memory* (LSTM). Adicionalmente, investiga-se a combinação desses modelos com técnicas de *Change Point Detection* (CPD), visando aumentar a sensibilidade na identificação de alterações sutis e sustentadas na resposta vibratória.

A validação das metodologias é realizada por meio de cinco *benchmarks*, envolvendo dados numéricos e experimentais de diferentes sistemas rotativos, incluindo rotores apoiados em mancais magnéticos, exaustor industrial e ensaios de degradação de rolamentos. O desempenho é avaliado por métricas clássicas de classificação, com ênfase no *F1 score* devido ao forte desbalanceamento entre classes. Os resultados indicam que não há um modelo universalmente superior, porém modelos baseados em LSTM e MLP destacam-se em falhas de evolução progressiva, enquanto a combinação com CPD contribui para a detecção de mudanças sutis. Assim, o trabalho fornece subsídios técnicos para aplicação industrial em manutenção preditiva.

Palavras-chave: Detecção de anomalias. Máquinas rotativas. Vibração de rotores. Inteligência Artificial.

MARQUES, V. G., **ANOMALY DETECTION IN ROTOR VIBRATION RESPONSES USING ARTIFICIAL INTELLIGENCE TECHNIQUES**. 2026. MsC Dissertation, Federal University of Uberlândia, Uberlândia, MG, Brazil.

Abstract

Rotating machinery plays an essential role in industrial sectors such as oil and gas, power generation and mining, and its failures may result in unplanned shutdowns, significant economic losses and safety risks. Vibration analysis stands out as one of the main condition monitoring techniques, enabling the early identification of mechanical problems and supporting predictive maintenance strategies. The increasing adoption of continuous monitoring systems has led to large volumes of data, imposing limitations on traditional approaches based exclusively on manual analysis.

This dissertation proposes and evaluates Machine Learning methodologies for anomaly detection in rotor vibration responses, with emphasis on unsupervised strategies aligned with the industrial reality of scarce labeled fault data. The work integrates and extends the *Edge Analytics* framework, developed by the Structural Mechanics Laboratory of the Federal University of Uberlândia (LMest - UFU), which combines simple statistical techniques and deep learning models for anomaly detection.

In addition to the original models of the *Edge Analytics* architecture — *Gates*, *Autoencoder*, *DeepAnT*, and *Histogram* — new models based on *Multilayer Perceptron* (MLP) and *Long Short-Term Memory* (LSTM) are developed and evaluated. Furthermore, the combination of these models with *Change Point Detection* (CPD) techniques is investigated in order to enhance sensitivity in identifying subtle and sustained changes in vibration responses.

The validation of the proposed methodologies is conducted through five *benchmarks*, involving numerical and experimental data from different rotating systems, including rotors supported by magnetic bearings, an industrial exhaust fan and bearing degradation tests. Performance is assessed using classical classification metrics, with emphasis on the *F1 score* due to the strong class imbalance. The results indicate that there is no universally superior model, however LSTM- and MLP-based models stand out in faults with progressive evolution, while the combination with CPD contributes to the detection of subtle changes. Thus, the study provides technical support for industrial applications in predictive maintenance.

Keywords: Anomaly Detection. Rotating machines. Rotor vibration. Artificial Intelligence.

Sumário

Lista de Figuras	xviii
Lista de Tabelas	xix
Lista de Simbolos	xx
1 Introdução	1
1.1 Contextualização	1
1.2 Contribuições Prévias ao Estudo no LMEst	2
1.3 Objetivos do Trabalho	5
1.4 Organização da Dissertação	6
2 Revisão Bibliográfica	7
3 <i>Machine Learning</i> e Redes Neurais	16
3.1 Conceitos fundamentais de rede neural simples	18
3.2 <i>Multilayer Perceptron</i> (MLP)	20
3.3 Redes Neurais Recorrentes	22
3.4 <i>Long Short-Term Memory</i> (LSTM)	24
3.5 <i>Change Point Detection</i> (CPD)	26
3.6 Métricas de Desempenho	28
3.7 Classificação de Anomalias por <i>Score</i> de Erro	30
4 Arquitetura <i>Edge Analytics</i>	32
4.1 Introdução e Objetivo	32
4.2 Camada <i>Gates</i>	34
4.3 <i>Autoencoder</i>	36
4.4 <i>DeepAnT</i>	38

4.5	Histograma	40
4.6	Arquitetura	40
5	Dinâmica de Rotores e <i>Benchmarks</i>	42
5.1	Dinâmica de rotores	42
5.1.1	Modelagem	43
5.1.2	Defeitos em Máquinas Rotativas	44
5.2	<i>Benchmarks</i>	47
5.2.1	<i>Benchmark 1</i>	47
5.2.2	<i>Benchmark 2</i>	49
5.2.3	<i>Benchmark 3</i>	50
5.2.4	<i>Benchmark 4</i>	51
5.2.5	<i>Benchmark 5</i>	51
6	Resultados	53
6.1	<i>Edge Analytics</i> - Modelos Originais	53
6.1.1	<i>Benchmark 1</i>	55
6.1.2	<i>Benchmark 2</i>	56
6.1.3	<i>Benchmark 3</i>	57
6.1.4	<i>Benchmark 4</i>	58
6.1.5	<i>Benchmark 5</i>	59
6.2	MLP, LSTM e CPD	60
6.2.1	MLP	60
6.2.2	LSTM	65
6.3	<i>Change Point Detection</i> (CPD)	70
6.3.1	Combinação MLP e CPD	72
6.3.2	Combinação LSTM e CPD	75
6.4	Comparação de Resultados	78
6.4.1	<i>Benchmark 1</i>	79
6.4.2	<i>Benchmark 2</i>	80
6.4.3	<i>Benchmark 3</i>	81
6.4.4	<i>Benchmark 4</i>	82
6.4.5	<i>Benchmark 5</i>	83

<i>SUMÁRIO</i>	xv
6.5 Avaliação complementar por ROC AUC	84
7 Considerações Finais	85
7.1 Propostas de Trabalhos Futuros	87
Referências	89

Lista de Figuras

3.1	Representação típica esquemática de neurônio de rede neural e de célula nervosa. Adaptado de GÉRON (2019)	17
3.2	Grupo de <i>Perceptrons</i> . Adaptado de GÉRON (2019)	18
3.3	Esquema ilustrativo do <i>feedforward</i> e do <i>backpropagation</i> em uma rede neural. Autoria própria.	19
3.4	Rede <i>Multi Layer Perceptron</i> (MLP) com duas camadas intermediárias. Adaptado de Faceli (2011)	20
3.5	Representação compacta de uma RNN, destacando o laço recorrente que permite a propagação de informação entre instantes sucessivos. Adaptado de Olah (2015)	23
3.6	RNN “desenrolada” no tempo, evidenciando o compartilhamento de parâmetros e a propagação do estado oculto entre instantes sucessivos. Adaptado de Olah (2015)	23
3.7	Arquitetura geral de uma rede LSTM, evidenciando os mecanismos de controle e os estados de memória. Adaptado de (AFRIDI et al., 2023).	25
3.8	Detalhe da estrutura interna das portas de uma rede LSTM, evidenciando o controle do fluxo de informação (AFRIDI et al., 2023).	26
3.9	Variação de parâmetros estatísticos em séries temporais. Adaptado de (GUPTA et al., 2024).	27
3.10	Série temporal com pontos de mudança detectados por CPD. Adaptado de (GUPTA et al., 2024).	27
3.11	Matriz de confusão para um problema de classificação binária. Adaptado de Lobão (2020)	28
4.1	Ilustração da Arquitetura do <i>Edge Analytics</i> . Adaptado de (BORGES et al., 2025).	33
4.2	Exemplo ilustrativo da camada <i>Gates</i>	36
4.3	Exemplo ilustrativo da camada <i>Autoencoder</i> . Adaptado de (BORGES et al., 2025).	38
4.4	Exemplo ilustrativo do modelo <i>DeepAnT</i> . Adaptado de (MUNIR et al., 2019).	39

5.1	<i>Benchmark 1: Amplitude de vibração no tempo.</i>	48
5.2	<i>Benchmark 1: Anomalias.</i>	48
5.3	Bancada experimental de rotor com mancais magnéticos.	49
5.4	<i>Benchmark 2: Sinal no tempo e classificação.</i>	49
5.5	Exaustor industrial, sensores e placa de aquisição (GONÇALVES, 2024)	50
5.6	<i>Benchmark 3: Sinal no tempo e classificação.</i>	50
5.7	<i>Benchmark 4: Sinal no tempo e classificação.</i>	51
5.8	Bancada de teste de mancais de rolamento da NASA IMS. Adaptado de (HOU et al., 2022)	52
5.9	<i>Benchmark 5: Sinal no tempo e classificação.</i>	52
6.1	<i>Benchmark 1: Sinal no tempo, gabarito e classificação dos modelos.</i>	55
6.2	<i>Benchmark 2: Sinal no tempo, gabarito e classificação dos modelos.</i>	56
6.3	<i>Benchmark 3: Sinal no tempo, gabarito e classificação dos modelos.</i>	57
6.4	<i>Benchmark 4: Sinal no tempo, gabarito e classificação dos modelos.</i>	58
6.5	<i>Benchmark 5: Sinal no tempo, gabarito e classificação dos modelos.</i>	59
6.6	Sinal no tempo, gabarito e classificação do modelo MLP para o <i>benchmark 1</i> .	63
6.7	Sinal no tempo, gabarito e classificação do modelo MLP para o <i>benchmark 2</i> .	63
6.8	Sinal no tempo, gabarito e classificação do modelo MLP para o <i>benchmark 3</i> .	64
6.9	Sinal no tempo, gabarito e classificação do modelo MLP para o <i>benchmark 4</i> .	64
6.10	Sinal no tempo, gabarito e classificação do modelo MLP para o <i>benchmark 5</i> .	65
6.11	Sinal no tempo, gabarito e classificação do modelo LSTM para o <i>benchmark 1</i> .	68
6.12	Sinal no tempo, gabarito e classificação do modelo LSTM para o <i>benchmark 2</i> .	68
6.13	Sinal no tempo, gabarito e classificação do modelo LSTM para o <i>benchmark 3</i> .	69
6.14	Sinal no tempo, gabarito e classificação do modelo LSTM para o <i>benchmark 4</i> .	69
6.15	Sinal no tempo, gabarito e classificação do modelo LSTM para o <i>benchmark 5</i> .	70
6.16	Erro de previsão e classificação por limiar do modelo LSTM para o <i>benchmark 1</i> .	71
6.17	Erro de previsão e classificação do modelo LSTM + CPD para o <i>benchmark 1</i> .	72
6.18	Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o <i>benchmark 1</i> .	73
6.19	Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o <i>benchmark 2</i> .	73

6.20	Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o <i>benchmark 3</i>	74
6.21	Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o <i>benchmark 4</i>	74
6.22	Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o <i>benchmark 5</i>	75
6.23	Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o <i>benchmark 1</i>	76
6.24	Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o <i>benchmark 2</i>	76
6.25	Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o <i>benchmark 3</i>	77
6.26	Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o <i>benchmark 4</i>	77
6.27	Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o <i>benchmark 5</i>	78
6.28	Ranking de <i>F1-score</i> dos modelos para o <i>benchmark 1</i>	79
6.29	Ranking de <i>F1-score</i> dos modelos para o <i>benchmark 2</i>	80
6.30	Ranking de <i>F1-score</i> dos modelos para o <i>benchmark 3</i>	81
6.31	Ranking de <i>F1-score</i> dos modelos para o <i>benchmark 4</i>	82
6.32	Ranking de <i>F1-score</i> dos modelos para o <i>benchmark 5</i>	83

Lista de Tabelas

6.1	Descrição dos <i>benchmarks</i> avaliados.	55
6.2	<i>Benchmark</i> 1: Métricas de desempenho por modelo.	56
6.3	<i>Benchmark</i> 2: Métricas de desempenho por modelo.	57
6.4	<i>Benchmark</i> 3: Métricas de desempenho por modelo.	58
6.5	<i>Benchmark</i> 4: Métricas de desempenho por modelo.	59
6.6	<i>Benchmark</i> 5: Métricas de desempenho por modelo.	60
6.7	Tamanho de amostra otimizado do modelo MLP por <i>benchmark</i>	62
6.8	Métricas de desempenho do modelo MLP para os <i>benchmarks</i> avaliados.	62
6.9	Tamanho de amostra otimizado do modelo LSTM por <i>benchmark</i>	67
6.10	Métricas de desempenho do modelo LSTM para os <i>benchmarks</i> avaliados.	67
6.11	Métricas de desempenho do modelo MLP + CPD para os <i>benchmarks</i> avaliados.	72
6.12	Métricas de desempenho do modelo LSTM + CPD para os <i>benchmarks</i> avaliados.	75
6.13	Métricas de desempenho dos modelos para o <i>benchmark</i> 1.	79
6.14	Métricas de desempenho dos modelos para o <i>benchmark</i> 2.	80
6.15	Métricas de desempenho dos modelos para o <i>benchmark</i> 3.	81
6.16	Métricas de desempenho dos modelos para o <i>benchmark</i> 4.	82
6.17	Métricas de desempenho dos modelos para o <i>benchmark</i> 5.	83
6.18	ROC AUC dos modelos para os <i>benchmarks</i> avaliados.	84

Lista de Simbolos

Abreviaturas

AE	<i>Autoencoder</i>
AIoT	<i>Artificial Intelligence of Things</i>
ANN	<i>Artificial Neural Network</i>
API	<i>American Petroleum Institute</i>
AS	<i>Anomaly Score</i>
BPTT	<i>Backpropagation Through Time</i>
CPD	<i>Change Point Detection</i>
DeepAnT	<i>Deep Anomaly Detection for Time Series</i>
DL	<i>Deep Learning</i>
DNN	<i>Deep Neural Networks</i>
DT	<i>Decision Trees</i>
EA	<i>Envelope Analysis</i>
FEMEC	Faculdade de Engenharia Mecânica
FFT	<i>Fast Fourier Transform</i>
FN	<i>False Negative</i>
FP	<i>False Positive</i>
IA	Inteligência Artificial

IFDP		<i>Intelligent Fault Diagnosis and Prognosis</i>
IMS		<i>Intelligent Maintenance Systems</i>
KNN		<i>k-Nearest Neighbors</i>
KS		Teste de Kolmogorov–Smirnov
LMEst		Laboratório de Mecânica de Estruturas
LR		<i>Logistic Regression</i>
LSTM		<i>Long Short-Term Memory</i>
MAE		<i>Mean Absolute Error</i>
MEF		Método de Elementos Finitos
ML		<i>Machine Learning</i>
MLP		<i>Multilayer Perceptron</i>
MSE		<i>Mean Squared Error</i>
PCA		<i>Principal Component Analysis</i>
PdM		<i>Predictive Maintenance</i>
RF		<i>Random Forest</i>
RMS		<i>Root Mean Square</i>
RMSE		<i>Root Mean Squared Error</i>
RNA		Redes Neurais Artificiais
RNN		<i>Recurrent Neural Network</i>
ROC	AUC	<i>Receiver Operating Characteristic Area Under the Curve</i>
ROSS		<i>Rotordynamic Open-Source Software</i>
RPM		Rotações por Minuto
SHM		Monitoramento da Integridade Estrutural

STFT	<i>Short-Time Fourier Transform</i>
SVM	<i>Support Vector Machine</i>
TN	<i>True Negative</i>
TP	<i>True Positive</i>
UFRJ	Universidade Federal do Rio de Janeiro
UFU	Universidade Federal de Uberlândia
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
XAI	<i>Explainable Artificial Intelligence</i>

Símbolos

$\ddot{\mathbf{q}}$	Vetor de acelerações generalizadas do rotor
$\dot{\mathbf{q}}$	Vetor de velocidades generalizadas do rotor
$\mathbf{C}$	Matriz global de amortecimento
$\mathbf{D}$	Matriz associada ao efeito giroscópico
$\mathbf{F}_d$	Vetor de forças externas aplicadas ao sistema
$\mathbf{F}_m$	Vetor de forças não lineares provenientes dos mancais
$\mathbf{F}_u$	Vetor de forças de excitação por desbalanceamento
$\mathbf{K}$	Matriz global de rigidez do rotor
$\mathbf{K}_{sv}$	Matriz de enrijecimento dependente da velocidade de rotação
$\mathbf{M}$	Matriz global de massa do rotor
$\mathbf{q}$	Vetor de deslocamentos generalizados do rotor
$\mathbf{W}$	Vetor de forças peso atuantes no rotor
μ	Média do erro de reconstrução no conjunto de treinamento

Ω	Velocidade angular de rotação do rotor
σ	Desvio padrão do erro de reconstrução no conjunto de treinamento
k	Parâmetro de sensibilidade utilizado no cálculo do limiar de decisão
L	Lagrangeano do sistema ($L = T - U$)
Q_i	Forças generalizadas associadas às coordenadas q_i
q_i	Coordenadas generalizadas do sistema
T	Energia cinética do sistema rotativo
U	Energia potencial do sistema rotativo

Capítulo 1

Introdução

1.1 Contextualização

Máquinas rotativas são equipamentos essenciais para diversos processos produtivos da indústria (DAS et al., 2023), estando presentes em setores como óleo e gás, geração de energia e mineração, com equipamentos como bombas, compressores, turbinas a gás e a vapor, motores elétricos e caixas de engrenagens. Dada sua criticidade operacional, falhas destes equipamentos podem resultar em paradas não planejadas, grandes perdas econômicas e riscos à segurança operacional. Assim, o monitoramento da condição operacional e a identificação precoce de degradações mecânicas constituem elementos essenciais para o aumento da confiabilidade dos ativos e para o suporte a estratégias de manutenção preditiva baseadas em condição das máquinas (GONÇALVES, 2024; LIU et al., 2018).

O monitoramento da condição pode ser realizado por meio da análise de diferentes parâmetros, como temperatura de mancais e de óleo lubrificantes, análise de óleo, medição de ruído, corrente elétrica e vibrações. A análise de vibração destaca-se por sua natureza não intrusiva e pela riqueza de informações que pode fornecer sobre o estado dinâmico do sistema rotativo (BRITO, 2022). Assim, o monitoramento de vibração de máquinas rotativas é fundamental para a manutenção preditiva, permitindo a detecção e diagnóstico de problemas como desbalanceamento, desalinhamento, folgas e defeitos em mancais e engrenagens (BENTLY et al., 2002).

Em equipamentos de grande porte e alta criticidade, como turbinas a vapor e a gás, compressores industriais e bombas centrífugas críticas para processos industriais, sistemas de monitoramento contínuo de vibração já são tradicionalmente empregados com a finalidade de proteção,

garantindo a parada automática da máquina caso níveis de vibração ultrapassem limites previamente estabelecidos. Com a redução de custos de sensores, é crescente o monitoramento também de máquinas de menor porte. Assim, além da finalidade de proteção, o monitoramento contínuo implica na geração de grande volume de dados de vibração de máquinas.

O aumento expressivo da quantidade dos dados de monitoramento de vibração impõe limitações às abordagens tradicionais baseadas exclusivamente em análise manual. Como alternativa, técnicas baseadas em Inteligência Artificial (IA) e Aprendizado de Máquina (ML) têm sido amplamente investigadas para automatizar processos de detecção de anomalias e diagnóstico de falhas de máquinas rotativas. Esses métodos são capazes de extrair padrões relevantes a partir de grandes conjuntos de dados, reduzindo a dependência de conhecimento especializado explícito e ampliando a escalabilidade das soluções de monitoramento (LEI et al., 2020).

No contexto industrial, as máquinas operam majoritariamente em condição normal e falhas reais são eventos raros, implicando em uma restrição à implementação de métodos de ML supervisionados, devido à consequente escassez de dados rotulados de falha (MISBAH et al., 2024). Diante dessa limitação, têm se destacado abordagens baseadas em aprendizado não supervisionado ou semissupervisionado, especialmente aquelas focadas em detecção de anomalias, que consistem em divergências em relação ao regime de operação considerado normal. Nessas abordagens, os modelos são treinados predominantemente com dados representativos do comportamento normal da máquina, de forma a serem capazes de identificar desvios que indiquem alterações no regime regular de funcionamento.

Assim, a integração entre análise de vibração e técnicas de Aprendizado de Máquina orientadas à detecção de anomalias representa uma direção promissora para o monitoramento inteligente de máquinas rotativas. Essa combinação permite explorar de forma sistemática o grande volume de dados gerados em ambientes industriais, permitindo maior robustez na identificação precoce de falhas e na disponibilização de informações para planejamento de manutenção preditiva, alinhando-se às demandas atuais da Indústria 4.0.

1.2 Contribuições Prévias ao Estudo no LMest

O Laboratório de Mecânica de Estruturas "José Eduardo Tannús Reis" (LMest) da Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia (FEMEC-UFU) tem mantido atuação contínua e relevante em pesquisa aplicada à dinâmica de máquinas rotativas. Em

particular, parte significativa dessa produção concentra-se em métodos de monitoramento e avaliação da integridade estrutural, com aplicações voltadas à detecção de anomalias e ao aumento da confiabilidade operacional de sistemas rotativos.

[Cavallini Jr. \(2013\)](#) investigou a detecção de trincas transversais em eixos, propondo duas técnicas de monitoramento de integridade estrutural, sendo uma baseada no comportamento não linear da trinca e outra na impedância eletromecânica. Foram analisados três modelos de trincas a partir de simulações computacionais.

Em linha semelhante, [SOUZA LEÃO \(2019\)](#) desenvolveu um modelo representativo de uma máquina real e explorou, de forma numérica e experimental, um método de detecção de falhas a partir de sinais vibratórios, além de propor uma estratégia de controle ativo capaz de atenuar componentes super-harmônicas associadas à presença de trinca.

[Rende \(2020\)](#) formulou um modelo de rotor associado a quatro vigas representativas de pás e investigou sua estabilidade empregando a teoria de Floquet e a transformação de coordenadas multi-pás.

Posteriormente, [Nascimento \(2022\)](#) estendeu esse tipo de análise incorporando defeitos do tipo trinca transversal e utilizando uma formulação FLEX para representar o dano nas palhetas, permitindo examinar alterações no comportamento dinâmico do conjunto.

[Silva \(2023\)](#), em sua tese, confrontou modelos de trinca reportados na literatura e avaliou técnicas de *Structural Health Monitoring* (SHM) fundamentadas em efeitos de ressonância e em medidas derivadas de formas modais, com validação por estudos numéricos e experimentais, evidenciando a capacidade de identificação e localização de trincas transversais em máquinas rotativas.

Paralelamente, observa-se uma tendência de aprofundamento na modelagem e no entendimento de componentes essenciais de máquinas rotativas, como mancais e selos, bem como no desenvolvimento de estratégias de controle e mitigação de vibração.

[Oliveira \(2019\)](#) sintetizou controladores para rotores supercríticos suportados por mancais magnéticos, avaliando controladores do tipo PID e empregando respostas ao desbalanceamento para organizar um metamodelo *Kriging* voltado ao projeto e à análise desses sistemas.

Com foco em balanceamento, [Carvalho \(2022\)](#) propôs uma metodologia baseada em *Kriging* alimentada por amostras derivadas do Método do Coeficiente de Influência, relacionando condições de desbalanceamento às massas e posições corretivas.

Em seguida, [Cabral \(2023\)](#) apresentou uma alternativa de balanceamento sem massas de

teste, empregando redes neurais treinadas a partir de respostas vibratórias obtidas em ensaios numéricos e experimentais.

[Rezende \(2025\)](#), partindo da utilização do filtro de Kalman aumentado para estimar as forças de desbalanceamento, desenvolveu a técnica de balanceamento de rotores sem o uso de massas de teste.

No mesmo sentido, a tese de [Pereira Neto \(2025\)](#), desenvolve uma metodologia de balanceamento baseada em modelo de redes neurais artificiais (RNA) para estimar as forças de desbalanceamento.

Adicionalmente, o desenvolvimento de metodologias baseadas em *Artificial Intelligence* (IA) para monitoramento e diagnóstico de falhas tem se consolidado como uma demanda crescente no LMEst. Esse movimento não apenas amplia as possibilidades de detecção e identificação de anomalias em máquinas rotativas, como também aproxima a pesquisa das tendências contemporâneas observadas na academia e na indústria, fortalecendo a integração entre técnicas de IA e problemas clássicos de dinâmica de rotores.

No contexto de detecção de anomalias em sinais de vibração de rotores, destaca-se a atuação do LMEst no projeto de pesquisa, desenvolvimento e inovação intitulado Aplicação de Técnica de Inteligência Artificial na Captura de Dados de Máquinas para Armazenamento de Dados de Anormalidades em Servidores (Edge Analytics). A detecção de anomalias em sinais de vibração de máquinas rotativas utilizando métodos de IA é parte integrante do software desenvolvido neste projeto de pesquisa.

A tese de doutorado de [Brito \(2022\)](#) investigou técnicas de Inteligência Artificial Explicável (XAI) aplicadas ao monitoramento de ativos industriais, com foco na interpretação de modelos de aprendizado de máquina utilizados para detecção de anomalias em séries temporais. O trabalho discute estratégias de extração de características e modelagem para sinais de vibração, e enfatiza como métodos de explicabilidade aumentam a confiabilidade no diagnóstico automatizado. São investigados métodos não supervisionados para detecção de anomalias em séries temporais de vibração de mancais de rolamento, destacando o bom desempenho do modelo *Isolation Forest*.

[Gonçalves \(2024\)](#), em sua tese, propôs uma metodologia de detecção de anomalias em sistemas rotativos baseada em aprendizado não supervisionado, integrando técnicas de extração de características nos domínios do tempo, frequência e tempo-frequência com métodos de redução de dimensionalidade como *PCA*. O estudo emprega algoritmos de clusterização — *k-means*, *DBSCAN*, *GMM* e métodos hierárquicos — organizados em um modelo *ensemble* aplicado em

janelas deslizantes para monitoramento contínuo. A abordagem foi validada em três plataformas: um rotor com mancais magnéticos, um exaustor industrial com mancais de rolamento e um sistema rotativo experimental suportado por mancais hidrodinâmicos. Em todos os casos, a metodologia apresentou elevada capacidade de distinguir alterações de comportamento, alcançando *F1-scores* superiores a 94% para os dois primeiros sistemas e permitindo, no terceiro, a identificação robusta de anomalias simuladas e reais (desbalanceamento, desalinhamento e trincas).

1.3 Objetivos do Trabalho

Este trabalho tem como objetivo principal desenvolver e comparar modelos de Aprendizado de Máquina para detecção de anomalias em respostas de vibração de rotores, integrando técnicas utilizadas no âmbito do projeto de pesquisa desenvolvido no LMEst denominado *Edge Analytics* (*Gates*, *DeepAnT*, *Autoencoder - AE*, *Histogram*), com o desenvolvimento de novos modelos (MLP e LSTM), bem como sua combinação com técnicas estatísticas de detecção de mudança (*Change Point Detection – CPD*), avaliando desempenho, robustez, sensibilidade e aplicabilidade industrial.

A utilização de treinamento não supervisionado tem por finalidade a adesão com a realidade industrial, onde são poucos os dados de falha disponíveis para máquinas rotativas. O desenvolvimento de modelo LSTM é consistente com desenvolvimentos recentes no campo de aplicação de RNN para detecção de anomalias de rotores, dada a capacidade da LSTM de lidar com dependências temporais de longo prazo em séries temporais na tarefa de predição de sequências. O desenvolvimento de um modelo MLP tem por finalidade a avaliação do desempenho de uma arquitetura de rede neural mais simples, com menor custo computacional, para a tarefa de detecção de anomalias.

A avaliação de modelos de ML combinados com a técnica de CPD tem por objetivo fornecer insumos para análise de desempenho de modelos combinados de detecção de anomalia, que combinam abordagem convencional de classificação de anomalia a partir de comparação com um *score* de erro com a observação da mudança de patamar do regime do erro nas predições.

Os modelos avaliados foram concebidos para a detecção generalista de anomalias, sem foco em modos de falha específicos, buscando identificar quaisquer desvios relevantes em relação à condição normal de operação.

A aplicação dos modelos em cinco diferentes conjuntos de dados, denominados *benchmarks*, obtidos a partir de equipamentos distintos, permite avaliar a generalização dos modelos em diferentes contextos.

1.4 Organização da Dissertação

Além deste capítulo introdutório, esta Dissertação inclui outros seis capítulos, brevemente descritos a seguir.

O [Capítulo 2](#) apresenta uma revisão bibliográfica de trabalhos científicos recentes que investigaram detecção de anomalias em máquinas rotativas utilizando técnicas de aprendizado de máquina.

O [Capítulo 3](#) apresenta descrição de conceitos fundamentais para este trabalho, tratando de aspectos teóricos de ML, descrição dos modelos MLP, LSTM e da técnica de CPD, além da aplicação de ML em tarefas de classificação de anomalias.

O [Capítulo 4](#) aborda a arquitetura *Edge Analytics*, Projeto de Pesquisa desenvolvido no LMEst/UFU para detecção de anomalias em rotores. São descritos os modelos e a arquitetura de votação do sistema.

O [Capítulo 5](#) descreve sucintamente os principais conceitos associados à vibração de rotores e caracteriza os conjuntos de dados de vibração (*benchmarks*) utilizados para avaliação dos modelos.

O [Capítulo 6](#) apresenta os resultados decorrentes da otimização dos modelos originais da arquitetura *Edge Analytics* e o desempenho dos modelos MLP e LSTM, individualmente e em combinação com a técnica CPD, na detecção de anomalia nos cinco conjuntos de dados de vibração utilizados. Também é apresentada uma avaliação comparativa de desempenho dos diferentes modelos.

O [Capítulo 7](#), por fim, elenca as principais conclusões, destacando os ganhos e limitações dos modelos abordados e desenvolvidos, bem como as perspectivas futuras.

Capítulo 2

Revisão Bibliográfica

Neste capítulo são apresentados trabalhos correlatos que tratam da aplicação de técnicas de ML para detecção de anomalias em sinais de vibração de máquinas rotativas. O trabalho de revisão da literatura constitui etapa essencial para compreensão dos avanços e tendências atuais, também fornecendo informações para identificação de métodos e técnicas mais adequadas para abordagem do problema em questão.

O aprendizado de máquina constitui uma subárea da Inteligência Artificial voltada ao desenvolvimento de sistemas com capacidade de adaptação com base em dados. Fundamentado em princípios matemáticos e estatísticos, esse conjunto de técnicas permite a extração de conhecimento a partir de observações, bem como o apoio à tomada de decisão em sistemas complexos. Por isso, sua aplicação em tarefas de monitoramento da condição e diagnóstico de falhas em máquinas rotativas tem sido amplamente reconhecida como promissora na literatura ([LIU et al., 2018](#)).

De acordo com [Liu et al. \(2022\)](#) e [Brito \(2022\)](#), a técnica de análise de vibração possui diversas vantagens: (i) permite realização de medições durante operação; (ii) facilidade de posicionamento dos sensores para coleta de dados; (iii) proporciona um amplo conhecimento a respeito das características dinâmicas para identificação de falhas mecânicas; (iv) viabiliza tempo de aquisição condizente para monitoramento de grande número de ativos; e (v) permite identificar informações detalhadas sobre os equipamentos, incluindo condições mecânicas, elétricas e até estruturais.

A ampliação do monitoramento de vibração em máquinas rotativas tem resultado na geração de grandes volumes de dados, cuja análise tradicional depende de especialistas. Contudo, a elevada quantidade de sinais e a limitação de recursos humanos e de sistemas de alarme

confiáveis dificultam a análise manual e totalmente automática, motivando o desenvolvimento de estratégias baseadas em inteligência artificial. Neste sentido, destaca-se a detecção de falhas não supervisionada, também conhecida como detecção de anomalias (BRITO, 2022). Apesar do avanço do monitoramento de máquinas rotativas, a literatura apresenta predominância de abordagens supervisionadas voltadas à classificação e prognóstico, sendo relativamente limitado o número de métodos não supervisionados, conforme indicado por Liu et al. (2022) e Stetco et al. (2019).

Sistemas de diagnóstico em máquinas rotativas são tipicamente estruturados em etapas de coleta de dados, processamento de sinais, extração de características, identificação de falhas e tomada de decisão pela equipe de manutenção (LEI, 2017). É nessa estrutura que são aplicados métodos de aprendizado de máquina que, após treinamento e ajuste de hiperparâmetros, têm seu desempenho avaliado por meio de indicadores específicos.

Em relação ao processamento de sinais de vibração e extração de características, a literatura apresenta técnicas e algoritmos numerosos e bem desenvolvidos que são tipicamente classificados em três categorias: domínios do tempo, da frequência e tempo-frequência (LEI, 2017; LIU et al., 2018; WU et al., 2023).

A detecção de anomalias consiste em identificar eventos e padrões inesperados que diferem significativamente dos eventos considerados normais (LIU et al., 2020; ALAM et al., 2025). No contexto de ML, anomalias também são definidas como *outliers* que, de acordo com Hawkins (1980), consistem em amostras cujos valores se desviam tanto das restantes que suscitam suspeitas sobre o mecanismo que as gerou. A capacidade de trabalhar de forma não supervisionada permite ultrapassar o problema da obtenção de rótulos em aplicações reais de detecção de anomalias. (BRITO, 2022). Um aspecto central nas abordagens de detecção de anomalias consiste na capacidade de sintetizar o comportamento de sistemas multivariados em um único indicador escalar, usualmente denominado *Anomaly Score (AS)*, o qual quantifica o grau de desvio em relação ao comportamento considerado normal (LI et al., 2022). Embora amplamente empregado no monitoramento de máquinas rotativas, o conceito de AS também é recorrente em aplicações bem-sucedidas de detecção de anomalias em diferentes domínios, incluindo séries temporais multivariadas, sistemas industriais, saúde, redes de comunicação e Internet das Coisas, conforme evidenciado por revisões recentes da literatura (ALAM et al., 2025).

O monitoramento e o diagnóstico de falhas em máquinas rotativas têm se beneficiado do

uso de modelos de Inteligência Artificial baseados em técnicas de Aprendizado de Máquina. Entre os métodos supervisionados, destacam-se redes neurais artificiais (ANN), árvores de decisão (DT), *k-nearest neighbors* (*k*-NN), *Naïve Bayes* (NB) e *support vector machines* (SVM), enquanto abordagens não supervisionadas incluem algoritmos como *k-means*, clusterização hierárquica, modelos de mistura Gaussiana estimados via *expectation-maximization* (GMM-EM) e *density-based spatial clustering of applications with noise* (DBSCAN). Além disso, técnicas de *Deep Learning*, como *Autoencoders* (AE), redes neurais convolucionais (CNN), *deep belief networks* (DBN) e redes neurais recorrentes (RNN), têm sido amplamente empregadas. Nessas arquiteturas, a extração de características é realizada de forma automática pelo próprio modelo, a partir do aprendizado de representações não lineares diretamente dos dados, o que as torna particularmente adequadas para abordagens de detecção de anomalias baseadas na identificação de desvios em relação ao comportamento operacional normal (GONÇALVES, 2024).

A detecção precoce e confiável de falhas em rotores é investigada por Tahir et al. (2016), com foco específico em desbalanceamento e desalinhamento, condições que frequentemente apresentam padrões de frequência semelhantes e, portanto, difíceis de diferenciar. Os autores ressaltam que intervenções de correção de desbalanceamento, como a adição ou remoção de massa, podem agravar o problema quando aplicadas equivocadamente em casos de desalinhamento. Para lidar com essa dificuldade, o estudo propõe o uso combinado de vibrações radiais e axiais, avaliando a sensibilidade estatística de características extraídas no domínio do tempo a partir de sinais multieixo. A eficácia do método foi avaliada por meio de *Support Vector Machines* (SVM), resultando em 100% de precisão na distinção entre as falhas analisadas, assegurando assim um diagnóstico preciso antes da adoção de ações corretivas.

Rai et al. (2017) propõem um método de avaliação da degradação de rolamentos baseado na extração de características via *Empirical Mode Decomposition* e posterior agrupamento por K medóides. Os centróides dos estados saudável e degradado são utilizados como referência para calcular uma curva de valor de confiança, que quantifica a dissimilaridade do sinal de teste em relação ao comportamento normal e atua como indicador de saúde. Aplicada a sinais de vibração de ensaios até a falha, a abordagem mostrou-se mais sensível que métricas clássicas (RMS, curtose) e que métodos de diagnóstico como análise de envelope, especialmente na detecção precoce da degradação. O estudo indica ainda que a análise de envelope (EA) pode complementar a técnica como verificação adicional, e destaca o potencial do K medóides como ferramenta eficaz para monitoramento de desempenho em rolamentos.

Amruthnath e Gupta (2018) destacam a crescente relevância da manutenção preditiva e o papel central da detecção de falhas nesse contexto, enfatizando a dificuldade recorrente na seleção de métodos robustos e precisos para cenários industriais. Diante da escassez ou ausência de dados históricos, os autores apontam algoritmos de *aprendizado não supervisionado* como alternativas viáveis para modelagem. O estudo avaliou diversos métodos — incluindo *PCA*, T^2 , clusterização hierárquica, *K-means*, *Fuzzy C-Means*, aplicados a sinais de vibração de um exaustor, comparando seu desempenho, precisão e robustez. Ao final, os autores propuseram uma metodologia sistemática para contrastar algoritmos e apoiar a escolha do modelo mais adequado para detecção de falhas em ambientes reais.

Em (LIU et al., 2018) é apresentada uma revisão abrangente das pesquisas recentes e do desenvolvimento de métodos de aprendizado aplicados ao diagnóstico de falhas em máquinas rotativas, considerando tanto os fundamentos teóricos quanto sua viabilidade prática. Os autores definem três tarefas essenciais nesse contexto: (i) identificar se o equipamento opera em condição normal ou apresenta desvios — caracterizando a detecção de anomalias; (ii) reconhecer a falha incipiente e determinar sua causa — correspondente à classificação de falhas; e (iii) prever a evolução futura do estado de degradação — associada ao prognóstico. O estudo também discute uma ampla gama de algoritmos, desde classificadores tradicionais, como *k Nearest Neighbors* (*k*-NN), *Naïve Bayes* (NB) e *Support Vector Machines* (SVM), até modelos mais avançados, como *Artificial Neural Networks* (ANNs) e *Deep Neural Networks* (DNN). Além de apresentarem as vantagens e limitações de cada abordagem, os autores ressaltam que as técnicas de aprendizado de máquina permanecem como alternativas promissoras e amplamente atrativas para o diagnóstico de falhas em máquinas rotativas.

Wang et al. (2018) propõem uma abordagem inteligente de diagnóstico de falhas em três estágios voltada ao monitoramento prático de processos industriais, buscando superar limitações de métodos tradicionais. A técnica incorpora um novo esquema de processamento de características, combinando detecção de tendências baseada em *Variational Mode Decomposition* e um algoritmo de ponderação adaptativa para aumentar a precisão com menor custo computacional. Os autores introduzem ainda o método *Adaptive Density Peak Search* (ADPS) para clusterização, que apresentou desempenho superior a técnicas convencionais como o *Density Peak Search* original, *Affinity Propagation* e *K-medoids*. A metodologia foi validada em três aplicações industriais distintas, demonstrando resultados consistentes no diagnóstico de falhas em rolamentos e engrenagens, mesmo sob conhecimento prévio limitado. Os autores destacam,

por fim, que a abordagem é adequada para monitoramento adaptativo em condições variáveis e pode ser aplicada a dados não rotulados obtidos até a falha, reforçando seu potencial para uso em ambientes industriais online.

Em (ZHANG et al., 2019) é discutida a relevância da manutenção preditiva (*Predictive Maintenance* – PdM) baseada em métodos de aprendizado de máquina para o diagnóstico de falhas e a estimativa da vida útil remanescente de equipamentos industriais. Os autores estruturam o processo de PdM orientado a dados em três subprocessos principais: (i) aquisição e pré-processamento dos sinais; (ii) engenharia de características, envolvendo extração, concatenação e seleção; e (iii) treinamento dos modelos e realização das previsões. O estudo destaca que a pesquisa em PdM ainda se encontra majoritariamente em estágio experimental, com grande parte dos trabalhos fundamentados em conjuntos de dados públicos amplamente utilizados, como os da *Case Western Reserve University* (CWRU) e do *Intelligent Maintenance Systems* (IMS). O artigo dá ênfase especial às abordagens orientadas a dados e compara o desempenho de diferentes algoritmos de aprendizado — desde classificadores tradicionais, como *Logistic Regression* (LR), *Support Vector Machines* (SVM), *Decision Trees* (DT) e *Random Forest* (RF), até métodos de *deep learning*, como *Artificial Neural Networks* (ANNs), *Deep Neural Networks* (DNNs) e *Autoencoders* (AEs), frequentemente considerados mais promissores. Os autores também sintetizam os principais tipos de características utilizadas e discutem tendências e direções futuras para a área.

Lei et al. (2020) apresenta uma revisão abrangente sobre as aplicações de métodos de aprendizado de máquina no diagnóstico de falhas, oferecendo uma visão histórica da área, seus avanços recentes e perspectivas futuras. Os autores discutem os principais métodos supervisionados e não supervisionados, bem como as técnicas de *deep learning*, destacando suas contribuições e limitações no contexto do monitoramento inteligente de máquinas. A revisão enfatiza a forte dependência de dados reais e rotulados provenientes dos equipamentos analisados, ressaltando que a qualidade e a disponibilidade desses dados são fatores determinantes para o desempenho dos modelos. Um ponto central do trabalho é a relevância crescente dos modelos explicáveis (*explainable models*) no diagnóstico de falhas, especialmente a partir de 2017, impulsionada pela necessidade de interpretar decisões de sistemas automatizados em cenários industriais críticos. Por fim, os autores concluem que, embora o aprendizado profundo tenha alcançado avanços significativos, os métodos tradicionais de aprendizado não devem ser descartados. A evolução da aprendizagem estatística, aliada ao aumento do *Big Data* industrial, pode favorecer

o desenvolvimento de sistemas diagnósticos com parâmetros, características e resultados mais transparentes e de fácil compreensão.

Em (WANG et al., 2020) é empregada uma abordagem de *clusterização K-means* para o diagnóstico de falhas em máquinas rotativas no contexto de *Big Data* e sistemas ciberfísicos. O método incluiu a extração de características de falhas e a seleção de autovetores relevantes, utilizando sinais de vibração de um compressor centrífugo sob condições de surge, fricção, desalinhamento e operação normal. O modelo de agrupamento apresentou taxas de reconhecimento de 94%, 100% e 80% para surge, fricção e desalinhamento, respectivamente, demonstrando viabilidade prática da *clusterização* para diagnóstico industrial. Os autores destacam, contudo, a necessidade de estudos adicionais para avaliar o desempenho da técnica em cenários mais complexos, envolvendo cinco ou mais estados operacionais.

O estudo desenvolvido por Lupea e Lupea (2023) apresenta um sistema para detecção de defeitos de montagem em um banco de testes rotativo composto por um eixo delgado com disco central, apoiado por rolamentos oscilantes e acionado em velocidade constante por correia dentada. Foram introduzidas diferentes falhas estruturais, como deslocamento axial e excentricidade do disco, além de erros de posicionamento do motor-redutor e da própria correia. Características no domínio do tempo e da frequência, extraídas de sinais de vibração, foram utilizadas como preditores em uma tarefa de classificação multiclasse com oito estados de saúde. A análise exploratória, com métodos supervisionados e não supervisionados, permitiu avaliar a relevância das características, correlações e dificuldades de separação entre classes. Nos experimentos com classificadores do MATLAB, o SVM quadrático apresentou o melhor desempenho, alcançando 99,18% de precisão com o conjunto completo de 41 características e 98,93% com o subconjunto das 18 características mais relevantes obtidas nos eixos X e Y do acelerômetro.

Saha et al. (2022) apresentam uma técnica de diagnóstico inteligente para identificar diferentes falhas em rolamentos de esferas, incluindo defeitos na pista interna, pista externa, gaiola e condição saudável. Foram realizados experimentos para gerar dados defeituosos e os dados de vibração foram convertidos para o domínio da frequência por meio da FFT e usados na tarefa de classificação. O diagnóstico foi realizado com *Support Vector Machines* (SVM), cujo desempenho foi aprimorado com *Particle Swarm Optimization* (PSO). O SVM otimizado alcançou 93,9% de precisão, superando o SVM ajustado por validação cruzada (92%) e outros métodos tradicionais avaliados.

Em (SINGH et al., 2023) é apresentada uma revisão de duas décadas (2000–2020) de aplicações de IA em monitoramento e diagnóstico de falhas de máquinas rotativas no contexto da Indústria 4.0, cobrindo *ANN*, *SVM*, *fuzzy logic* e abordagens de *deep learning* para motores de indução, rolamentos, engrenagens e bombas centrífugas, com síntese de resultados, lacunas e tendências de pesquisa. A revisão destaca o fluxo típico de aquisição de dados, extração de características e decisão assistida por IA, o papel de sinais de vibração, corrente e acústica, e o ganho de desempenho com DL quando a engenharia de características é limitada. Embora o foco principal seja diagnóstico supervisionado, o mapeamento dos estágios e sinais mais informativos é diretamente aproveitável em detecção de anomalias (treinando modelos com condição normal e medindo desvios), apontando oportunidades para ampliar métodos não supervisionados em cenários com poucos rótulos.

Das et al. (2023) apresentam uma revisão ampla de diagnóstico e prognóstico inteligente de falhas (IFDP) em máquinas rotativas no contexto da Indústria 4.0, destacando desafios centrais (dados escassos/rotulados, falhas compostas, adaptação de domínio, fusão multissensor, seleção/otimização de algoritmos) e a necessidade de métricas além da acurácia para avaliação confiável. Cada componente da máquina (rolamentos, engrenagens, rotor, estator, eixo, entre outros) é analisado individualmente, ressaltando como suas particularidades estruturais e dinâmicas afetam sinais, indicadores e o desempenho dos modelos de diagnóstico. Embora o foco histórico seja diagnóstico/prognóstico supervisionados, os autores salientam que detecção de anomalias (especialmente com dados não rotulados ou distribuídos de forma desequilibrada) é essencial para identificar desvios precoces do comportamento normal, contornar a escassez de rótulos e dar suporte a PdM em cenários reais, e que inclui abordagens não supervisionadas/semissupervisionadas, *transfer learning* e pipelines com *anomaly scores* (densidade/distância/erro de reconstrução) integrados a métricas mais informativas (matriz de confusão, *precision/recall*, AUC).

Mian et al. (2023) desenvolveu um *framework AIoT* para detecção de anomalias em máquinas rotativas, integrando aquisição por acelerômetro e decisão em *edge* (*Raspberry Pi 4B*) via *SVM*, com monitoramento e atuação remotos via plataforma web. O método emprega janelas deslizantes e nove características estatísticas no tempo (*RMS*, pico a pico, *kurtosis*, *skewness*, etc.) e foi validado em três bancos (engrenagens, rolamentos e motor de indução), obtendo 100% em tarefa binária e, em problemas multiclasse, *F1-scores* elevados ($\approx 91\text{--}99\%$). O estudo destaca a viabilidade industrial de modelos não profundos e de baixo custo computacional para

detecção distribuída de anomalias.

Em (AFRIDI et al., 2023) é apresentado um sistema de prognóstico de rolamentos baseado em *LSTM* aplicado diretamente ao sinal bruto de vibração, eliminando a necessidade de seleção de característica no tempo ou na frequência. O modelo foi treinado de forma supervisionada para predição futura do sinal (*time-series forecasting*) usando um *dataset run-to-failure* de rolamentos (Teste 02 do *IMS Bearing Data*) e posteriormente validado/aplicado em um *dataset* distinto, também *run-to-failure* (Teste 03). Apesar do foco principal em prognóstico, pode ser integrado a esquemas de detecção de anomalias via desvio de erro (por exemplo, limiares de *RMSE/MAE* sobre o resíduo de previsão) para indicar degradações incipientes. Nos experimentos, o método obteve *RMSE* muito baixo ($\approx 0,014-0,010$) e erros médios reduzidos (*MAE*, *NMAE*, *MAPE*), superando abordagens com *features* e outras arquiteturas, e demonstrou generalização em dados de usinas hidrelétrica e eólica ($RMSE \approx 0,11-0,12$). O estudo evidencia que *LSTM* com dados brutos simplifica o *pipeline*, mantém a precisão em *run-to-failure* e é adequado a prognóstico em tempo real em ambientes industriais.

Em (EGAJI et al., 2020) é proposta uma estratégia de detecção de anomalias em motores elétricos a partir de vibração de quatro rolamentos, extraíndo estatísticas no tempo (desvio padrão, variância, *skewness*, *kurtosis*, mínimo, máximo) e reduzindo as por *PCA* a uma dimensão. Em aprendizado não supervisionado/semisupervisionado, os autores treinaram regressores (*k NN*, *SVR*, *RF*) apenas com dados saudáveis e detectaram anomalias por desvio de reconstrução ($MAE > \text{média} + 2 * \text{desvio padrão}$). Testado em um ensaio *run to failure* (20 kHz; janelas de 1 s a cada 10 min), o método antecipou a falha de pista externa com 1,3–1,1 dias de antecedência, com melhor desempenho do *Random Forest* (menos falsos positivos) em relação a *SVR* e *k NN*.

A revisão bibliográfica apresentada evidencia o potencial das técnicas de Aprendizado de Máquina para monitoramento da condição e detecção de anomalias em máquinas rotativas, abrangendo desde métodos clássicos de classificação e agrupamento até arquiteturas de *Deep Learning*, como Autoencoders, CNNs, RNNs e LSTMs. Entretanto, observa-se que os trabalhos revisados frequentemente concentram-se na avaliação de modelos específicos ou em aplicações voltadas a diagnóstico supervisionado e prognóstico. Adicionalmente, verificou-se uma oportunidade de investigação na comparação de diferentes abordagens não supervisionadas de detecção de anomalias aplicadas diretamente a séries temporais de vibração de distintos sistemas rotativos, bem como na avaliação do uso de técnicas de *Change Point Detection* (CPD) em conjunto com modelos de aprendizado de máquina.

Esta Dissertação de Mestrado busca contribuir por meio do desenvolvimento e da avaliação comparativa de múltiplos modelos não supervisionados, incluindo arquiteturas baseadas em MLP e LSTM, aplicados a cinco *benchmarks* de vibração, além da análise do potencial da técnica de CPD como mecanismo complementar para identificação de mudanças persistentes no comportamento vibratório. A comparação dos resultados obtidos em diferentes conjuntos de dados permite extrair conclusões úteis para o desenvolvimento de estratégias de manutenção preditiva mais robustas, combinando maior eficácia na detecção de anomalias com uso mais eficiente de recursos computacionais e de análise especializada.

Capítulo 3

Machine Learning e Redes Neurais

Machine Learning (ML) é um ramo da Inteligência Artificial que permite que modelos computacionais aprendam padrões a partir de dados, sem a necessidade de regras explícitas previamente programadas. Em aplicações industriais, como o monitoramento de vibração em rotores, o ML é especialmente relevante devido à sua capacidade de lidar com grandes volumes de dados, extrair conhecimento a partir das informações disponíveis e gerar subsídios para a tomada de decisões.

Os métodos de *Machine Learning* podem ser classificados, de forma geral, de acordo com o tipo de supervisão disponível durante o treinamento. No aprendizado supervisionado, o modelo é treinado a partir de dados rotulados, nos quais as entradas estão associadas às saídas desejadas, sendo amplamente utilizado em tarefas de classificação e regressão. Já no aprendizado não supervisionado, não há rótulos explícitos, e o objetivo consiste em identificar padrões, estruturas ou comportamentos atípicos nos dados. Entre esses dois extremos, encontram-se abordagens semissupervisionadas, nas quais apenas uma parte do conjunto de dados possui rótulos. Em aplicações de detecção de anomalias, especialmente em contextos industriais, é comum o uso de estratégias não supervisionadas ou semissupervisionadas, uma vez que dados representativos de falhas são escassos ou inexistentes durante a fase de treinamento (LEI, 2017). Nesse contexto, diferentes técnicas podem ser empregadas conforme a disponibilidade de rótulos e o objetivo da aplicação.

Entre as técnicas mais simples de *Machine Learning*, destacam-se métodos clássicos amplamente utilizados em função de sua facilidade de implementação e interpretação. A regressão, em suas diferentes variantes, é empregada para modelar relações entre variáveis e realizar previsões de valores contínuos. O algoritmo *k-nearest neighbors* (kNN) baseia-se na similaridade entre

amostras para tarefas de classificação ou regressão, sendo particularmente eficaz em conjuntos de dados de menor complexidade. Já o *Isolation Forest* é uma técnica voltada à detecção de anomalias, que identifica padrões atípicos por meio do isolamento de observações raras, apresentando bom desempenho em cenários industriais de monitoramento de condições, como a análise de vibração de máquinas rotativas. Além dessas técnicas, outros métodos clássicos de aprendizado de máquina, como árvores de decisão, máquinas de vetores de suporte e classificadores bayesianos, também são amplamente utilizados em problemas de classificação e regressão (HASTIE et al., 2009).

Apesar de sua ampla utilização, essas técnicas apresentam limitações na modelagem de relações altamente não lineares e em problemas que envolvem grande complexidade de dados, o que motiva o uso de modelos mais sofisticados, como as redes neurais artificiais. As redes neurais artificiais (RNAs) foram desenvolvidas tomando como referência a estrutura e o funcionamento do sistema nervoso biológico, com o objetivo de simular a capacidade de aprendizado do cérebro humano (GÉRON, 2019). Dessa forma, a arquitetura das redes neurais é inspirada nos neurônios humanos, sendo apresentado um esquema típico de ambos os tipos de neurônios na figura 3.1.

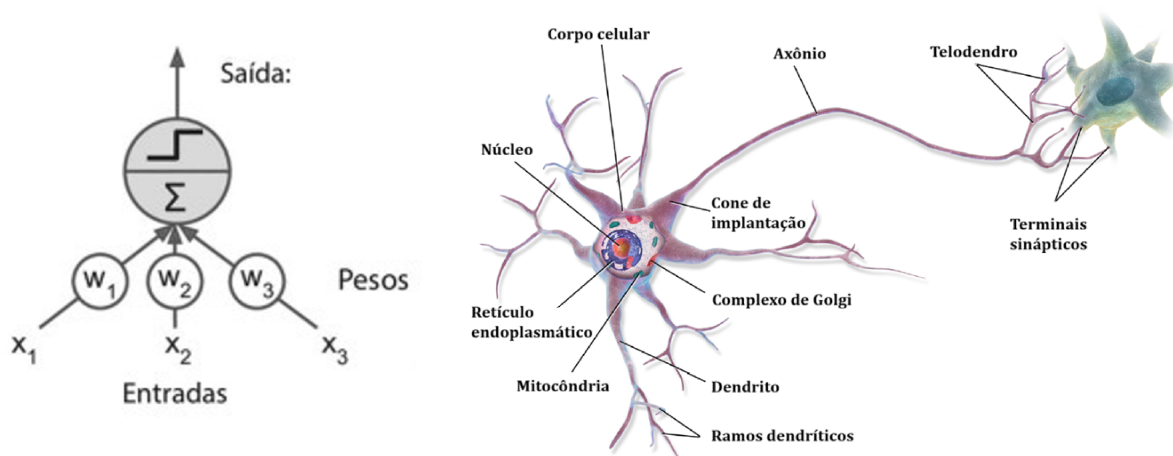


Figura 3.1: Representação típica esquemática de neurônio de rede neural e de célula nervosa. Adaptado de GÉRON (2019).

A aplicação de redes neurais artificiais está diretamente associada ao conceito de aprendizado de máquina, no qual técnicas computacionais são empregadas para identificar padrões e correlações nos dados, permitindo sua utilização em tarefas de previsão, agrupamento e classificação. Adicionalmente, as RNAs possuem capacidade adaptativa de aprendizado, que pode ser aprimorada ao longo do tempo por meio da realimentação do processo de treinamento com novas informações, resultando na otimização de seu desempenho (CABRAL, 2023).

As redes neurais artificiais são compostas, de forma geral, por três elementos principais:

a camada de entrada, uma ou mais camadas ocultas e a camada de saída. A primeira RNA a ser proposta foi o modelo *perceptron* desenvolvido por Frank Rosenblatt que, ao empregar apenas uma camada de neurônios, apresentou bons resultados em problemas de classificação com aprendizado supervisionado.

3.1 Conceitos fundamentais de rede neural simples

Uma das redes neurais artificiais mais simples é o modelo *Perceptron*, desenvolvido em 1957 por Frank Rosenblatt. Esse modelo constitui uma estrutura básica de rede neural, fundamentada na manipulação matemática dos dados de entrada com o objetivo de produzir resultados de saída com a maior precisão possível. O funcionamento do algoritmo baseia-se na soma ponderada das entradas, na qual cada valor de entrada é multiplicado por um peso associado e, posteriormente, somado a um termo de viés. Em seguida, o resultado dessa operação é aplicado a uma função de ativação, responsável por transformar o valor obtido de acordo com o tipo de função adotada e a natureza dos dados processados. A função de ativação pode ser entendida como um mecanismo de transformação não linear, essencial para possibilitar a modelagem de problemas mais complexos (CABRAL, 2023).

A figura 3.2 apresenta um esquema ilustrativo do modelo *Perceptron*. À esquerda, é mostrada a Unidade Linear com *Threshold* (*Linear Threshold Unit* – LTU), responsável por realizar a multiplicação dos dados de entrada pelos respectivos pesos, efetuar a soma desses valores e aplicar a função de ativação. À direita, observa-se o diagrama de um conjunto de *perceptrons*, composto por uma camada de entrada, incluindo um neurônio de viés, e uma camada de saída formada pelas LTUs.

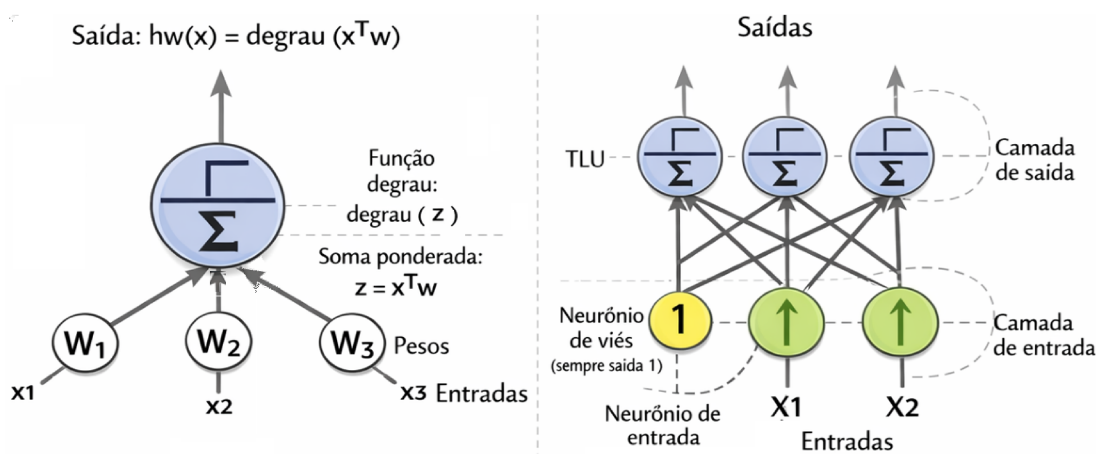


Figura 3.2: Grupo de *Perceptrons*. Adaptado de GÉRON (2019).

O processo de manipulação matemática, que envolve a multiplicação dos pesos e a soma com os vieses, é propagado de camada em camada, desde a entrada até a saída da rede. Esse procedimento é conhecido como *feedforward*, caracterizando um fluxo direto da informação da camada de entrada para a camada de saída. Em arquiteturas mais simples, como o *Perceptron*, esse processo pode ser suficiente para a realização do treinamento.

Entretanto, em redes neurais mais complexas, é necessário o uso do processo de *backpropagation*, responsável pela atualização dos pesos e dos vieses ao longo do treinamento. O *backpropagation* é executado a partir da comparação entre a saída produzida pela rede e os valores de referência presentes no conjunto de dados, o que define a chamada função de perda, ou *loss function*. Com base no valor da função de perda, os pesos e vieses são ajustados de forma retrógrada, isto é, da camada de saída em direção à camada de entrada, permitindo a minimização do erro e a melhoria progressiva do desempenho da rede.

Portanto, o treinamento de uma rede neural envolve um ciclo iterativo composto por três etapas principais: (i) a propagação direta (*feedforward*), na qual os dados de entrada são combinados por meio da multiplicação pelos pesos e da soma com os vieses; (ii) o cálculo da função de perda ao final da rede, a partir da comparação entre a saída predita e os valores de referência; e (iii) a atualização dos pesos e vieses por meio da propagação reversa do erro (*backpropagation*). Esse procedimento é repetido ao longo de múltiplas iterações até que o erro seja minimizado ou que um critério de parada seja atingido. A figura 3.3 ilustra, de forma esquemática, o fluxo de informação no *feedforward* e o caminho inverso de correção dos parâmetros no *backpropagation*.

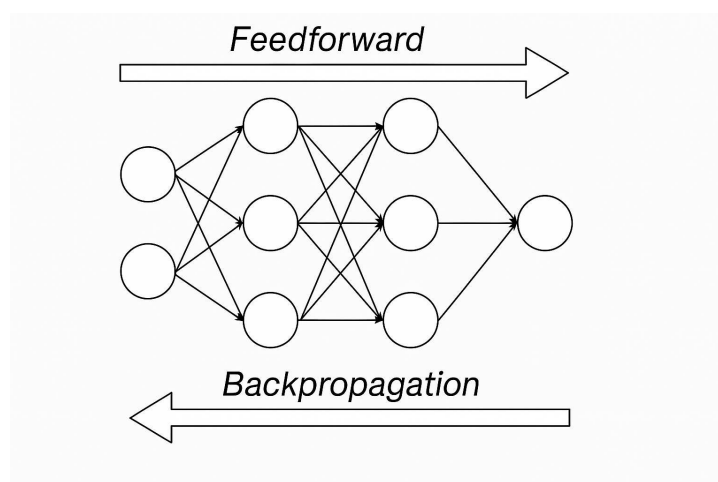


Figura 3.3: Esquema ilustrativo do *feedforward* e do *backpropagation* em uma rede neural. Autoria própria.

A evolução das redes neurais artificiais levou ao desenvolvimento de arquiteturas mais complexas, caracterizadas pela adição de novas camadas de unidades lineares com *threshold* (LTUs) entre a camada de entrada e a camada de saída. Essas camadas intermediárias, denominadas camadas ocultas, dão origem às redes do tipo *Perceptron* de múltiplas camadas, conhecidas como *Multi Layer Perceptron* (MLP). Quando mais de uma camada oculta está presente na arquitetura, a rede passa a ser classificada como profunda (*deep neural network*). A figura 3.4 apresenta um esquema ilustrativo de uma rede MLP.

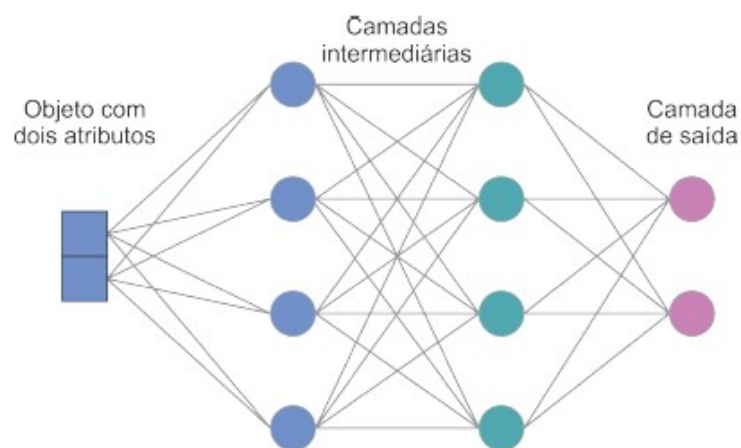


Figura 3.4: Rede *Multi Layer Perceptron* (MLP) com duas camadas intermediárias. Adaptado de [Faceli \(2011\)](#).

O aumento do número de camadas ocultas nas arquiteturas de redes neurais deu origem ao conceito de *deep learning*, no qual modelos com múltiplas camadas são capazes de aprender representações hierárquicas dos dados. Nessas arquiteturas profundas, as camadas iniciais tendem a extrair características mais simples, enquanto camadas mais profundas capturam padrões progressivamente mais complexos e abstratos. O *deep learning* tem se destacado em aplicações que envolvem grandes volumes de dados e elevada complexidade, como processamento de sinais, visão computacional e análise de séries temporais, ampliando significativamente a capacidade de generalização dos modelos em comparação com redes neurais rasas.

3.2 Multilayer Perceptron (MLP)

As redes neurais do tipo *Multilayer Perceptron* (MLP) constituem uma das arquiteturas mais tradicionais e amplamente empregadas no contexto de *Machine Learning*, sendo particularmente adequadas para modelar relações não lineares entre conjuntos de variáveis de entrada e saída. Um MLP é composto por camadas totalmente conectadas (*fully connected*), organizadas em uma

camada de entrada, uma ou mais camadas ocultas e uma camada de saída (ver figura 3.4), nas quais cada neurônio realiza uma combinação linear dos sinais de entrada seguida da aplicação de uma função de ativação não linear. Essa estrutura confere ao MLP elevada capacidade de aproximação de funções complexas, o que justifica seu amplo uso em problemas de classificação, regressão e diagnóstico inteligente (LEI, 2017).

Na aplicação de detecção de falhas em máquinas rotativas, os MLPs são tipicamente empregados a partir de características extraídas de sinais de vibração, que permitem discriminar o estado do equipamento, ao mesmo tempo em que reduzem variações irrelevantes e ruídos presentes nos sinais brutos. Conforme discutido por Lei (2017), essas características constituem as entradas do modelo, enquanto as saídas correspondem à classificação da condição da máquina. O objetivo fundamental do treinamento do MLP é minimizar a discrepância entre as saídas estimadas pelo modelo e os rótulos desejados, processo realizado por meio do ajuste iterativo dos parâmetros da rede a partir de algoritmos de otimização baseados em retropropagação do erro (*backpropagation*) (LEI, 2017).

Em aplicações industriais reais a disponibilidade de dados rotulados de falha é limitada, motivando o uso de abordagens não supervisionadas. Nesse cenário, destaca-se o emprego do MLP na forma de *Autoencoder*, onde a rede é treinada com amostras consideradas normais para reconstruir apenas sinais associados à condição normal de operação. O *Autoencoder* MLP é estruturado a partir de um codificador (*encoder*), responsável por mapear o vetor de entrada para uma representação de menor dimensionalidade, e de um decodificador (*decoder*), que reconstrói o sinal original a partir dessa representação comprimida. A presença de uma camada de gargalo, denominada *bottleneck*, força o modelo a aprender representações compactas e relevantes do sinal, princípio amplamente explorado em aplicações de detecção de anomalias.

A principal vantagem do MLP reside em sua simplicidade estrutural e baixo custo computacional quando comparado a arquiteturas mais complexas, como redes recorrentes profundas, tornando o modelo atrativo para aplicações industriais. Dessa forma, o uso do MLP neste trabalho justifica-se por permitir estabelecer uma base comparativa consistente com outras arquiteturas e técnicas de detecção de anomalias avaliadas no contexto desta dissertação.

3.3 Redes Neurais Recorrentes

As Redes Neurais Recorrentes (*Recurrent Neural Networks* – RNNs) são uma classe de modelos especialmente adequada ao processamento de dados sequenciais, como séries temporais, linguagem natural e sinais amostrados ao longo do tempo. De forma distinta das redes do tipo *feedforward*, as RNNs incorporam conexões recorrentes que permitem que informações de instantes anteriores influenciem o processamento no instante atual, oferecendo ao modelo uma forma de “memória” por meio do estado oculto. Em termos gerais, a cada passo temporal t , a rede recebe uma entrada x_t e produz um estado oculto h_t , que resume informações relevantes do histórico recente da sequência (GOODFELLOW et al., 2016).

De maneira simplificada, o funcionamento de uma RNN pode ser descrito por relações do tipo:

$$h_t = \phi(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \quad (3.1)$$

$$y_t = \psi(W_{hy}h_t + b_y), \quad (3.2)$$

onde $\phi(\cdot)$ e $\psi(\cdot)$ representam funções de ativação (ou mapeamentos) e as matrizes de pesos W_{xh} , W_{hh} e W_{hy} são compartilhadas ao longo do tempo. Isto é, os mesmos parâmetros são aplicados em todos os instantes da sequência. Esse compartilhamento de pesos é uma característica central das RNNs e contribui para que o modelo lide com sequências de comprimento variável (GOODFELLOW et al., 2016).

A figura 3.5 apresenta uma representação compacta de uma célula recorrente. No diagrama, o bloco A recebe a entrada x_t e produz o estado oculto h_t , enquanto o laço indica que o estado no instante t depende do estado no instante anterior, permitindo a propagação de informação ao longo da sequência. Essa forma de representação enfatiza a natureza recorrente do modelo, na qual o processamento atual é condicionado pela dinâmica interna acumulada nos passos anteriores.

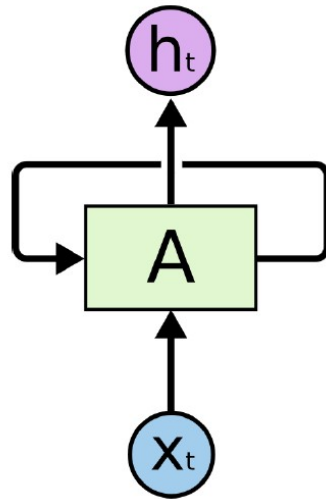


Figura 3.5: Representação compacta de uma RNN, destacando o laço recorrente que permite a propagação de informação entre instantes sucessivos. Adaptado de Olah (2015).

Uma maneira equivalente e de maior didática para interpretar uma RNN consiste em expandir o laço no tempo, obtendo uma cadeia de blocos repetidos, como ilustrado na figura 3.6. Nessa representação, a mesma célula A é replicada ao longo dos instantes $t-1, t, t+1$, etc., evidenciando que cada bloco possui os mesmos parâmetros e repassa seu estado ao bloco subsequente. Essa visão em cadeia torna explícita a dependência temporal e facilita a compreensão de como a informação flui pela sequência, bem como de como erros podem ser propagados ao longo dos passos temporais durante o treinamento (GOODFELLOW et al., 2016).

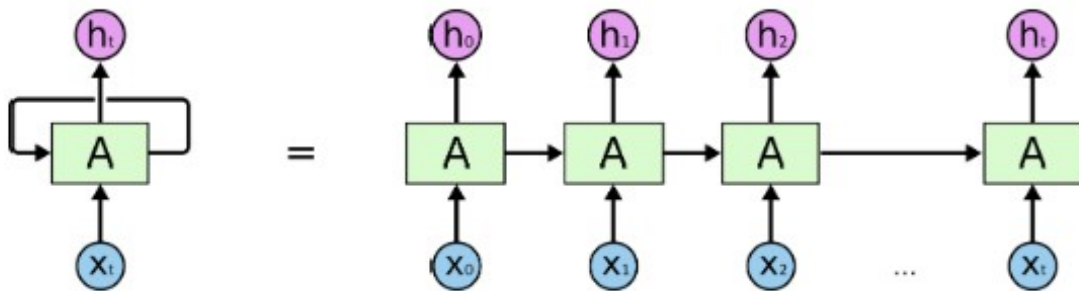


Figura 3.6: RNN “desenrolada” no tempo, evidenciando o compartilhamento de parâmetros e a propagação do estado oculto entre instantes sucessivos. Adaptado de Olah (2015).

Em geral, o treinamento de RNNs é realizado por meio do procedimento conhecido como *backpropagation through time* (BPTT), no qual o modelo desenrolado é tratado como uma rede profunda ao longo do eixo temporal, e os gradientes são computados retropropagando o erro a partir dos instantes mais recentes para os mais antigos. Contudo, RNNs convencionais podem apresentar dificuldades para capturar dependências temporais de longo alcance, associadas prin-

principalmente aos problemas de desvanecimento e explosão de gradientes ao longo de sequências extensas ([PASCANU et al., 2013](#); [GOODFELLOW et al., 2016](#)). Essas limitações motivaram o desenvolvimento de arquiteturas recorrentes aprimoradas, capazes de reter informação relevante por intervalos maiores, como as LSTMs (*Long Short-Term Memory*) e as GRUs (*Gated Recurrent Units*), que introduzem mecanismos baseados em portas para regular o fluxo de informação e permitir a preservação de estados internos relevantes por intervalos mais longos.

3.4 Long Short-Term Memory (LSTM)

As redes LSTM *Long Short-Term Memory* (LSTM) são um tipo especial de redes neurais recorrentes, sendo desenvolvidas para superar as limitações práticas das RNNs em aprender dependências de longo prazo em sequências. Embora RNNs sejam, em princípio, adequadas para dados temporais por incorporarem realimentação, seu treinamento por métodos baseados em gradiente tende a se tornar ineficiente quando a informação relevante está separada por muitos passos de tempo. Esse comportamento decorre do escoamento inadequado do erro ao longo do tempo, levando a sinais de gradiente que podem desaparecer (*vanishing gradients*) ou explodir (*exploding gradients*), o que dificulta a atribuição de crédito a eventos distantes na sequência. Foi nesse contexto que Hochreiter e Schmidhuber propuseram a LSTM como uma arquitetura recorrente especificamente projetada para preservar e controlar o fluxo de informação e de erro em longos intervalos temporais, superando o gargalo de aprendizagem observado em RNNs convencionais ([HOCHREITER; SCHMIDHUBER, 1997](#)).

A contribuição central da LSTM é introduzir um mecanismo de memória interna com controle explícito do que deve ser mantido, atualizado ou exposto ao restante da rede. Em termos conceituais, a LSTM pode ser entendida como uma RNN cuja unidade recorrente é enriquecida por um estado de memória de longo prazo (o *cell state*) e por mecanismos de controle (portas) que regulam a passagem de informação. Essa concepção permite que o modelo retenha contexto por longos horizontes, ao mesmo tempo em que evita que informações irrelevantes se acumulem indefinidamente. A motivação original enfatizada por Hochreiter e Schmidhuber é justamente garantir um fluxo de erro mais estável ao longo do tempo, tornando factível aprender relações temporais com defasagens que seriam proibitivas para algoritmos recorrentes tradicionais ([HOCHREITER; SCHMIDHUBER, 1997](#)).

No contexto de monitoramento de condição e prognóstico de falhas, essa capacidade de

modelar dependências de longo prazo é particularmente relevante, pois sinais de vibração frequentemente apresentam padrões informativos intercalados por trechos ruidosos ou pouco discriminativos. Afridi et al. (2023) destaca que, em aplicações de monitoramento de condição baseadas em séries temporais de sensores, pode haver um grande intervalo temporal entre pontos realmente informativos, o que reduz a eficácia de modelos sequenciais clássicos e compromete RNNs devido às dificuldades de treinamento associadas a gradientes instáveis.

A rede LSTM é composta fundamentalmente por três mecanismos de controle, denominados *Forget gate*, *Input gate* e *Output gate*, conforme evidenciado na figura 3.7. Essas portas são responsáveis, respectivamente, por descartar informações irrelevantes, incorporar ou atualizar novas informações e controlar a transmissão do conteúdo armazenado a cada instante da sequência temporal. Esse processo se dá pela atuação sobre dois estados principais: o estado de memória de longo prazo, representado pelo *cell state*, e o estado de curto prazo, representado pelo *hidden state*. O *cell state* pode ser interpretado como um caminho contínuo de informação que percorre toda a rede ao longo do tempo, na qual os dados relevantes são preservados, como amostras com dados de vibração. A adição de informações significativas e a remoção de conteúdos pouco relevantes desse estado são reguladas dinamicamente pelas portas.



Figura 3.7: Arquitetura geral de uma rede LSTM, evidenciando os mecanismos de controle e os estados de memória. Adaptado de (AFRIDI et al., 2023).

A figura 3.8 detalha a estrutura interna das portas da rede LSTM, evidenciando que o controle do fluxo de informação é realizado por operações que determinam o descarte, aproveitamento, armazenamento e a transmissão seletiva das informações recebidas na entrada da célula LSTM, permitindo ao modelo capturar dependências temporais de curto e longo prazo de forma estável.

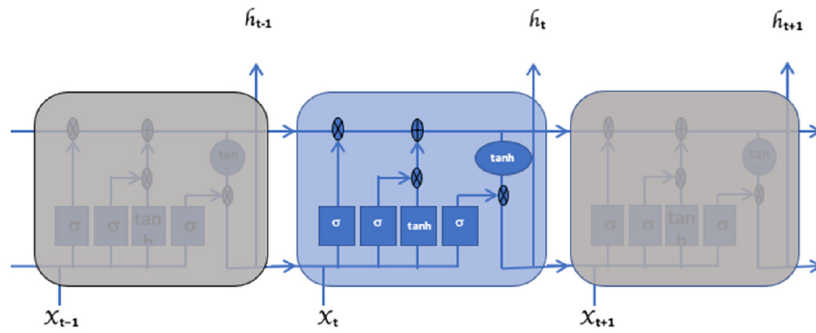


Figura 3.8: Detalhe da estrutura interna das portas de uma rede LSTM, evidenciando o controle do fluxo de informação (AFRIDI et al., 2023).

O trabalho de Afridi et al. (2023) analisou o desempenho de uma arquitetura de rede LSTM para detecção de falhas em rolamentos, utilizando dados brutos de vibração treinados de forma supervisionada. Os autores compararam o desempenho com outros modelos de *deep learning* sobre o mesmo conjunto de dados e sendo obtido melhores resultados com a LSTM, tomando como referência a métrica de RMSE.

Assim, a motivação para o desenvolvimento do modelo LSTM neste trabalho foi avaliar o desempenho de uma arquitetura capaz de explorar dependências temporais de longo prazo com maior eficácia, característica potencialmente relevante para a detecção de anomalias associadas à degradação progressiva de máquinas rotativas.

3.5 Change Point Detection (CPD)

A detecção do ponto de mudança (CPD) em séries temporais consiste na identificação de alterações estruturais em sequências de dados (LIU et al., 2013), utilizando parâmetros estatísticos para inferência como média e desvio padrão. A figura 3.9 ilustra a variação de parâmetros estatísticos em séries temporais. O principal objetivo consiste em analisar os dados levando em consideração valores predecessores e sucessores, de forma a detectar mudanças no padrão de comportamento e identificar os pontos de mudança (GUPTA et al., 2024).

De acordo com Xu et al. (2025), existem registros da aplicação do conceito de CPD nos anos 1950 para controle de qualidade na indústria e, desde então, distintos métodos de CPD vêm sendo utilizados em outros campos como monitoramento ambiental, saúde, finanças e monitoramento de atividades humanas.

Os métodos de *Change Point Detection* (CPD) têm como objetivo identificar a ocorrência, a quantidade e a posição de mudanças estruturais em séries temporais, podendo também segmentar

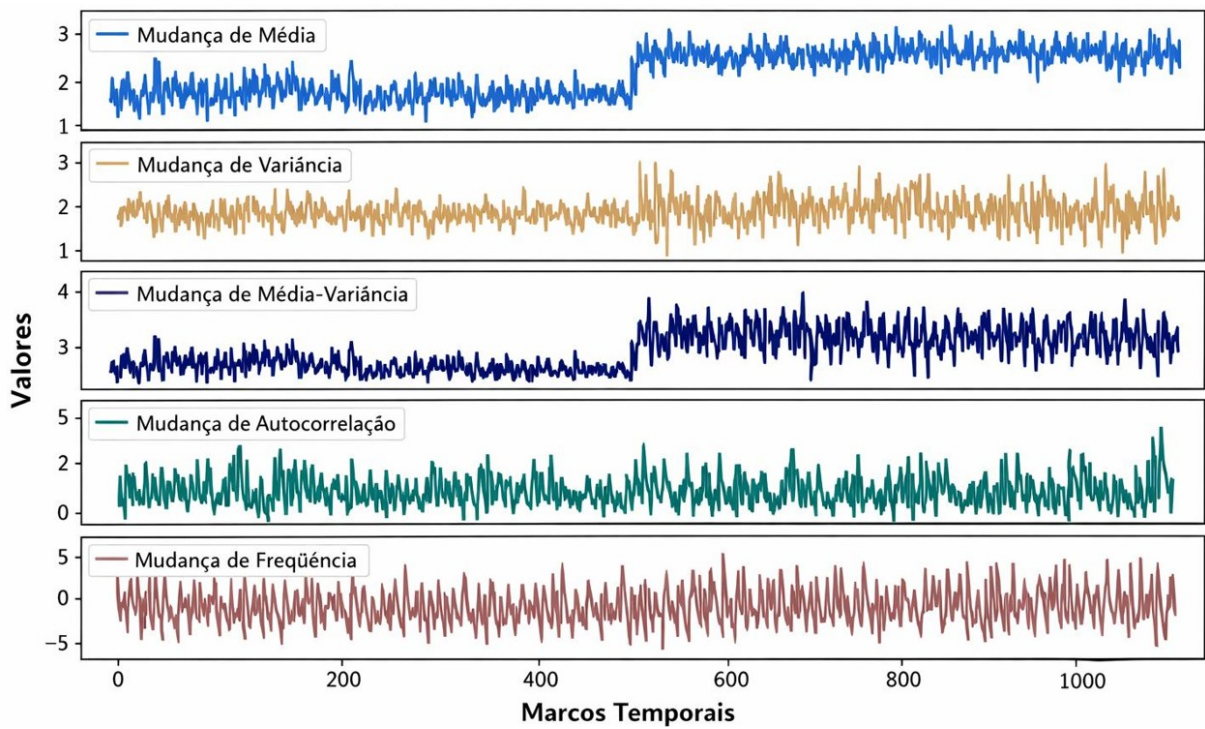


Figura 3.9: Variação de parâmetros estatísticos em séries temporais. Adaptado de (GUPTA et al., 2024).

os dados em intervalos com comportamentos distintos (XU et al., 2025). No contexto de monitoramento de vibração em máquinas rotativas, o CPD permite detectar transições sutis no comportamento do sistema, que podem estar associadas a estágios iniciais de falhas mecânicas. Nessas aplicações, o modelo deve identificar os últimos pontos de comportamento considerado normal detectando transições na série temporal.

A figura 3.10 apresenta uma série temporal com pontos de mudança detectados por CPD.

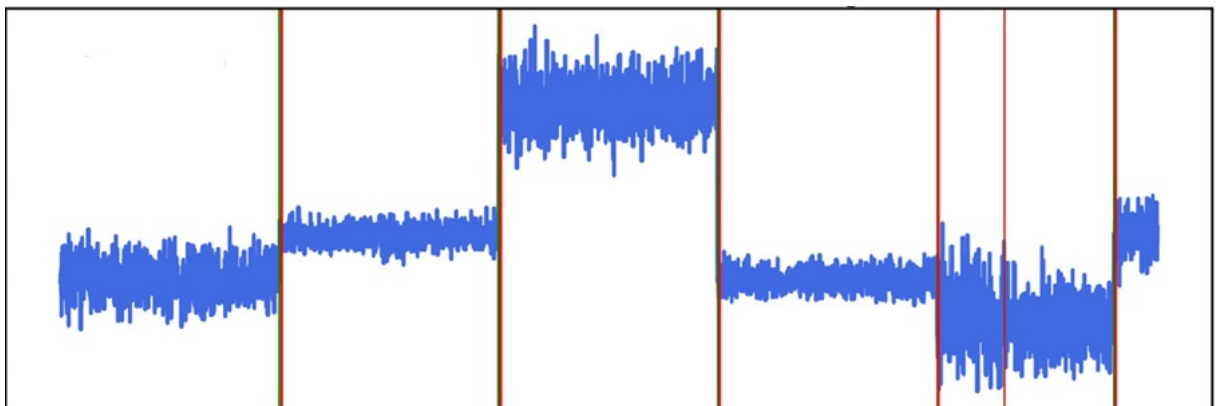


Figura 3.10: Série temporal com pontos de mudança detectados por CPD. Adaptado de (GUPTA et al., 2024).

3.6 Métricas de Desempenho

Em problemas de classificação, como a detecção de anomalias em sinais de vibração de rotores, o desempenho dos modelos de *Machine Learning* pode ser avaliado por meio da matriz de confusão. Essa matriz reúne informações referentes às predições corretas e incorretas realizadas pelo modelo para cada classe, permitindo uma análise detalhada de seu comportamento. A partir da matriz de confusão, diversas métricas podem ser derivadas para a avaliação quantitativa do desempenho dos classificadores (GÉRON, 2019).

A figura 3.11 apresenta a matriz de confusão associada a um problema de classificação binária, no qual a classe positiva é representada pelo sinal (+) e a classe negativa pelo sinal (-). Nesse contexto, os elementos da matriz são definidos da seguinte forma:

- **TP (True Positive) ou VP (Verdadeiro Positivo)**: número total de amostras pertencentes à classe positiva corretamente classificadas como positivas;
- **TN (True Negative) ou VN (Verdadeiro Negativo)**: número total de amostras pertencentes à classe negativa corretamente classificadas como negativas;
- **FP (False Positive ou Falso Positivo)**: número total de amostras pertencentes à classe negativa que foram incorretamente classificadas como positivas;
- **FN (False Negative ou Falso Negativo)**: número total de amostras pertencentes à classe positiva que foram incorretamente classificadas como negativas.

		Classe Predita	
		+	-
Classe Verdadeira	+	TP	FN
	-	FP	TN

Figura 3.11: Matriz de confusão para um problema de classificação binária. Adaptado de Lobão (2020).

O número total de amostras analisadas é dado por

$$n = TP + TN + FP + FN, \quad (3.3)$$

A partir da matriz de confusão, podem ser obtidas diversas métricas de desempenho, amplamente utilizadas na avaliação de modelos de classificação. As principais métricas consideradas

neste trabalho são descritas a seguir.

- **Acurácia** (*accuracy*): corresponde à taxa global de acerto do modelo, sendo calculada pela razão entre a soma dos elementos corretamente classificados (valores da diagonal principal da matriz de confusão) e o número total de amostras analisadas. Essa métrica é expressa por:

$$accuracy = \frac{TP + TN}{n}. \quad (3.4)$$

- **Precisão** (*precision*): indica a proporção de exemplos corretamente classificados como positivos em relação ao total de amostras preditas como positivas pelo modelo. Essa métrica é definida por:

$$precision = \frac{TP}{TP + FP}. \quad (3.5)$$

- **Sensibilidade** (*recall*): também conhecida como taxa de verdadeiros positivos, representa a capacidade do modelo em identificar corretamente as amostras pertencentes à classe positiva. É calculada por:

$$recall = \frac{TP}{TP + FN}. \quad (3.6)$$

- **F1-score**: corresponde à média harmônica entre a precisão e a sensibilidade, sendo uma métrica especialmente útil em cenários com classes desbalanceadas. O *F1-score* é definido como:

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}. \quad (3.7)$$

Convém ressaltar que, diferente da média aritmética, a média harmônica é mais sensível a valores extremos (GÉRON, 2019), o que torna o *F1-score* uma métrica mais rigorosa para avaliar o desempenho de modelos em situações onde há um desequilíbrio significativo entre as classes.

Em problemas de detecção de anomalias em sinais de vibração, é comum a ocorrência de conjuntos de dados fortemente desbalanceados, nos quais a quantidade de amostras normais

é significativamente superior ao número de amostras anômalas. Nesses cenários, a utilização exclusiva da acurácia pode conduzir a interpretações equivocadas do desempenho do modelo, uma vez que classificações corretas da classe majoritária tendem a dominar essa métrica. O *F1-score*, por sua vez, mostra-se mais adequado para esse tipo de aplicação, pois combina a precisão e a sensibilidade em uma única medida, penalizando modelos que apresentam elevado número de falsos alarmes ou baixa capacidade de detecção de eventos anômalos. Dessa forma, o *F1-score* torna-se particularmente relevante em sistemas de monitoramento de vibração, nos quais a correta identificação de poucas falhas reais é mais importante do que o simples acerto da condição normal.

Outra métrica amplamente utilizada para avaliação de classificadores binários é a área sob a curva ROC (*Receiver Operating Characteristic Area Under the Curve*). A curva ROC representa a relação entre a taxa de verdadeiros positivos (*True Positive Rate* ou *Recall*) e a taxa de falsos positivos (*False Positive Rate*) para diferentes limiares de decisão. A métrica ROC AUC resume essa curva em um único valor entre 0 e 1, sendo valores próximos de 1 indicativos de elevada capacidade de discriminação entre as classes.

Embora o *F1-score* tenha sido adotado como principal métrica deste trabalho em função do forte desbalanceamento entre as classes, a ROC AUC fornece uma medida complementar da capacidade discriminatória dos modelos ao longo de diferentes limiares de decisão.

3.7 Classificação de Anomalias por *Score* de Erro

Em aplicações de detecção de anomalias em sinais de vibração de máquinas rotativas, é comum a utilização de abordagens baseadas em aprendizado do comportamento normal do sistema. Nesses métodos, os modelos são treinados predominantemente com dados representativos da operação normal, de modo a aprender padrões característicos do funcionamento saudável da máquina. Durante a fase de operação, o modelo é utilizado para reconstruir ou prever o sinal de vibração, e a discrepância entre o sinal observado e o sinal estimado é utilizada como indicativo da presença de anomalias. Dessa forma, amostras cujo erro de reconstrução ou previsão excede um determinado limiar (*threshold*) são classificadas como anômalas.

O erro entre o sinal real e o sinal estimado pode ser quantificado por diferentes métricas, sendo as mais comuns o erro quadrático médio (*Mean Squared Error* – MSE), o erro absoluto médio (*Mean Absolute Error* – MAE) e suas variações. Essas métricas fornecem um *score* de

anomalia associado a cada amostra ou janela temporal, permitindo a conversão do problema de detecção em uma tarefa de classificação binária a partir da definição de um limiar. A escolha desse limiar exerce influência direta sobre o desempenho do sistema, afetando o compromisso entre sensibilidade à detecção de falhas e a taxa de falsos alarmes. O *threshold* pode ser definido de forma empírica, com base no conhecimento do processo, ou de maneira automática, a partir de critérios estatísticos aplicados à distribuição dos erros no conjunto de dados de treinamento, como percentis, médias acrescidas de múltiplos do desvio padrão ou técnicas de otimização orientadas por métricas de desempenho.

Capítulo 4

Arquitetura *Edge Analytics*

4.1 Introdução e Objetivo

O Laboratório de Mecânica de Estruturas “José Eduardo Tannús Reis” (LMEst) da Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia (FEMEC-UFU) tem feito contribuições significativas em pesquisas realizadas na área de dinâmica de máquinas rotativas, incluindo o desenvolvimento de metodologias baseadas em IA para monitoramento e diagnóstico de falhas em máquinas rotativas ([GONÇALVES, 2024](#)).

Neste contexto, a detecção de anomalias em sinais de vibração de máquinas rotativas utilizando métodos de IA é parte integrante do *software* desenvolvido no LMEst para o projeto de pesquisa, desenvolvimento e inovação intitulado **Aplicação de Técnica de Inteligência Artificial na Captura de Dados de Máquinas para Armazenamento de Dados de Anormalidades em Servidores (Edge Analytics)**.

O *software Edge Analytics* propõe uma arquitetura multicamadas para a detecção de anomalias em máquinas rotativas, combinando métodos tradicionais de análise de sinais com modelos de aprendizado profundo, de forma a constituir um sistema robusto e adaptativo para aplicação em tempo real. A estrutura proposta busca equilibrar eficiência computacional e capacidade de generalização, características fundamentais para aplicações em ambientes industriais ([BORGES et al., 2025](#)).

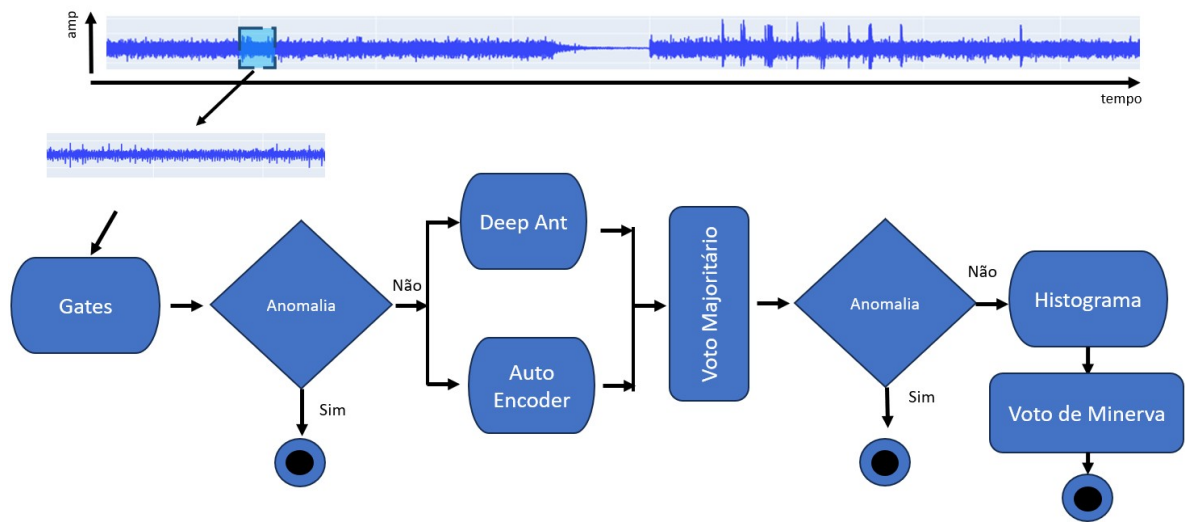


Figura 4.1: Ilustração da Arquitetura do *Edge Analytics*. Adaptado de (BORGES et al., 2025).

A arquitetura é composta por um conjunto de etapas sequenciais, conforme ilustrado na figura 4.1. A primeira camada, denominada *Gates*, utiliza atributos simples extraídos nos domínios do tempo e da frequência, como valores eficazes (RMS) e amplitudes pico a pico, com o objetivo de realizar uma avaliação rápida sobre possíveis desvios do comportamento esperado dos sinais de vibração. Essa abordagem permite uma filtragem inicial eficiente, reduzindo o volume de dados encaminhados para análises mais complexas.

As camadas subsequentes incorporam modelos de aprendizado profundo, incluindo um modelo do tipo *DeepAnT* e um *Autoencoder*, responsáveis por uma análise mais aprofundada dos sinais. Esses modelos são empregados para capturar padrões não lineares e mudanças sutis no comportamento dinâmico do sistema. Em situações nas quais ocorre divergência entre os modelos quanto à identificação de uma anomalia, uma camada estatística adicional, baseada na análise de histogramas, é utilizada para a tomada de decisão final.

A estrutura do *Edge Analytics* foi desenvolvida com o objetivo de atender às exigências de ambientes industriais, possibilitando o processamento local dos dados e a detecção de anomalias de forma descentralizada. A implementação adotada considera etapa única de treinamento dos modelos no período inicial de operação.

A principal contribuição do *framework* proposto reside em sua capacidade de detectar anomalias de forma confiável, oferecendo uma solução abrangente para a prevenção de falhas e minimização de custos associados à manutenção corretiva.

O desenvolvimento da metodologia de detecção de anomalias envolveu a obtenção e a exploração de diferentes conjuntos de dados, denominados *benchmarks*, dos quais cinco são

contemplados no presente trabalho. Ressalta-se que, em atendimento às exigências do projeto para a integração da metodologia ao *software* aplicado na indústria, o processo de detecção de anomalias foi conduzido de forma dinâmica, simulando o monitoramento de uma máquina em tempo real, com atualizações contínuas dos sinais de vibração por meio da utilização de janelas deslizantes.

Borges et al. (2025) apresenta resultados do *Edge Analytics* aplicado em três *benchmarks* distintos obtidos no LMEst, obtendo acurácias de até 91%, porém identificando oportunidades de melhorias na arquitetura. O presente trabalho contribui ao avaliar o desempenho do *Edge Analytics* em dois conjuntos de dados adicionais, além de promover uma otimização dos hiperparâmetros dos modelos de forma individualizada para cada um dos *benchmarks*.

4.2 Camada *Gates*

A camada *Gates* constitui o primeiro nível de inteligência do *framework* do *Edge Analytics* e tem como principal objetivo realizar uma triagem inicial dos sinais de vibração adquiridos. Essa camada atua como um filtro rápido e de baixo custo computacional, capaz de identificar desvios significativos em relação ao comportamento esperado do sistema antes que análises mais complexas sejam acionadas. Tal abordagem é especialmente relevante em aplicações industriais, nas quais restrições de tempo de processamento e recursos computacionais são fatores críticos.

O funcionamento da camada *Gates* baseia-se na extração e análise de parâmetros fundamentais nos domínios do tempo e da frequência, amplamente utilizados no monitoramento da integridade de máquinas rotativas. Dentre os atributos empregados destacam-se: (i) a amplitude pico a pico, dada por

$$x_{p2p} = \max_n(x[n]) - \min_n(x[n]), \quad (4.1)$$

e (ii) o valor eficaz (*Root Mean Square* – RMS),

$$x_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{n=1}^N x^2[n]}, \quad (4.2)$$

onde $x[n]$ representa o sinal de vibração na janela analisada e N é o número de amostras na janela. Esses descritores fornecem uma caracterização global do nível de energia e da amplitude do sinal, sendo amplamente utilizados para indicar alterações abruptas no regime de operação.

Adicionalmente, quando habilitado, o parâmetro de rotação (RPM) é estimado a partir do sinal e de informações nominais do sistema, permitindo a detecção de variações significativas na velocidade de operação.

Para cada janela deslizante do sinal de vibração, a camada *Gates* constrói um vetor de parâmetros que reúne os descritores extraídos naquela janela, definido por

$$\mathbf{p}_t = \begin{bmatrix} x_{p2p} \\ x_{RMS} \\ RPM \end{bmatrix}_t, \quad (4.3)$$

onde o índice t indica a janela temporal corrente e cada componente representa um descritor calculado a partir do sinal de vibração correspondente àquela janela.

O vetor de referência, definido inicialmente na primeira janela analisada, é dado por

$$\mathbf{p}_{base} = \begin{bmatrix} x_{p2p} \\ x_{RMS} \\ RPM \end{bmatrix}_{base}. \quad (4.4)$$

Esse vetor representa o estado de referência do sistema, associado à condição normal inicial de operação, contra o qual as janelas subsequentes são comparadas ao longo do processo de monitoramento.

A decisão de anomalia é tomada por comparação entre o vetor de parâmetros extraídos na janela corrente, $\mathbf{p}_t$, e um vetor de referência $\mathbf{p}_{base}$. Calcula-se a variação relativa (normalizada) de cada parâmetro por

$$\delta_t = \left(\frac{\mathbf{p}_t - \mathbf{p}_{base}}{\mathbf{p}_{base}} \right), \quad (4.5)$$

e classifica-se a janela como anômala caso qualquer componente exceda o limiar de sensibilidade τ (hiperparâmetro percentual),

$$\exists k \text{ tal que } \delta_t^{(k)} > \tau. \quad (4.6)$$

O valor de sensibilidade foi considerado como 0.5 para todos os *benchmarks*, mantendo o valor *default* que vinha sendo considerado no *Edge Analytics*.

A principal vantagem da camada *Gates* reside em sua eficiência computacional e em sua

capacidade de identificar anomalias evidentes de forma quase imediata. Ao filtrar sinais claramente discrepantes logo na primeira etapa, reduz-se significativamente o volume de dados encaminhados às camadas subsequentes, que empregam modelos de aprendizado profundo mais complexos. Dessa forma, a camada *Gates* desempenha um papel fundamental na arquitetura proposta, contribuindo para a robustez, escalabilidade e viabilidade do framework de *Edge Analytics* em aplicações de monitoramento em tempo real.

A Figura 4.2 apresenta um exemplo de funcionamento da camada *Gates*.

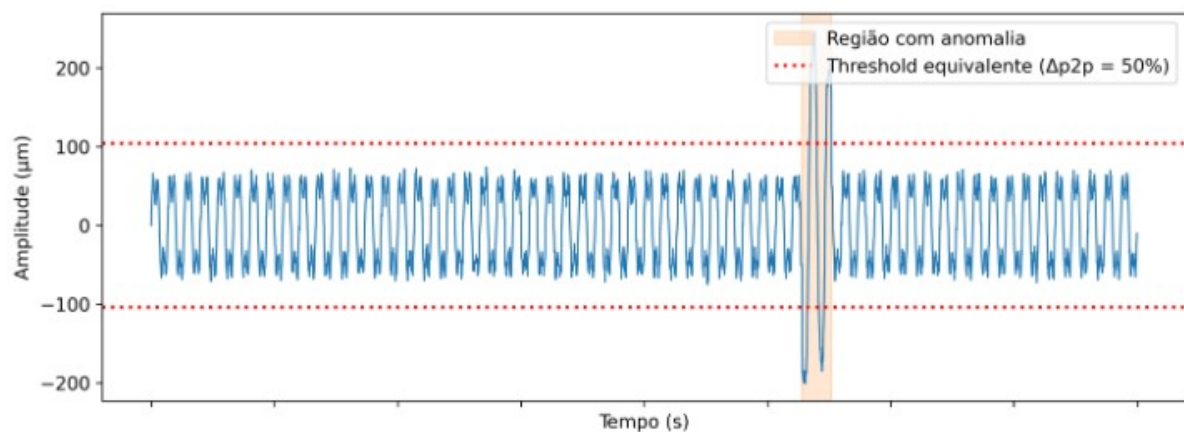


Figura 4.2: Exemplo ilustrativo da camada *Gates*.

4.3 *Autoencoder*

A camada *Autoencoder* se constitui num nível mais sofisticado do *framework Edge Analytics*, sendo utilizada para avaliar sinais de vibração que demandam uma análise mais aprofundada. Sua premissa segue o princípio clássico de detecção de anomalias por reconstrução, onde um *Autoencoder* treinado predominantemente com dados representativos da condição normal tende a reconstruir adequadamente padrões compatíveis com o comportamento saudável, enquanto sinais com desvios significativos da distribuição aprendida apresentam maior erro de reconstrução. Assim, a decisão sobre classificação como anomalia é realizada a partir do erro entre o sinal de entrada e a sua reconstrução (AHMAD et al., 2021).

Na implementação adotada, os sinais adquiridos são pré-processados para adequação ao treinamento e à inferência. Cada janela de sinal é segmentada em subblocos (*sub-packs*) de tamanho fixo, com alta sobreposição durante o treinamento para aumentar a variabilidade dos exemplos, e menor sobreposição durante a fase de operação para reduzir o custo computacional. Em seguida, cada subbloco é normalizado para o intervalo $[0, 1]$ com base nos valores mínimo

e máximo observados, garantindo consistência de escala entre treino e sua aplicação.

A arquitetura do modelo é baseada em camadas convolucionais 1D com dilatações temporais, buscando capturar dependências locais e padrões temporais relevantes presentes em sinais de vibração. No codificador, o sinal é processado por convoluções causais com taxas de dilatação crescentes e, em seguida, é comprimido por camadas com *stride*, produzindo uma representação latente de menor dimensionalidade. O decodificador realiza o processo inverso, empregando camadas *Conv1DTranspose* para reamostragem e convoluções adicionais para reconstruir o sinal original a partir do espaço latente, sendo o treinamento conduzido de forma não supervisionada com função de perda do tipo erro quadrático médio (MSE) e otimizador Adam, com *Early Stopping* baseado na perda de validação.

Após o treinamento, o modelo opera reconstruindo os subblocos de cada janela e calculando o erro de reconstrução. Esse erro é quantificado pela média do erro absoluto (MAE) entre o subbloco de entrada e sua reconstrução, acumulando-se os erros ao longo dos subblocos para produzir um indicador agregado. A partir disso, define-se um índice de variação percentual do erro agregado em relação a um estado de referência (*baseline*), permitindo transformar o erro de reconstrução em uma métrica comparável ao longo do tempo e adequada ao monitoramento dinâmico por janelas deslizantes.

A decisão final de anomalia é feita por comparação desse índice com um limiar τ (*threshold*). Quando o limiar não é previamente fornecido, é realizada uma etapa de ajuste automático a partir de um *buffer* de *scores* calculados nas primeiras janelas em operação. O limiar é então estimado pela regra:

$$\tau = \mu + \kappa \sigma, \quad (4.7)$$

em que μ e σ representam, respectivamente, a média e o desvio padrão dos *scores* armazenados, e κ é um hiperparâmetro de sensibilidade. Esse procedimento está alinhado com abordagens baseadas em erro de reconstrução e limiarização para detecção de anomalias em máquinas rotativas.

A Figura 4.3 apresenta um esquema da camada *Autoencoder*.

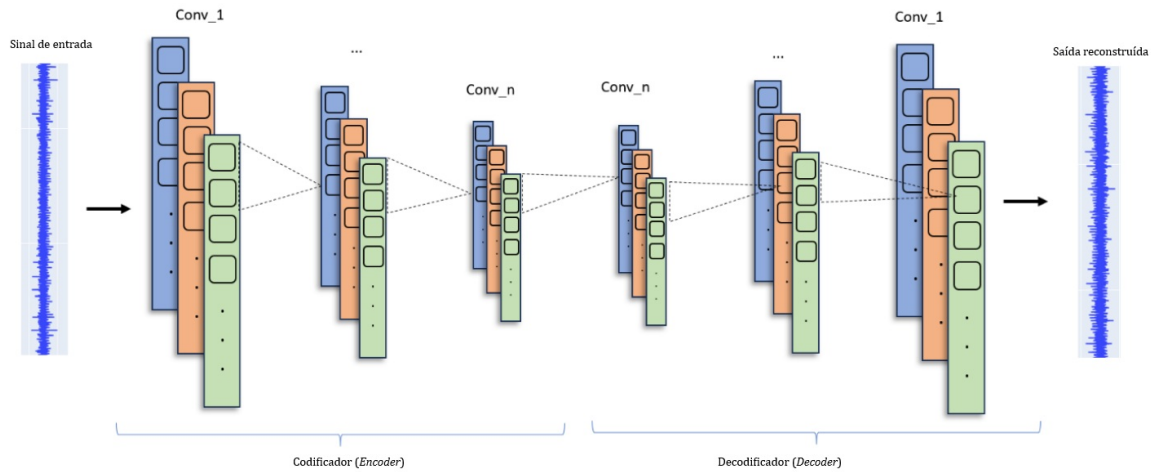


Figura 4.3: Exemplo ilustrativo da camada *Autoencoder*. Adaptado de (BORGES et al., 2025).

4.4 DeepAnT

O modelo *DeepAnT* é responsável por detectar anomalias a partir de um paradigma preditivo, no qual o comportamento normal do sinal é modelado por meio da previsão de amostras futuras e a anomalia é inferida quando o erro de previsão excede um limiar. Essa abordagem, proposta por (MUNIR et al., 2019), organiza o método em dois módulos: (i) um preditor de séries temporais baseado em redes convolucionais profundas e (ii) um detector que transforma o desvio entre o valor previsto e o valor observado em um *score* de anomalia. Por não depender de rótulos de falha, o método se adequa a cenários industriais nos quais a rotulagem de eventos anômalos é escassa ou inviável, utilizando os dados históricos para capturar a dinâmica predominante do regime de operação (MUNIR et al., 2019).

Na implementação adotada no *Edge Analytics*, esta camada recebe janelas do sinal de vibração e mantém um *buffer* com três janelas consecutivas, de modo a prover contexto suficiente para a etapa de previsão. O pré-processamento é realizado no domínio da frequência: a janela temporal é transformada via FFT, e apenas metade do espectro é utilizada na composição do vetor de entrada. Esse procedimento reduz a dimensionalidade e enfatiza componentes espectrais relevantes para máquinas rotativas, mantendo consistência com o fluxo de dados empregado em tempo real. O módulo preditor segue a arquitetura descrita por (MUNIR et al., 2019), baseada em uma DCNN (*Deep Convolutional Neural Network*, isto é, Rede Neural Convolucional Profunda) com duas camadas convolucionais 1D, cada uma seguida por *max-pooling*, além de uma camada densa para regressão. No código, são utilizados 32 filtros em cada camada convolucional, com *kernel* de tamanho 2 e ativação ReLU; após as operações de convolução e *pooling*, os mapas

de características são achatados (*Flatten*) e processados por uma camada totalmente conectada com 40 neurônios (ReLU), seguida de *dropout* para reduzir sobreajuste, e uma camada de saída com dimensão compatível com a janela de previsão. O treinamento é realizado com função de perda do tipo MAE e otimizador Adam.

Após o treinamento inicial, a operação consiste em prever a janela mais recente e comparar a previsão com a janela observada. O erro entre as janelas é quantificado por MAE e armazenado em um *buffer*, cujo tamanho é definido por um hiperparâmetro de vizinhança (*anomaly_neighborhood*). Para reduzir flutuações de curto prazo, a implementação aplica ainda uma média móvel sobre os valores de MAE (com janela *moving_avg_size*). Em seguida, é calculado um *score* de anomalia que mede a discrepância entre o MAE central da vizinhança e a média dos MAEs na própria vizinhança. Dessa forma, o método não se limita ao erro instantâneo, mas incorpora contexto local e suavização temporal, tornando a decisão mais estável em regime de monitoramento contínuo.

A decisão binária (normal/anômalo) é obtida por limiarização do *score*. Durante uma fase de calibração (*threshold_calibration_window*), os *scores* são acumulados para formar uma distribuição de referência. Após essa etapa, o limiar passa a ser calculado de maneira adaptativa utilizando a mesma expressão empregada na camada *Autoencoder* (Eq. (4.7)), em que μ e σ representam, respectivamente, a média e o desvio padrão dos *scores* recentes, e κ é um hiperparâmetro de sensibilidade. Adicionalmente, a implementação mantém um vetor de limiares suavizado (*threshold_smooth*) para evitar variações abruptas na fronteira de decisão.

Por fim, a janela é sinalizada como anômala quando o *score* corrente supera o limiar calculado, alinhando-se ao princípio do *DeepAnT* de que desvios preditivos persistentes refletem incompatibilidade com o padrão normal aprendido pelo modelo.

A Figura 4.4 apresenta um esquema do modelo *DeepAnT*.

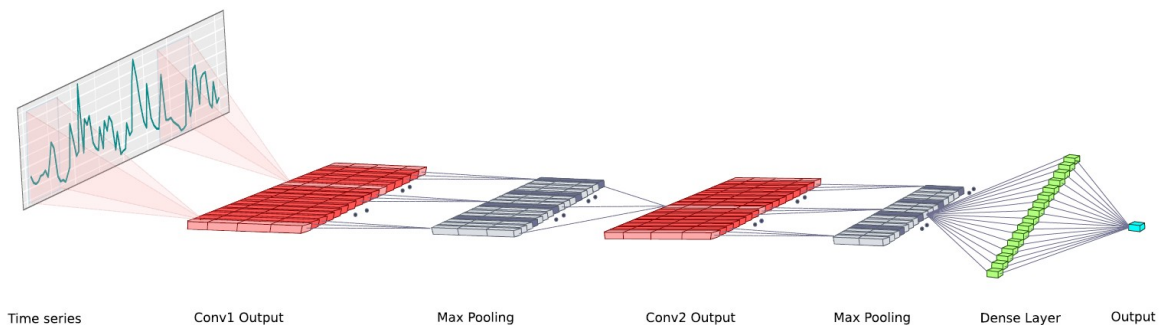


Figura 4.4: Exemplo ilustrativo do modelo *DeepAnT*. Adaptado de (MUNIR et al., 2019).

4.5 Histograma

A camada de histograma é empregada como um mecanismo estatístico de decisão final da arquitetura para situações nas quais os modelos baseados em aprendizado profundo apresentam decisões conflitantes. Nessa etapa, o objetivo é avaliar se o sinal corrente permanece consistente com o comportamento normal previamente observado, comparando distribuições estimadas sob condição de referência com as distribuições do sinal de interesse. Assim, a decisão não depende diretamente do *score* de um modelo específico, mas de uma medida estatística de compatibilidade entre distribuições.

Na implementação, é inicialmente definido um conjunto de referência a partir das amostras de treinamento associadas à condição normal de operação. Com base nesse conjunto, são construídos histogramas de referência que servem como padrão estatístico de comparação. Para cada novo sinal analisado, os histogramas são gerados utilizando o mesmo número de classes e os mesmos limites da distribuição de referência, garantindo a comparabilidade direta entre os dados. A avaliação da discrepância entre as distribuições é realizada por um teste não paramétrico do tipo Kolmogorov–Smirnov, aplicado sobre os histogramas normalizados, produzindo um *p-value* como estatística de decisão. O menor *p-value* entre as representações analisadas é adotado como medida conservadora, e a amostra é classificada como anômala quando esse *p-value* é inferior ao limite definido (*p_value_limit*). Em regime de operação, esse limite pode ser mantido fixo (quando especificado) ou ajustado automaticamente após uma etapa de calibração inicial, na qual o *p_value_limit* é estimado a partir da estatísticas de primeira e segunda ordem (média e desvio padrão) dos *p-values* observados, sendo diretamente controlado por um parâmetro de sensibilidade.

Do ponto de vista conceitual, essa camada está alinhada ao princípio discutido por (CANDY et al., 2017), de que em sistemas vibratórios a condição normal pode ser caracterizada por assinaturas dinâmicas e por suas distribuições estatísticas associadas. Assim, mudanças relevantes nessas distribuições em relação ao comportamento de referência podem ser interpretadas como indícios de anomalia.

4.6 Arquitetura

Conforme figura 4.1, o processo de detecção de anomalias inicia-se com a aquisição de sinais de vibração provenientes de máquinas rotativas. Em seguida, os primeiros n sinais,

sendo n um hiperparâmetro definido previamente, são utilizados na fase de inicialização do sistema, com o objetivo de treinar os modelos de aprendizado profundo e estabelecer valores de referência empregados na avaliação subsequente de anomalias. Portanto, a arquitetura considera a premissa de que as amostras iniciais representam o regime normal de operação da máquina, sendo utilizadas para calibrar os modelos e definir os limiares de decisão.

Após a etapa de inicialização, cada novo sinal recebido é submetido inicialmente à camada *Gates*, responsável por uma análise rápida baseada em descritores fundamentais nos domínios do tempo e da frequência. Nessa camada, características simples são extraídas e comparadas com limiares pré-definidos. Caso seja identificada uma variação significativa em relação ao comportamento esperado, o sinal é imediatamente classificado como anômalo, dispensando o encaminhamento para as etapas seguintes e reduzindo o custo computacional global do processo.

Quando não há indicação de anomalia na camada *Gates*, o sinal avança para a camada de aprendizado profundo, na qual dois modelos (*DeepAnT* e *Autoencoder*) realizam, de forma independente, a avaliação de anomalia. Se ambos os modelos concordarem quanto à presença de anomalia, o sinal é classificado diretamente como anômalo. Em caso de divergência entre os modelos, o sinal é encaminhado para a camada estatística final baseada em histogramas, a qual compara características estatísticas do sinal com distribuições históricas associadas ao regime normal, resolvendo a ambiguidade e sendo definida a decisão final.

De forma geral, a arquitetura do *Edge Analytics* busca equilibrar rapidez e acurácia: nas etapas iniciais, prioriza-se eficiência por meio de critérios simples e de baixa complexidade, enquanto sinais mais difíceis de classificar são analisados por modelos mais expressivos e por um verificador estatístico final. Essa estratégia permite detectar anomalias com maior robustez sem sobrecarregar recursos computacionais, aspecto essencial para aplicações em monitoramento contínuo e em ambientes industriais (BORGES et al., 2025).

Capítulo 5

Dinâmica de Rotores e *Benchmarks*

5.1 Dinâmica de rotores

O desenvolvimento e aplicação de máquinas rotativas de menor peso e em aplicações de elevadas velocidades de rotação está correlacionado com a modelagem mais precisa do comportamento dinâmico de rotores. Tomando como exemplo compressores centrífugos, tratam-se de equipamentos dinâmicos projetados especificamente por aplicação, exigindo que o projeto leve em consideração um compromisso entre aspectos aerodinâmicos e rotodinâmicos ([LUTKE, 2004](#)). Além disso, os limites de valores de vibração de rotores impostos por normas adotadas na indústria como [American Petroleum Institute \(API 617\) \(2022\)](#) implicam na necessidade de precisão de modelagem rotodinâmica, que deve prever adequadamente a resposta de vibração dos rotores.

A modelagem de rotores relaciona os esforços aos quais o rotor é submetido à sua resposta em vibração. Assim, a análise da resposta de vibração de rotores fornece informações sobre a condição do equipamento, permitindo a identificação de uma condição anômala na operação em comparação com a condição considerada normal.

Este capítulo apresenta os principais conceitos da modelagem da dinâmica de rotores e da análise de defeitos em máquinas rotativas, sendo apresentadas informações válidas para equipamentos dinâmicos de diferentes tipos.

Convém ressaltar que os modelos de ML abordados nesse trabalho não utilizam o modelo matemático de rotores. No entanto, um modelo numérico pode ser utilizado para geração de dados de validação dos modelos, o que de fato foi realizado no *benchmark* 1, conforme descrito posteriormente.

5.1.1 Modelagem

De maneira geral e conforme [Ferraris e Lalanne \(1998\)](#), o comportamento dinâmico de um rotor pode ser descrito através de equações obtidas a partir das energias associadas (cinéticas e de deformação) e do trabalho virtual das forças externas para cada componente do sistema. A associação desta abordagem com o Método de Elementos Finitos (MEF) permite a determinação das equações de movimento do rotor.

Um modelo matemático que descreve o comportamento dinâmico de uma máquina rotativa leva em conta a geometria e materiais de componentes como eixo, impelidores e acoplamentos, além de condições de operação e dependência da velocidade na determinação dos efeitos de rigidez e amortecimento impostos por mancais e selos. Também considera-se o efeito giroscópico para determinação da resposta em vibração do rotor.

A técnica do MEF permite discretizar os elementos do rotor de forma que, para cada elemento discretizado, são obtidas suas matrizes elementares de massa, amortecimento e rigidez. A partir da conexão dos nós de cada elemento, essas matrizes elementares são concatenadas (respeitando sua ordem na discretização) e então são obtidas as matrizes globais de massa, giroscópica, de rigidez e de amortecimento.

[Cavallini Jr. \(2013\)](#) desenvolve para os componentes eixo, disco e mancais de uma máquina rotativa as matrizes elementares utilizadas para compor as matrizes globais da equação do movimento do rotor, baseado no MEF.

A partir da equação de Lagrange e da aplicação do MEF, é possível obter a equação diferencial que descreve numericamente o comportamento dinâmico de um rotor flexível suportado por mancais hidrodinâmicos ou de rolamentos, conforme apresentado na Eq. 5.1.

$$\mathbf{M} \ddot{\mathbf{q}} + [\mathbf{D} + \Omega \mathbf{D}_g] \dot{\mathbf{q}} + [\mathbf{K} + \dot{\Omega} \mathbf{K}_{sv}] \mathbf{q} = \mathbf{W} + \mathbf{F}_u + \mathbf{F}_m + \mathbf{F}_d \quad (5.1)$$

A matriz $\mathbf{M}$ representa a massa, enquanto $\mathbf{D}$ corresponde ao amortecimento e $\mathbf{D}_g$ descreve o efeito giroscópico. A matriz $\mathbf{K}$ caracteriza a rigidez e $\mathbf{K}_{sv}$ expressa o enrijecimento adicional provocado pela variação da velocidade de rotação. Todas essas matrizes estão associadas aos componentes girantes da máquina, como eixo, discos e acoplamentos. O vetor de deslocamentos é indicado por $\mathbf{q}$, e a velocidade angular de operação é representada por Ω . A força peso, $\mathbf{W}$, atua apenas sobre as partes rotativas; a força $\mathbf{F}_u$ modela o desbalanceamento; $\mathbf{F}_m$ reúne as forças não lineares provenientes dos mancais que sustentam o eixo; e $\mathbf{F}_d$ representa a força de excitação

externa, caso tal excitação seja imposta ao sistema.

Tais matrizes globais são utilizadas na solução das equações do movimento do rotor, de modo que por meio da solução de autovalor e de autovetor encontram-se as frequências naturais da máquina rotativa, acompanhada de seus modos de vibrar. Além disso, utilizando-se da integração numérica é possível determinar a resposta do deslocamento do rotor em condições de desbalanceamento, que constitui uma análise essencial no projeto de máquinas.

A modelagem e análise dinâmica de sistemas rotativos pode ser realizada no ROSS (*Rotordynamic Open-Source Software*), uma biblioteca de código aberto desenvolvida no python (Timbó et al., 2020) e que é parte de uma iniciativa de pesquisa da Petrobras em colaboração com a Universidade Federal de Uberlândia (UFU) e a Universidade Federal do Rio de Janeiro (UFRJ).

O ROSS está disponível publicamente no repositório do GitHub¹, colaborando para oportunidades de integração da comunidade científica e fortalecendo a integração entre indústria e academia (Timbó et al., 2020).

A biblioteca emprega o método dos elementos finitos (MEF) para a modelagem dos sistemas, representando componentes como eixos, mancais, selos e impelidores. Adicionalmente, o ROSS realiza simulações avançadas, como análise modal, resposta em frequência, diagrama de Campbell, cálculo de velocidades críticas, resposta temporal devido a forças assíncronas ou harmônicas (EMRICH, 2026). O ROSS constitui-se como uma solução acessível e robusta, aplicável tanto em contextos acadêmicos quanto industriais com destaque para análises de rotores de grandes máquinas como turbinas, compressores e geradores (Timbó et al., 2020).

5.1.2 Defeitos em Máquinas Rotativas

De acordo com Lei (2017), o processo de diagnóstico pode ser entendido como a tarefa de identificar e estabelecer a relação entre as informações obtidas no espaço de medição e os padrões de falha existentes no espaço de falhas. O diagnóstico envolve três etapas principais: detecção, isolamento e identificação da falha.

A detecção de falhas consiste em indicar se alguma anomalia já ocorreu na máquina monitorada. O isolamento de falhas busca determinar qual componente falhou e onde a falha se encontra. Por fim, a identificação de falhas procura caracterizar o tipo e a severidade da falha identificada.

¹ <<https://github.com/ross-rotordynamics/ross>>

As três etapas são interdependentes e formam um fluxo contínuo de análise, sendo que apenas a primeira, a detecção de falhas, está dentro do escopo do presente trabalho.

Máquinas rotativas são projetadas para que a resposta de vibração fique dentro de níveis permitidos por normas de projeto, como as publicações da API, e/ou critérios do fabricante, levando-se em consideração valores máximos de desbalanceamento admissível durante projeto da máquina.

Assim, rotores operando em condições regulares exibem uma resposta de vibração considerada como normal. Diferentes fatores podem implicar na alteração deste comportamento, como variação de condições operacionais, montagem incorreta, impactos, desgaste, admissão de corpos estranhos e outros. Nestes cenários a resposta de vibração do rotor irá apresentar um comportamento distinto, com possível aumento de amplitudes e alteração do espectro de frequências, permitindo a identificação de anomalias.

[Bently et al. \(2002\)](#) apresentam como exemplos de falhas que implicam no aumento da amplitude de vibração de rotores: desbalanceamento, desalinhamento, instabilidade fluido induzida, folgas mecânicas excessivas, roçamento, trincas em eixo.

A resposta de vibração pode ser avaliada a partir de diferentes técnicas e gráficos, que fornecem uma variada gama de informações para a análise de comportamento do rotor ([BENTLY et al., 2002](#)). A etapa de processamento de sinais trata da extração de informações a serem utilizadas na análise de condição da máquina a partir dos sinais originalmente medidos. De maneira geral, as técnicas e algoritmos de processamento de sinais podem ser divididos em três categorias de análise: domínio do tempo, domínio da frequência e domínio do tempo-frequência ([LEI, 2017](#)).

Os sinais originais de medição, geralmente amostrados repetidamente em intervalos de tempo predefinidos, são obtidos no domínio do tempo. Assim, a análise no domínio do tempo é feita diretamente sobre esses sinais brutos. As abordagens tradicionais nesse domínio calculam características estatísticas capazes de descrever o estado de saúde das máquinas, tais como: média, valor de pico, valor eficaz (RMS), curtose e assimetria (*skewness*). Esses indicadores estatísticos são usualmente denominados características no domínio do tempo.

A análise no domínio da frequência, por sua vez, é realizada a partir da transformação dos sinais para esse domínio. Sua principal vantagem em relação à análise no domínio do tempo é a capacidade de decompor o sinal original em seus componentes harmônicos. A técnica mais amplamente utilizada é a análise espectral por meio da transformada rápida de Fourier (FFT). O

objetivo central da análise de espectro é isolar e identificar componentes de frequência associadas às características de falha das máquinas. Entre os métodos baseados em FFT, o espectro de potência é um dos mais empregados. Adicionalmente, ferramentas auxiliares como filtros de frequência, análise de envelope e análise de bandas laterais são frequentemente utilizadas para aprofundar a interpretação dos sinais.

Uma limitação importante da análise no domínio da frequência é que ela assume que as propriedades estatísticas do sinal permanecem aproximadamente constantes ao longo do intervalo analisado, o que restringe sua aplicação a sinais estacionários ou quase-estacionários. Entretanto, sinais de máquinas rotativas frequentemente exibem comportamento não estacionário. Para lidar com essa limitação, desenvolveu-se a análise no domínio tempo-frequência, que permite avaliar simultaneamente a evolução temporal e o conteúdo espectral dos sinais. Essa abordagem gera representações bidimensionais capazes de evidenciar padrões de falha com maior clareza.

Entre as ferramentas mais utilizadas na análise tempo-frequência está a transformada de Fourier de tempo curto (STFT), que segmenta o sinal em janelas de curta duração e aplica a transformada de Fourier a cada segmento. Características típicas extraídas nesse domínio — como entropia — são obtidas a partir de transformadas *wavelet*, *wavelet packet* e decomposições empíricas de modo. Essas características têm se mostrado eficazes para representar o estado de saúde de máquinas operando em condições não estacionárias (BRITO, 2022).

Modelos de ML desenvolvidos para detecção de anomalias podem trabalhar com sinais brutos de vibração ou com atributos extraídos dos sinais nos diferentes domínios descritos previamente. Para o segundo caso, se faz necessária a etapa de extração de atributos, onde devem ser escolhidos parâmetros que reflitam a condição da máquina para que sejam calculados a partir do sinal original. De acordo com Lei (2017), os atributos podem conter informações pouco úteis e insensíveis à saúde do equipamento, além de afetar o resultado do diagnóstico e a eficiência computacional.

Conforme destacado por Brito (2022), um dos principais motivos para a ampla adoção de modelos de DL em diversas tarefas é que essas abordagens realizam, de forma implícita, o processo de extração de características. Isso ocorre porque suas arquiteturas são capazes de aprender automaticamente atributos discriminativos a partir das relações não lineares modeladas internamente, eliminando a necessidade, muitas vezes trabalhosa e custosa do ponto de vista computacional, de realizar a extração manual de características.

5.2 Benchmarks

A validação e avaliação de desempenho das metodologias presentes no *Edge Analytics* e desenvolvidas no presente trabalho se dá através da análise de métricas de desempenho da tarefa de classificação dos conjuntos de dados nas classes normal e anômala. Para tal, foram utilizados diferentes conjuntos de dados, denominados *benchmarks*, constituídos por dados de amplitude de vibração no tempo, com respectiva classificação entre normalidade e anomalia por amostra.

Nesse contexto, é essencial que os dados sejam rotulados corretamente para avaliação de desempenho dos modelos e esta tarefa foi realizada por especialistas do LMEst para diferentes conjuntos de dados. Estes profissionais classificaram as amostras manualmente, tomando como referência análise de parâmetros como amplitude no domínio da frequência em 1X, 2X, 3X, RMS, valor máximo do sinal no tempo e variação com base no vetor de frequência. Dessa forma, para cada conjunto de dados foram definidos os parâmetros mais relevantes para a detecção de anomalias e, assim, adotados limites para esta classificação.

Convém destacar que os modelos desenvolvidos buscam identificar mudanças no comportamento normal da resposta vibratória. Assim, além de falhas mecânicas, alterações de regime operacional, como variações de rotação, acelerações e desacelerações, também podem ser classificadas como anomalias quando resultam em desvios significativos em relação ao padrão de referência aprendido pelo modelo.

Esta seção irá descrever e apresentar os cinco *benchmarks* utilizados no desenvolvimento desta dissertação.

5.2.1 Benchmark 1

Este conjunto de dados foi obtido a partir de modelo numérico-computacional discretizado via elementos finitos. Para a aquisição de sinais foi considerada uma rotação de 3600 rpm, frequência base de 60 Hz, frequência de aquisição de 1000 Hz e tempo de 14400 s. A figura 5.1 exibe a resposta de vibração no tempo.

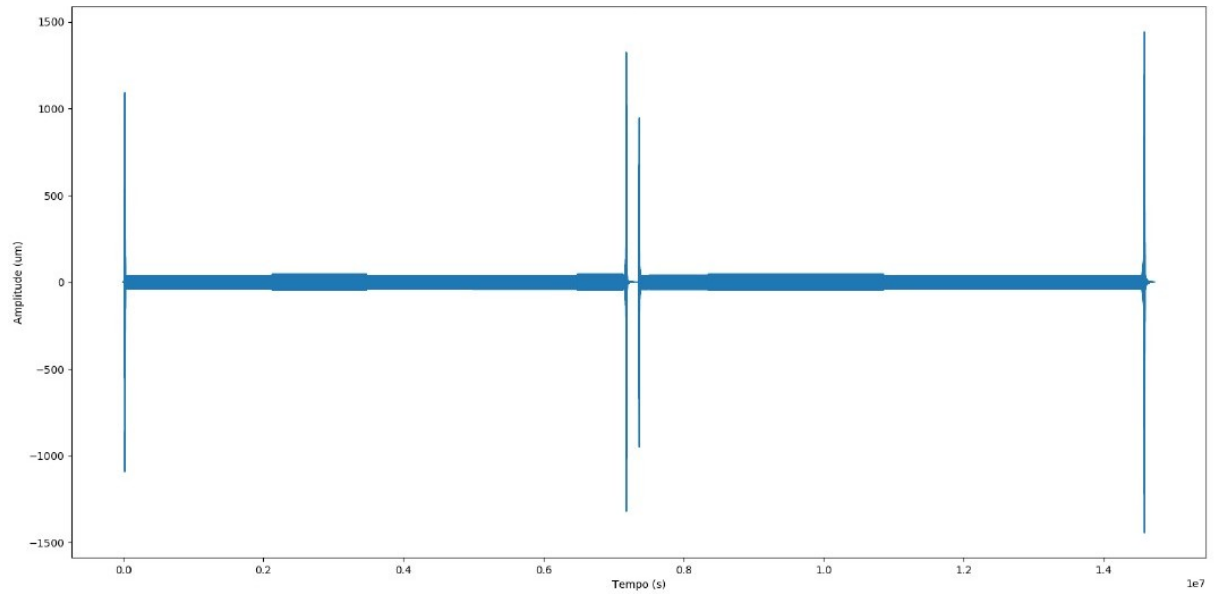


Figura 5.1: *Benchmark 1*: Amplitude de vibração no tempo.

Foram inseridas variações como desbalanceamento, mudança de velocidade, aceleração, desaceleração e desalinhamento para verificação do desempenho dos modelos de detecção de anomalias. A figura 5.2 apresenta o gabarito adotado, onde as anomalias inseridas são: A – Aceleração de 0 a 3600 rpm; B, C, E, H, J – Desbalanceamento; D, G, I, K – Variação de velocidade; F – Desaceleração de 3540 a 0 rpm e aceleração até 3720 rpm; L – Desaceleração de 3600 a 0 rpm.

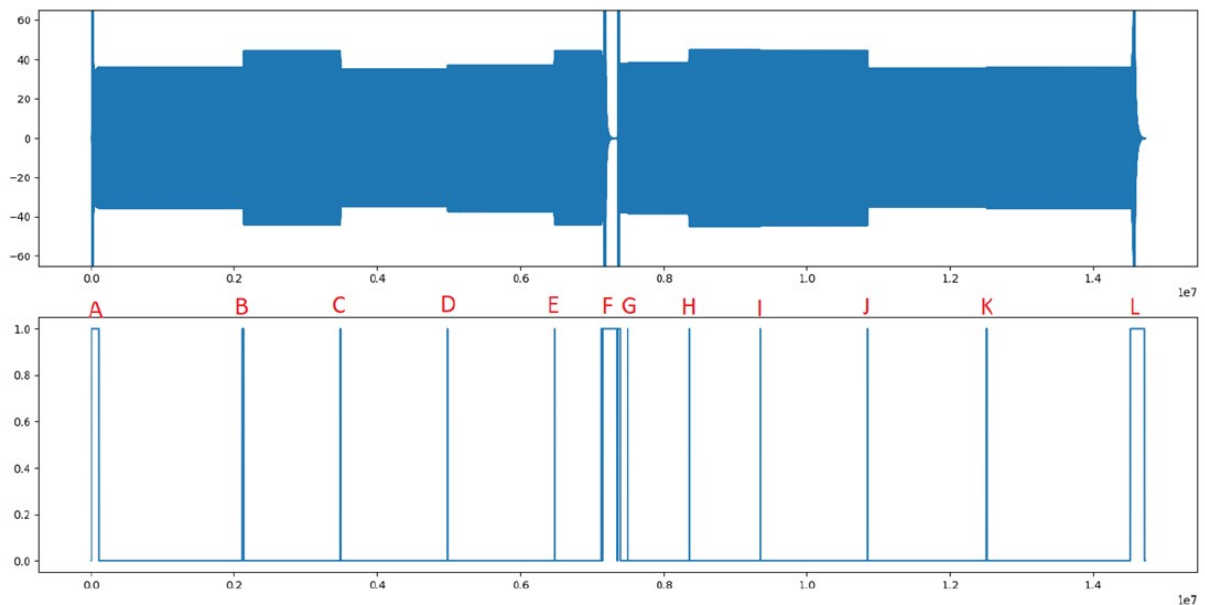


Figura 5.2: *Benchmark 1*: Anomalias.

5.2.2 Benchmark 2

Este conjunto de dados foi obtido a partir de uma bancada instalada no LMEst que consiste em um rotor apoiado em mancais magnéticos, representada na figura 5.3. Foram considerados 10 minutos para aquisição de dados, sendo 2 s para cada pacote de sinal, rotação de 4500 rpm e frequência de amostragem de 19638,6 Hz.

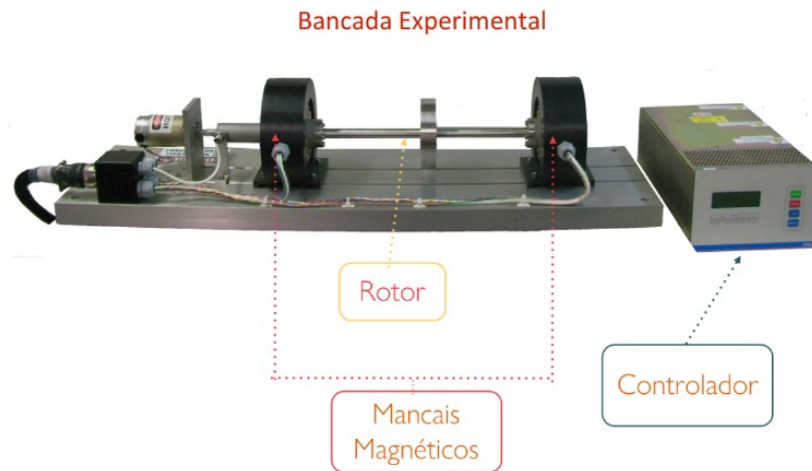


Figura 5.3: Bancada experimental de rotor com mancais magnéticos.

Para geração do *Benchmark 2* foram introduzidos efeitos de desbalanceamento e impactos no rotor, e a figura 5.4 apresenta o sinal e a respectiva classificação das amostras.

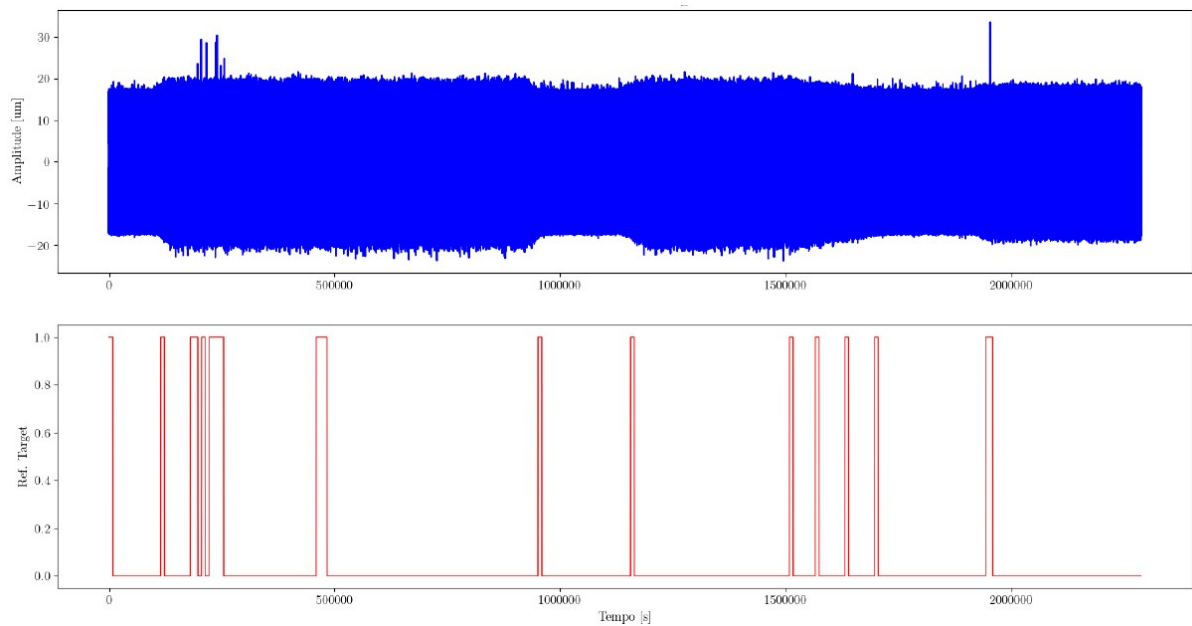


Figura 5.4: *Benchmark 2*: Sinal no tempo e classificação.

5.2.3 Benchmark 3

Este conjunto de dados foi obtido em um exaustor industrial apoiado em mancais de rolamentos e acionado por polia, conforme exibido na figura 5.5.

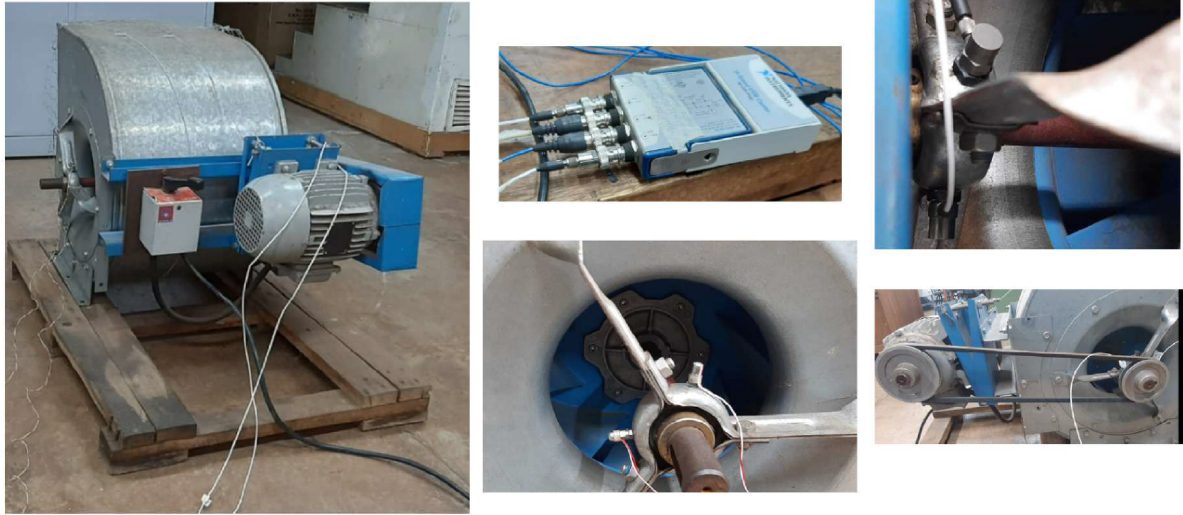


Figura 5.5: Exaustor industrial, sensores e placa de aquisição (GONÇALVES, 2024)

A aquisição de dados foi realizada com uma placa da National Instruments, com frequência de amostragem de 16384 Hz, rotação nominal de 2160 rpm e duração de 100 s. A condição normal de operação é representada nos primeiros 40 s, sendo introduzidas perturbações posteriores como desaceleração, aceleração e impactos (GONÇALVES, 2024). O valor RMS de vibração foi utilizado para classificação das amostras e a figura 5.6 apresenta o sinal e a respectiva classificação das amostras.

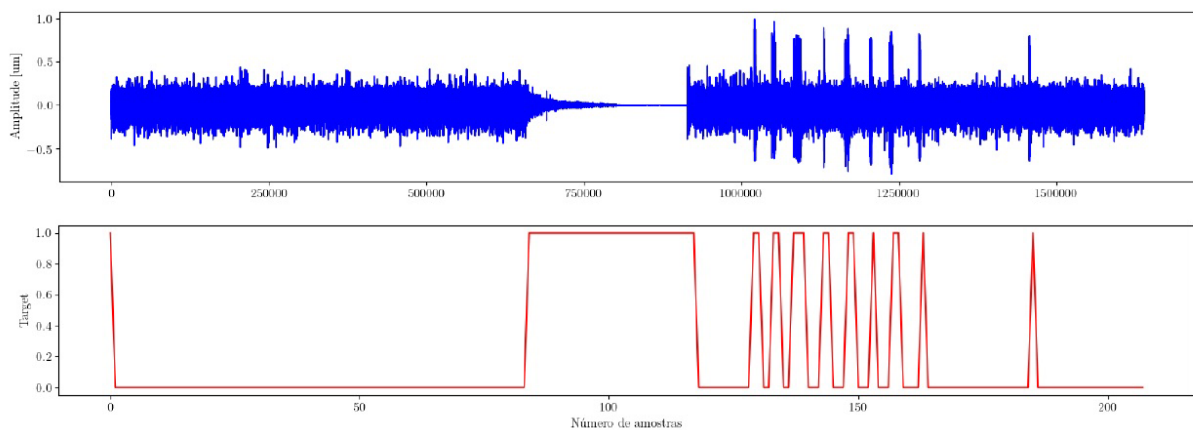


Figura 5.6: Benchmark 3: Sinal no tempo e classificação.

5.2.4 Benchmark 4

Neste *benchmark* também foi utilizada a bancada de rotor de mancais magnéticos, utilizando uma placa de aquisição fabricada no LMEst. A aquisição de dados teve duração de aproximadamente 180 segundos, rotação de 4500 rpm e frequência de amostragem de 10000 Hz.

Para geração do *Benchmark 4* foram introduzidos efeitos aceleração, desaceleração e impactos no rotor. A figura 5.7 apresenta o sinal e a respectiva classificação das amostras.

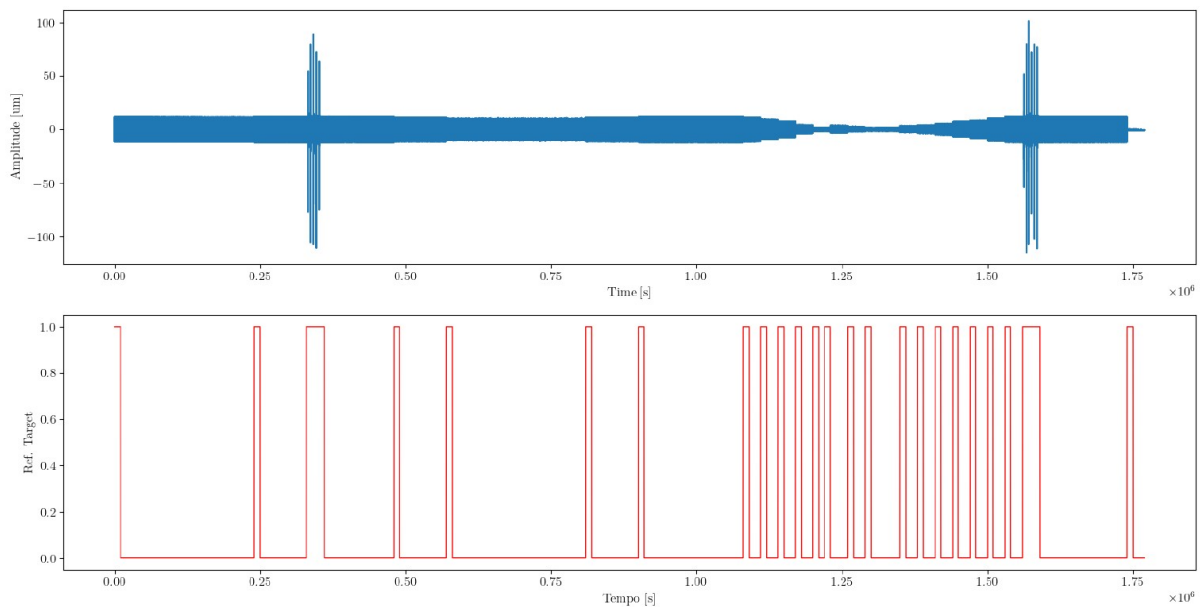


Figura 5.7: *Benchmark 4*: Sinal no tempo e classificação.

5.2.5 Benchmark 5

Este conjunto de dados é referente a um teste representativo da evolução de falhas em mancais de rolamentos, onde a bancada de teste consiste em um eixo apoiado em quatro mancais na velocidade de 2000 rpm com aplicação de carga radial, conforme exibido na figura 5.8. Os dados de vibração foram coletados a cada 10 minutos com taxa de aquisição de 20 kHz e 20480 pontos (HOU et al., 2022), sendo o sinal bruto obtido pelo IMS (*Intelligent Maintenance Systems Center*) da Universidade de Cincinnati e disponibilizado pela NASA.

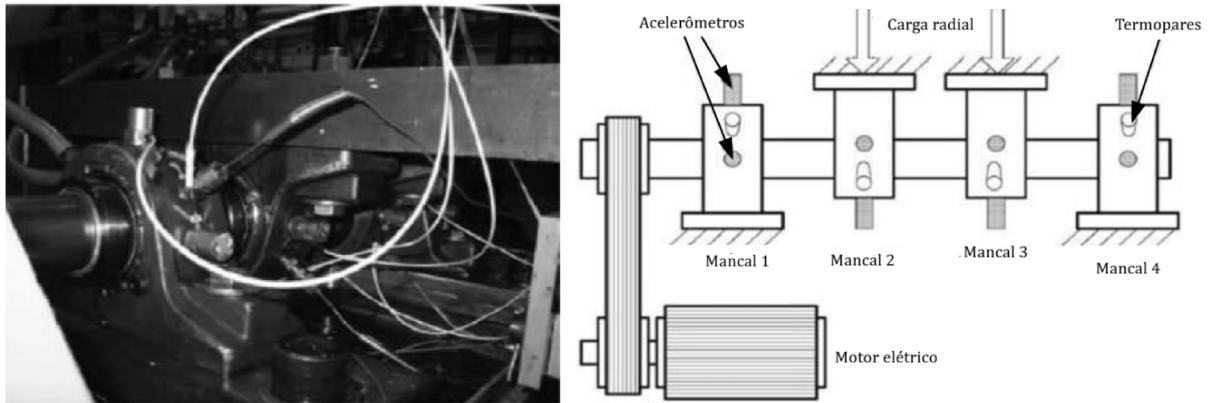


Figura 5.8: Bancada de teste de mancais de rolamento da NASA IMS. Adaptado de (HOU et al., 2022)

O IMS disponibiliza três conjuntos de dados para esse ensaio, tendo sido utilizado o *Dataset 2* neste trabalho. A classificação das amostras foi realizada pelo LMEst e a figura 5.9 apresenta o sinal e a respectiva classificação.

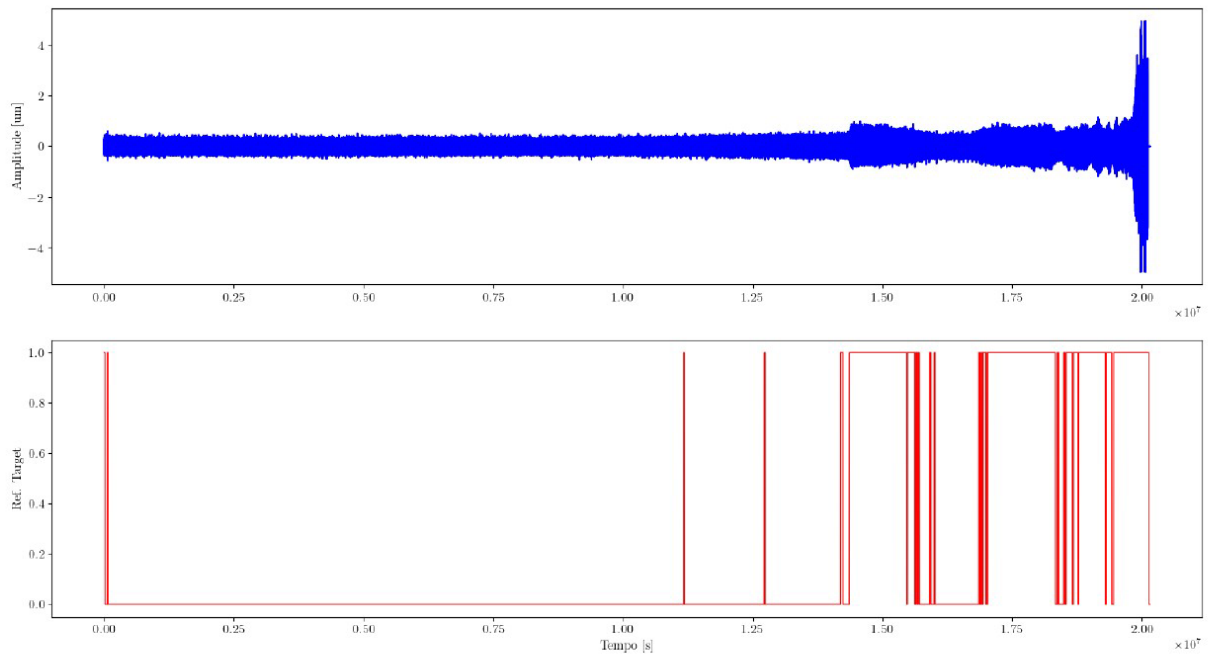


Figura 5.9: *Benchmark 5*: Sinal no tempo e classificação.

Capítulo 6

Resultados

Neste capítulo são apresentados, para os cinco *benchmarks* descritos previamente, os resultados da aplicação da arquitetura *Edge Analytics* e dos modelos individuais originalmente considerados na aplicação, bem como os resultados obtidos para modelos MLP e LSTM desenvolvidos no âmbito do presente trabalho. O capítulo se encerra com uma avaliação comparativa de desempenho dos modelos.

6.1 *Edge Analytics* - Modelos Originais

Esta seção apresenta os resultados da arquitetura *Edge Analytics* e dos modelos individuais obtidos para cada *benchmark* após otimização individualizada por conjunto de dados.

Utilizando os modelos desenvolvidos no LMEst, foi realizada a otimização para cada *benchmark* através da técnica de *Grid Search* para os hiperparâmetros de entrada de cada modelo, como o parâmetro de sensibilidade utilizado para ajuste do limiar utilizado na classificação de anomalia. Assim, os modelos foram executados para diferentes combinações das faixas de valores dos hiperparâmetros, sendo avaliadas as métricas de desempenho para cada combinação e, ao final, sendo adotada a combinação de melhores resultados. Convém observar que foi mantida a mesma configuração original do modelo *Gates*.

Convém destacar um aspecto metodológico do processo de otimização realizado neste trabalho. Embora os modelos avaliados sejam treinados em regime não supervisionado, utilizando exclusivamente dados considerados normais durante a etapa de aprendizado, a seleção dos hiperparâmetros foi realizada por meio de *Grid Search* orientado pelas métricas de desempenho calculadas com base no gabarito completo de cada *benchmark*.

Dessa forma, a etapa de otimização utilizou informações provenientes dos rótulos de classificação das amostras, buscando identificar a combinação de hiperparâmetros que maximizasse principalmente o *F1-score*. Assim, apesar de o treinamento dos modelos permanecer não supervisionado, a definição dos hiperparâmetros caracteriza uma etapa supervisionada de calibração experimental.

Consequentemente, os resultados apresentados devem ser interpretados como uma estimativa do desempenho máximo potencial dos modelos para cada conjunto de dados analisado, e não como uma avaliação totalmente independente de informações de classificação. Essa estratégia foi adotada com o objetivo de permitir uma comparação consistente entre diferentes arquiteturas e identificar o potencial de desempenho de cada abordagem.

A seguir é apresentada, para cada *benchmark*, uma figura com o sinal de vibração no tempo, o gabarito da classificação por amostra, classificações por modelo e conforme arquitetura completa. A região sombreada no início corresponde às amostras utilizadas durante treinamento da arquitetura *Edge Analytics*. Também é exibida para cada conjunto de dados uma tabela com as métricas de desempenho.

Conforme poderá ser observado, um aspecto a ser destacado é a capacidade de adaptação dos modelos de IA. Uma vez que uma anomalia é detectada e se torna persistente, este estado é reconhecido como a nova normalidade pelo modelo e, assim, uma nova anomalia só é contabilizada caso ocorra uma nova variação no sinal.

Durante a fase de treinamento dos modelos de *deep learning* as amostras são classificadas como anômalas, porém esta classificação não é considerada no cálculo das métricas de desempenho, que são calculadas somente considerando as amostras após etapa de treinamento.

As pequenas diferenças observadas na quantidade de amostras nas matrizes de confusão entre os modelos individuais e a arquitetura completa decorrem do mecanismo de sincronização da arquitetura. Os modelos que compõem o sistema possuem períodos de inicialização distintos, pois alguns necessitam de múltiplas amostras para calibrar *buffers* internos ou completar operações de janela deslizante. A arquitetura completa implementa um mecanismo de votação sincronizado que só produz um resultado quando todos os modelos estão aptos a contribuir com uma predição. Dessa forma, amostras iniciais onde qualquer modelo ainda não alcançou seu ponto de funcionamento estável são desconsideradas no cálculo das métricas da arquitetura, enquanto os modelos individuais, uma vez em operação, já começam a gerar predições.

A tabela 6.1 apresenta, para cada *benchmark*, o número total de pontos do sinal, o tamanho

de janela utilizado e o número de amostras resultante.

Tabela 6.1: Descrição dos *benchmarks* avaliados.

Benchmark	Pontos	Tamanho janela	Nº amostras
1	14.369.000	10.000	1436
2	1.915.142	20.000	95
3	1.523.400	17.000	89
4	1.740.000	10.000	174
5	20.152.320	20.480	984

6.1.1 Benchmark 1

A figura 6.1 apresenta as classificações por modelo, enquanto a tabela 6.2 apresenta as métricas de desempenho.

É possível observar que a camada *Gates* capta corretamente as anomalias mais significativas e que o desempenho individual da *Autoencoder* é similar ao da arquitetura completa, sendo obtido 89 % de *F1-score*.

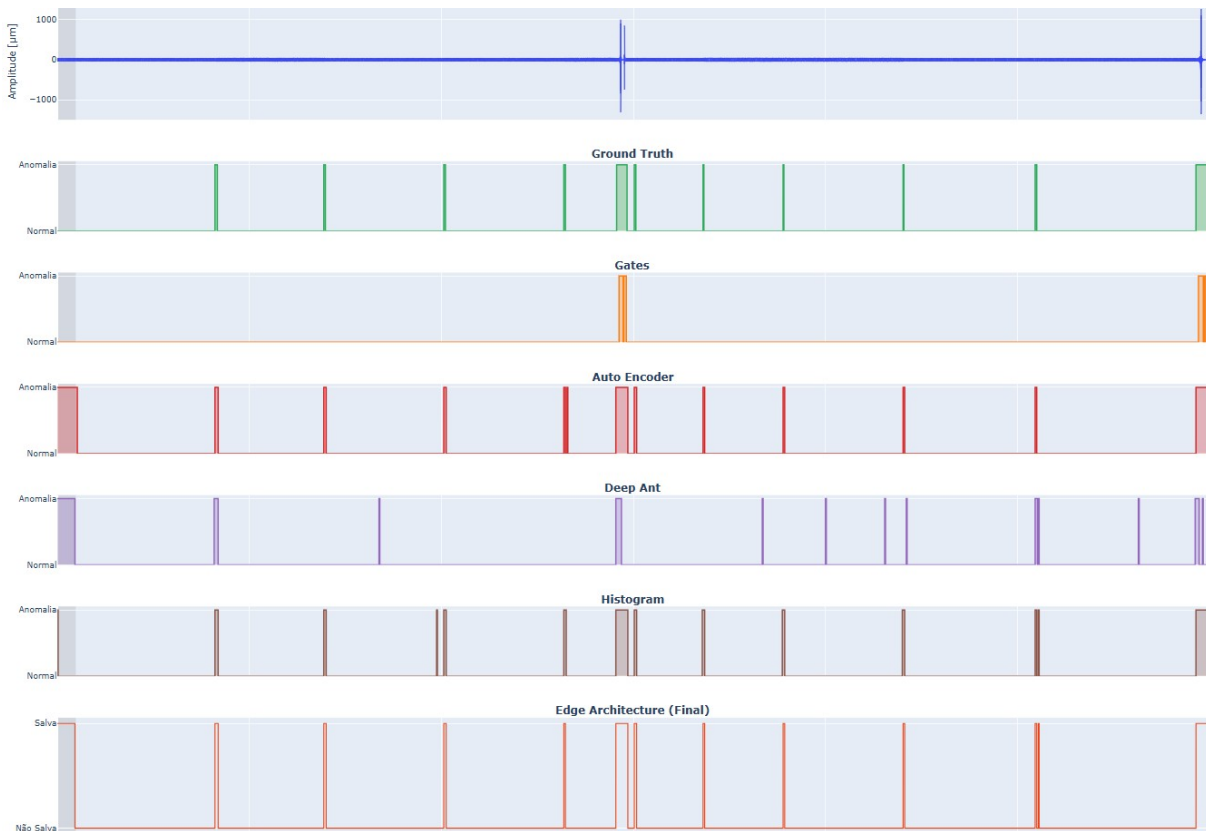


Figura 6.1: Benchmark 1: Sinal no tempo, gabarito e classificação dos modelos.

Tabela 6.2: *Benchmark 1*: Métricas de desempenho por modelo.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Gates	8	0	1361	21	98	100	28	43
Autoencoder	42	10	1359	0	99	81	100	89
DeepAnT	16	12	1360	26	97	57	38	46
Histogram	42	15	1356	0	99	74	100	85
Arquitetura Edge	42	10	1361	0	99	81	100	89

6.1.2 *Benchmark 2*

A figura 6.2 apresenta as classificações por modelo, enquanto a tabela 6.3 apresenta as métricas de desempenho.

Este sinal apresenta anomalias mais sutis, de forma que a camada *gates* não detectou nenhuma anomalia. O modelo *histogram* obteve o melhor desempenho individual, inclusive superando o da arquitetura completa.

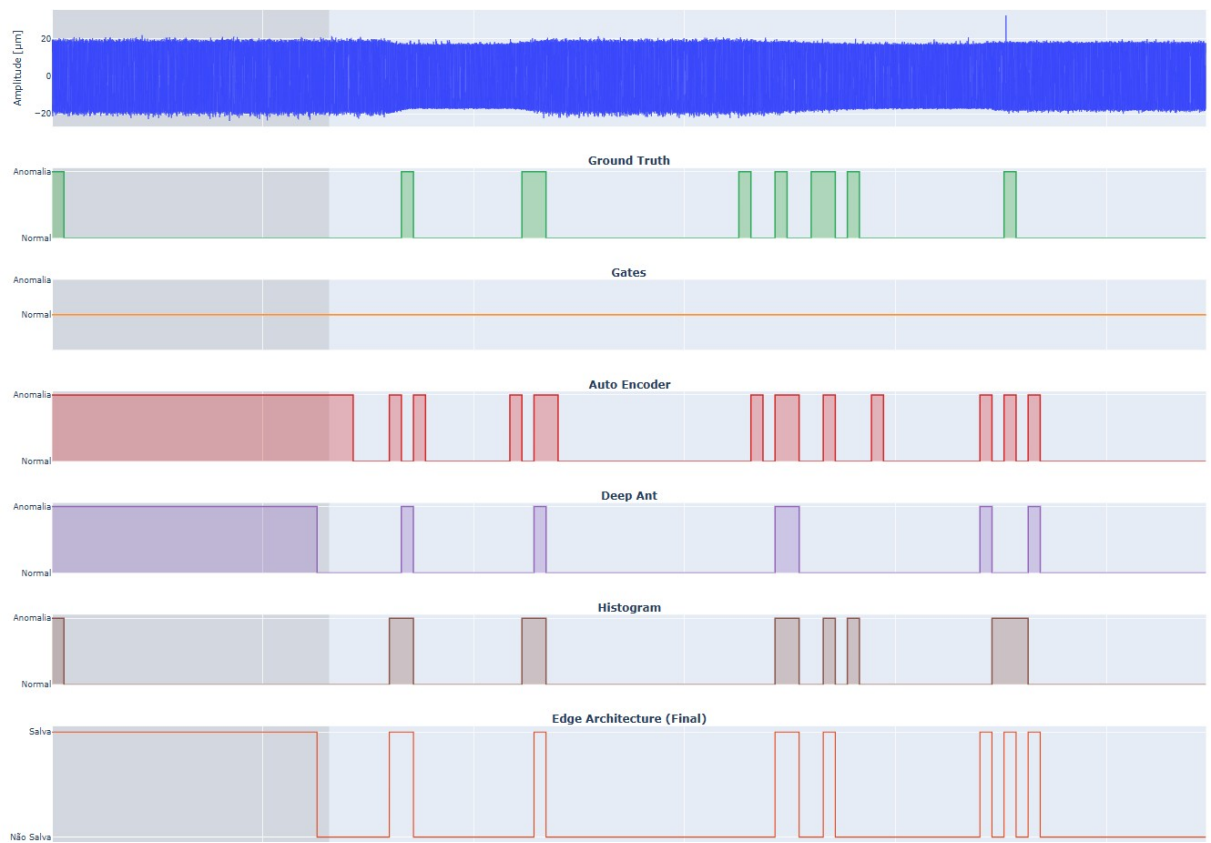


Figura 6.2: *Benchmark 2*: Sinal no tempo, gabarito e classificação dos modelos.

Tabela 6.3: *Benchmark 2*: Métricas de desempenho por modelo.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Gates	0	0	40	8	83	0	0	0
Autoencoder	4	9	52	5	80	31	44	36
DeepAnT	3	3	61	6	88	50	33	40
Histogram	7	4	58	2	92	64	78	70
Arquitetura Edge	5	4	59	4	89	56	56	56

6.1.3 *Benchmark 3*

A figura 6.3 apresenta as classificações por modelo, enquanto a tabela 6.4 apresenta as métricas de desempenho.

Este *benchmark* apresenta anomalias significativas, de forma que a camada *gates* tem desempenho próximo da arquitetura completa, evidenciando a capacidade da arquitetura de classificar corretamente anomalias deste tipo com baixo custo computacional, sem necessidade de aplicação de modelos de DL. Também é possível observar que o modelo *histogram* tem o melhor desempenho entre todos os modelos.

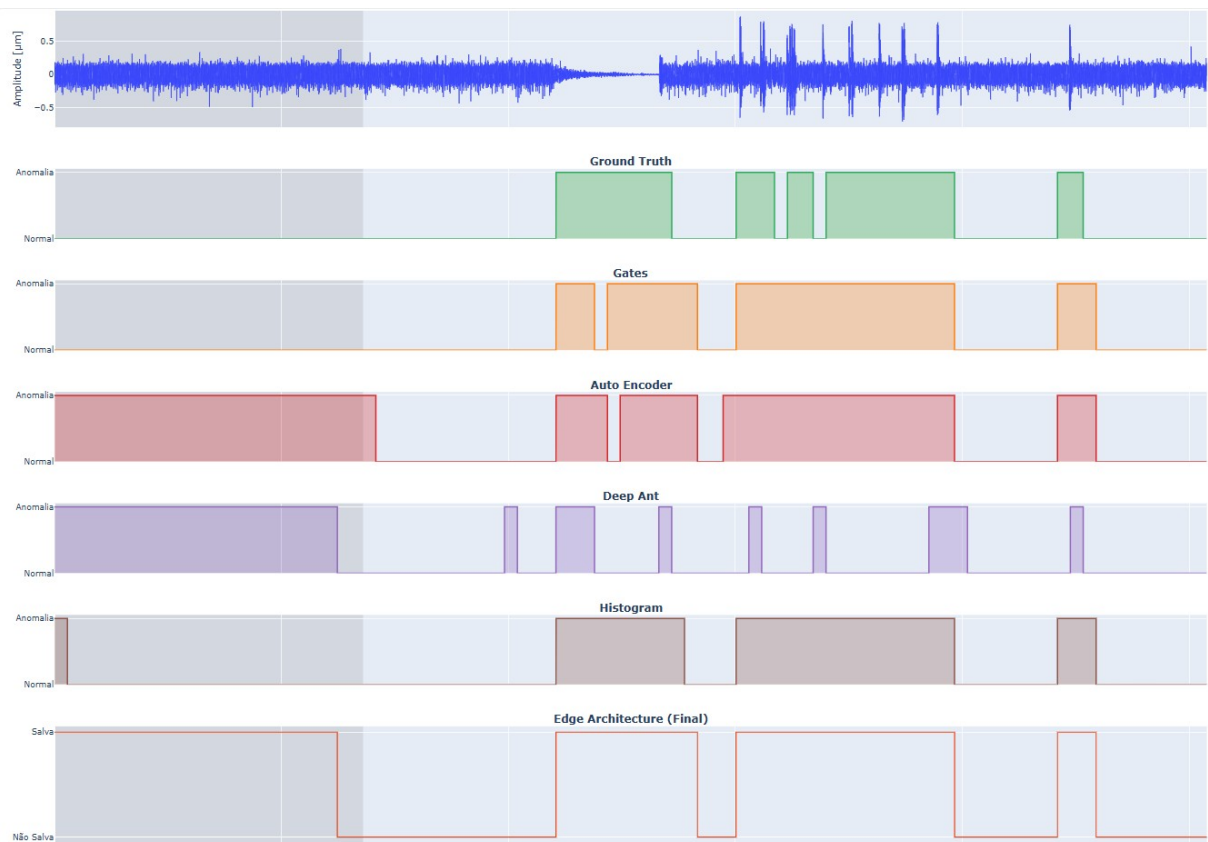
Figura 6.3: *Benchmark 3*: Sinal no tempo, gabarito e classificação dos modelos.

Tabela 6.4: *Benchmark 3*: Métricas de desempenho por modelo.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Gates	18	4	18	1	88	82	95	88
Autoencoder	25	6	32	1	89	81	96	88
DeepAnT	8	3	38	18	69	73	31	43
Histogram	26	4	35	0	94	87	100	93
Arquitetura Edge	26	5	34	0	92	84	100	91

6.1.4 *Benchmark 4*

A figura 6.4 apresenta as classificações por modelo, enquanto a tabela 6.5 apresenta as métricas de desempenho.

Observa-se que a camada *histogram* e a arquitetura completa apresentam desempenho similar.

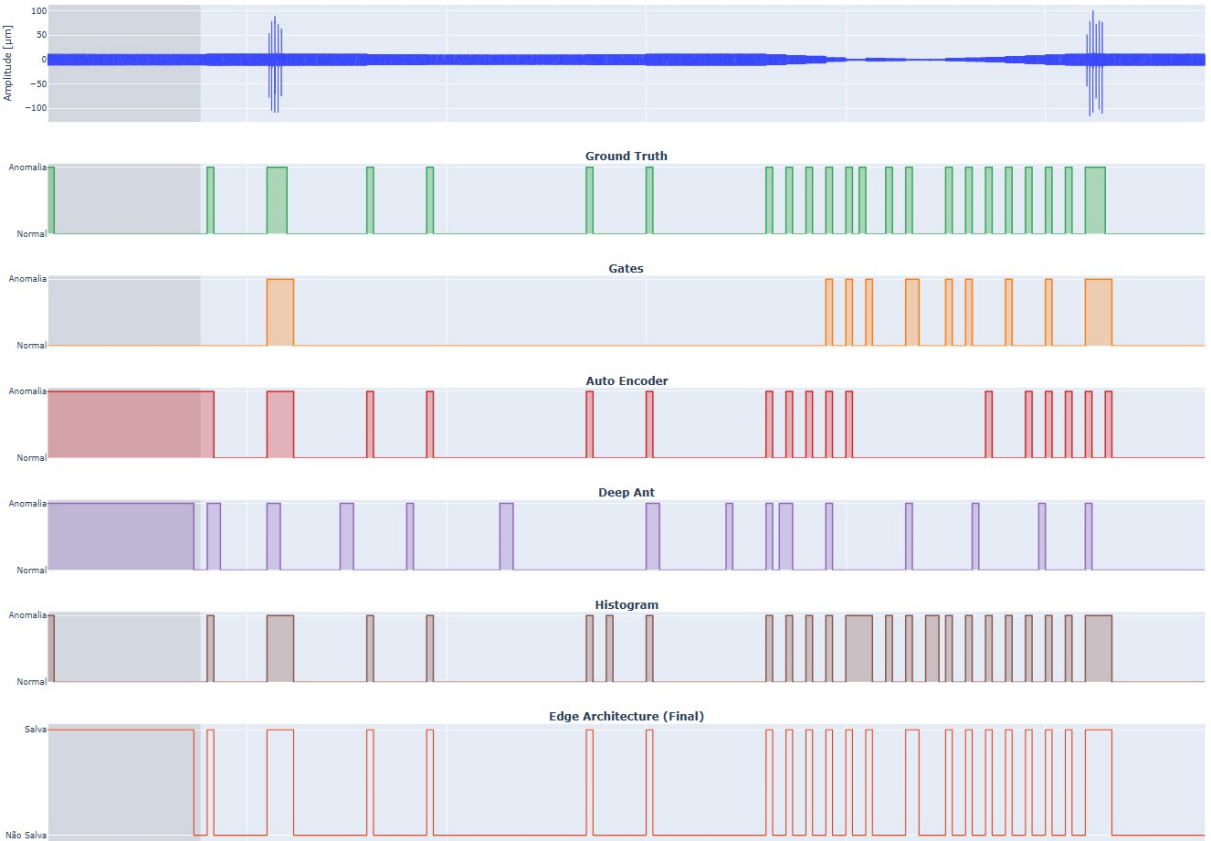


Figura 6.4: *Benchmark 4*: Sinal no tempo, gabarito e classificação dos modelos.

Tabela 6.5: *Benchmark 4*: Métricas de desempenho por modelo.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Gates	10	3	103	12	88	77	45	57
Autoencoder	17	2	122	8	93	89	68	77
DeepAnT	9	11	115	17	82	45	35	39
Histogram	26	7	118	0	95	79	100	88
Arquitetura Edge	24	4	121	2	96	86	92	89

6.1.5 *Benchmark 5*

A figura 6.5 apresenta as classificações por modelo, enquanto a tabela 6.6 apresenta as métricas de desempenho.

O modelo *DeepAnT* apresenta baixa performance na detecção de anomalias, enquanto a camada de histogramas apresenta o melhor desempenho individual e supera de forma significativa a arquitetura completa.

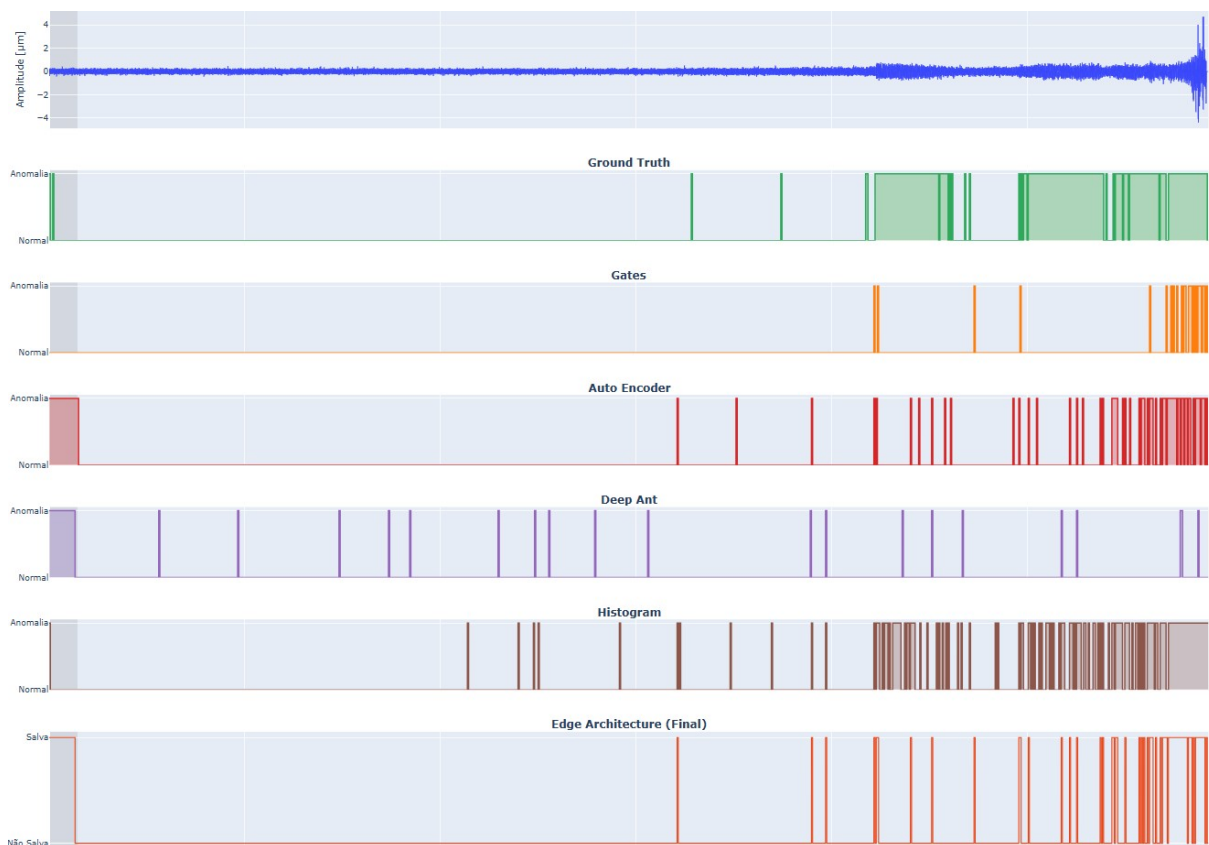
Figura 6.5: *Benchmark 5*: Sinal no tempo, gabarito e classificação dos modelos.

Tabela 6.6: *Benchmark 5*: Métricas de desempenho por modelo.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Gates	5	4	742	185	80	56	3	5
Autoencoder	55	10	736	158	82	85	26	40
DeepAnT	7	13	736	206	77	35	3	6
Histogram	123	23	724	90	88	84	58	68
Arquitetura Edge	56	8	739	157	83	88	26	40

6.2 MLP, LSTM e CPD

Um dos objetivos do presente trabalho foi de desenvolver modelos de ML para aplicação nos mesmos *benchmarks* e comparação com os resultados dos modelos originais do *Edge Analytics*.

Assim, foram desenvolvidos códigos computacionais na linguagem *python* utilizando a API Keras do *Tensorflow* para os modelos de MLP (*Multilayer Perceptron*) e LSTM (*Long Short Term Memory*) que, de maneira similar aos modelos originais do *Edge Analytics*, operam de forma não supervisionada.

Também foi avaliado o desempenho da técnica de CPD (*Change Point Detection*) aplicada em conjunto com esses modelos de ML.

6.2.1 MLP

A arquitetura do modelo MLP adotado neste trabalho foi baseada em um *Autoencoder* não supervisionado, treinado exclusivamente com dados associados à condição normal de operação. O treinamento foi realizado utilizando o mesmo número de amostras empregado no treinamento da arquitetura do *Edge Analytics*, de forma a garantir comparabilidade entre as diferentes abordagens avaliadas. O sinal de entrada do modelo corresponde diretamente ao sinal de vibração no domínio do tempo, o qual é previamente segmentado em janelas de tamanho fixo N_w , com sobreposição controlada pelo parâmetro *STRIDE*. Cada janela é reorganizada em um vetor unidimensional, que constitui a entrada do MLP.

A rede é composta por duas camadas densas no codificador (*encoder*), seguidas por uma camada de gargalo (*bottleneck*) responsável pela compressão da informação, e duas camadas densas no decodificador (*decoder*), responsáveis pela reconstrução do sinal original. A dimensionalidade da camada de gargalo é definida pelo parâmetro *bottleneck_ratio*, que controla o grau de compressão imposto ao modelo. Utiliza-se *early stopping* durante treinamento baseado na perda de validação, de modo a evitar *overfitting*. A otimização dos pesos da rede é realizada

utilizando o algoritmo Adam, devido à sua eficiência e robustez em problemas de aprendizado profundo.

Após o treinamento, a detecção de anomalias é realizada a partir do erro de reconstrução, calculado como o valor RMS entre o sinal de entrada e o sinal reconstruído pelo *Autoencoder*. O limiar de decisão é definido de forma estatística a partir dos dados de treinamento, segundo a expressão $\text{thr} = \mu + k \cdot \sigma$, onde μ e σ representam, respectivamente, a média e o desvio padrão do erro de reconstrução, e k é o parâmetro *threshold_k*. Adicionalmente, o modelo permite ajuste opcional de sensibilidade aplicado aos erros de reconstrução.

Para fins de comparação direta com o *Edge Analytics*, os resultados são avaliados considerando blocos de amostras com o mesmo tamanho utilizado na arquitetura de referência. Nessa avaliação, se ao menos uma janela N_w for classificada como anômala todo o bloco é contabilizado como anômalo.

A etapa de otimização do modelo foi implementada utilizando a técnica de *Grid Search*, sendo avaliadas combinações de hiperparâmetros (como janelamento, sensibilidade, gargalo) para cada *benchmark* e sendo computadas as métricas de desempenho para cada combinação. Ao final do processo, são adotados os hiperparâmetros que implicam nos melhores resultados para cada conjunto de dados, sendo utilizado o *F1-score* como referência.

De forma análoga à arquitetura *Edge Analytics*, a otimização dos hiperparâmetros do modelo MLP foi realizada utilizando o gabarito de cada *benchmark* para cálculo das métricas de desempenho empregadas como critério de seleção. Assim, embora a rede tenha sido treinada exclusivamente com dados considerados normais e sem utilização de rótulos durante o aprendizado, a escolha final dos hiperparâmetros foi conduzida com auxílio das informações de classificação presentes no conjunto de dados. Essa abordagem permitiu avaliar o potencial máximo do modelo em cada cenário analisado, porém introduz uma dependência do gabarito que pode não estar disponível em aplicações industriais reais.

A tabela 6.7 apresenta o tamanho de janela otimizado para cada *benchmark* avaliado, considerando a melhor combinação de hiperparâmetros obtida por meio do processo de otimização.

Tabela 6.7: Tamanho de amostra otimizado do modelo MLP por *benchmark*.

<i>Benchmark</i>	Tamanho janela
1	512
2	1.024
3	16.384
4	16
5	128

A seguir são apresentados os resultados obtidos com o modelo MLP para os *benchmarks* avaliados. A tabela 6.8 apresenta as métricas de desempenho para cada conjunto de dados, considerando a melhor combinação de hiperparâmetros obtida por meio do processo de otimização. Na sequência são apresentados, para cada *benchmark*, o sinal de vibração no tempo, o gabarito da classificação por amostra e as classificações obtidas pelo modelo MLP.

Tabela 6.8: Métricas de desempenho do modelo MLP para os *benchmarks* avaliados.

Benchmark	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
1	8	0	1371	34	98	100	19	32
2	1	0	62	8	89	100	11	20
3	18	3	36	8	83	86	69	77
4	14	14	111	12	83	50	54	52
5	206	22	725	7	97	90	97	93

Observa-se que o modelo MLP apresentou o melhor desempenho no *benchmark* 5, obtendo *F1-score* de 93 % e superando os modelos do *Edge Analytics*. O modelo também superou o desempenho da *DeepAnT* nos *benchmarks* 3 e 4, obtendo performance inferior aos demais modelos. Já nos *benchmarks* 1 e 2, o modelo foi superado pelas demais abordagens.

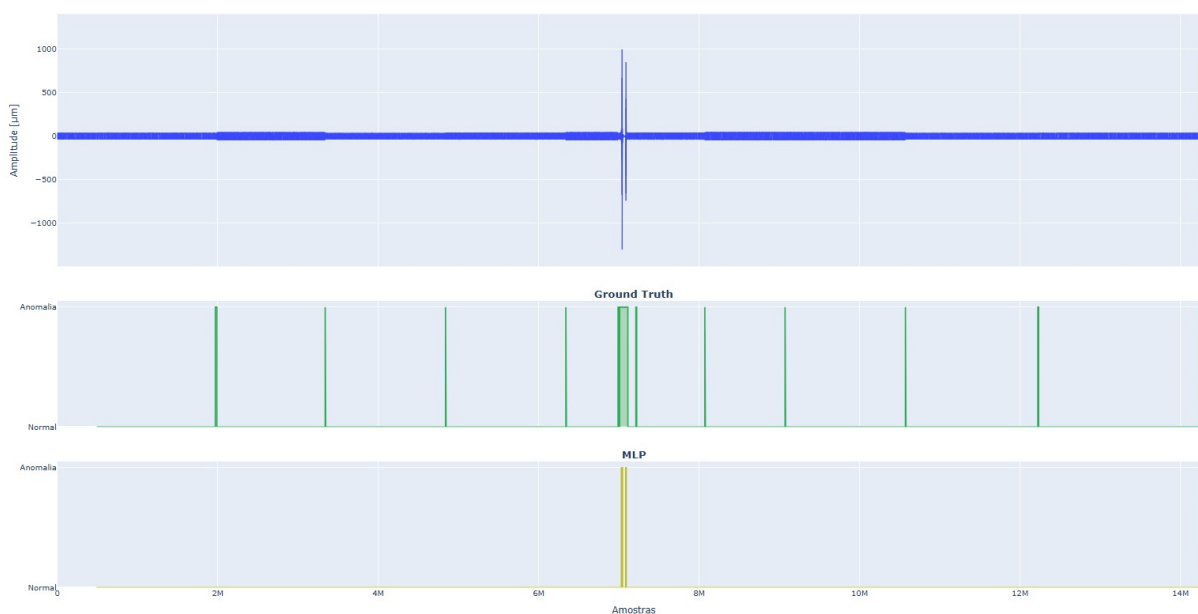


Figura 6.6: Sinal no tempo, gabarito e classificação do modelo MLP para o *benchmark 1*.

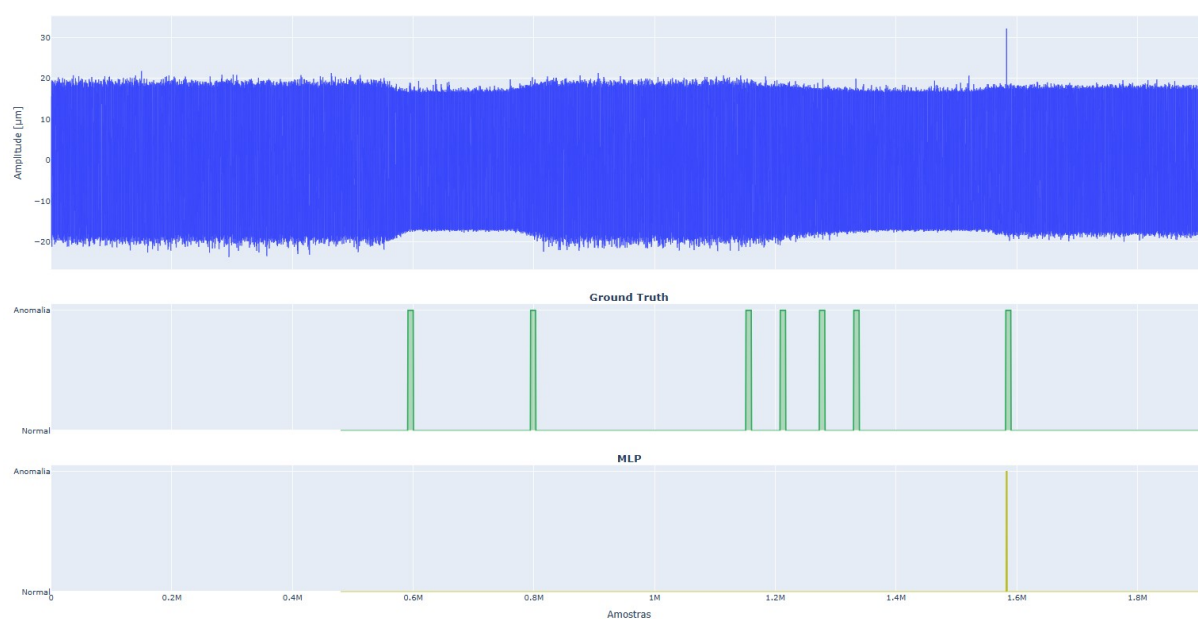


Figura 6.7: Sinal no tempo, gabarito e classificação do modelo MLP para o *benchmark 2*.

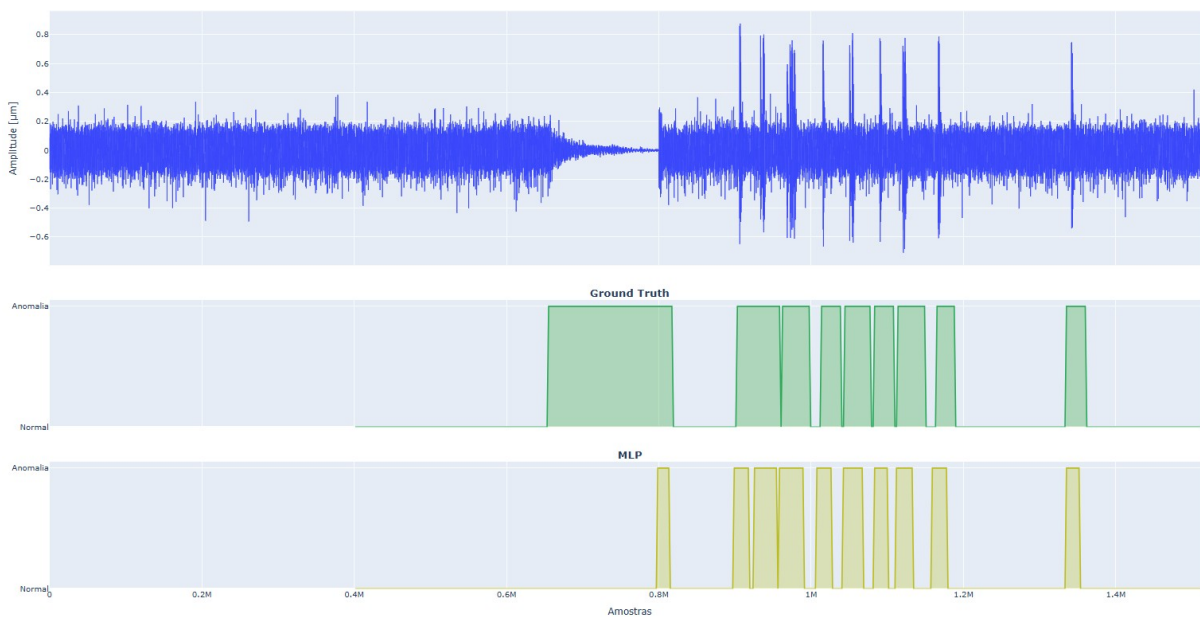


Figura 6.8: Sinal no tempo, gabarito e classificação do modelo MLP para o *benchmark 3*.

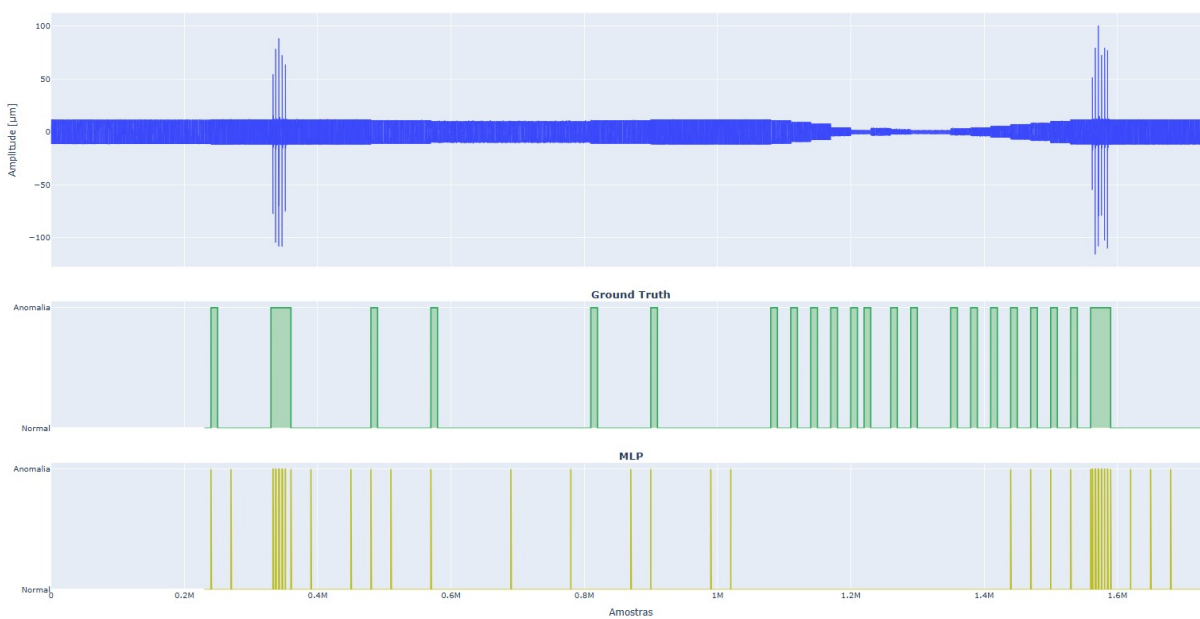


Figura 6.9: Sinal no tempo, gabarito e classificação do modelo MLP para o *benchmark 4*.

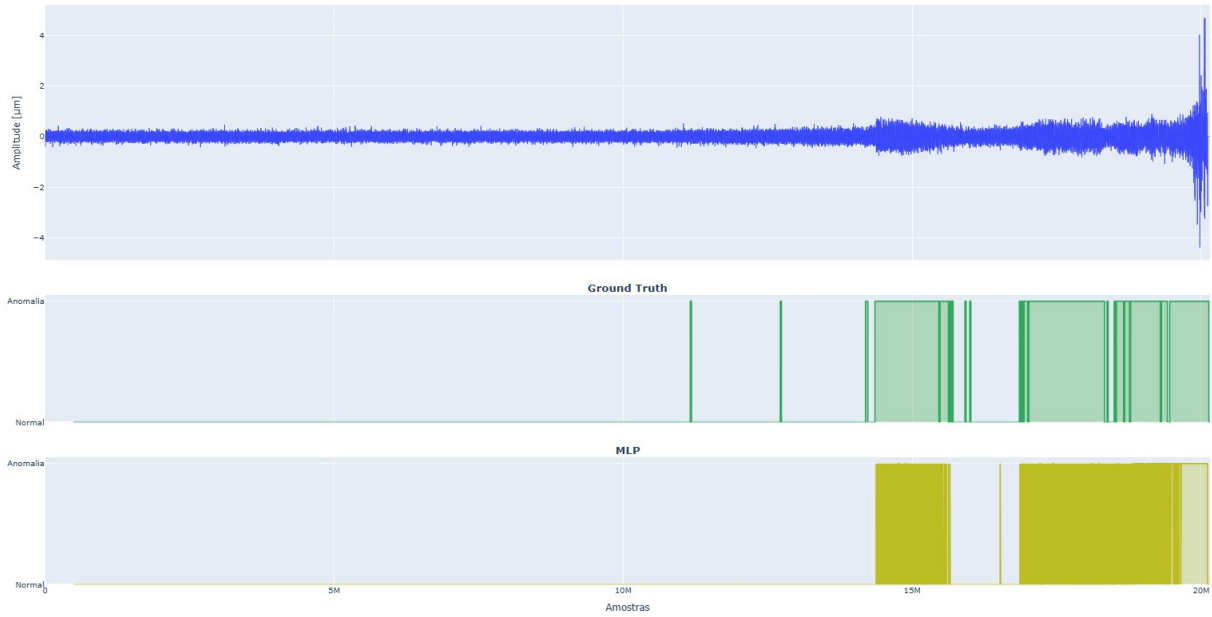


Figura 6.10: Sinal no tempo, gabarito e classificação do modelo MLP para o *benchmark 5*.

6.2.2 LSTM

A arquitetura do modelo LSTM adotado neste trabalho foi implementada como um preditor não supervisionado, treinado exclusivamente com dados associados à condição normal de operação. O treinamento foi realizado utilizando o mesmo número de amostras empregado no treinamento da arquitetura do *Edge Analytics*, de forma a garantir comparabilidade entre as diferentes abordagens avaliadas. O sinal de entrada corresponde ao sinal de vibração no domínio do tempo, o qual é previamente segmentado em janelas de tamanho fixo N_w , com sobreposição controlada pelo parâmetro *STRIDE*. A partir dessas janelas, são formadas sequências temporais de comprimento L_{SEQ} , nas quais cada sequência contém L_{SEQ} blocos consecutivos do sinal. O objetivo do modelo é realizar predição do próximo bloco de amostras a partir do histórico recente representado pela sequência de entrada.

A rede é composta por N_{LAYERS} camadas recorrentes do tipo *LSTM*, cada uma com $LSTM_UNITS$ unidades de memória, e com aplicação de *dropout* para redução de *overfitting*. Na implementação utilizada, as camadas LSTM são empilhadas (*stacked LSTM*), em que as camadas intermediárias retornam sequências e a última camada retorna apenas o estado final. Após cada camada LSTM, é empregada uma camada densa com ativação *ReLU* para enriquecimento da representação aprendida, seguida de uma camada densa linear responsável por produzir a estimativa do bloco futuro, com dimensionalidade igual à do bloco de entrada. O treinamento é realizado com o otimizador Adam e adota-se *early stopping*.

Após o treinamento, a detecção de anomalias é efetuada a partir do erro de predição do modelo, calculado entre o bloco real e o bloco previsto para cada janela analisada. Esse erro constitui um *score* por janela, que é comparado a um limiar obtido a partir de um percentil do erro observado no conjunto de treinamento realizado com amostras normais, com a possibilidade de ajuste para atender uma precisão mínima especificada e de aplicação de um fator de sensibilidade. Como a predição só é possível após a formação completa das sequências de comprimento L_{SEQ} , as janelas iniciais são descartadas. Por fim, as decisões de classificação binárias (normal/anômalo) são atribuídas às janelas previstas e podem ser comparadas diretamente com os alvos por janela.

Para atender o objetivo de comparação direta com o *Edge Analytics*, os resultados são avaliados considerando blocos de amostras com o mesmo tamanho utilizado na arquitetura de referência. Nessa avaliação, se ao menos uma janela N_w for classificada como anômala todo o bloco é contabilizado como anômalo.

A etapa de otimização do modelo foi implementada utilizando a técnica de *Grid Search*, sendo avaliadas combinações de hiperparâmetros (como janelamento, sobreposição, sequência, quantidade de unidades de LSTM, sensibilidade) para cada *benchmark* e sendo computadas as métricas de desempenho para cada combinação. Ao final do processo, são adotados os hiperparâmetros que implicam nos melhores resultados para cada conjunto de dados, sendo utilizado o *F1-score* como referência.

Cabe ressaltar que o processo de otimização dos hiperparâmetros da rede LSTM também utilizou o gabarito dos *benchmarks* para avaliação das diferentes configurações testadas. Portanto, apesar do treinamento da rede ocorrer de forma não supervisionada, utilizando apenas dados representativos da condição normal, a seleção dos hiperparâmetros foi orientada por métricas calculadas a partir dos rótulos de classificação. Essa característica deve ser considerada na interpretação dos resultados, uma vez que fornece ao processo de calibração informações que eventualmente não estariam disponíveis em aplicações industriais de monitoramento contínuo.

A tabela 6.9 apresenta o tamanho de janela otimizado para cada *benchmark* avaliado, considerando a melhor combinação de hiperparâmetros obtida por meio do processo de otimização.

Tabela 6.9: Tamanho de amostra otimizado do modelo LSTM por *benchmark*.

<i>Benchmark</i>	Tamanho janela
1	4.096
2	8.192
3	16.384
4	128
5	1.024

A seguir são apresentados os resultados obtidos com o modelo LSTM para os *benchmarks* avaliados. A tabela 6.10 apresenta as métricas de desempenho para cada conjunto de dados, considerando a melhor combinação de hiperparâmetros obtida por meio do processo de otimização. Na sequência são apresentados, para cada *benchmark*, o sinal de vibração no tempo, o gabarito da classificação por amostra e as classificações obtidas pelo modelo LSTM.

Tabela 6.10: Métricas de desempenho do modelo LSTM para os *benchmarks* avaliados.

Benchmark	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
1	15	0	1371	27	98	100	36	53
2	3	9	53	6	79	25	33	29
3	18	3	36	8	83	86	69	77
4	13	8	117	13	86	62	50	55
5	204	12	735	9	98	94	96	95

Observa-se que o modelo LSTM apresentou o melhor desempenho no *benchmark* 5, obtendo *F1-score* de 95 % e superando os modelos do *Edge Analytics*. O modelo também superou o desempenho da *DeepAnT* nos *benchmarks* 1, 3 e 4, obtendo performance inferior aos demais modelos. Já no *benchmark* 2, o modelo foi superado pelas demais abordagens.

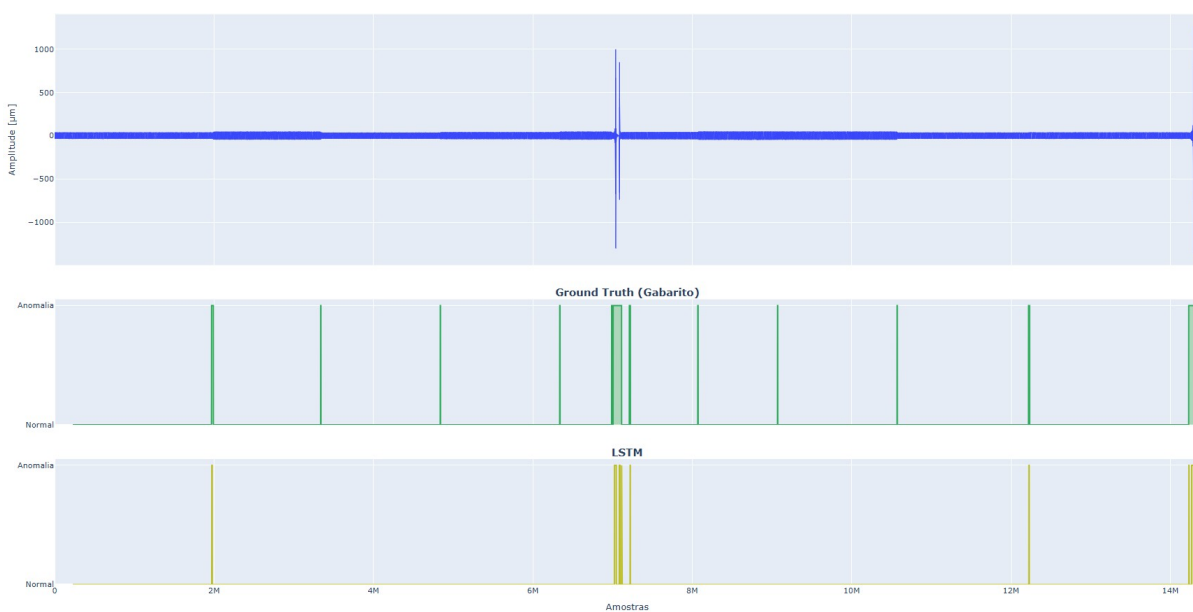


Figura 6.11: Sinal no tempo, gabarito e classificação do modelo LSTM para o *benchmark 1*.

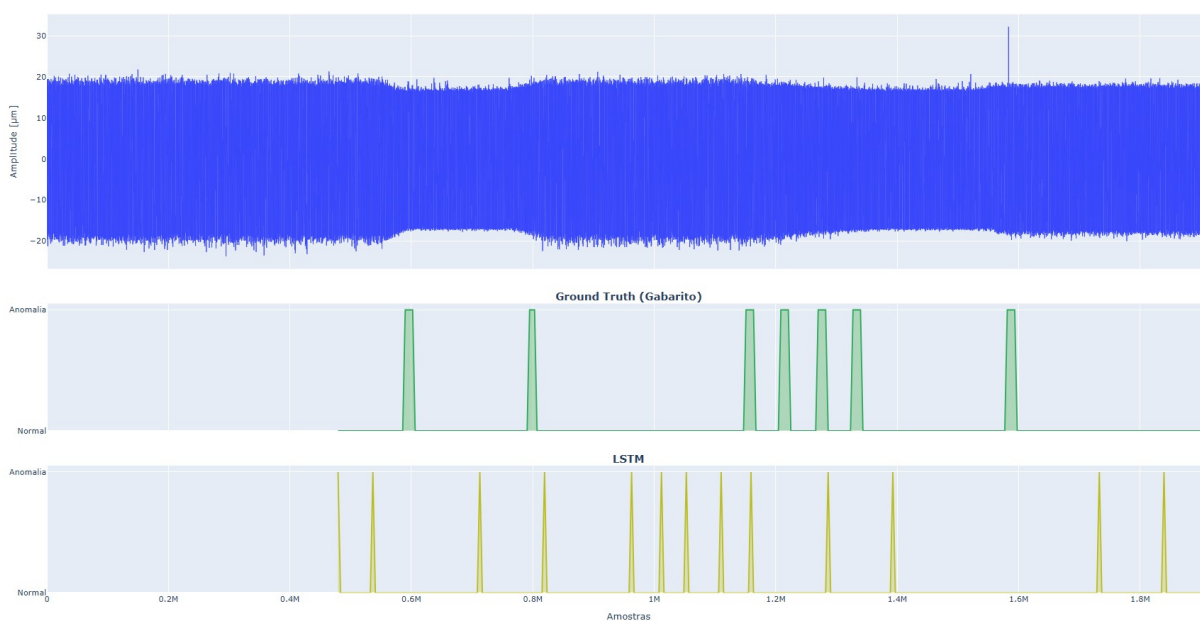


Figura 6.12: Sinal no tempo, gabarito e classificação do modelo LSTM para o *benchmark 2*.

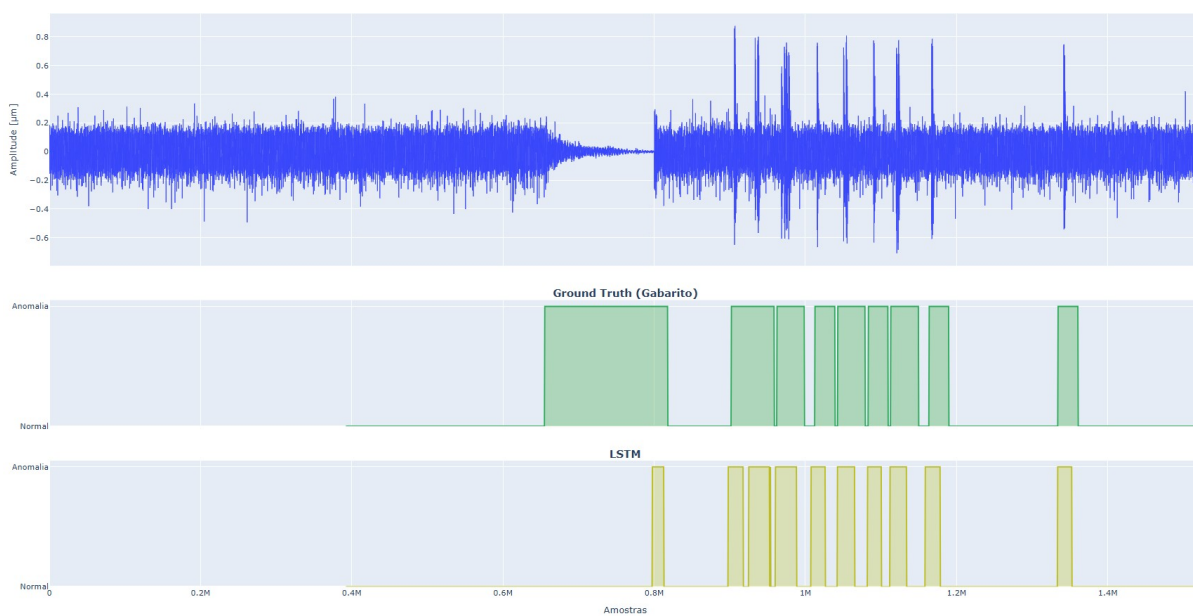


Figura 6.13: Sinal no tempo, gabarito e classificação do modelo LSTM para o *benchmark 3*.

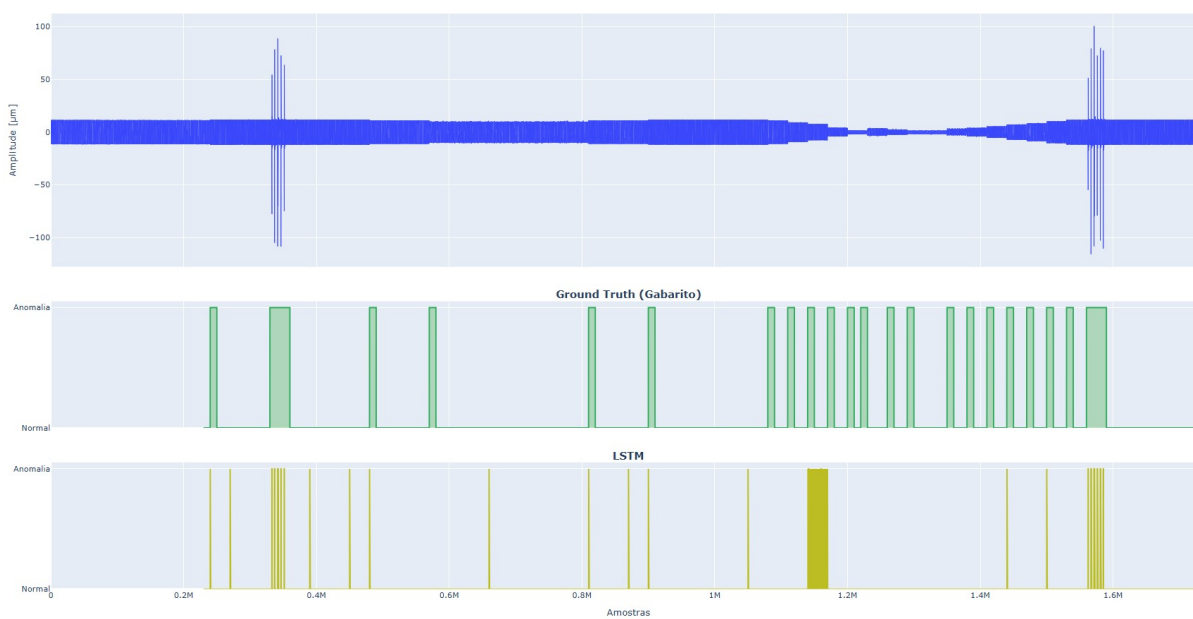


Figura 6.14: Sinal no tempo, gabarito e classificação do modelo LSTM para o *benchmark 4*.

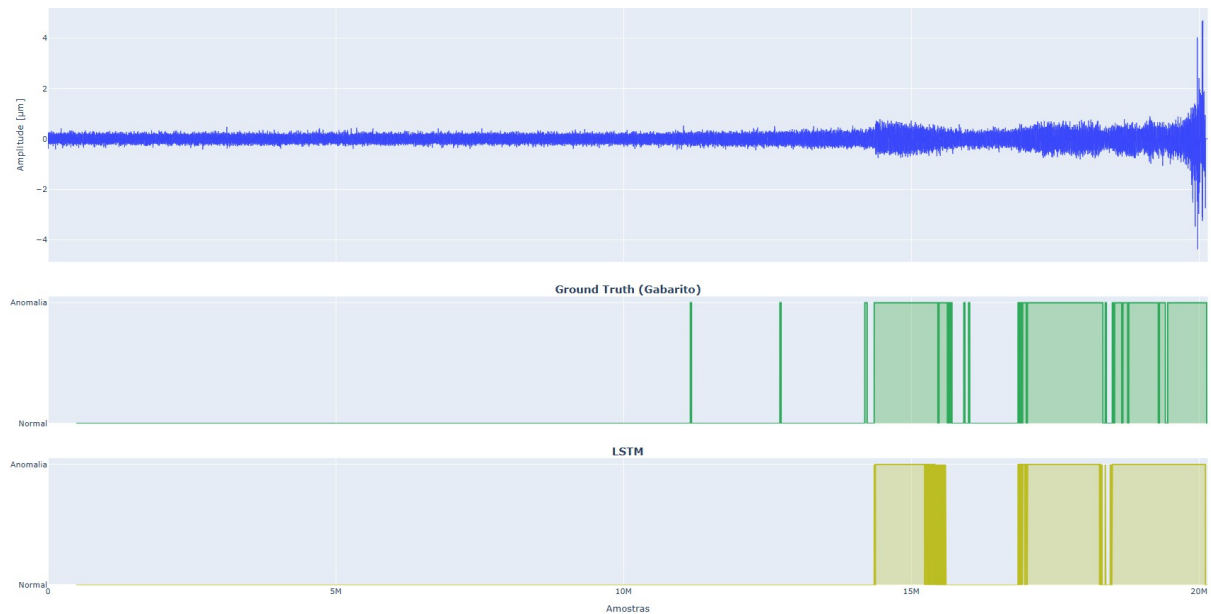


Figura 6.15: Sinal no tempo, gabarito e classificação do modelo LSTM para o *benchmark 5*.

6.3 *Change Point Detection (CPD)*

Analisando os resultados dos modelos LSTM e MLP, observa-se eficácia na detecção de anomalias mais significativas em relação à tendência dos valores de vibração no tempo, enquanto os modelos falham na detecção de anomalias mais sutis e nas mudanças sustentadas de comportamento da resposta de vibração.

De posse desta percepção, foi adicionada à cada modelo uma camada adicional inspirada na técnica de CPD (*Change Point Detection*).

A implementação do CPD foi acoplada aos *pipelines* de MLP Autoencoder e LSTM como uma etapa pós-modelo, atuando diretamente sobre a série temporal de *scores* (erro de reconstrução na MLP e erro de predição na LSTM). Assim, a camada adicional de CPD consistiu na inserção de detecção de anomalias baseada em estatísticas do *score* de erro calculadas em janelas da série temporal. Após suavização do *score* por mediana móvel, quantifica-se as mudanças estruturais pela diferença absoluta entre medianas consecutivas e define-se um limiar adaptativo usando um percentil dessas variações, complementado por uma magnitude mínima para descartar flutuações consideradas normais. Em seguida, cada candidato à mudança é validado comparando o nível do *score* antes e depois do ponto (critério de variação relativa), marcando uma banda curta como região anômala quando ocorre alteração sustentada de patamar.

As arquiteturas combinadas LSTM + CPD e MLP + CPD mantém a detecção clássica por limiar global do *score* e a decisão final da classificação considera uma amostra como anômala

caso seja assim classificada por uma das duas estratégias. Portanto, foram consideradas duas abordagens complementares para detecção de anomalia, onde a detecção por limiar captura anomalias mais abruptas e o CPD detecta anomalias associadas a variações sustentadas de comportamento da vibração.

Tomando como exemplo o *benchmark* 1, o modelo originalmente desenvolvido de LSTM apresentou dificuldade na detecção de anomalias sutis associadas a mudança de comportamento de vibração, como é evidenciado na figura 6.16. A aplicação do CPD sobre o *score* de predição do modelo LSTM permitiu a detecção destas anomalias, como é evidenciado na figura 6.17, onde os pontos de mudança detectados por CPD coincidem com as transições no comportamento da vibração, mesmo quando o *score* de predição não ultrapassa o limiar global.

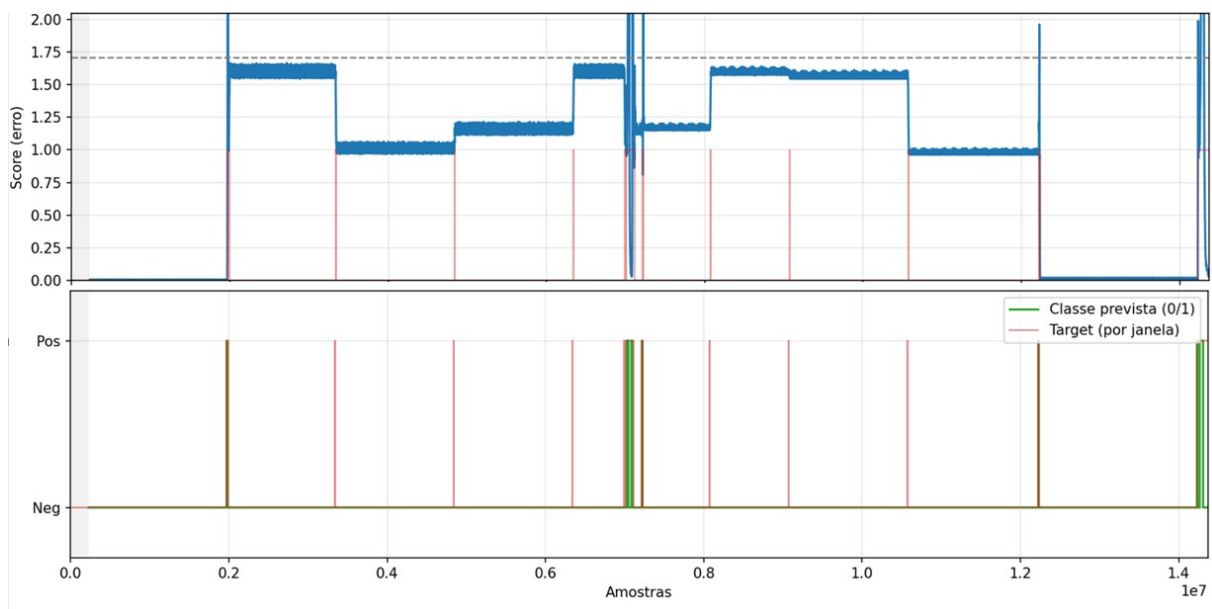


Figura 6.16: Erro de previsão e classificação por limiar do modelo LSTM para o *benchmark* 1.

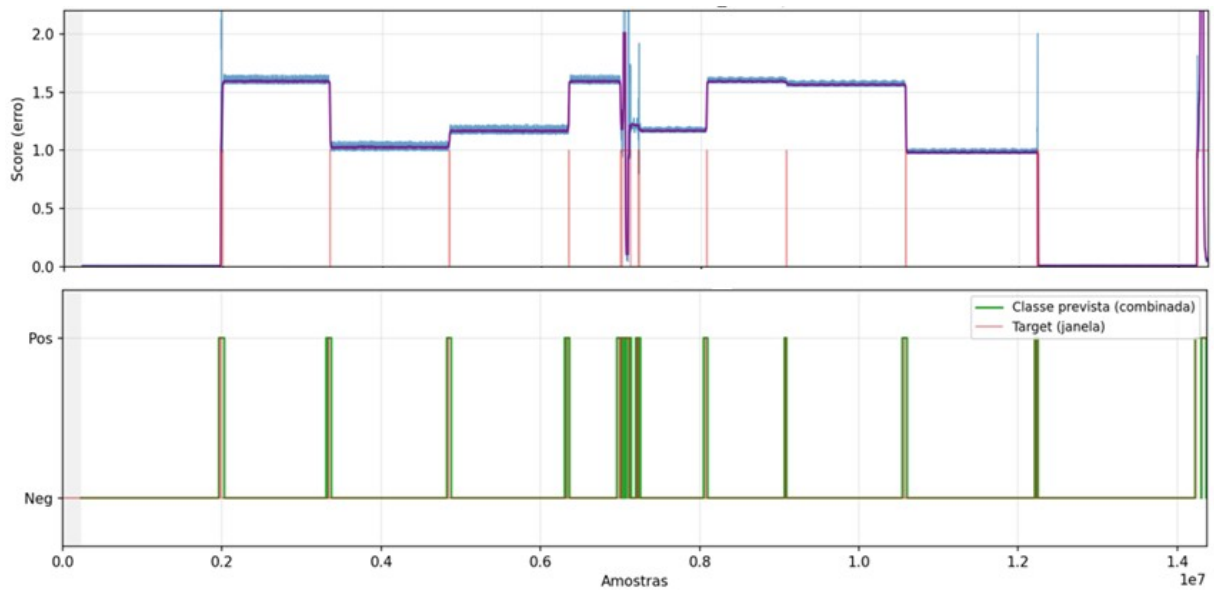


Figura 6.17: Erro de previsão e classificação do modelo LSTM + CPD para o *benchmark* 1.

6.3.1 Combinação MLP e CPD

Os parâmetros do CPD foram ajustados individualmente por *benchmark* e a seguir são exibidos os resultados obtidos.

Tabela 6.11: Métricas de desempenho do modelo MLP + CPD para os *benchmarks* avaliados.

Benchmark	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
1	41	33	1338	1	98	55	98	71
2	5	5	57	4	87	50	56	53
3	24	8	31	2	85	75	92	83
4	21	13	112	5	88	62	81	70
5	204	23	724	9	97	90	96	93

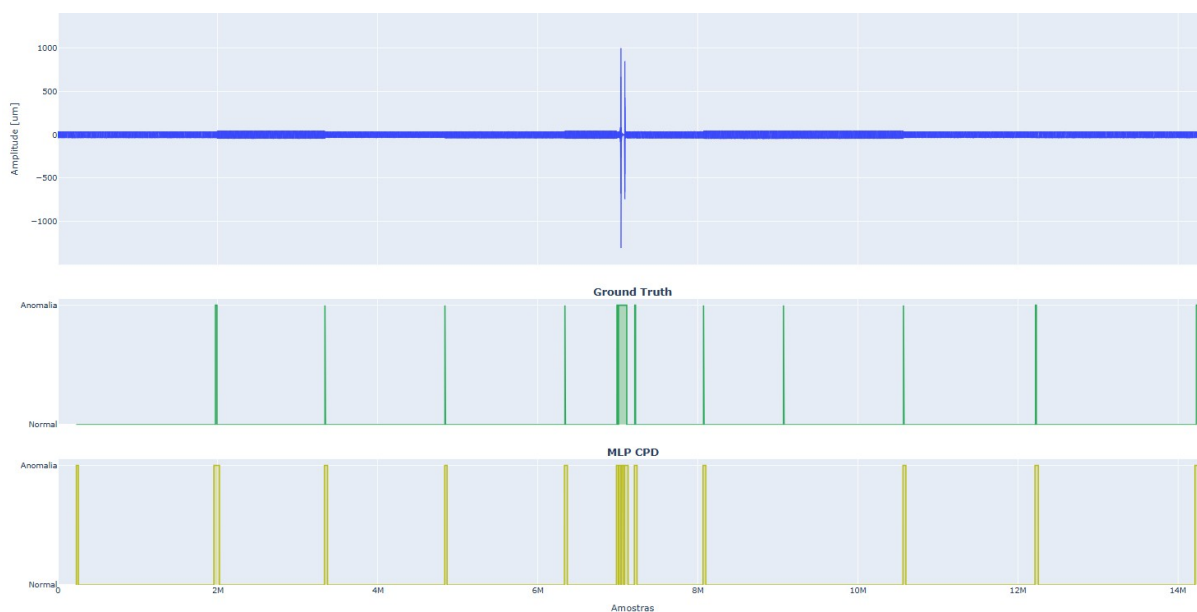


Figura 6.18: Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o *benchmark* 1.

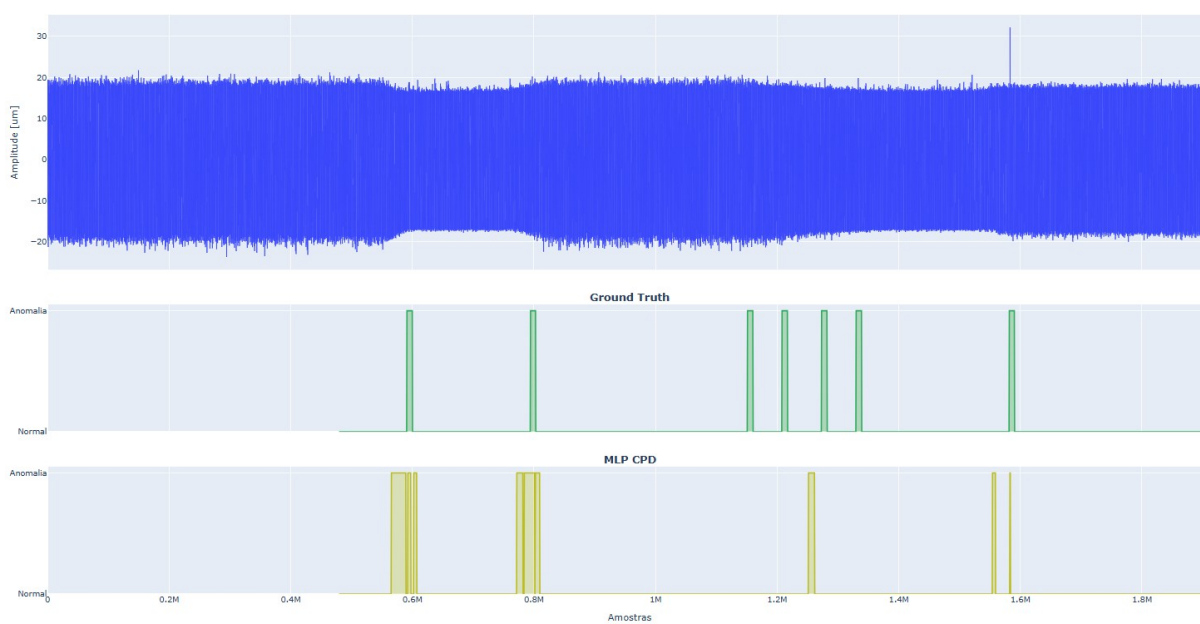


Figura 6.19: Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o *benchmark* 2.

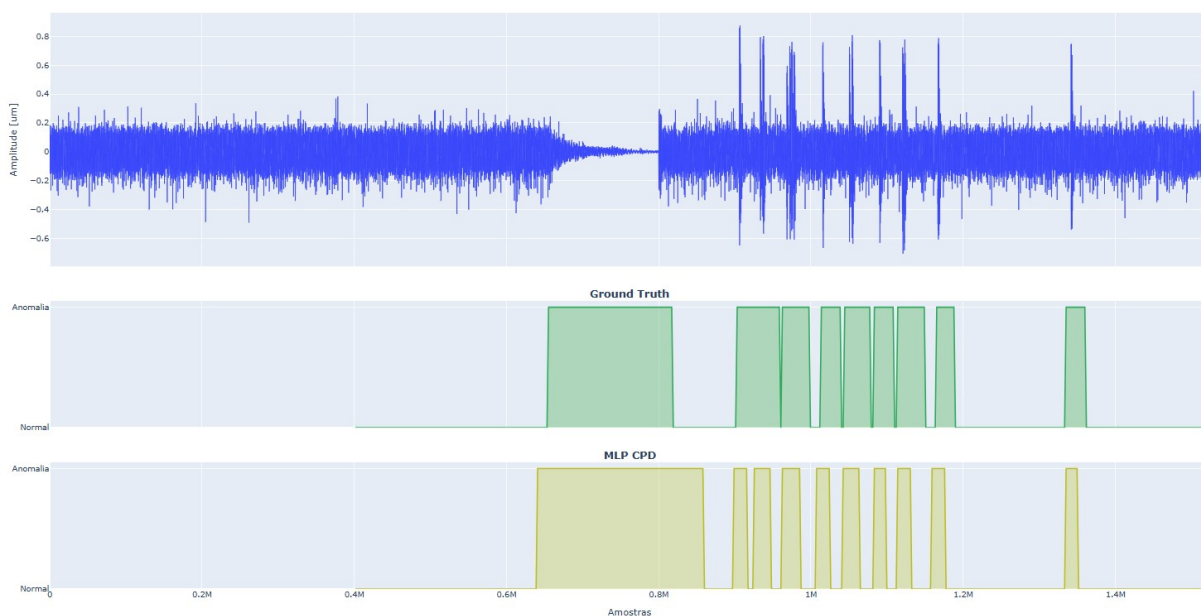


Figura 6.20: Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o *benchmark* 3.

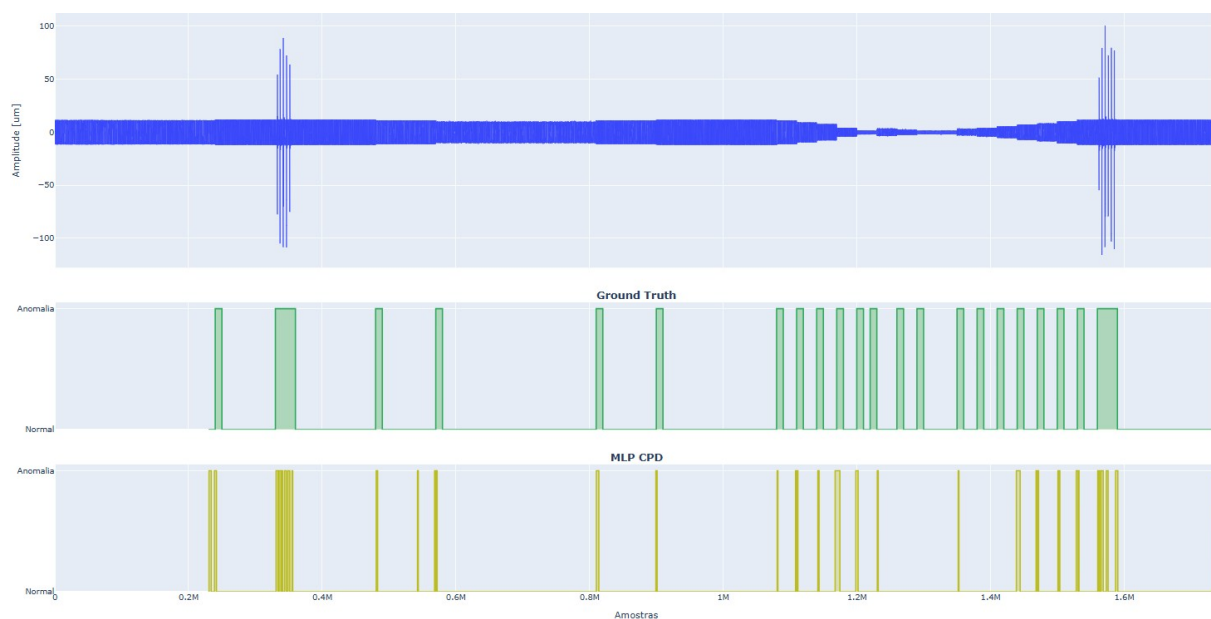


Figura 6.21: Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o *benchmark* 4.

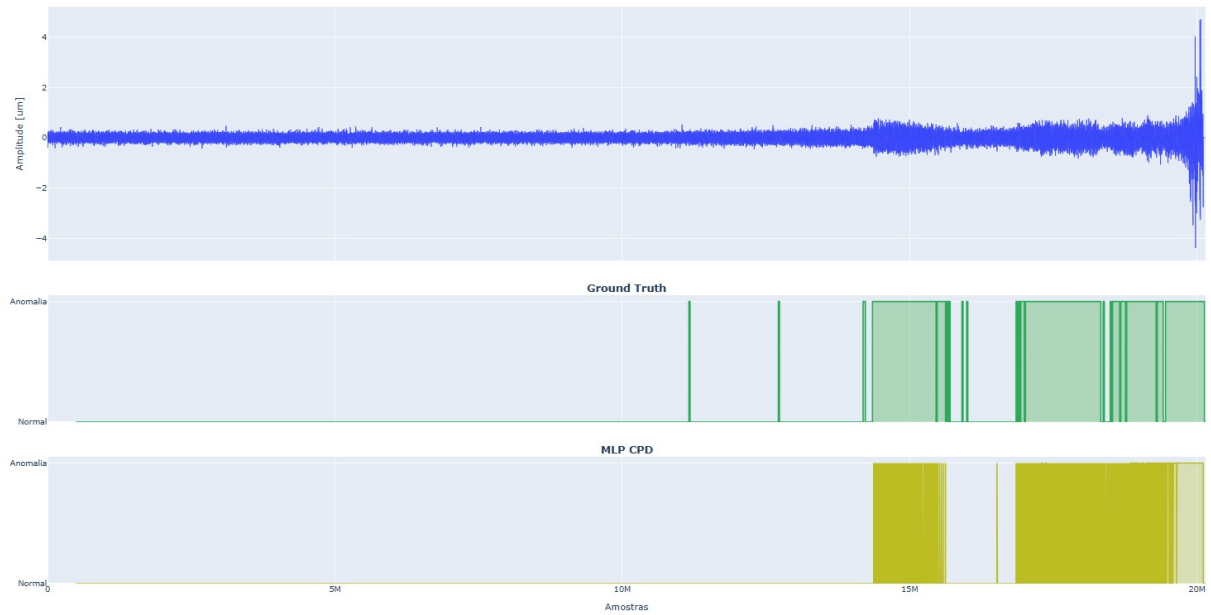


Figura 6.22: Sinal no tempo, gabarito e classificação do modelo MLP + CPD para o *benchmark* 5.

Observa-se que a combinação do modelo MLP com CPD leva à melhoria significativa de desempenho para os *benchmarks* 1 a 4. O desempenho para o *benchmark* 5 foi similar, porém convém ressaltar que já haviam sido obtidas métricas de desempenho significativamente elevadas para este conjunto de dados.

6.3.2 Combinação LSTM e CPD

Tabela 6.12: Métricas de desempenho do modelo LSTM + CPD para os *benchmarks* avaliados.

Benchmark	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
1	42	42	1329	0	97	50	100	67
2	4	9	53	5	80	31	44	36
3	25	6	33	1	89	81	96	88
4	22	17	108	4	86	56	85	68
5	204	12	735	9	98	94	96	95

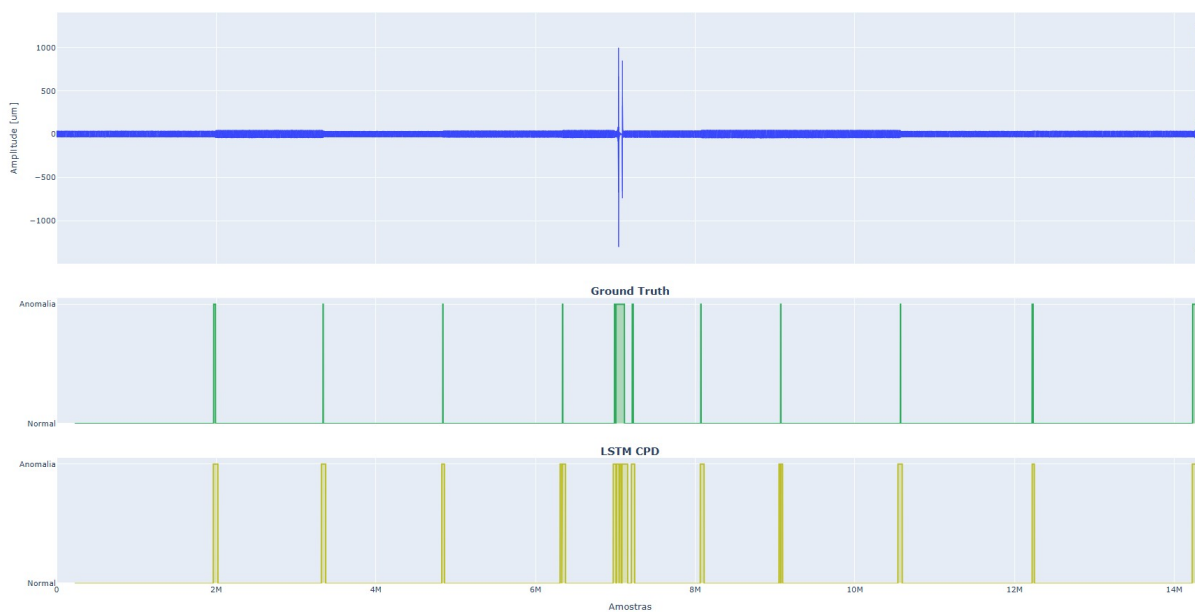


Figura 6.23: Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o *benchmark* 1.

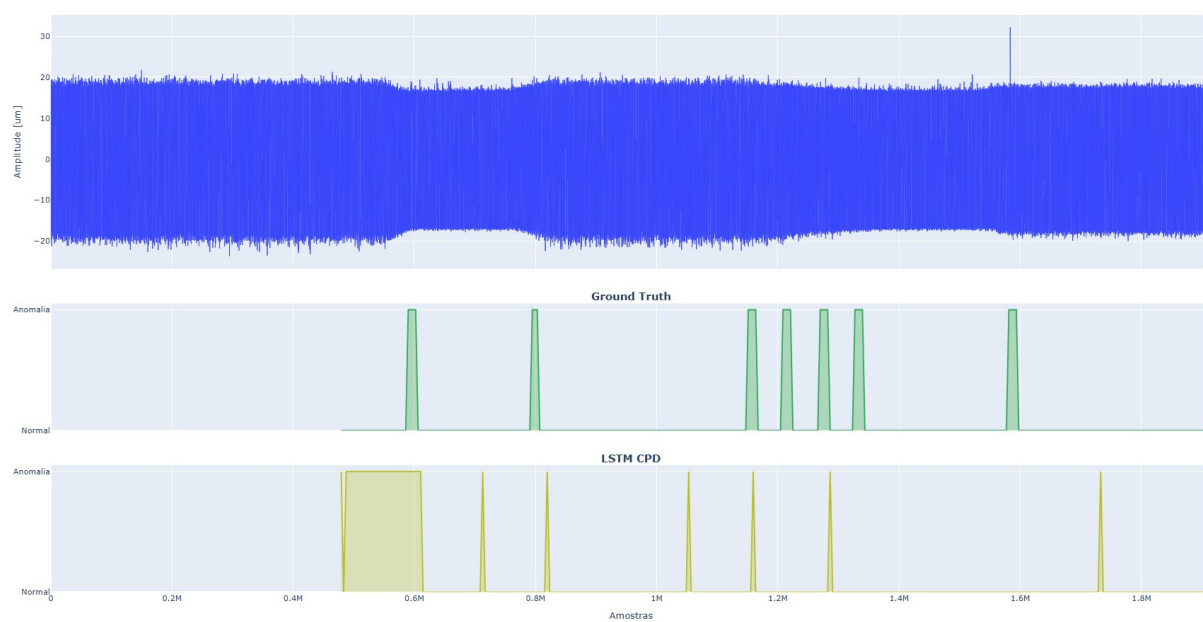


Figura 6.24: Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o *benchmark* 2.

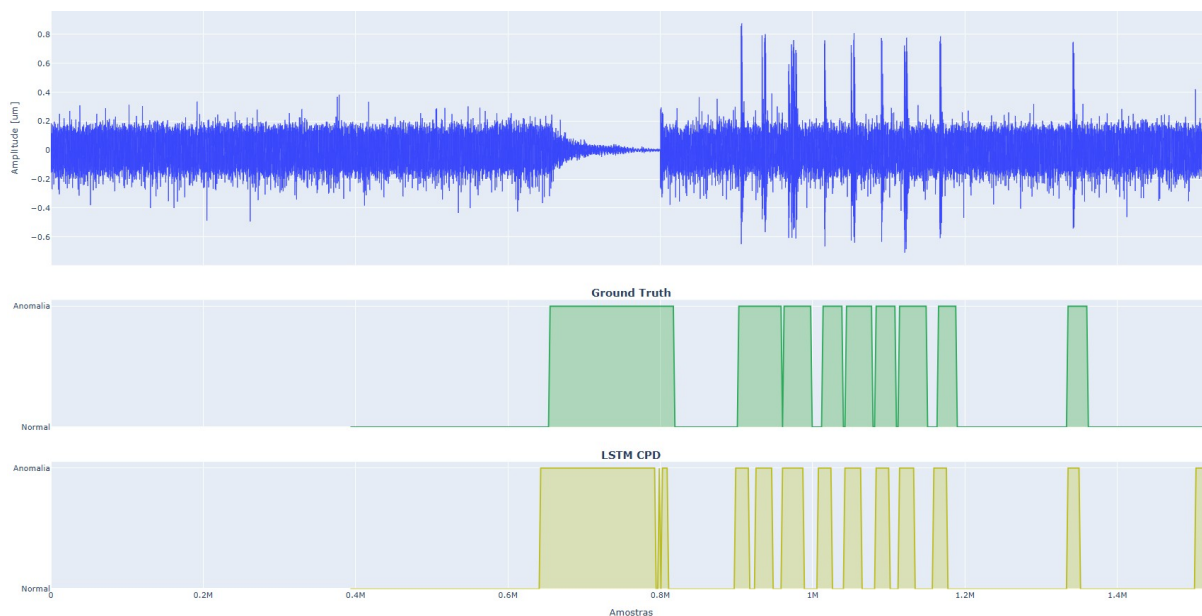


Figura 6.25: Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o *benchmark* 3.

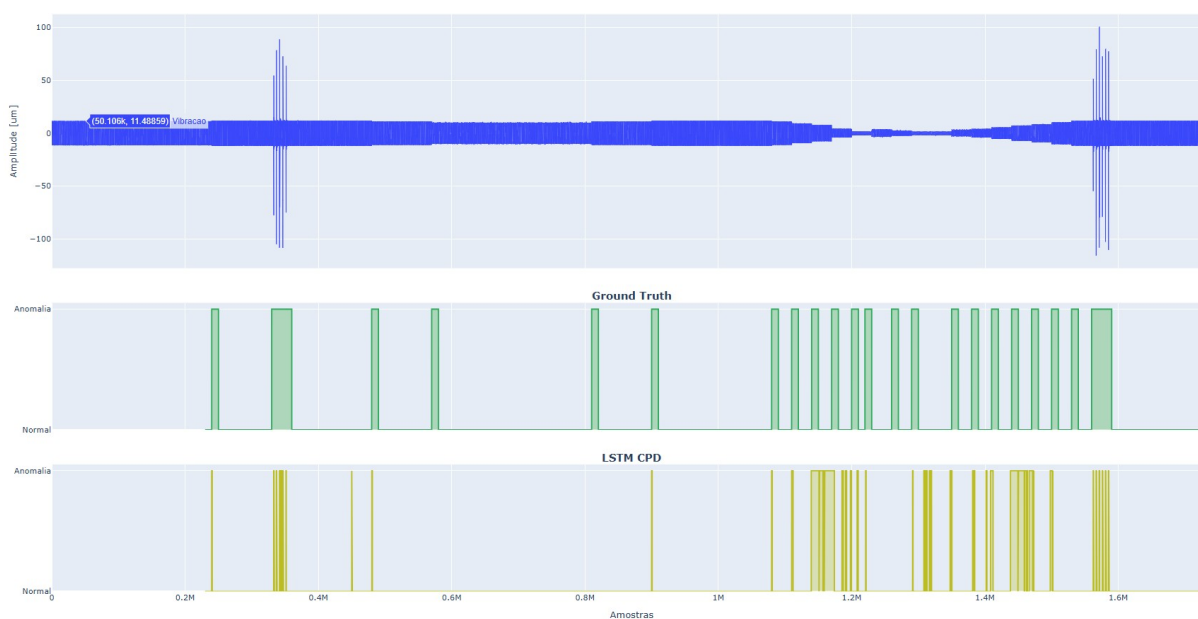


Figura 6.26: Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o *benchmark* 4.

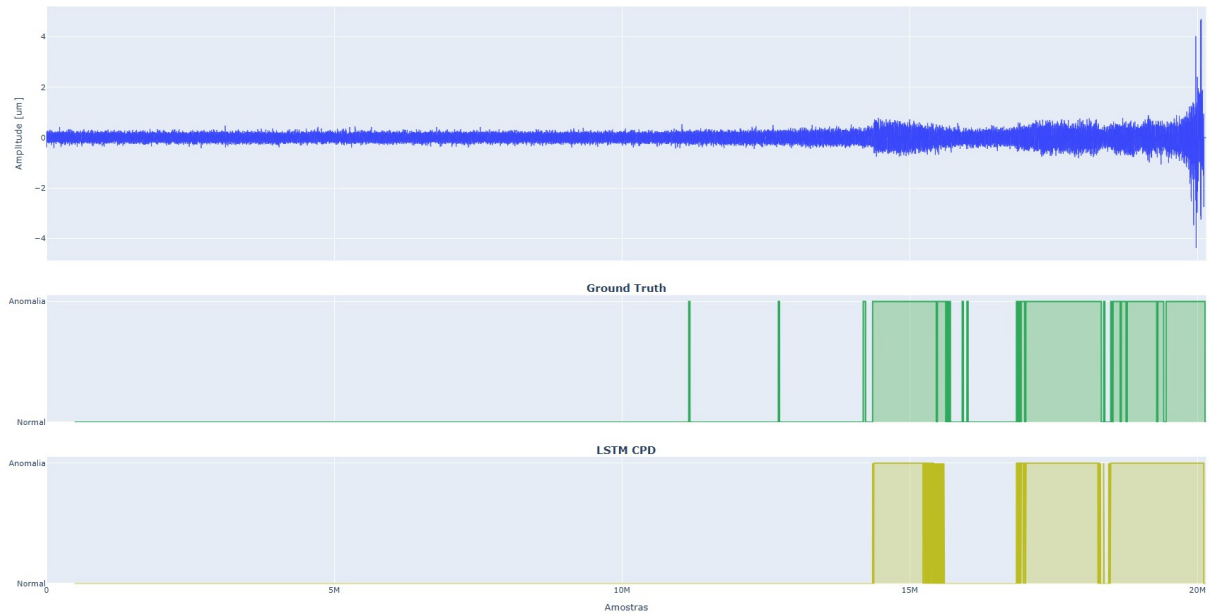


Figura 6.27: Sinal no tempo, gabarito e classificação do modelo LSTM + CPD para o *benchmark* 5.

De maneira similar à combinação MLP e CPD, a integração de LSTM com CPD leva à melhoria significativa de desempenho para os *benchmarks* 1 a 4 e manutenção de desempenho elevado para o *benchmark* 5.

6.4 Comparação de Resultados

Nesta seção são apresentados os resultados de desempenho dos modelos de ML avaliados: *DeepAnT*, *Autoencoder*, *Histogram*, MLP, LSTM, MLP + CPD, LSTM + CPD e arquitetura original *Edge Analytics*.

São apresentadas, em formato tabular, as métricas de desempenho e os valores de matriz de confusão para os cinco *benchmarks* utilizados na avaliação.

Também é apresentado um ranking comparativo dos modelos, considerando como referência a métrica de *F1-score*, que combina a precisão e o *recall*, ponderando de forma equilibrada a capacidade do modelo em identificar corretamente eventos anômalos e, simultaneamente, evitar falsos positivos. Essa métrica mostra-se particularmente adequada para o tema de estudo, uma vez que os conjuntos de dados apresentam forte desbalanceamento entre as classes normal e anômala, sendo uma característica típica do problema de detecção de falhas em máquinas rotativas. Nesse cenário, métricas como a acurácia podem levar a interpretações equivocadas do desempenho global do modelo, enquanto o *F1-score* fornece uma avaliação mais representativa

e robusta, permitindo uma comparação mais justa e consistente entre diferentes arquiteturas e estratégias de detecção de anomalias.

6.4.1 Benchmark 1

Tabela 6.13: Métricas de desempenho dos modelos para o *benchmark 1*.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Arquitetura Edge	42	10	1361	0	99	81	100	89
DeepAnT	16	12	1360	26	97	57	38	46
Autoencoder	42	10	1359	0	99	81	100	89
Histogram	42	15	1356	0	99	74	100	85
LSTM	15	0	1371	27	98	100	36	53
LSTM + CPD	42	42	1329	0	97	50	100	67
MLP	8	0	1371	34	98	100	19	32
MLP + CPD	41	33	1338	1	98	55	98	71

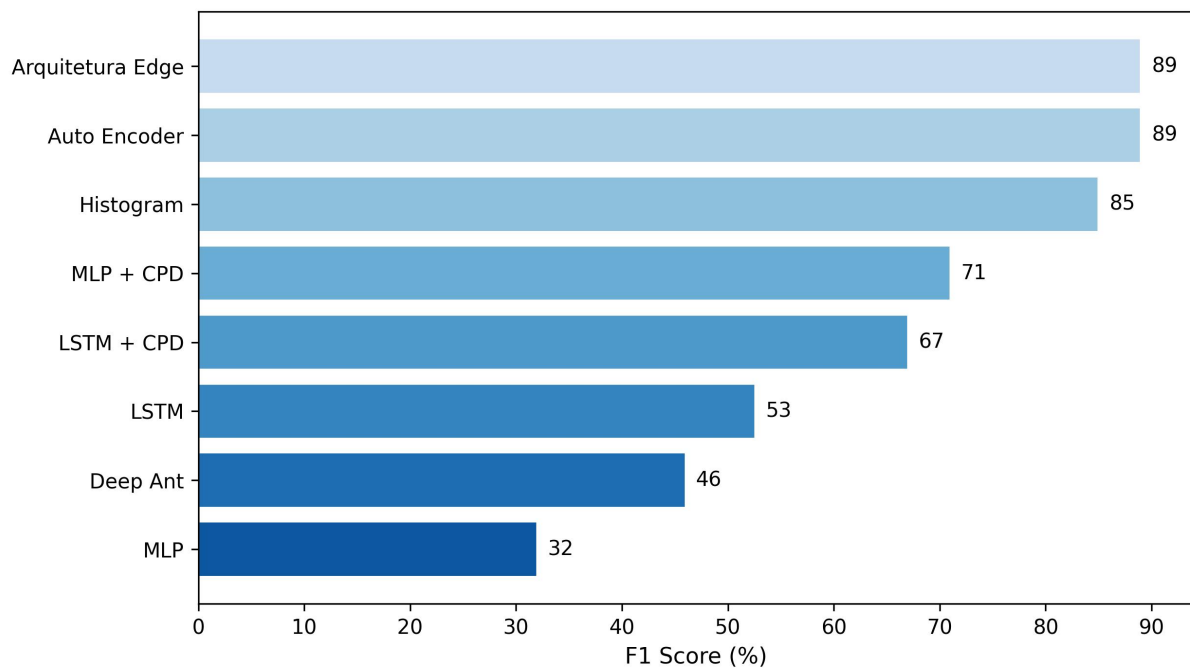


Figura 6.28: Ranking de *F1-score* dos modelos para o *benchmark 1*.

É possível observar que os melhores desempenhos são da arquitetura *Edge Analytics* e da rede *Autoencoder*, que apresentaram performances similares, levando à conclusão que esta rede, de forma individual, é capaz de ter desempenho similar à combinação de diferentes modelos. Os modelos LSTM + CPD, MLP + CPD e *histogram* obtiveram performance similar na identificação de verdadeiros positivos, porém classificando número maior de falsos positivos.

As redes LSTM, MLP e *DeepAnT* apresentaram os piores desempenhos, com baixa capacidade de identificar verdadeiros positivos.

6.4.2 Benchmark 2

Tabela 6.14: Métricas de desempenho dos modelos para o *benchmark 2*.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Arquitetura Edge	5	4	59	4	89	56	56	56
DeepAnT	3	3	61	6	88	50	33	40
Autoencoder	4	9	52	5	80	31	44	36
Histogram	7	4	58	2	92	64	78	70
LSTM	3	9	53	6	79	25	33	29
LSTM + CPD	4	9	53	5	80	31	44	36
MLP	1	0	62	8	89	100	11	20
MLP + CPD	5	5	57	4	87	50	56	53

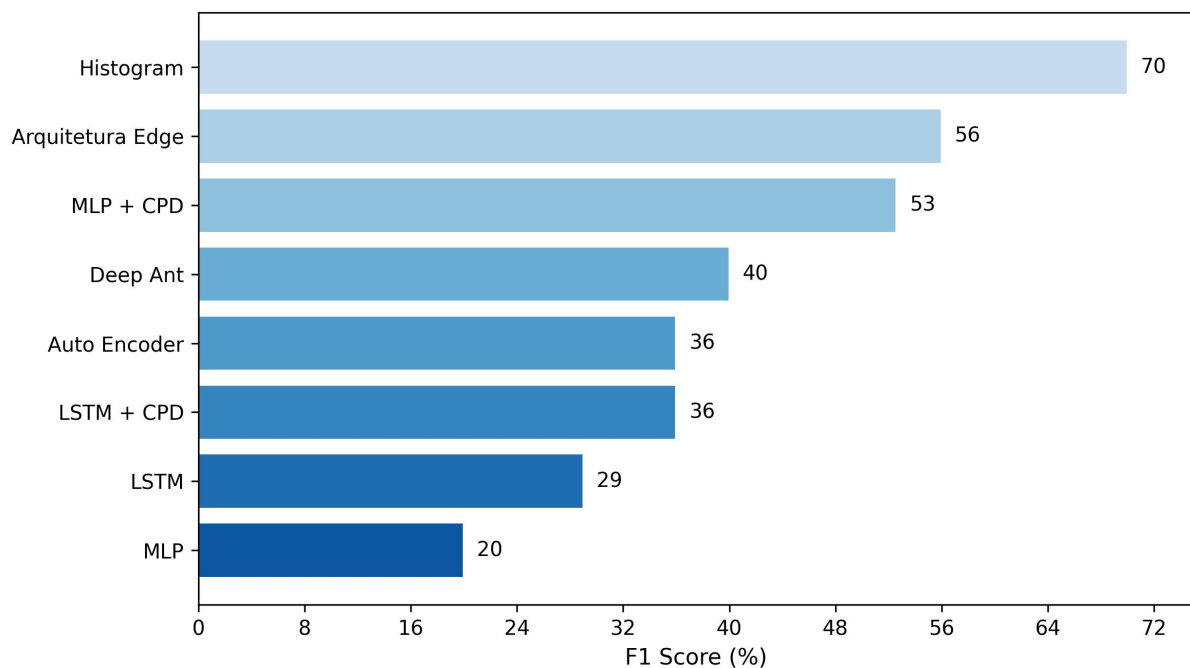


Figura 6.29: Ranking de *F1-score* dos modelos para o *benchmark 2*.

O modelo *histogram* obteve o melhor desempenho, superando inclusive o desempenho da arquitetura *Edge Analytics* que obteve uma performance bem próxima da rede MLP + CPD. LSTM e MLP apresentaram os piores desempenhos, com destaque para a rede MLP, que identificou apenas um verdadeiro positivo.

6.4.3 Benchmark 3

Tabela 6.15: Métricas de desempenho dos modelos para o *benchmark 3*.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Arquitetura Edge	26	5	34	0	92	84	100	91
DeepAnT	8	3	38	18	69	73	31	43
Autoencoder	25	6	32	1	89	81	96	88
Histogram	26	4	35	0	94	87	100	93
LSTM	18	3	36	8	83	86	69	77
LSTM + CPD	25	6	33	1	89	81	96	88
MLP	18	3	36	8	83	86	69	77
MLP + CPD	24	8	31	2	85	75	92	83

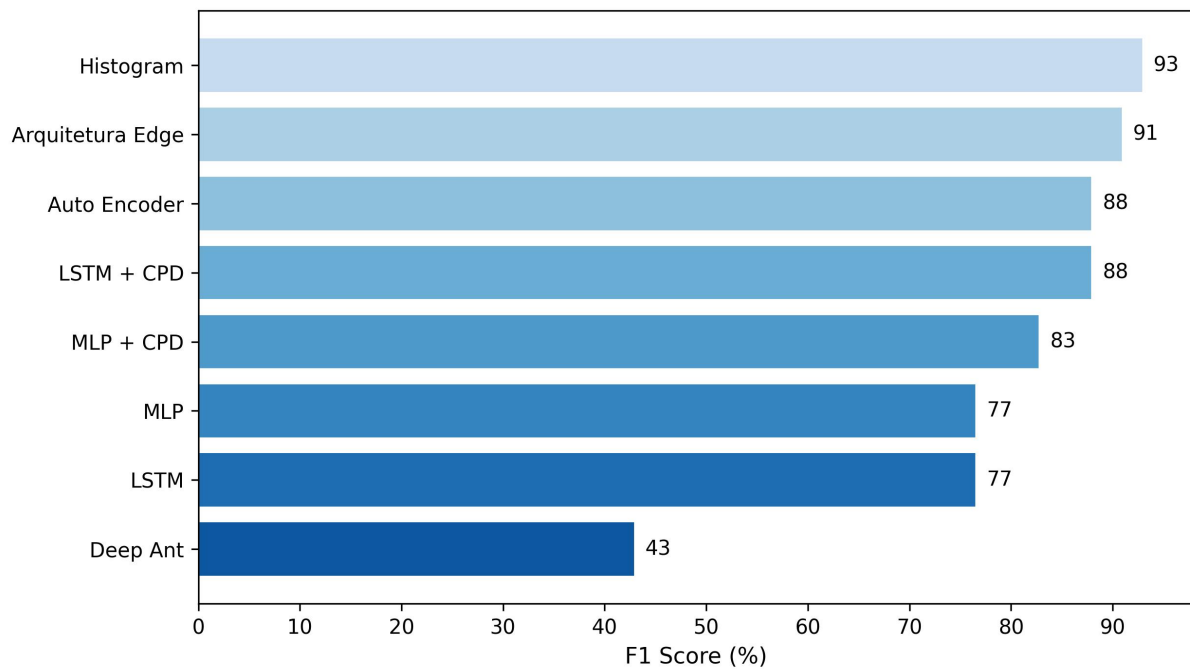


Figura 6.30: Ranking de *F1-score* dos modelos para o *benchmark 3*.

O modelo *histogram* obteve o melhor desempenho, superando ligeiramente o desempenho da arquitetura *Edge Analytics* e dos modelos *Autoencoder*, *LSTM + CPD* e *MLP + CPD*, que apresentaram desempenho elevado e resultados muito próximos. O modelo *DeepAnT* se destacou por apresentar o pior desempenho, com baixa capacidade de identificar verdadeiros positivos.

6.4.4 Benchmark 4

Tabela 6.16: Métricas de desempenho dos modelos para o *benchmark 4*.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Arquitetura Edge	24	4	121	2	96	86	92	89
DeepAnT	9	11	115	17	82	45	35	39
Autoencoder	17	2	122	8	93	89	68	77
Histogram	26	7	118	0	95	79	100	88
LSTM	13	8	117	13	86	62	50	55
LSTM + CPD	22	17	108	4	86	56	85	68
MLP	14	14	111	12	83	50	54	52
MLP + CPD	21	13	112	5	88	62	81	70

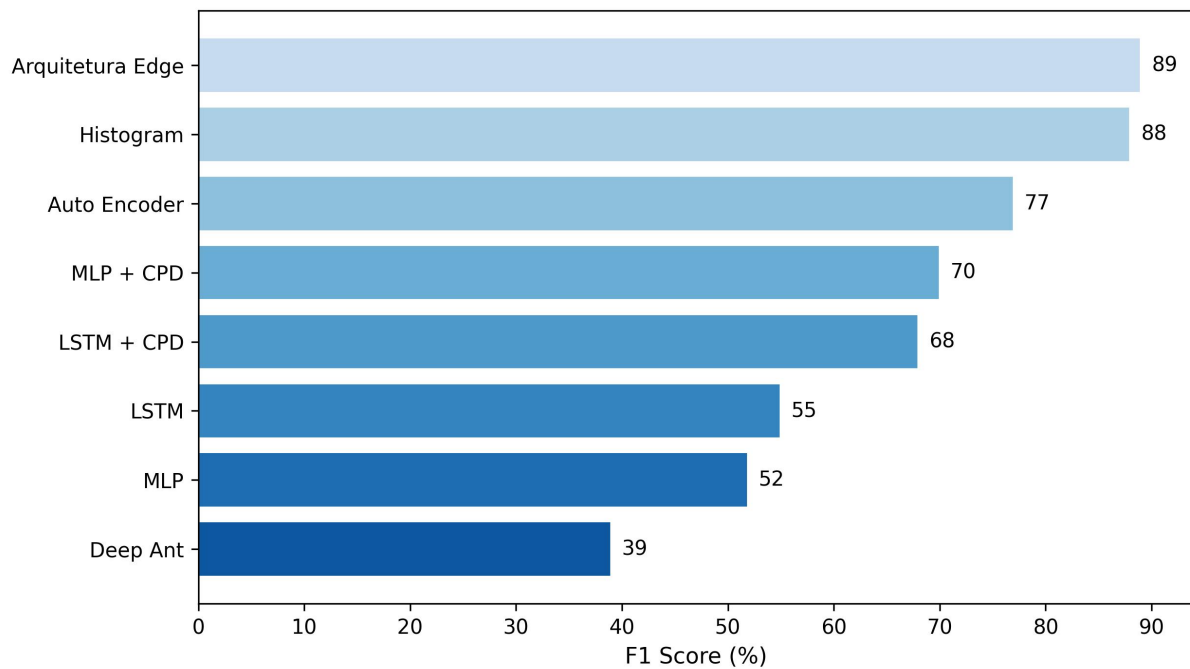


Figura 6.31: Ranking de *F1-score* dos modelos para o *benchmark 4*.

A arquitetura *Edge Analytics* obteve o melhor desempenho, sendo ligeiramente superior ao modelo *histogram*. Entre os demais modelos, a *Autoencoder* obteve a performance mais equilibrada, enquanto MLP e *DeepAnT* apresentaram os piores desempenhos, com baixa capacidade de identificar verdadeiros positivos e número elevado de falsos positivos.

6.4.5 Benchmark 5

Tabela 6.17: Métricas de desempenho dos modelos para o *benchmark 5*.

Modelo	VP	FP	VN	FN	Acurácia (%)	Precisão (%)	Recall (%)	F1-score (%)
Arquitetura Edge	56	8	739	157	83	88	26	40
DeepAnT	7	13	736	206	77	35	3	6
Autoencoder	55	10	736	158	82	85	26	40
Histogram	123	23	724	90	88	84	58	68
LSTM	204	12	735	9	98	94	96	95
LSTM + CPD	204	12	735	9	98	94	96	95
MLP	206	22	725	7	97	90	97	93
MLP + CPD	204	23	724	9	97	90	96	93

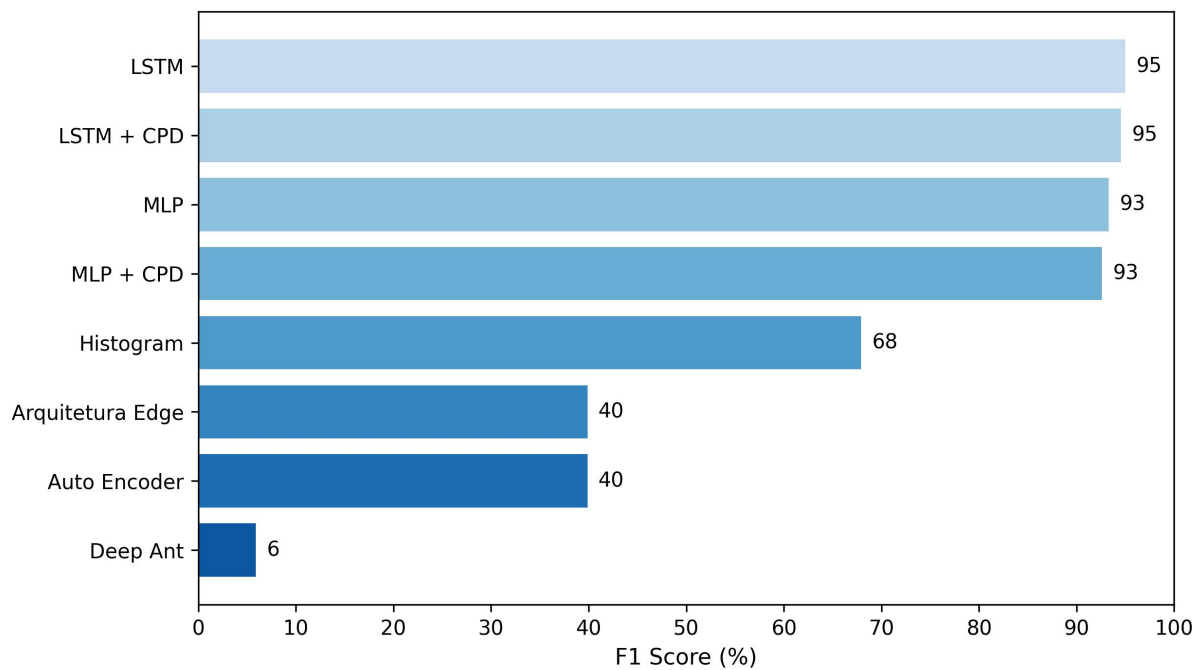


Figura 6.32: Ranking de *F1-score* dos modelos para o *benchmark 5*.

Os modelos LSTM e MLP obtiveram os melhores desempenhos, que demonstraram desempenho similar com a inclusão da técnica de CPD. Os demais modelos apresentaram performance bem inferior, com destaque negativo para a arquitetura *DeepAnT*. Uma provável hipótese para justificar estes resultados é o fato destes modelos terem sido implementados com tamanhos de amostras significativamente menores, aumentando a sensibilidade na detecção de anomalias associadas a evolução progressiva de falhas.

Em comparação com os demais modelos, a rede *DeepAnT* apresentou baixa performance para todos os *benchmarks*.

6.5 Avaliação complementar por ROC AUC

A tabela 6.18 apresenta os valores de ROC AUC obtidos para os modelos avaliados. Diferentemente do *F1-score*, que depende de um limiar específico de classificação, a ROC AUC avalia a capacidade de discriminação entre as classes ao longo de diferentes limiares de decisão.

A métrica ROC AUC foi calculada somente para os modelos individuais, uma vez que estes produzem *scores* contínuos passíveis de limiarização. A arquitetura *Edge Analytics* realiza uma decisão final baseada em votação entre camadas, não fornecendo diretamente um score contínuo único para construção de curvas ROC. O CPD foi aplicado como etapa de pós-processamento das predições, não interferindo no *score* contínuo usado no cálculo do ROC AUC. Por esse motivo, os valores de ROC AUC apresentados referem-se aos modelos individuais MLP e LSTM.

Tabela 6.18: ROC AUC dos modelos para os *benchmarks* avaliados.

Benchmark	Autoencoder	DeepAnT	Histograma	Gates	LSTM	MLP
1	0,979	0,928	0,995	0,726	0,590	0,578
2	0,757	0,516	0,947	0,638	0,430	0,492
3	0,899	0,762	0,949	0,895	0,622	0,532
4	0,941	0,699	0,964	0,835	0,644	0,568
5	0,663	0,599	0,767	0,650	0,986	0,970

De modo geral, os resultados de ROC AUC mostraram-se consistentes com as conclusões previamente obtidas a partir do *F1-score*. Os maiores valores foram observados para o modelo *Histograma*, que apresentou o melhor desempenho nos *benchmarks* 1 a 4, com ROC AUC variando entre 0,947 e 0,995. Esse resultado reforça a elevada capacidade discriminatória desse modelo para os conjuntos de dados avaliados.

Para o *benchmark* 5, representativo da degradação progressiva de rolamentos, o modelo LSTM apresentou o melhor resultado, com ROC AUC de 0,986, corroborando a superioridade já observada por meio do *F1-score*. De forma semelhante, o modelo Autoencoder apresentou valores elevados de ROC AUC em diferentes conjuntos de dados, confirmando sua robustez para a tarefa de detecção de anomalias.

Assim, *F1-score* foi considerada como principal referência deste trabalho devido ao forte desbalanceamento entre as classes, no entanto a avaliação complementar por ROC AUC confirma as principais conclusões obtidas anteriormente, indicando que os modelos que apresentaram melhor desempenho em termos de *F1-score* também possuem elevada capacidade de separação entre amostras normais e anômalas.

Capítulo 7

Considerações Finais

Nesta dissertação, investigou-se a aplicação de técnicas de *Machine Learning* para a detecção de anomalias em respostas de vibração de rotores, com ênfase em estratégias predominantemente não supervisionadas, em consonância com o cenário industrial em que dados rotulados de falha são escassos. Essa motivação e o enquadramento do problema no contexto do monitoramento por vibração são discutidos no Capítulo 1, enquanto uma revisão bibliográfica do tema é apresentada no Capítulo 2. Os conceitos essenciais de *Machine Learning*, redes neurais e métricas de desempenho adotadas são apresentados no Capítulo 3, estabelecendo a base para a avaliação experimental conduzida no restante do trabalho.

O desenvolvimento experimental foi estruturado a partir da integração e expansão do *framework Edge Analytics*, cuja arquitetura multicamadas é detalhada no Capítulo 4. Nesse capítulo, descrevem-se as camadas *Gates*, *Autoencoder*, *DeepAnT* e *Histogram*, bem como a lógica de decisão da arquitetura e a estratégia de operação em janelas deslizantes.

Os cinco *benchmarks* utilizados para validação são detalhados no Capítulo 5, que também sintetiza os fundamentos de dinâmica de rotores e de análise de sinais aplicados ao diagnóstico. O Capítulo 6 apresenta o desempenho da arquitetura *Edge Analytics* e de cada camada individual para os cinco conjuntos de dados de vibração. Os resultados consolidados e discutidos ao final do Capítulo 6 indicam que a arquitetura pode atingir *F1-score* de até aproximadamente 91%, evidenciando que a combinação de camadas é competitiva quando o perfil de anomalia é bem capturado pelos modelos de reconstrução/decisão estatística.

Complementarmente, no Capítulo 6 foram desenvolvidos e avaliados modelos adicionais baseados em MLP e LSTM, implementados em regime não supervisionado com abordagens por reconstrução e predição, respectivamente, mantendo coerência com o *pipeline* de detecção por

score de erro e limiar apresentado no Capítulo 3. A escolha da LSTM fundamenta-se em sua capacidade de modelar dependências temporais de longo prazo em séries temporais, enquanto a MLP foi escolhida pela simplicidade e baixo custo computacional.

Ainda no Capítulo 6, investigou-se a combinação de MLP e LSTM com uma etapa inspirada em *Change Point Detection* (CPD), aplicada sobre a série temporal do *score* de erro para aumentar a sensibilidade à identificação de mudanças sutis e sustentadas no comportamento vibratório. Essa estratégia foi motivada pela observação nos resultados de que certas anomalias podem não ultrapassar um limiar global de erro, mas manifestam-se como mudança estrutural de patamar no *score*, o que é compatível com a formulação clássica de CPD e sua aplicação a séries temporais.

A análise comparativa consolidada dos modelos foi apresentada no Capítulo 6, no qual se discutem, para cada *benchmark*, as métricas de acurácia, precisão, *recall* e *F1-score*, além de matrizes de confusão. Os resultados confirmam que a detecção de anomalias em vibração é fortemente dependente das características do sinal, do tipo de anomalia e do regime operacional representado em cada conjunto de dados, não havendo um modelo universalmente superior. Por exemplo, no *benchmark 1*, a arquitetura *Edge* e o *Autoencoder* apresentaram desempenho semelhante e elevado (cerca de 89%), enquanto o *DeepAnT* apresentou desempenho significativamente inferior (cerca de 46%), evidenciando limitações da configuração adotada para este conjunto. De modo análogo, no *benchmark 3*, observou-se *Histogram* e arquitetura *Edge* com *F1-score* elevado e próximo (aproximadamente 93% e 91%, respectivamente), reforçando a competitividade da decisão estatística e/ou híbrida para anomalias bem representadas por mudanças de distribuição e padrões globais do sinal.

No que se refere ao *Edge Analytics*, os resultados e a análise comparativa do Capítulo 6 indicam que a arquitetura multicamadas pode alcançar desempenho competitivo ao combinar um filtro inicial de baixo custo (*Gates*) com modelos mais expressivos (*Autoencoder* e *DeepAnT*) e uma etapa estatística de desempate (*Histogram*). Em cenários específicos, verificou-se que camadas individuais podem atingir desempenho comparável ou superior ao da combinação total. O *Histogram* apresentou melhor desempenho no *benchmark 2* e no *benchmark 3*, enquanto no *benchmark 1* a própria camada *Autoencoder* atinge *F1-score* similar ao da arquitetura completa. Esses resultados sugerem oportunidades de simplificação da arquitetura e de seleção dinâmica de camadas conforme o perfil do sinal e as restrições computacionais de implantação em *edge*.

A comparação também evidenciou limitações importantes do *DeepAnT* na configuração

adotada, com desempenho inferior em relação a outras abordagens em diferentes *benchmarks*. Em contrapartida, os modelos MLP e LSTM mostraram desempenho destacado no *benchmark* representativo de degradação de rolamentos, sugerindo que arquiteturas que exploram de forma mais direta a estrutura temporal do sinal podem ser particularmente eficazes na detecção de anomalias incipientes. No entanto, convém destacar que estes modelos consideraram tamanhos de amostras significativamente inferiores aos da arquitetura combinada que, dada a maior complexidade, implicaria em custo computacional elevado para processamento similar.

Por fim, a inclusão de CPD como etapa complementar de decisão mostrou ganhos consistentes em múltiplos *benchmarks*, ao combinar a detecção clássica por limiar global com a identificação de mudanças estruturais no comportamento do *score* de erro, reforçando a utilidade de estratégias híbridas para capturar tanto eventos abruptos quanto variações sustentadas do regime vibratório. Esses resultados indicam que técnicas de detecção de mudança podem atuar como um componente de robustez em arquiteturas de detecção de anomalias para monitoramento contínuo.

Uma limitação do presente estudo consiste no fato da etapa de otimização dos hiperparâmetros ter utilizado o gabarito dos *benchmarks* para seleção das melhores configurações. Embora essa estratégia tenha permitido avaliar o potencial máximo de cada modelo e realizar comparações consistentes entre arquiteturas, futuras investigações devem buscar metodologias de calibração completamente não supervisionadas.

Em síntese, este trabalho contribui ao: (i) consolidar e detalhar uma arquitetura *edge* multicamadas aplicada a sinais de vibração; (ii) propor e avaliar modelos adicionais (MLP e LSTM) em regime não supervisionado; (iii) demonstrar ganhos relevantes ao integrar CPD como etapa complementar de decisão; e (iv) fornecer uma análise comparativa abrangente para apoiar a seleção de modelos conforme características do sinal e restrições computacionais.

7.1 Propostas de Trabalhos Futuros

Apesar dos resultados obtidos, diversas extensões podem ser exploradas para ampliar a robustez, a generalização e a aplicabilidade industrial das metodologias desenvolvidas. As principais propostas de trabalhos futuros incluem:

- **Revisão da arquitetura *DeepAnT*:** reavaliar as escolhas de pré-processamento (por exemplo, domínio do tempo versus frequência, ou abordagens tempo-frequência), o desenho

do preditor convolucional e as estratégias de construção do *score* e calibração de limiares, buscando tornar o modelo mais sensível a padrões típicos de vibração de rotores e reduzir sua variabilidade de desempenho entre *benchmarks*.

- **Análise de generalização com hiperparâmetros unificados:** conduzir uma nova rodada de experimentos em que um mesmo conjunto de hiperparâmetros (ou faixas reduzidas) seja aplicado de forma consistente a todos os *datasets*, visando avaliar a capacidade de generalização dos modelos e reduzir a dependência de otimização individual por *benchmark*. Essa análise permitiria discutir, de forma mais objetiva, a robustez dos modelos frente à variabilidade de sinais e regimes operacionais.
- **Avaliação em um conjunto ampliado de *benchmarks*:** incorporar novos *benchmarks* com maior diversidade de máquinas, condições operacionais, tipos de falhas e incluindo dados de máquinas instaladas. Isso permitiria fortalecer as conclusões sobre aplicabilidade industrial e mapear limites de desempenho em cenários mais realistas.
- **Proposição de uma nova arquitetura *edge* baseada em modelos generalizados:** a partir dos resultados de generalização (com hiperparâmetros unificados) e do desempenho em mais *benchmarks*, propor uma arquitetura *edge* revisada, na qual a seleção/encadeamento de camadas e a lógica de decisão sejam derivadas do comportamento de modelos mais generalizáveis. Essa arquitetura pode incluir mecanismos adaptativos para escolha dinâmica do modelo (ou *ensemble* leve) conforme indicadores de regime, além de estratégias de calibração de limiar.
- **Avaliação do custo computacional e viabilidade de implantação em *edge*:** realizar uma comparação sistemática dos modelos desenvolvidos considerando consumo de memória, utilização de recursos computacionais, tempo de treinamento e tempo de inferência. Essa análise permitiria quantificar os compromissos entre desempenho de detecção e custo computacional, fornecendo critérios adicionais para seleção dos modelos em aplicações industriais de monitoramento contínuo.
- **Desenvolvimento de metodologias orientadas à diferenciação entre falhas mecânicas e mudanças de regime operacional:** investigar estratégias capazes de distinguir anomalias decorrentes de falhas mecânicas (como desbalanceamento, desalinhamento, defeitos em rolamentos e trincas) de anomalias associadas a mudanças normais de operação, como

partidas, paradas e variações de velocidade. Para isso, podem ser exploradas técnicas de incorporação de variáveis operacionais (rotação, carga e processo), aprendizado contextual e classificação hierárquica, visando aumentar a especificidade dos alarmes e reduzir a ocorrência de detecções associadas apenas a alterações operacionais esperadas.

- **Otimização de hiperparâmetros sem utilização do gabarito completo:** desenvolver uma metodologia de calibração capaz de preservar integralmente o caráter não supervisionado da abordagem, dispensando o uso direto e integral dos rótulos durante a seleção de hiperparâmetros. Uma possível abordagem consiste em segmentar os dados em conjuntos independentes de treinamento, calibração e avaliação, utilizando apenas amostras consideradas normais para ajuste dos modelos e definição dos limiares de decisão. Alternativamente, podem ser investigados critérios de otimização baseados exclusivamente em estatísticas dos erros de reconstrução ou previsão, estabilidade temporal dos scores de anomalia e métricas de detecção de mudanças não supervisionadas.
- **Avaliação dos modelos por múltiplas métricas de desempenho:** investigar o comportamento dos modelos utilizando métricas complementares ao *F1-score*, incluindo ROC AUC, *Precision-Recall AUC* e métricas específicas para cenários fortemente desbalanceados, buscando uma caracterização mais abrangente da capacidade de discriminação dos algoritmos.
- **Avaliação de técnicas de explicabilidade (*Explainable Artificial Intelligence* – XAI):** investigar métodos de explicabilidade aplicados aos modelos de detecção de anomalias, visando aumentar a interpretabilidade das decisões, facilitar a identificação dos fatores que contribuíram para a classificação de anomalias e ampliar a confiança dos usuários na aplicação industrial das soluções desenvolvidas.

Referências

AFRIDI, Y. S. et al. Lstm-based condition monitoring and fault prognostics of rolling element bearings using raw vibrational data. **Machines**, MDPI, v. 11, 5 2023. ISSN 20751702. Disponível em: <<https://doi.org/10.3390/machines11050531>>.

AHMAD, S. et al. Autoencoder-based condition monitoring and anomaly detection method for rotating machines. 1 2021. Disponível em: <<http://arxiv.org/abs/2101.11539>>.

ALAM, M. N. et al. Unsupervised machine learning for anomaly detection: A systematic review. **SSRG International Journal of Electrical and Electronics Engineering**, Seventh Sense Research Group, v. 12, p. 34–72, 7 2025. ISSN 23488379. Disponível em: <<https://doi.org/10.14445/23488379/IJEEE-V12I7P104>>.

American Petroleum Institute (API 617). **Axial and Centrifugal Compressors and Expander-Compressors for Petroleum, Chemical and Gas Industry Services**. 9ed. [S.l.], 2022.

AMRUTHNATH, N. et al. **A Research Study on Unsupervised Machine Learning Algorithms for Early Fault Detection in Predictive Maintenance**. IEEE Press, 2018. ISBN 9781538657485. Disponível em: <<https://doi.org/10.1109/IEA.2018.8387124>>.

BENTLY, D. E. et al. **Fundamentals of Rotating Machinery Diagnostics**. [S.l.], 2002.

BORGES, E. et al. **EDGE ANALYTICS: A FRAMEWORK FOR ANOMALY DETECTION IN ROTATING MACHINERY**. [S.l.], 2025.

BRITO, L. C. **Explainable Artificial Intelligence Approaches for Fault Diagnosis in Rotating Machinery**. [S.l.], 2022. Disponível em: <<https://doi.org/10.1016/j.ymssp.2021.108105>>.

CABRAL, L. **Balanceamento de Máquinas Rotativas utilizando Redes Neurais**. Tese (Doutorado) — Universidade Federal de Uberlândia, 7 2023. Disponível em: <<https://repositorio.ufu.br/handle/123456789/38769>>.

CANDY, J. V. et al. Anomaly detection for a vibrating structure: A subspace identification/tracking approach. **The Journal of the Acoustical Society of America**, Acoustical Society of America (ASA), v. 142, p. 680–696, 8 2017. ISSN 0001-4966. Disponível em: <<https://doi.org/10.1121/1.4996142>>.

CARVALHO, V. N. **Balanceamento de Máquinas Rotativas com Eixos Flexíveis utilizando Metamodelos Kriging**. [S.l.], 2022.

CAVALLINI Jr., A. A. **Detecção e Identificação de Trincas Transversais Incipientes em Eixos Horizontais Flexíveis de Máquinas Rotativas**. [S.l.], 2013.

DAS, O. et al. **Machine learning for fault analysis in rotating machinery: A comprehensive review**. Elsevier Ltd, 2023. Disponível em: <<https://doi.org/10.1016/j.heliyon.2023.e17584>>.

EGAJI, O. A. et al. **A Data Mining based Approach for Electric Motor Anomaly Detection Applied on Vibration Data**. IEEE, 2020. ISBN 9781728168234. Disponível em: <<https://doi.org/10.1109/WorldS450073.2020.9210318>>.

EMRICH, T. C. **Balanceamento de Máquinas Rotativas Usando Metamodelagem Cokriging**. [S.l.], 2026.

FACELI, K. **Inteligência Artificial: Uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2011.

FERRARIS, M. et al. **Rotordynamics Prediction in Engineering**. [S.l.]: [S.l.]: John Wiley e Sons, 1998.

GÉRON, A. **Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems**. [S.l.], 2019.

GONÇALVES, D. F. **Contribuição ao uso de inteligência artificial para detecção e diagnóstico de falhas em máquinas rotativas**. [S.l.], 2024.

GOODFELLOW, I. et al. **Deep Learning**. [S.l.]: MIT Press, 2016.

GUPTA, M. et al. **Comprehensive analysis of change-point dynamics detection in time series data: A review**. Elsevier Ltd, 2024. Disponível em: <<https://doi.org/10.1016/j.eswa.2024.123342>>.

HASTIE, T. et al. **Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction**. [S.l.], 2009. Disponível em: <<https://doi.org/10.1007/978-0-387-84858-7>>.

HAWKINS, D. M. **Identification of Outliers**. [S.l.]: Chapman and Hall, 1980.

HOCHREITER, S. et al. Hochreiter-1997 - Long Short-Term Memory. **Neural Comput**, v. 9, p. 1735, 1997. Disponível em: <<https://doi.org/10.1162/neco.1997.9.8.1735>>.

HOU, B. et al. Osesgram: Data aided method for selection of informative frequency bands for bearing fault diagnosis. **IEEE Transactions on Instrumentation and Measurement**, v. 71, 01 2022.

LEI, Y. **Intelligent fault diagnosis and remaining useful life prediction of rotating machinery**. Butterworth-Heinemann Ltd, : Xi'an Jiaotong University Press, 2017. 366 p. ISBN 9780128115343. Disponível em: <<https://doi.org/10.1016/B978-0-12-811534-3.00006-8>>.

LEI, Y. et al. **Applications of machine learning to machine fault diagnosis: A 1 review and roadmap 2**. [S.l.], 2020. Disponível em: <<https://doi.org/10.1016/j.ymssp.2019.106587>>.

LI, J. et al. Rotating machinery anomaly detection using data reconstruction generative adversarial networks with vibration energy analysis. **AIP Advances**, American Institute of Physics Inc., v. 12, 3 2022. ISSN 21583226. Disponível em: <<https://doi.org/10.1063/5.0085354>>.

LIU, Q. et al. **Feature extraction and classification algorithm, which one is more essential? An experimental study on a specific task of vibration signal diagnosis**. [S.l.], 2022. Disponível em: <<https://doi.org/10.1007/s13042-021-01477-4>>.

LIU, R. et al. **Artificial intelligence for fault diagnosis of rotating machinery: A review**. Academic Press, 2018. 33-47 p. Disponível em: <<https://doi.org/10.1016/j.ymssp.2018.02.016>>.

LIU, S. et al. Improving anomaly detection fusion method of rotating machinery based on ann and isolation forest. In: **Proceedings - 2020 International Conference on Computer Vision, Image and Deep Learning, CVIDL 2020**. Institute of Electrical and Electronics Engineers Inc., 2020. p. 581–584. ISBN 9781728194813. Disponível em: <<https://doi.org/10.1109/CVIDL51233.2020.00-23>>.

LIU, S. et al. Change-point detection in time-series data by relative density-ratio estimation. **Neural Networks**, v. 43, p. 72–83, 7 2013. ISSN 08936080. Disponível em: <<https://doi.org/10.1016/j.neunet.2013.01.012>>.

Lobão, D. A. **Técnicas de aprendizado de máquinas aplicadas ao diagnóstico de falhas em equipamentos industriais**. [S.l.], 2020.

LUPEA, I. et al. Machine learning techniques for multi-fault analysis and detection on a rotating test rig using vibration signal. **Symmetry**, MDPI, v. 15, 1 2023. ISSN 20738994. Disponível em: <<https://doi.org/10.3390/sym15010086>>.

LUTKE, K. H. **Process centrifugal compressors - basics, function, operation, design, application**. Springer, 2004.

MIAN, T. et al. Artificial intelligence of things based approach for anomaly detection in rotating machines. **Computers and Electrical Engineering**, Elsevier Ltd, v. 109, 7 2023. ISSN 00457906. Disponível em: <<https://doi.org/10.1016/j.compeleceng.2023.108760>>.

MISBAH, I. et al. Fault diagnosis in rotating machines based on transfer learning: Literature review. **Knowledge-Based Systems**, Elsevier B.V., v. 283, 1 2024. ISSN 09507051. Disponível em: <<https://doi.org/10.1016/j.knosys.2023.111158>>.

MUNIR, M. et al. Deepant: A deep learning approach for unsupervised anomaly detection in time series. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 1991–2005, 2019. ISSN 21693536. Disponível em: <<https://doi.org/10.1109/ACCESS.2018.2886457>>.

NASCIMENTO, K. B. do. **Análise do Comportamento Dinâmico de um Disco de Palhetas Flexíveis na Presença de uma Trinca Transversal Modelada pela Abordagem FLEX**. [S.l.], 2022.

OLAH, C. **Understanding LSTM Networks**. 2015. Disponível em: <<https://colah.github.io/posts/2015-08-Understanding-LSTMs>>.

OLIVEIRA, M. V. F. de. **Controle de rotores flexíveis suportados por mancais magnéticos**. [S.l.], 2019.

PASCANU, R. et al. On the difficulty of training recurrent neural networks. **Proceedings of the 30th International Conference on Machine Learning (ICML)**, 2 2013. Disponível em: <<https://arxiv.org/abs/1211.5063>>.

Pereira Neto, R. V. **Balanceamento de máquinas rotativas sem o uso de massas de teste através de redes neurais artificiais**. [S.l.], 2025.

RAI, A. et al. Bearing performance degradation assessment based on a combination of empirical mode decomposition and k-medoids clustering. **Mechanical Systems and Signal Processing**, Academic Press, v. 93, p. 16–29, 9 2017. ISSN 10961216. Disponível em: <<https://doi.org/10.1016/j.ymssp.2017.02.003>>.

RENDE, B. R. F. **Análise numérica e experimental do comportamento dinâmico de um sistema rotor-palhetas**. [S.l.], 2020.

REZENDE, S. W. F. de. **Novas contribuições na aplicação do filtro de Kalman aumentado-para o balanceamento de rotores**. [S.l.], 2025.

SAHA, D. K. et al. Development of intelligent fault diagnosis technique of rotary machine element bearing: A machine learning approach. **Sensors**, MDPI, v. 22, 2 2022. ISSN 14248220. Disponível em: <<https://doi.org/10.3390/s22031073>>.

SILVA, I. B. da. **Identificação de Trincas em Máquinas Rotativas Baseada em Técnicas de Segunda Derivada do Modo de Vibrar e Combinação de Ressonância**. [S.l.], 2023.

SINGH, V. et al. **Artificial intelligence application in fault diagnostics of rotating industrial machines: a state-of-the-art review**. Springer, 2023. 931-960 p. Disponível em: <<https://doi.org/10.1007/s10845-021-01861-5>>.

SOUZA LEÃO, L. de. **Detection and control of transverse cracks on rotating machines**. [S.l.], 2019.

STETCO, A. et al. **Machine learning methods for wind turbine condition monitoring: A review**. Elsevier Ltd, 2019. 620-635 p. Disponível em: <<https://doi.org/10.1016/j.renene.2018.10.047>>.

TAHIR, M. et al. Classification of unbalance and misalignment faults in rotor using multi-axis time domain features. In: . [S.l.: s.n.], 2016. p. 1–4.

Timbó, R. et al. Ross - rotordynamic open source software. **Journal of Open Source Software**, v. 5, n. 48, p. 2120, 2020. Disponível em: <<https://doi.org/10.21105/joss.02120>>.

WANG, Q. et al. Investigating the construction, training, and verification methods of k-means clustering fault recognition model for rotating machinery. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 8, p. 196515–196528, 2020. ISSN 21693536. Disponível em: <<https://doi.org/10.1109/ACCESS.2020.3028146>>.

WANG, Y. et al. Feature trend extraction and adaptive density peaks search for intelligent fault diagnosis of machines. **IEEE Trans. Ind. Informatics**, v. 15, p. 105–115, 2018.

WU, J. et al. Intelligent fault diagnosis of rolling bearings based on clustering algorithm of fast search and find of density peaks. **Quality Engineering**, Taylor and Francis Ltd., v. 35, p. 399–412, 2023. ISSN 15324222. Disponível em: <<https://doi.org/10.1080/08982112.2022.2140436>>.

XU, R. et al. **Change-point detection with deep learning: A review**. Higher Education Press Limited Company, 2025. 154-176 p. Disponível em: <<https://doi.org/10.1007/s42524-025-4109-z>>.

ZHANG, W. et al. Data-driven methods for predictive maintenance of industrial equipment: A survey. **IEEE Systems Journal**, Institute of Electrical and Electronics Engineers Inc., v. 13, p. 2213–2227, 9 2019. ISSN 19379234. Disponível em: <<https://doi.org/10.1109/JSYST.2019.2905565>>.