

UNIVERSIDADE NOVE DE JULHO - UNINOVE

**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA E GESTÃO DO
CONHECIMENTO**

ANDERSON FERREIRA DE LIMA

**FUSÃO DE DADOS EM ANÁLISE DE SÉRIES TEMPORAIS BINÁRIAS PARA
DETECÇÃO E LOCALIZAÇÃO DE FALHAS EM MÁQUINAS FERRAMENTAS**

SÃO PAULO

2025

ANDERSON FERREIRA DE LIMA

**FUSÃO DE DADOS EM ANÁLISE DE SÉRIES TEMPORAIS BINÁRIAS PARA
DETECÇÃO E LOCALIZAÇÃO DE FALHAS EM MÁQUINAS FERRAMENTAS**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Informática e Gestão do Conhecimento da Universidade Nove de Julho – UNINOVE, como requisito parcial para a obtenção do título de Doutor em Informática e Gestão do Conhecimento.

Prof. Dr. Fábio Henrique Pereira - Orientador

SÃO PAULO

2025

Lima, Anderson Ferreira de.

Fusão de dados em análise de séries temporais binárias para detecção e localização de falhas em máquinas ferramentas. / Anderson Ferreira de Lima. 2025.

245 f.

Tese (Doutorado) - Universidade Nove de Julho - UNINOVE, São Paulo, 2025.

Orientador (a): Prof. Dr. Fábio Henrique Pereira.

1. Máquina ferramenta. 2. Falha. 3. Manutenção preditiva. 4. Autoencoder e series temporais binárias.
- I. Pereira, Fábio Henrique. II. Título.

CDU 004

PARECER – EXAME DE DEFESA

Parecer da Comissão Examinadora designada para o exame de defesa do Programa de Pós-Graduação em Informática e Gestão do Conhecimento a qual se submeteu o aluno ANDERSON FERREIRA DE LIMA.

Tendo examinado o trabalho apresentado para obtenção do título de "Doutor em Informática e Gestão do Conhecimento", com Tese intitulada FUSÃO DE DADOS EM ANÁLISE DE SÉRIES TEMPORAIS BINÁRIAS PARA DETECÇÃO E LOCALIZAÇÃO DE FALHAS EM MÁQUINAS FERRAMENTAS, a Comissão Examinadora considerou o trabalho:

Aprovado

Aprovado condicionalmente

Reprovado com direito a novo exame

Reprovado

Examinadores

Prof. Dr. Fabio Henrique Pereira

Prof. Dr. Arthur Henrique de Andrade Melani

Prof. Dr. Walter Cardoso Sátiro

Prof. Dr. Cleber Gustavo Dias

Prof. Dr. Peterson Adriano Belan

Dedico este trabalho a
minha família.

AGRADECIMENTOS

Primeiramente agradeço a Deus pelo cuidado amoroso de todas as necessidades físicas e espirituais e por ser “refúgio e força” principalmente em épocas de aflição.

Aos meus queridos pais, João e Maria Tereza, pela sua incansável força, apoio, exemplo e dedicação.

À minhas irmãs, Solange e Kelly, meu agradecimento pelo encorajamento e apoio.

À minha namorada, Juliana, pela paciência, carinho e companheirismo.

Ao Prof. Dr. Fábio Henrique Pereira, por seu incentivo e orientação essenciais na realização deste trabalho.

À Capes pelo apoio financeiro por meio de bolsa Prosup.

Agradeço também a Universidade Nove de Julho pela oportunidade de participar deste programa de doutorado e aos docentes e funcionários pela colaboração e incentivo.

"In God we trust, all others must bring data."

W. Edwards Deming

LISTA DE FIGURAS

Figura 01 – Curva de desempenho do sistema.....	30
Figura 02 – Tipos de taxa de probabilidade de falha.....	31
Figura 03 – Curva P-F.....	33
Figura 04 – Ilustração dos posicionamentos dos sensores do <i>magazine</i>	104
Figura 05 – Ilustração das etapas de funcionamento do trocador de ferramentas..	105
Figura 06 – Dispositivo montado para simular o funcionamento de um <i>magazine</i> de ferramentas.....	107
Figura 07 – Aparato experimental montado para coletar dados.....	108
Figura 08 – Ilustração da coleta de dados em uma máquina ferramenta.....	111
Figura 09 – Ilustração da coleta de dados no aparato experimental.....	112
Figura 10 – Equivalência do funcionamento mecânico entre <i>magazine</i> e aparato experimental em condição normal.....	103
Figura 11 – Queda de ferramenta durante falha na liberação da fixação no spindle.....	114
Figura 12 – Simulação de falha no aparato experimental.....	115
Figura 13 – Simulação de falha no aparato experimental.....	116
Figura 14 – Início da degradação do sistema.....	116
Figura 15 – Abordagem proposta para análise e detecção de falhas em máquinas ferramenta.....	119
Figura 16 – Resultado da reconstrução da série no experimento 01.....	121
Figura 17 – Resultado da reconstrução da série no experimento 41.....	145
Figura 18 – Resultado da reconstrução da série no experimento 67.....	146
Figura 19 – Resultado da reconstrução da série no experimento 1.....	148
Figura 20 – Resultado da reconstrução da série no experimento 2.....	149

Figura 21 – Resultado da reconstrução da série no experimento 3.....	150
Figura 22 – Resultado da detecção de falhas nas amostras no experimento 42....	151
Figura 23 – Resultado da reconstrução da série no experimento 42.....	152
Figura 24 – Resultado da reconstrução da série no experimento 65.....	153
Figura 25 – Saída do modelo AE identificando os sensores específicos que falham no experimento 42 do capítulo 6.3.....	154
Figura 26 – Matriz de confusão do modelo AE.....	155
Figura 27 – Deteção de falha no modelo univariado.....	160
Figura 28 – Plotagem de uma janela de dados de cada variável do modelo multivariado.....	161
Figura 29 – Fluxograma de revisão da literatura I.....	234
Figura 30 – Análise temporal da distribuição dos estudos.....	236
Figura 31 – Fluxograma de revisão da literatura II.....	239
Figura 32 – Análise temporal da distribuição dos artigos.....	240
Figura 33 – Distribuição de citações dos artigos.....	241

LISTA DE TABELAS

Tabela 01 – Algoritmos utilizados na detecção de falhas e máquinas e equipamentos industriais.....	78
Tabela 02 – Metodologias de análise de séries temporais binárias.....	90
Tabela 03 – Especificações do dispositivo.....	109
Tabela 04 – Especificações do Arduino Mega.....	110
Tabela 05 – Primeiros 30 registros resultantes de um experimento.....	117
Tabela 06 – Primeiros 20 registros resultantes de uma série temporal binária aplicando-se a da técnica de fusão.....	125
Tabela 07 – Valores de limiar testados para identificação de tamanho de ciclos....	140
Tabela 08 – Parâmetros utilizados no experimento 01.....	141
Tabela 09 – Parâmetros utilizados no experimento 04.....	143
Tabela 10 – Parâmetros utilizados no experimento 67.....	145
Tabela 11 – Métricas de desempenho dos modelos.....	156
Tabela 12 – Parâmetros utilizados nos modelos.....	159
Tabela 13 – Comparação das Métricas e Adequação.....	162
Tabela 14 – Capacidade de generalização das abordagens.....	163
Tabela 15 – Protocolo de revisão da literatura I.....	233
Tabela 16 – Protocolo de revisão da literatura II.....	238

LISTA DE SÍMBOLOS

AE (Autoencoder): Autoencoder.

ANN (Artificial Neural Networks): Redes Neurais Artificiais.

BLSTM (Bi-Directional Long Short-Term Memory): Memória Longa e Curta Bidirecional.

BO (Bayesian Optimization): Otimização Bayesiana.

CBMT (Condition-Based Maintenance): Monitoramento Baseado em Condições.

CLP (Programmable Logic Controller): Controlador Lógico Programável.

CNC (Computerized Numerical Control): Comando Numérico Computadorizado.

CNN (Convolutional Neural Networks): Redes Neurais Convolucionais.

DBN (Dynamic Bayesian Networks): Redes Bayesianas Dinâmicas.

DCT (Discrete Cosine Transform): Transformada Discreta do Cosseno.

DNN (Deep Neural Networks): Redes Neurais Profundas.

DT (Decision Tree): Árvore de Decisão.

ELM (Extreme Learning Machine): Máquina de Aprendizado Extremo.

FCN (Fully Convolutional Networks): Redes Totalmente Convolucionais.

FIM (Fault Isolation Module): Módulo de Isolamento de Falhas.

FLM (Fault Location Module): Módulo de Localização de Faltas.

FNN (Feed-Forward Neural Network): Rede Neural Feed-Forward.

GA (Genetic Algorithm): Algoritmo Genético.

GAN (Generative Adversarial Networks): Redes Generativas Adversariais.

GBM (Gradient Boosting Machines): Máquinas de Boosting de Gradiente.

GGM (Gaussian Graphical Model): Modelo Gráfico Gaussiano.

GPR (Gaussian Process Regression): Regressão de Processo Gaussiano.

GRU (Gated Recurrent Unit): Unidade Recorrente Fechada.

GSA (Gravitational Search Algorithm): Algoritmo de Busca Gravitacional.

GSX (Gumbel-Sigmoid eXplanator): Explicador Gumbel-Sigmoid.

HHT (Hilbert-Huang Transform): Transformada de Huang-Hilbert.

HMM (Hidden Markov Model): Modelo de Markov Oculto.

IIoT (Industrial Internet of Things): Internet Industrial das Coisas.

KNN (K-Nearest Neighbors): K-Vizinhos Mais Próximos.

L2 (Ridge Classifier): Classificador Ridge.

LDA (Linear Discriminant Analysis): Análise Discriminante Linear.

LSTM (Long Short-Term Memory): Memória Longa e Curta.

LSTM-RNN (Long Short-Term Memory Recurrent Neural Network): Rede Neural Recorrente de Memória Longa e Curta.

MAE (Mean Absolute Error): Erro Médio Absoluto.

MFCC (Mel Frequency Cepstral Coefficients): Coeficientes Cepstrais na Frequência de Mel.

MG (Microgrid): Microrrede.

ML (Machine Learning): Aprendizagem de Máquina.

MLP (Multi-Layer Perceptron): Perceptron Multicamadas.

MLR (Multiple Linear Regression): Regressão Linear Múltipla.

mRMR (Maximum Relevance and Minimum Redundancy): Máxima Relevância e Mínima Redundância.

MSE (Mean Squared Error): Erro Quadrático Médio.

NB (Naive Bayes): Naive Bayes.

PdM (Predictive Maintenance): Manutenção Preditiva.

PNN (Probabilistic Neural Network): Rede Neural Probabilística.

PNP (Positive-Negative-Positive): Positivo-Negativo-Positivo.

PSD (Power Spectral Density): Densidade Espectral de Potência.

PSO (Particle Swarm Optimization): Otimização por Enxame de Partículas.

PWM (Pulse Width Modulation): Modulação por Largura de Pulso.

Q-learning (Quality Learning): Aprendizado de Qualidade.

RF (Random Forest): Floresta Aleatória.

RL (Logistic Regression): Regressão Logística.

RMS (Root Mean Square): Raiz Média Quadrada.

RMSE (Root Mean Squared Error): Raiz do Erro Quadrático Médio.

RNN (Recurrent Neural Networks): Redes Neurais Recorrentes.

RSSF (Wireless Sensor Networks): Redes de Sensores Sem Fio.

SAE (Sparse Autoencoder): Autoencoder Esparsos.

SVM (Support Vector Machine): Máquina de Vetores de Suporte.

SVR (Support Vector Regression): Regressão de Vetores de Suporte.

TCN (Temporal Convolutional Networks): Redes Convolucionais Temporais.

T-PLS (Total Projection to Latent Structures): Projeção Total para Estruturas Latentes.

TPR (True Positive Rate): Taxa de Verdadeiros Positivos.

USB (Universal Serial Bus): Barramento Serial Universal.

Vca (Alternating Current Voltage): Tensão em Corrente Alternada.

Vcc (Direct Current Voltage): Tensão em Corrente Contínua.

WST (Wavelet Scattering Transform): Transformada de Espalhamento Wavelet.

XGBoost (Extreme Gradient Boosting): Boosting de Gradiente Extremo.

UNIDADES

<i>A</i>	Amperes
<i>Hz</i>	Hertz
<i>rpm</i>	rotações por minuto
<i>V</i>	Volts
<i>W</i>	Watt
Kgf/cm	quilograma força por centímetro
<i>k</i>	quilo (10^3)
<i>M</i>	mega (10^3)
bar	pressão

RESUMO

As máquinas ferramentas com controle numérico computadorizado (CNC) são essenciais nos processos de fabricação modernos devido à sua alta precisão e velocidade. No entanto, essas características estão sujeitas a falhas ocultas e deterioração, reduzindo a confiabilidade operacional. Para aumentar o tempo médio entre falhas, são utilizados dispositivos de detecção que monitoram o estado operacional dos componentes da máquina, permitindo a análise e processamento de informações para maximizar a confiabilidade. Muitas indústrias adotam a manutenção preditiva (PdM), que utiliza monitoramento inteligente para evitar falhas futuras. Nos últimos anos, o aprendizado de máquina (ML) tem se mostrado fundamental na previsão de falhas, contribuindo para a PdM. Técnicas de ML baseadas em dados de sensores, incluindo abordagens de aprendizagem profunda como Autoencoders (AE), têm mostrado resultados promissores. A análise de dados de séries temporais de sensores, especialmente dados binários, tem recebido atenção significativa, permitindo a detecção eficiente de falhas ou atividades anormais e facilitando ações de manutenção imediatas. Este trabalho visa desenvolver métodos precisos e eficientes de detecção de falhas para melhorar a confiabilidade e eficiência dos sistemas de produção industrial. A pesquisa inclui uma análise abrangente dos modelos de ML para detecção de falhas, focando em dados de séries temporais binárias. Além de avaliar o desempenho dos modelos, foi desenvolvido um dispositivo para capturar dados dos sensores e um aparato experimental para simular dados de falha com segurança. Esta tese propôs uma abordagem inovadora que combina a fusão de séries temporais binárias multivariadas com modelos autoencoder, demonstrando, por meio de experimentos práticos com dados reais, que é possível detectar falhas em máquinas-ferramenta com alta precisão, ao mesmo tempo em que contribui teoricamente para o avanço da análise de séries binárias no contexto da manutenção preditiva.

Palavras-chave: máquina ferramenta, falha, manutenção preditiva, autoencoder e séries temporais binárias

ABSTRACT

Computer Numerical Control (CNC) machine tools are essential in modern manufacturing processes due to their high precision and speed. However, these characteristics are susceptible to hidden faults and degradation, reducing operational reliability. To increase the Mean Time Between Failures (MTBF), detection devices are used to monitor the operational state of machine components, enabling the analysis and processing of information to maximize reliability. Many industries adopt Predictive Maintenance (PdM), which uses intelligent monitoring to prevent future failures. In recent years, Machine Learning (ML) has proven to be fundamental in failure prediction, contributing significantly to PdM. ML techniques based on sensor data, including deep learning approaches such as Autoencoders (AE), have shown promising results. The analysis of sensor time series data, especially binary data, has received significant attention, enabling efficient detection of faults or abnormal activities and facilitating immediate maintenance actions. This work aims to develop accurate and efficient fault detection methods to improve the reliability and efficiency of industrial production systems. The research includes a comprehensive analysis of ML models for fault detection, focusing on binary time series data. In addition to evaluating model performance, a device was developed to capture sensor data and an experimental apparatus was built to safely simulate fault conditions. This thesis proposed an innovative approach that combines the fusion of multivariate binary time series with autoencoder models, demonstrating—through practical experiments with real data—that it is possible to detect faults in machine tools with high precision, while also contributing theoretically to the advancement of binary time series analysis in the context of predictive maintenance.

Keywords: machine tool, fault, predictive maintenance, autoencoder, binary time series

SUMÁRIO

1. INTRODUÇÃO	18
1.1 IDENTIFICAÇÃO DE LACUNAS.....	20
1.2 OBJETIVOS	22
1.2.1 Objetivos Específicos	22
1.3 JUSTIFICATIVA.....	22
1.4 DELIMITAÇÕES	25
1.4.1 Cenário de Aplicação	25
1.4.2 Simplificações no Tratamento de Dados	25
1.4.3 Integração com a Interface de Coleta de Dados	26
1.4.4 Exclusão de Abordagens Tradicionais de Validação	26
1.4.5 Limitações e Perspectivas Futuras para Aplicações em Sistemas Embarcados	26
1.5 ESTRUTURA DA TESE	27
2. FUNDAMENTAÇÃO TEÓRICA	29
2.1 CONCEITUAÇÃO DE FALHA	29
2.2 CONCEITUAÇÃO DE MANUTENÇÃO	30
2.3 CONCEITUAÇÃO DE SÉRIES TEMPORAIS.....	34
2.4 CONCEITUAÇÃO DE APRENDIZADO DE MÁQUINA	35
2.5 APRENDIZADO PROFUNDO	39
2.6 APRENDIZADO DE MÁQUINA APLICADO A MANUTENÇÃO DE MÁQUINAS E EQUIPAMENTOS.....	41
2.7 AUTOENCODER	44
2.7.1 Fundamentos do autoencoder.....	47
2.7.2 Aplicação de Autoencoders	51
2.7.3 Detalhes de Implementação	53
2.7.4 Otimizadores	56
2.7.5 Hiperparâmetros	58
2.7.6 Regularização	59
2.8 AUTOENCODERS LSTM	60
2.8.1 Treinamento de Autoencoders LSTM	61
2.8.2 Aplicações de Autoencoders LSTM	62
2.8.3 Detalhes de Implementação	63
3. REVISÃO DA LITERATURA	64

3.1 ALGORITMOS UTILIZADOS EM ANÁLISE E DETECÇÃO DE FALHAS EM MÁQUINAS E EQUIPAMENTOS INDUSTRIAIS	64
3.1.1 Análise e detecção de padrões em séries temporais binárias	82
3.1.2 Abordagens estatísticas	84
3.1.3 Abordagens por algoritmos clássicos de aprendizado de máquina	85
3.1.4 Abordagens por aprendizado profundo	87
3.2 ANÁLISE DE SÉRIES TEMPORAIS MULTIVARIADAS E APLICAÇÕES EM AMBIENTES INDUSTRIAIS	96
4. AMBIENTE DE DESENVOLVIMENTO E ENSAIO DO MODELO PROPOSTO	104
4.1 CARACTERÍSTICAS PRINCIPAIS DO AMBIENTE DE ENSAIO	105
4.2 COLETA DE DADOS	109
4.3 DETALHAMENTO DOS CONJUNTOS DE DADOS	111
4.4 COLETA DE DADOS NO APARATO EXPERIMENTAL	114
4.5 DELIMITAÇÃO DO ESCOPO DOS DADOS	118
5. MATERIAIS E MÉTODOS	120
5.1 CARATERIZAÇÃO METODOLÓGICA	120
5.2 DETECÇÃO DE FALHAS	120
5.2.1 Passo 1 - Coleta dos dados	122
5.2.2 Passo 2 - Transformação dos dados	124
5.2.3 Passo 3 - Detecção da falha	125
5.2.4 Passo 4 - Isolamento da falha	132
5.2.5 Desempenho	136
6. RESULTADOS	138
6.1 IDENTIFICAÇÃO DO TAMANHO DE CICLOS DE SÉRIES TEMPORAIS UTILIZANDO AUTOCORRELAÇÃO	138
6.2 TREINAMENTO DO MODELO	140
6.3 DETEÇÃO DE FALHAS	147
6.3.1 Localização de falhas	153
6.4 MÉTRICAS DE DESEMPENHO	154
6.5 TESTES COMPARATIVOS ENTRE AUTOENCODER PARA SÉRIES TEMPORAIS UNIVARIADAS E MULTIVARIADAS	158
6.6 EXPERIMENTOS COM ABORDAGEM ESTATÍSTICA	165
6.7 EXPERIMENTOS COMPARATIVOS ENTRE AE, CNN E RNN NA DETEÇÃO DE FALHAS	169
6.7.1 Metodologia dos experimentos	169
6.7.2 Resultados dos experimentos	171

6.7.3 Conclusões dos experimentos	173
7. DISCUSSÕES	175
7.1 ANÁLISE COMPARATIVA COM ESTUDOS RECENTES	175
7.2 AVANÇO PROPORCIONADO PELO MÉTODO DESENVOLVIDO	182
7.3 APLICABILIDADE E EFICÁCIA PRÁTICA DAS FERRAMENTAS E ABORDAGENS EM CENÁRIOS REAIS.....	183
8. CONCLUSÕES	185
9. REFERÊNCIA BIBLIOGRÁFICAS	188
APÊNDICE I – PROTOCOLO DE REVISÃO DA LITERATURA I	232
SELEÇÃO DOS ARTIGOS PARA IDENTIFICAR AS APLICAÇÕES DE APRENDIZADO DE MÁQUINA EM PREDIÇÃO DE FALHAS	232
ANÁLISE DA DIVERSIDADE DE ALGORITMOS APLICADO A PREDIÇÃO DE FALHAS EM MÁQUINAS E EQUIPAMENTOS.	235
APÊNDICE II – PROTOCOLO DE REVISÃO DA LITERATURA II	238
SELEÇÃO DOS ARTIGOS SOBRE SÉRIES TEMPORAIS BINÁRIAS	238
ANÁLISE DE SÉRIES TEMPORAIS BINÁRIAS.....	240
ANÁLISE DAS DESCRIÇÕES DOS DADOS	241
TÉCNICAS DE ANÁLISE DE SÉRIES TEMPORAIS BINÁRIAS	242
ANÁLISE DOS OBJETIVO DOS ARTIGOS	244

1. INTRODUÇÃO

As máquinas ferramentas equipadas com controle numérico computadorizado (CNC) fazem parte da base dos processos de usinagem e fabricação modernos das indústrias de manufatura. Por serem máquinas de alta tecnologia, possuem altas taxas de precisão e velocidade, aspectos estes que por estarem sobre os riscos de falhas ocultas e deterioração, fazem com que os índices de desempenho não possam ser mantidos e a confiabilidade da operação é gradualmente reduzida (ZHANG et al., 2019; JIN et al., 2023).

Para aumentar o tempo médio entre falhas (MTTF), é essencial empregar dispositivos de monitoramento que permitam acompanhar em tempo real as condições operacionais dos componentes da máquina, possibilitando a detecção precoce de degradações. Dessa forma, as informações coletadas podem ser analisadas e processadas para antecipar possíveis falhas e otimizar a confiabilidade do equipamento. A manutenção preditiva (PdM), que faz uso de tais tecnologias para monitorar e prever falhas antes que ocorram, tem sido uma escolha comum em muitas indústrias para maximizar o MTTF e evitar interrupções inesperadas (KUNZER et al., 2022).

A manutenção preditiva se esforça para maximizar a disponibilidade de máquinas e equipamentos. Na última década, o aprendizado de máquina (ML) começou a desempenhar um papel fundamental no domínio de prever falhas em máquinas e assim contribuir para a PdM (KANE et al., 2022; WAHID; BRESLIN; INTIZAR, 2022). Amplas abordagens foram propostas para explorar o ML baseado em dados de sensores. Recentemente, também abordagens de aprendizagem profunda foram utilizadas, incluindo Autoencoders (AE), mostrando resultados promissores (KHOSRAVINIA, 2023).

Em virtude disto, a análise de dados de séries temporais de sensores em domínios industriais ganhou atenção significativa. Dentre estes, os dados binários de série temporal são uma classe especial que são gerados em grandes volumes, especialmente em máquinas ferramentas. E o ML se apresenta como uma tecnologia promissora para a análise e deteção de falhas em séries temporais binárias. Os modelos de ML podem detectar com eficiência falhas ou atividades anormais,

facilitando ações imediatas de manutenção para mitigar possíveis interrupções na produção (PRINCZ; SHALOO; EROL, 2024).

O desenvolvimento de métodos precisos e eficientes de detecção de falhas pode melhorar muito a confiabilidade e eficiência de sistemas de produção industrial (WANG et al., 2022). É aqui que este trabalho pretende começar e fornecer uma abordagem prática no campo da manutenção baseada em dados. Este trabalho conduzirá uma análise detalhada dos principais modelos de ML aplicados à detecção de falhas, focando nas suas vantagens e desvantagens, com o objetivo de identificar lacunas de pesquisa em séries temporais binárias de máquinas ferramenta.

O foco não foi apenas na construção de modelos, mas também no desenvolvimento de uma interface para capturar os dados dos sensores das máquinas ferramentas e desenvolvimento de um aparato experimental para simular dados de falha com segurança. Para avaliação do modelo foram coletados dados do aparato experimental montado com sensores industriais. Uma variedade de falhas foi implementada em experimentos, para os quais o modelo foi projetado para detectar. Por fim, os resultados do modelo foram contrastados com os experimentos e, assim, comparados entre si.

Observou-se que a dependência temporal das séries temporais apresenta uma complexidade elevada. A análise e detecção de falhas em séries temporais multivariadas exigem a captura tanto das dependências temporais dentro de cada série temporal quanto das intercorrelações entre diferentes pares de séries temporais. (ILIOPOULOS et al., 2023; CHEN; FENG; WIRJANTO, 2023). Em contraste, observou-se que os trabalhos identificados pela revisão da literatura, utilizam em sua maioria a análise de séries temporais univariadas que simplifica o problema ao se concentrar em uma única variável, potencialmente negligenciando importantes dependências cruzadas.

Com base na revisão da literatura, não foram identificados estudos que abordem o funcionamento de máquinas ferramenta e examinem as causas de falhas utilizando dados de séries temporais binárias multivariadas capturadas diretamente dos sensores das máquinas, corroborando a conclusão de Princz, Shaloo e Erol

(2024), que também ressaltam a necessidade de aprofundar as investigações sobre métodos de detecção de falhas com base em dados de séries temporais binárias multivariadas capturados diretamente dos sensores das máquinas. Além disso, não foram encontrados registros de pesquisas que explorem o uso de modelos de aprendizado profundo para cobrir todo o processo, desde a captura dos dados até a análise, identificação e localização das falhas.

A partir da revisão da literatura realizada, foi possível identificar algumas abordagens para a análise de séries temporais multivariadas, incluindo métodos baseados em redes neurais, modelos probabilísticos e algoritmos de inferência de topologia. No entanto, nenhum dos métodos encontrados na literatura se mostrou diretamente aplicável aos dados coletados de sensores de máquinas ferramenta.

Este trabalho propõe uma abordagem teórica inovadora para a transformação de séries temporais binárias multivariadas e demonstra sua aplicação prática na detecção e localização de falhas em máquinas-ferramenta, contribuindo para a melhoria da manutenção preditiva.

Os experimentos realizados comparando a técnica desenvolvida nesta tese aplicada a três modelos diferentes para detecção de falhas em séries temporais binárias, mostraram que o Autoencoder Long Short-Term Memory (LSTM) apresentou o melhor desempenho geral, além de minimizar falsos positivos. No contexto da manutenção preditiva, a minimização de falsos positivos é um fator muito importante. Reduzir falsos alarmes significa que menos componentes da máquina precisarão ser desmontados para verificação, diminuindo custos operacionais e o tempo de inatividade. Embora falsos negativos possam resultar na não detecção de falhas reais, estratégias de monitoramento contínuo podem mitigar esse risco.

1.1 IDENTIFICAÇÃO DE LACUNAS

A crescente integração de tecnologias avançadas no ambiente industrial tem impulsionado a busca por métodos inovadores para otimizar a operação e manutenção de máquinas ferramentas (KUNZER *et al.*, 2022). Nesse contexto, as lacunas de pesquisa que motivaram este trabalho incluem:

- Desafios na detecção de falhas em máquinas ferramenta: A necessidade de lidar com a crescente complexidade dos dados gerados por sensores e atuadores em máquinas ferramenta, onde padrões anômalos podem ser difíceis de identificar usando métodos estatísticos convencionais (PRINCZ; SHALOO; EROL, 2024).
- Limitações dos métodos convencionais: A falta de eficácia dos métodos convencionais de detecção de falhas em lidar com a diversidade e complexidade dos dados de séries temporais binárias, destacando a necessidade de abordagens mais avançadas e adaptáveis (PAN *et al.*, 2022; BODNER; CEULEMANS, 2023).
- Escassez de pesquisas em detecção de falhas em dados binários: A falta de estudos abordando especificamente a detecção de falhas em séries temporais binárias, deixando uma lacuna no conhecimento sobre a aplicação de modelos de aprendizado de máquina não supervisionado nesse contexto (PRINCZ; SHALOO; EROL, 2024).
- Necessidade de avaliação de modelos de aprendizado de máquina: A importância de avaliar e comparar o desempenho de diferentes modelos de aprendizado de máquina não supervisionado para detecção de falhas, a fim de identificar suas forças, fraquezas e aplicabilidade em cenários industriais (KHOSRAVINIA, 2023; LI *et al.*, 2023).

Como Princz, Shaloo e Erol (2024) também discutem, durante a pesquisa descobriu-se que uma ampla gama de modelos de aprendizado de máquina são usados para detecção de falhas em diferentes áreas, incluindo áreas em que grandes quantidades de dados podem ser coletadas dos sensores. Com base na literatura encontrada, não foi identificado nenhum estudo que trate do funcionamento de máquinas ferramentas e examine a causa da falha usando dados de séries temporais binárias, capturadas dos próprios sensores da máquina, através de modelos de aprendizado profundo, e que descreva desde a captura dos dados até a análise e identificação da falha.

Essas lacunas de pesquisa destacam a motivação por trás deste trabalho, buscando preencher essas lacunas e contribuir para o avanço do monitoramento de

condições de máquinas ferramentas por meio de técnicas inovadoras de análise e detecção de falhas em dados de séries temporais binárias.

1.2 OBJETIVOS

O objetivo principal é utilizar técnicas de aprendizado de máquina para identificar padrões nos dados gerados pela ativação e desativação de sensores e detectar e localizar falhas em máquinas e equipamentos industriais. O estudo visa melhorar a capacidade de detecção de falhas e otimizar o desempenho dos sistemas de produção por meio do desenvolvimento de uma técnica de fusão de dados, ao qual seja possível analisar séries temporais binárias multivariadas de forma simultânea e simplificada.

1.2.1 Objetivos Específicos

- Desenvolver uma interface eletrônica para capturar os dados dos sensores das máquinas ferramentas.
- Desenvolver um aparato experimental, composto por uma estrutura metálica projetada para ativar e desativar sensores de forma controlada, a fim de simular dados de falha em experimentos.
- Desenvolver modelos de previsão baseados em redes neurais Autoencoder, Redes Neurais Convolucionais (CNN) e Redes Neurais Recorrentes (RNN).
- Desenvolver uma técnica para análise de séries temporais binárias multivariadas.
- Aplicar os modelos em bases de dados reais coletadas em máquinas ferramentas e no aparato experimental para verificar a acurácia dos modelos.
- Contrastar os resultados dos modelos.
- Analisar o efeito da técnica de fusão de dados proposta em contraste a análise direta de séries temporais binárias multivariadas.

1.3 JUSTIFICATIVA

Na indústria, a manutenção das máquinas ferramentas é um dos pontos importantes pois afeta o tempo de operação e a eficiência dos equipamentos,

portanto, as falhas nesse tipo de equipamentos precisam ser identificadas e resolvidas para eliminar paradas no processo produtivo (WAN *et al.*, 2017; CHEN *et al.*, 2022).

Apesar das máquinas ferramentas serem equipadas com um Controle Numérico Computadorizado, e um Controlador Lógico Programável (CLP) que opera com uma programação. Estes recursos têm a função de monitorar e avaliar se a ativação e desativação dos sinais dos sensores ocorreram dentro de uma margem de tolerância aceitável, em relação ao momento esperado para cada evento. A margem de tolerância, que varia de acordo com a finalidade e sensibilidade de cada sensor, pode ser da ordem de milissegundos ou segundos, dependendo do processo específico que está sendo controlado.

Essa lógica de programação permite que o algoritmo do CLP detecte falhas na máquina ferramenta apenas no exato momento em que elas ocorrem, acionando uma resposta imediata para impedir danos maiores. Contudo, esse método de monitoramento reativo não antecipa a degradação progressiva do sistema. A análise das séries temporais binárias geradas pelos sensores oferece uma abordagem mais proativa, permitindo identificar padrões que indicam o início de uma possível falha antes que ela cause uma parada inesperada.

Além disso, esse método possibilita a identificação de falhas esporádicas, que podem passar despercebidas pelo monitoramento tradicional, e auxilia na localização precisa do componente defeituoso, sem a necessidade de desmontar a máquina inteira em busca da falha. Tudo isso é possível porque os modelos de detecção foram previamente treinados com dados da respectiva máquina em condição normal de funcionamento, permitindo uma comparação precisa entre o comportamento esperado e a situação atual da máquina.

Dessa forma, a degradação gradual ou falhas intermitentes do sistema podem ser detectadas e tratadas previamente, otimizando a manutenção preventiva e evitando tanto paradas inesperadas quanto a desmontagem desnecessária do equipamento.

Ao aplicar métodos analíticos baseados em dados, podem ser encontrados resultados interpretativos para decisões estratégicas, proporcionando benefícios como redução de custos de manutenção, redução de tempo de inatividade para manutenção, redução de quebras de máquinas, redução de estoque de peças de

reposição, aumento da produção, vida útil prolongada de peças de reposição e maior segurança para o operador. (SEZER *et al.*, 2018; ZHANG *et al.*, 2022).

Quanto ao desenvolvimento de modelos baseados em Autoencoder, justifica-se por representarem uma abordagem avançada para a previsão de falhas, e a aplicação destes em conjunto com o desenvolvimento da técnica de transformação de séries temporais binárias multivariadas permite explorar representações latentes de todos os canais de sinais de forma simultânea, a cada *timestamp*, potencialmente melhorando a precisão das estimativas de falhas.

Referente a validação dos modelos com dados reais provenientes de máquinas ferramentas, juntamente com dados obtidos de um aparato experimental, permite avaliar a robustez e a eficácia dos modelos em condições práticas. Ao treinar os modelos com esses dados, torna-se possível comparar os resultados obtidos pelo modelo com os dados experimentais e as condições reais das máquinas, permitindo verificar se o modelo foi capaz de identificar corretamente a falha inserida propositalmente em determinado ponto. Essa abordagem possibilita uma avaliação mais precisa do desempenho dos modelos, garantindo que eles sejam eficazes na detecção de falhas em cenários reais.

Dessa forma, a execução desse trabalho não apenas ampliará o conhecimento teórico sobre técnicas de previsão de falhas em dados de séries temporais binárias, mas também fornecerá contribuições práticas e aplicáveis ao monitoramento de máquinas ferramentas, com potencial impacto positivo na eficiência operacional e na manutenção preditiva desses equipamentos.

Além disso, na revisão da literatura realizada, embora diversos estudos abordem a aplicação de técnicas de aprendizado de máquina para detecção de falhas em séries temporais, nenhum dos artigos identificados considerava a criação de modelos especificamente otimizados para aplicações embarcadas e soluções em tempo real. Essa lacuna é particularmente relevante, uma vez que a implementação de modelos eficientes em dispositivos com recursos limitados e a necessidade de detecção rápida para evitar falhas em processos industriais são desafios cada vez mais presentes nas áreas de manutenção preditiva e automação (GRØNBECH *et al.*, 2020). A ausência de pesquisas voltadas para a binarização e otimização de modelos para esses contextos destaca a necessidade de um estudo que contemple essas demandas.

Portanto, a justificativa para a escrita deste trabalho está fundamentada na necessidade de avançar no monitoramento de condições de máquinas ferramentas por meio da aplicação de técnicas inovadoras de aprendizado de máquina para detecção de falhas em dados de séries temporais binárias. Embora os algoritmos de aprendizado de máquina têm mostrado eficácia na detecção de falhas em dados de séries temporais, faltam pesquisas que abordem especificamente a detecção de falhas a partir de dados binários, proveniente de sensores industriais (PRINCZ; SHALOO; EROL, 2024).

1.4 DELIMITAÇÕES

Nesta tese, estabelecem-se algumas delimitações que guiam o escopo e a aplicação da solução proposta, que se concentra na manutenção de máquinas ferramenta em ambiente in loco. A seguir, são detalhados os principais aspectos que caracterizam essas delimitações.

1.4.1 Cenário de Aplicação

A solução proposta foi idealizada para ser utilizada em cenários de manutenção de máquinas industriais no próprio local de operação. O processo inicia-se com a coleta de dados da máquina em perfeito funcionamento, logo após sua instalação. Esses dados, coletados em um estado de operação sem falhas, são utilizados para treinar o modelo autoencoder, que servirá posteriormente como base de comparação.

Esse método difere de abordagens convencionais, onde o treinamento de modelos costuma ser feito com divisões padronizadas de conjuntos de dados de treinamento e teste. No caso desta abordagem, o treinamento ocorre exclusivamente com dados de máquinas em perfeitas condições, criando um modelo de referência. Dados posteriores, coletados durante solicitações de assistência técnica, serão analisados e comparados com o modelo treinado para detectar possíveis falhas ou o início de um processo de deterioração.

1.4.2 Simplificações no Tratamento de Dados

Foram feitas simplificações intencionais no tratamento dos dados utilizados tanto para o treinamento quanto para a análise posterior do modelo. A qualidade dos

dados foi garantida na etapa de desenvolvimento da interface de coleta de dados, bem como no algoritmo de coleta. Portanto, não foram necessárias técnicas de preparação de dados como normalização ou outras abordagens comuns para ajustar os dados antes do treinamento.

1.4.3 Integração com a Interface de Coleta de Dados

A solução aqui proposta está fortemente vinculada ao uso da interface de coleta de dados desenvolvida. Esta interface desempenha um papel central na coleta de dados da máquina em dois momentos específicos:

1. Durante a instalação – quando os dados em condições ótimas de funcionamento da máquina são capturados e usados para o treinamento do modelo.
2. Durante a assistência técnica – quando novos dados da máquina, possivelmente em condições de falha, são coletados para análise.

Essa integração permite que o modelo autoencoder seja continuamente atualizado e aplicado para detecção de falhas com base nos dados em campo.

1.4.4 Exclusão de Abordagens Tradicionais de Validação

Diferentemente das abordagens tradicionais que utilizam divisões de dados em porcentagens pré-definidas para treinamento e teste, nesta tese adotou-se uma estratégia distinta devido à abundância de dados disponíveis. Foi coletado um conjunto de dados exclusivo de funcionamento normal, sem falhas, para o treinamento do autoencoder. Além disso, um ou dois conjuntos de dados adicionais, contendo registros de condições de funcionamento normal, sem falhas, foram coletados separadamente e utilizados para testar o modelo treinado. Essa abordagem permitiu treinar o autoencoder apenas com dados normais sem a necessidade de dividir os dados originalmente coletados.

1.4.5 Limitações e Perspectivas Futuras para Aplicações em Sistemas Embarcados

Embora este trabalho tenha como objetivo futuro a implementação de modelos mais eficientes para aplicações embarcadas, ela não avança para a área de

binarização e otimização de modelos, que são etapas críticas para reduzir ainda mais o consumo de recursos e melhorar o desempenho em dispositivos de baixo consumo. No entanto, o foco principal deste trabalho é na criação de uma técnica que visa simplificar o modelo, de modo a facilitar sua execução em sistemas de computação convencional e no futuro em sistemas embarcados, mantendo a eficácia na detecção de falhas. Assim, busca-se reduzir a complexidade do modelo sem comprometer a precisão da detecção, preparando o terreno para futuras pesquisas voltadas à otimização e binarização.

1.5 ESTRUTURA DA TESE

Este trabalho está organizado em nove capítulos: Introdução, Fundamentação Teórica, Revisão da Literatura, Ambiente de Desenvolvimento e Ensaio do Modelo Proposto, Materiais e Métodos, Resultados, Discussões e Cronograma. No primeiro capítulo é apresentado uma introdução ao tema, lacuna de pesquisa, justificativa e os objetivos. No segundo capítulo é apresentada a fundamentação teórica sobre aprendizado de máquina e aprendizado de máquina aplicado a manutenção em máquinas ferramenta.

No terceiro capítulo, é realizada uma revisão da literatura, discutindo os principais trabalhos e estudos que abordam a detecção de falhas em máquinas-ferramenta, com ênfase nos modelos baseados em séries temporais binárias e aprendizado profundo. O quarto capítulo, apresenta o Ambiente de Desenvolvimento e Ensaio do Modelo Proposto, detalha a configuração do ambiente computacional e experimental necessário para implementar e testar o modelo sugerido e o ambiente de desenvolvimento e ensaio dos modelos propostos. O quinto capítulo, Materiais e Métodos, descreve os dados, ferramentas e abordagens metodológicas empregadas para o desenvolvimento e teste dos modelos.

No sexto capítulo, são apresentados os resultados obtidos a partir da aplicação do modelo proposto, incluindo as métricas de desempenho e a comparação com outros métodos. No sétimo capítulo, são analisados os resultados obtidos, com ênfase nas implicações dos achados, desafios encontrados, limitações do estudo e possíveis

melhorias. Finalmente, o oitavo capítulo apresenta o Cronograma, que descreve a execução da tese em termos de prazos e etapas concluídas.

2. FUNDAMENTAÇÃO TEÓRICA

Para a abordagem deste trabalho buscou-se referencias nos temas: falha, manutenção e aprendizado de máquina para explicar o conceito de aprendizado de máquina aplicado à predição de falhas em máquinas ferramenta utilizando dados de sensores.

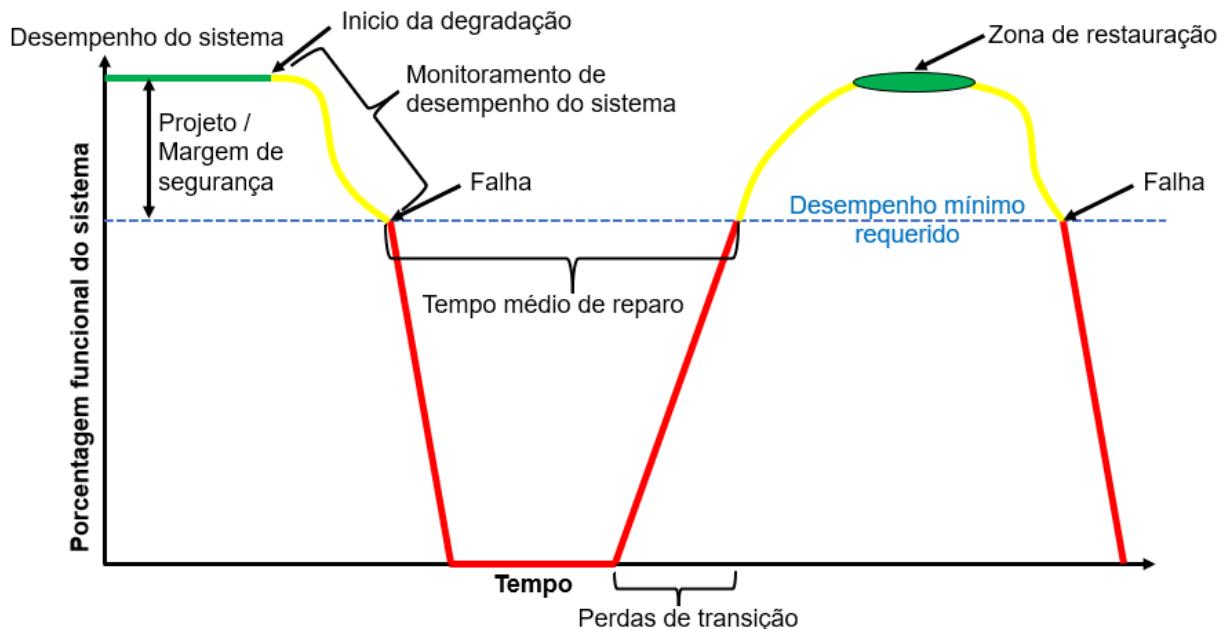
2.1 CONCEITUAÇÃO DE FALHA

Uma conceituação importante no contexto deste estudo é a distinção entre defeito e falha. Enquanto o defeito é caraterizado como qualquer desvio de uma característica de um item em relação aos seus requisitos (ABNT, 1994, p.3), a ABNT define o termo falha como término da capacidade de um item desempenhar a função requerida (ABNT, 1994, p. 3). Isto significa que a abordagem de predição de falhas deve basear-se numa compreensão clara da falha em cada um dos níveis do sistema (AHMAD *et al.*, 2022).

Os componentes do sistema podem estar degradados, apresentarem defeito ou até mesmo falhar e ainda assim não causar falha do sistema. De forma que o papel das técnicas de predição de falhas é reconhecer a margem para falha e estimar o tempo de falha a fim de possibilitar o planejamento prévio dos reparos necessários, a fim de minimizar o Tempo Médio para Reparo (MTTR) e o tempo de inatividade associado. (VELIBEYOGLU; NOH; POZZI, 2019).

O ciclo de vida de um sistema, desde o momento da intenção do projeto (desempenho mínimo exigido + margem de segurança do projeto), passando pela degradação, até a falha funcional e subsequente restauração é apresentado na Figura 01, a falha ocorre quando o desempenho do sistema cai abaixo do ponto de desempenho mínimo requerido (WU; LIU; XU, 2022).

Figura 01 – Curva de desempenho do sistema.



Fonte: Adaptado de NASA (2008).

A prevenção de falhas é uma abordagem proativa que visa prevenir uma falha prevista e iniciar uma ação de correção para evitar a falha funcional do sistema (ZHANG *et al.*, 2022).

2.2 CONCEITUAÇÃO DE MANUTENÇÃO

A ABNT define o termo manutenção na NBR 5462 como a Combinação de todas as ações técnicas e administrativas, incluindo as de supervisão, destinadas a manter ou recolocar um item em um estado no qual possa desempenhar uma função requerida (ABNT, 1994, p. 6). A progressão das estratégias de manutenção atuais evoluiu da manutenção corretiva para a manutenção preventiva e depois para a manutenção baseada na condição (CBM). A manutenção corretiva é a forma mais antiga de manutenção em que nenhuma ação é tomada para manter o equipamento até que ele quebre e necessite de reparo ou substituição. Para evitar falhas catastróficas e paradas de emergência, a manutenção preventiva foi introduzida na década de 1950 (AIWINA *et al.*, 2009; VELIBEYOGLU; NOH; POZZI, 2019).

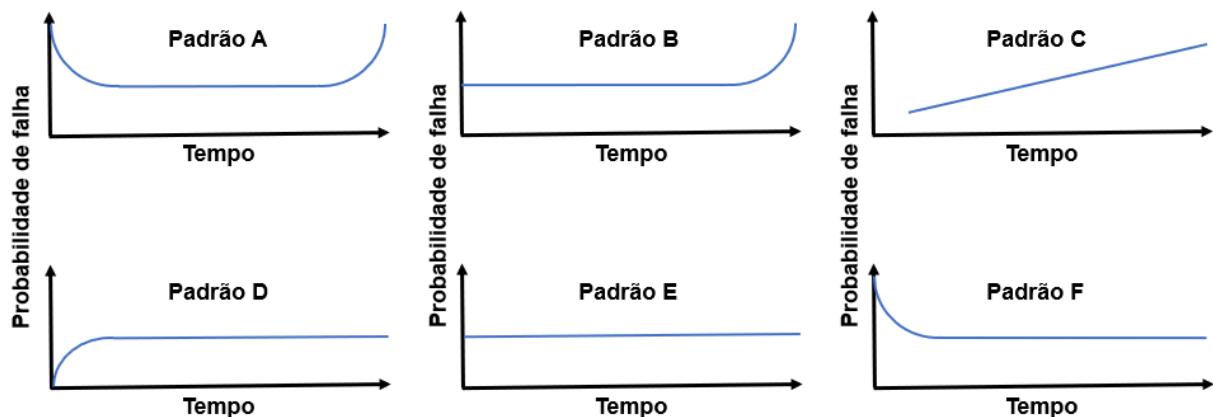
A manutenção preventiva TBM (manutenção baseada no tempo) baseia-se no princípio de que a falha do sistema/equipamento está sempre relacionada ao tempo

de operação, ou seja, a probabilidade de falha de cada sistema aumenta com o tempo. Porém, na prática observa-se que este comportamento não é válido para todos os componentes. Segundo Maubray (1997) existem seis tipos de padrões de probabilidade de falha relacionados ao tempo de execução:

- Padrão A, Curva da Banheira, representação tradicional do comportamento de falha de vários itens como componentes mecânicos, eletrônicos, materiais e até mesmo seres humanos, em que três comportamentos diferentes podem ser observados: falha prematura ou mortalidade infantil; falha durante a vida útil; ou falha durante o envelhecimento.
- Padrão B, Desgaste, probabilidade de falha constante e aumento repentino ao final da vida útil do equipamento;
- Padrão C, Fadiga, aumento constante da probabilidade de falha desde o início da operação, sem ser possível identificar um momento de aumento drástico da probabilidade de falha;
- Padrão D, Aumento Inicial, probabilidade mínima no início da operação, aumento repentino até alcançar um determinado valor onde se mantém constante;
- Padrão E, Falha aleatória, probabilidade de falha sempre constante;
- Padrão F, Mortalidade infantil, probabilidade de falha alta no início da operação, seguido por queda repentina até atingir um determinado valor e se mantém constante;

As curvas de probabilidade de falha versus tempo para todos os tipos são apresentadas na Figura 02.

Figura 02 – Tipos de taxa de probabilidade de falha.



Fonte: Adaptado de MAUBRAY (1997).

Os padrões de probabilidade de falhas A e B são comumente associados a componentes simples e únicos, tais como pneus, pastilhas de freio e elementos estruturais. Por outro lado, itens mais complexos tendem a seguir os padrões de probabilidade condicional representados pelos modelos C, D, E e F. Esta distinção fundamental entre os padrões de falha de componentes complexos e simples tem repercussões significativas na gestão da manutenção (NASA, 2008; AHMAD *et al.*, 2022).

Componentes simples e de peça única geralmente exibem uma correlação clara entre a confiabilidade e a idade. Isso se torna ainda mais evidente em casos em que fatores como fadiga do metal ou desgaste mecânico são predominantes, ou quando os componentes são projetados para serem consumíveis, com uma vida útil curta ou predefinida. Em tais situações, estabelecer um limite de idade baseado na duração de uso ou nos ciclos de estresse pode ser uma estratégia eficaz para aumentar a confiabilidade do sistema complexo ao qual esses componentes pertencem (NASA, 2008; ZHANG *et al.*, 2022).

Componentes complexos frequentemente apresentam uma fase inicial de mortalidade infantil, seguida por um aumento gradual ou estabilização na probabilidade de falha, sendo raro o envelhecimento acelerado. Em diversas situações, a manutenção programada pode inadvertidamente elevar a taxa de falhas ao introduzir um alto índice de mortalidade infantil em um sistema que, de outra forma, seria estável (NASA, 2008; AIWINA *et al.*, 2009).

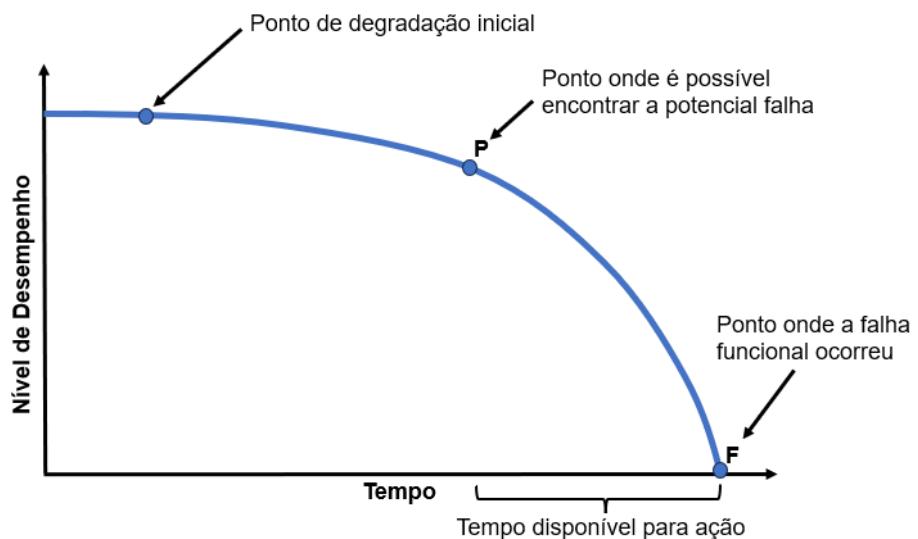
Portanto, para componentes cujo comportamento de falha segue os padrões C, D, E e F, são necessárias estratégias alternativas. Estas podem incluir manutenção corretiva, aceitação do risco de falha ou modificações no projeto, como a inclusão de redundâncias. Apesar de tais políticas ocasionalmente diminuírem as falhas em equipamentos, elas exigem maior esforço de mão de obra, não erradicam completamente falhas catastróficas e podem resultar em manutenções desnecessárias (AIWINA *et al.*, 2009; BIGGIO; KASTANIS, 2020).

É neste contexto que entra a Manutenção Baseada em Condição (CBM), que representa uma abordagem de manutenção que rastreia continuamente a saúde operacional das máquinas e baseia as decisões de manutenção nas informações obtidas desse monitoramento, ou seja, manutenção preditiva. A CBM procura acompanhar a condição das máquinas por meio de medições que não interferem em

sua operação normal. Essa estratégia tem se mostrado eficiente para diminuir manutenções desnecessárias e para aumentar a confiabilidade dos equipamentos, ganhando popularidade crescente nos últimos anos (YAGUO et al., 2018).

Neste procedimento, testes são conduzidos ou dados são coletados para avaliar o estado do equipamento, permitindo que as tarefas de manutenção sejam agendadas para um momento em que a diminuição da confiabilidade já começou, mas o equipamento ainda mantém sua funcionalidade (JARDINE et al., 2006; VACHTSEVANOS et al., 2006). A Figura 03 demonstra o intervalo de tempo entre a detecção do início de uma falha, ponto P, e o momento em que ocorre a falha funcional real do equipamento ponto F.

Figura 03 – Curva P-F.



Fonte: Adaptado de Upasane et al. (2024).

O acompanhamento do nível de desempenho e a previsão do estado de saúde de equipamentos críticos constituem elementos chave da estratégia de manutenção preditiva. A evolução acelerada no processamento de informações tem facilitado a obtenção de resultados quase em tempo real operacional. Desenvolvimentos recentes no campo do aprendizado de máquina e do aprendizado profundo revelaram uma série de algoritmos eficazes aplicados ao prognóstico da condição de máquinas. Distanciando-se de regras programadas manualmente, estes algoritmos são capazes de assimilar a progressão das condições das máquinas, viabilizando a geração de previsões mais precisas e fundamentadas (VOGL; WEISS; HELU, 2019; BIGGIO; KASTANIS, 2020).

2.3 CONCEITUAÇÃO DE SÉRIES TEMPORAIS

Uma série de dados é uma sequência ordenada de pontos de dados. Os pontos de dados descrevem alguma propriedade de objeto ou processo com base em uma medida contínua, como temperatura, massa, ângulo, posição e velocidade. Se a ordem for baseada no tempo, a sequência é geralmente chamada de série temporal. Independentemente da medida de ordenação, o registro dos pontos de dados geralmente segue intervalos discretos e igualmente espaçados (SCHMIDL; WENIG; PAPENBROCK, 2022).

Desta forma, séries temporais são conjuntos de observações de uma variável ao longo do tempo, organizadas em uma sequência cronológica uniforme (BRAEI; WAGNER, 2020). Uma série temporal é um conjunto ordenado, onde:

$$(T = \{T_1, T_2, \dots, T_m\}) \quad (01)$$

Uma série temporal é um conjunto ordenado de (m) pontos de dados potencialmente multidimensionais com valor real.

$$(T_i \in R^n) \quad (02)$$

Uma subsequência de uma série temporal é um segmento contíguo de (T) com comprimento .

$$\begin{aligned} (T_{i,j} = \{T_i, \dots, T_j\} \subseteq T) \\ (|T_{i,j}| = j - i + 1) \\ (|T_{i,j}| \geq 1) \end{aligned} \quad (03)$$

Diversos autores assumem que os pontos de dados são equidistantes, o que é verdade para a maioria das séries temporais do mundo real e dispensa os algoritmos de interpretar as medidas contínuas concretas (tempo, massa, ângulo, etc.); séries de dados que não seguem essa suposição precisam ser discretizadas (SCHMIDL; WENIG; PAPENBROCK, 2022).

A definição de falha no contexto de séries temporais difere na literatura. Pontos de dados individuais, falhas e anomalias são geralmente chamados de outliers (AGGARWAL, 2017).

Como se pretende avaliar abordagens de detecção de falhas, usará-se a seguinte definição geral neste trabalho:

Uma falha de série temporal é uma sequência de pontos de dados (Ti, j) de comprimento $(j - i + 1 \geq 1)$ que desvia alguma característica, modelo e/ou medida de similaridade de padrões frequentes na série temporal (T) .

2.4 CONCEITUAÇÃO DE APRENDIZADO DE MÁQUINA

De acordo com Mitchell (1997), o aprendizado de máquina é um campo de pesquisa que foca no desenvolvimento de métodos que possam melhorar seu desempenho com os dados obtidos. Ainda que o conceito de aprendizagem de máquina seja vasto, quando relaciona com programas de computador, o que se necessita aprender geralmente abrange um conjunto de tarefas. O programa aprende obtendo experiência com essas tarefas, objetivando melhorar determinadas métricas de desempenho, utilizando-se de estatística, probabilidade e álgebra linear. (MORRIE; ROSTAMIZAD; TALWALKAR, 2018; IQBAL *et al.*, 2020).

De forma que, a aplicação de aprendizado de máquina em análise preditiva, por exemplo, tem a função principal de estimar o risco de acontecimentos futuros a partir de experiências passadas, a fim de guiar a tomada de decisão atual. Para executar tarefas e obter melhores resultados, as máquinas usam um conjunto de ferramentas e algoritmos para entender os dados existentes e gerar regras preditivas. No entanto, o problema que um algoritmo busca solucionar pode variar, mas geralmente tem uma entrada que precisa ser transformada para produzir uma saída (IQBAL *et al.*, 2020; KAZMIERCZAK, 2022).

Alpaydin (2010) elucida um problema resolvido por um algoritmo de aprendizado de máquina, a ordenação de um conjunto de números. Nesse caso, a entrada será um conjunto não ordenado de números, onde alguns processos de classificação serão executados e a saída será um conjunto ordenado de números. No

entanto, algumas tarefas podem ser diferentes, como o exemplo de análise preditiva mencionado anteriormente. Em geral, as ferramentas de aprendizado de máquina podem ser classificadas em uma das seguintes categorias: aprendizado supervisionado e aprendizado não supervisionado (YE, 2018).

No aprendizado supervisionado, a conclusão de um conjunto de dados é conhecida e rotulada por um especialista na área (GUERRA *et al.*, 2022). A diferenciação entre o tipo de variável resposta resulta em duas subcategorias de aprendizagem supervisionada: algoritmos para problemas de classificação, quando o resultado é uma variável categórica (qualitativa) e algoritmos para problemas de regressão, quando o resultado é uma variável quantitativa (IQBAL *et al.*, 2020).

No contexto de aprendizagem supervisionada, cada observação ($i = (1, 2, 3, \dots, n)$) de um conjunto de dados possui um vetor de medição de variáveis preditoras (variáveis de entrada ou independentes) (x_i) e uma medição correspondente à resposta de interesse (y_i) (*output*, desfecho ou variável dependente) e o modelo que liga a resposta à variável preditora é ajustado com o objetivo de prever essa resposta em observações futuras, para as quais apenas dados envolvendo a variável preditora estão disponíveis (JAMES *et al.*, 2014; SARKER, 2021). Um conjunto de dados representado por um vetor (x) pode ser observado, onde cada variável preditora (x_i) descreve uma característica do conjunto de dados, como cor, tamanho ou forma. Neste exemplo, um vetor (x_1), com variáveis independentes ($x_{i1}, x_{i2}, x_{i3},$) o algoritmo classifica as observações na classe estimada (y_i) (KORDNOORI *et al.*, 2021).

Para a construção de modelos de aprendizado supervisionado, após rotular e definir o conjunto de dados inicial, uma parte das amostras é separada para treinamento. As divisões mais usadas são 60/40, 70/30 ou 80/20, dependendo do tamanho inicial do conjunto de dados, quanto maior o número de observações, maior a proporção do conjunto inicial usado para treinamento do modelo (RASCHKA, 2017). Em seguida, ajuste-se os atributos que controlam o treinamento do modelo, hiperparâmetros, e a outra parte da amostra é utilizada para testar o desempenho do modelo. Esta etapa inicial tem como objetivo determinar se o algoritmo funciona bem não apenas no conjunto de dados usado para treinamento, mas também em sua capacidade de generalizar novas observações durante o teste (SARKER, 2021).

No caso do aprendizado não-supervisionado, não existe uma variável resposta conhecida responsável por guiar a análise e o objetivo compreende, a título de exemplo, agrupar populações parecidas, *clustering*, ou reduzir a dimensionalidade de um conjunto de dados, onde conjuntos com muitas variáveis independentes são reduzidos a dimensões menores para análise sem perder as características dos dados (NOVIA *et al.*, 2022). No caso de agrupamento, os dados são analisados quanto à similaridade ou dissimilaridade e agrupados usando uma medida de distância, como a distância euclidiana, que é um vetor ($P = (p_1, p_2, p_3, \dots, n)$) e outro vetor ($Q = (q_1, q_2, q_3, \dots, n)$), uma forma de agrupamento não supervisionado é o agrupamento particional, onde cada ponto deve pertencer a pelo menos uma partição ou cluster (k), onde o desenvolvedor do modelo inicialmente define o número de partições (ALLOGHANI *et al.*, 2020; SARKER, 2021).

Pode-se dizer que os algoritmos de aprendizado de máquina divergem por serem dependentes de dados, pois são os requisitos das etapas necessárias para o algoritmo aprender pesos e estimar parâmetros de acordo com a tarefa desejada. Para fazer isso, geralmente é necessária uma etapa chamada treinamento. Após o treinamento, a saída do algoritmo é o modelo, que pode ser utilizado, por exemplo, para prever novas instâncias dos dados inseridos (SATINET; FOUSS, 2022; MOLNAR, 2022).

Vale ressaltar que existem outras formas de aprendizado como o aprendizado semi-supervisionado e o aprendizado por reforço. No aprendizado semi-supervisionado, uma pequena porção de dados é rotulada e combinada com uma grande porção de dados não rotulados (ZHONG; HUANG, 2018; VAN ENGELEN; HOOS, 2020). No aprendizado por reforço, um algoritmo aprende o que fazer para maximizar uma recompensa numérica. Nesse caso, o algoritmo não recebe uma tarefa específica e deve descobrir por tentativa e erro quais ações geram a maior recompensa (SUTTON; BARTO, 2018; CHOI *et al.*, 2020).

Os algoritmos de aprendizado de máquina foram criados com o propósito de solucionar desafios complexos, integrando diversas técnicas para aprimorar sua eficácia. O objetivo primordial é alcançar a estimativa mais precisa por meio de um modelo matemático capaz de abranger a variabilidade presente nos dados e manter sua efetividade em novos conjuntos de dados. Existem modelos que empregam a

fusão de múltiplos outros modelos na busca pela otimização máxima dos resultados. Essas fusões de diversos modelos preditivos são conhecidas como *ensembles* (SARKER, 2021; SATINET; FOUSS, 2022).

Antes de aplicar os modelos de aprendizado de máquina, é importante pré-processar as variáveis de forma a otimizar o desempenho dos algoritmos. Variáveis quantitativas geralmente passam por uma padronização para evitar problemas de escala, utilizando, por exemplo, a técnica do escore-z (*z-score*). No caso de variáveis qualitativas com mais de duas categorias, é comum a conversão em variáveis "*dummy*," onde cada categoria se transforma em uma nova variável binária com valores 0 ou 1. Uma das variáveis é mantida como referência (RASCHKA, 2017; SARKER, 2021).

Em geral, as técnicas de pré-processamento de dados incluem modificar dados brutos, selecionar variáveis relevantes, lidar com dados ausentes e organizar dados categóricos. A transformação de dados desempenha um papel importante na otimização do desempenho de muitos algoritmos que dependem de variáveis de previsão medidas na mesma escala. Dois métodos amplamente usados para dimensionar variáveis são a normalização (escalonar uma variável para um intervalo entre 0 e 1, criando um intervalo bem definido) e a padronização (centrar uma variável em torno da média igual a zero e do desvio padrão igual a um (SARKER, 2021).

Adicionalmente, as transformações de dados podem ser valiosas para corrigir distribuições distorcidas, por exemplo, usando uma estatística assimétrica ou a relação entre os valores mais altos e mais baixos de uma variável preditora. De acordo com a estatística de assimetria, quanto mais longe de 0, mais assimétrica é a distribuição da variável. No caso da relação maior-menor valor, os valores maiores que 20 indicam uma distribuição assimétrica. (SATINET; FOUSS, 2022).

Em geral, o processo de aprendizagem de um modelo preditivo consiste em dois objetivos principais: seleção e avaliação do modelo. Na primeira etapa, o objetivo é avaliar o desempenho de vários modelos para selecionar o modelo com o melhor desempenho. Na segunda etapa, após a seleção do modelo, o foco se volta para a

estimativa do erro de predição do modelo quando aplicado a novos dados, o erro de generalização (SARKER, 2021; MEESAD; BOONMATHAM, 2023).

Quando se têm grandes conjuntos de dados, o método mais eficiente de seleção de modelo é dividir aleatoriamente o conjunto de treinamento em três partes distintas: treinamento, validação e teste. Portanto, o conjunto de dados original é dividido em três subconjuntos. Os dados de treinamento são usados para ajustar o modelo, enquanto os dados de validação são usados para selecionar o modelo com melhor desempenho na previsão. Por fim, os dados de teste cumprem o papel de avaliar o desempenho geral da capacidade de generalização do modelo selecionado (DUCHANOY et al., 2020; PANCHAL; 2020).

2.5 APRENDIZADO PROFUNDO

Aprendizado Profundo (DL) é uma metodologia avançada dentro do campo do Aprendizado de Máquina. Ela se destaca por empregar redes neurais artificiais profundas, estruturas compostas por múltiplas camadas de unidades de processamento, para extrair e aprender padrões complexos de dados. Estas redes neurais, devido à sua profundidade, são capazes de capturar representações hierárquicas dos dados, processando-os em diversos níveis de abstração (XINYUAN et al., 2023).

O termo "profundo" nesse contexto refere-se à quantidade significativa de camadas através das quais os dados são processados, permitindo que as redes neurais aprendam desde padrões simples até aspectos muito complexos e abstratos dos dados. Essa característica as torna extremamente eficientes na modelagem de dados em níveis elevados de abstração, por meio de transformações não lineares sucessivas (ALI et al., 2023).

O Deep Learning também é fundamental em aplicações críticas como em carros autônomos, onde contribui para a tomada de decisão e previsões complexas, integrando técnicas como Aprendizado por Reforço Profundo e Aprendizado de Transferência. Na indústria, auxilia na otimização de custos operacionais, melhoria da produtividade, diagnóstico de falhas e otimização do tempo de inatividade (YOTOV; ALEKSIEVA-PETROVA, 2023).

No contexto de manutenção de máquinas e equipamentos, as abordagens de aprendizado profundo e aprendizado de máquina superficial diferem em sua complexidade, capacidade de extração de recursos e desempenho:

- Complexidade: modelos de aprendizagem profunda, como redes neurais convolucionais (CNN) e redes de memória de longo e curto prazo (LSTM), são mais complexos do que algoritmos tradicionais de aprendizagem de máquina superficial, como Naïve Bayes ou Random Forest. Os modelos de aprendizagem profunda podem aprender automaticamente representações hierárquicas de dados, tornando-os mais adequados para capturar padrões intrincados em dados relacionados a manutenção (WASON, 2018; MAHARA *et al.*, 2023).
- Extração de recursos: os modelos de aprendizado profundo se destacam na extração automática de recursos relevantes de dados brutos, eliminando a necessidade de engenharia manual de recursos. Algoritmos superficiais de aprendizado de máquina geralmente exigem recursos artesanais, que podem não capturar as nuances presentes nos dados manutenção (LIANG *et al.*, 2017).
- Desempenho: Estudos demonstraram que os algoritmos de aprendizagem profunda geralmente superam os algoritmos tradicionais de aprendizagem de máquina na detecção de notícias falsas sobre saúde. Por exemplo, no contexto de manutenção, as abordagens de aprendizagem profunda alcançaram maior precisão e pontuações F1 em comparação com métodos superficiais de aprendizagem automática (ZHAO *et al.*, 2019).
- Adaptabilidade: Os modelos de aprendizagem profunda têm a capacidade de se adaptar e aprender com grandes quantidades de dados. Algoritmos superficiais de aprendizado de máquina podem ter dificuldades para generalizar bem para padrões de dados novos e invisíveis, limitando sua eficácia na detecção falhas (QIU *et al.*, 2023).

Em suma, o Deep Learning é uma técnica de aprendizado de máquina poderosa e versátil, capaz de lidar com dados multidimensionais e realizar aprendizado de representação em diversos níveis. Seu sucesso em áreas diversas, desde a análise de sentimentos no *trading* financeiro até aplicações em saúde e

indústria, demonstra sua capacidade de processar e analisar grandes volumes de dados, oferecendo soluções precisas e inovadoras para uma ampla gama de desafios (SAMINE *et al.*, 2023).

2.6 APRENDIZADO DE MÁQUINA APLICADO A MANUTENÇÃO DE MÁQUINAS E EQUIPAMENTOS

A manutenção preditiva evoluiu de métodos iniciais de inspeção visual para métodos automatizados que utilizam técnicas avançadas de processamento de sinais baseadas em aprendizado de máquina, reconhecimento de padrões, redes neurais e lógica difusa (LUO *et al.*, 2020). Através de dados obtidos de sensores integrados as máquinas ferramentas, a manutenção preditiva pode evitar a substituição desnecessária de equipamentos, reduzir o tempo de inatividade da máquina, deteção de falha e, desta forma, economizar custos e melhorar a eficiência (FRONTONI *et al.*, 2018).

Portanto, os algoritmos são importantes para uma manutenção preditiva eficaz, e existem vários tipos de técnicas que podem ser aplicadas em diferentes estágios de implementação do aprendizado de máquina aplicado a máquinas ferramenta. O aprendizado de máquina aplicado à manutenção de máquinas ferramenta pode ser dividido em duas grandes categorias: supervisionado, onde informações sobre a ocorrência de falhas estão presentes no conjunto de dados de modelagem, e não supervisionado, onde informações lógicas e/ou de processo estão disponíveis, mas nenhum dado de manutenção. (LOK *et al.*, 2022).

De acordo com Silvestrin, Hoogendoorn e Koole (2019) os algoritmos tradicionais de aprendizado de máquina são frequentemente combinados com técnicas de engenharia de recursos para extrair informações relevantes dos dados dos sensores em cenários de manutenção preditiva, estes algoritmos incluem: *Random Forrest* (RF); Árvore de Decisão (DT) e K-vizinhos mais próximos (KNN).

Silvestrin, Hoogendoorn e Koole (2019) também destacam que abordagens de aprendizagem profunda diferem dos algoritmos tradicionais de aprendizagem de máquina na manutenção preditiva das seguintes maneiras:

- Engenharia de recursos: algoritmos tradicionais de aprendizado de máquina geralmente exigem engenharia manual de recursos, onde o conhecimento do domínio é usado para extrair recursos relevantes dos dados. Em contraste, as abordagens de aprendizagem profunda podem aprender recursos automaticamente a partir dos dados brutos, eliminando a necessidade de engenharia manual de recursos (KUHN; JOHNSON, 2019).
- Requisitos de dados: técnicas de aprendizagem profunda, como redes *Long Short-Term Memory* (LSTM) e redes neurais convolucionais (CNNs), são conhecidas por exigirem que grandes quantidades de dados rotulados sejam treinadas de forma eficaz. Em contraste, algoritmos tradicionais de aprendizado de máquina, como Random Forest e Árvores de decisão, podem funcionar bem em conjuntos de dados menores com engenharia de recursos adequada (Krizhevsky; Sutskever; Hinton, 2017).
- Complexidade: Os modelos de aprendizado profundo são normalmente mais complexos e computacionalmente intensivos em comparação com algoritmos tradicionais de aprendizado de máquina. Essa complexidade permite que modelos de aprendizagem profunda capturem padrões intrincados nos dados, mas também requer mais recursos computacionais para treinamento e inferência (Buscema; Sacco, 2017).
- Desempenho: As abordagens de aprendizagem profunda mostraram resultados promissores em diversas tarefas de manutenção preditiva, como estimativa da vida útil restante de turbinas eólicas e avaliação da degradação de rolamentos. No entanto, a eficácia dos modelos de aprendizagem profunda depende fortemente da disponibilidade de conjuntos de dados grandes e equilibrados, o que pode nem sempre ser viável em cenários de manutenção preditiva da vida real (Nguyen; Medjaher, 2019).

Desta forma a comparação dos algoritmos apontando que embora as abordagens de aprendizagem profunda ofereçam a vantagem da extração automática de recursos e o potencial para capturar padrões complexos em dados, elas também apresentam desafios relacionados aos requisitos de dados, complexidade computacional e desempenho em cenários com disponibilidade limitada de dados (Silvestrin; Hoogendoorn; Koole, 2019).

Silvestrin, Hoogendoorn e Koole (2019) apresentam alguns desafios potenciais na aplicação de abordagens de aprendizagem profunda à manutenção preditiva, que incluem:

- Disponibilidade de dados: Os modelos de aprendizagem profunda normalmente requerem grandes quantidades de dados rotulados para treinamento, que podem nem sempre estar prontamente disponíveis em cenários de manutenção preditiva. Para enfrentar esse desafio, técnicas como aumento de dados, aprendizagem por transferência e aprendizagem semi-supervisionada podem ser empregadas para aproveitar ao máximo os dados limitados (LUO *et al.*, 2015)
- Dados desequilibrados: Na manutenção preditiva, a ocorrência de eventos de falha pode ser rara em comparação com condições normais de operação, levando a conjuntos de dados desequilibrados. Os modelos de aprendizagem profunda podem ter dificuldade em aprender com dados desequilibrados, resultando em previsões tendenciosas. Técnicas como sobre amostragem, sub amostragem e uso de métricas de avaliação apropriadas podem ajudar a mitigar o impacto de dados desequilibrados no desempenho do modelo (DAS; DATTA; CHAUDHURI, 2018).
- Interpretabilidade: Os modelos de aprendizagem profunda são frequentemente considerados caixas pretas devido às suas arquiteturas complexas e representações de alta dimensão. Compreender como esses modelos fazem previsões é importante para obter insights sobre os mecanismos subjacentes à degradação da integridade das máquinas. Técnicas como visualização de modelos, análise de importância de recursos e métodos de explicação de modelos podem melhorar a Interpretabilidade de modelos de aprendizado profundo em manutenção preditiva (MONTAVON; SAMEK; MÜLLER, 2018).
- Recursos computacionais: O treinamento de modelos de aprendizagem profunda pode ser intensivo em termos computacionais, exigindo hardware poderoso e tempo e recursos significativos. Para enfrentar esse desafio, técnicas como otimização de modelos, computação distribuída e aproveitamento de serviços baseados em nuvem podem ajudar a melhorar a eficiência dos processos de treinamento e inferência (CAMPOS *et al.*, 2017).

De forma que ao enfrentar esses desafios por meio de técnicas apropriadas de manipulação de dados, métodos de Interpretabilidade de modelos e estratégias de otimização, a aplicação de abordagens de aprendizagem profunda na manutenção preditiva pode ser aprimorada para fornecer previsões mais precisas e confiáveis que as técnicas tradicionais de aprendizado de máquina para monitoramento da integridade das máquinas e programação de manutenção (SILVESTRIN; HOOGENDOORN; KOOLE, 2019).

2.7 AUTOENCODER

Os autoencoders (AE) são uma classe de redes neurais artificiais utilizadas para aprender representações eficientes de dados não supervisionados, geralmente para redução de dimensionalidade ou extração de características. Introduzidos inicialmente na década de 1980, eles ganharam destaque com o avanço das técnicas de *deep learning* e têm sido amplamente aplicados em áreas como processamento de imagens, reconhecimento de padrões e detecção de anomalias (FINK et al., 2020).

A ideia fundamental por trás dos autoencoders é aprender uma representação compacta (codificação) dos dados de entrada, de forma que seja possível reconstruir a entrada original a partir dessa representação. Essa capacidade de compressão e reconstrução torna os autoencoders ferramentas poderosas para entender e manipular dados complexos.

Algumas dessas aplicações dos autoencoder incluem detecção de falhas, e eles são particularmente eficazes na redução da dimensionalidade de dados de alta dimensão, na captura de características importantes e na reconstrução da entrada original. (PUTATUNDA, 2018). Ao se utilizar os autoencoder com estratégia de reconstrução de sinal, esta arquitetura específica pode assimilar o comportamento regular do sistema e distingui-lo dos estados do sistema que deviam divergir dos observados na condição rotulada (FINK et al., 2020).

Um autoencoder treinado pode detectar falhas e identificar pontos de dados com alto erro de reconstrução como valores discrepantes, indicando desvio substancial dos dados nominais. Este modelo é particularmente eficaz para dados

complexos e de alta dimensão (PUTATUNDA, 2018; FINK et al., 2020). Os autoencoders são capazes de aprender os padrões latentes de dados de maneira não supervisionada, tornando-os adequados para tarefas de detecção de falhas onde dados de falhas rotulados podem ser escassos (Wang et al., 2019).

A arquitetura básica de um autoencoder consiste em duas partes principais:

1. Codificador (Encoder): Transforma o input de alta dimensionalidade em uma representação de dimensionalidade mais baixa, também chamada de código ou representação latente.
2. Decodificador (Decoder): Reconstrói os dados originais a partir da representação latente.

Entre o codificador e o decodificador, há uma camada central chamada de *bottleneck*, que é responsável por forçar a rede a aprender uma representação comprimida dos dados. A dimensionalidade dessa camada é menor do que a da entrada, o que impõe uma restrição que obriga o modelo a capturar as características mais salientes dos dados (HINTON; SALAKHUTDINOV, 2006).

A arquitetura específica da rede do autoencoder depende da aplicação e da natureza dos dados de entrada. Por exemplo, Cheng et al. (2018) utiliza autoencoder convolucional profundo em compressão de imagens e Dizaji et al. (2017) faz uma incorporação conjunta de autoencoder convolucional e minimização de entropia relativa para análise e visão computacional. Embora este trabalho se concentre principalmente em dados de séries temporais, para os quais normalmente seriam utilizadas redes recorrentes à implementação específica do autoencoder para realizar a detecção de falhas e suporte para diagnóstico pode apresentar-se como uma solução mais eficiente quando se lida com dados de alta frequência. (SHABANI; IOSIFIDIS, 2020).

A detecção de falhas é uma área ativa de pesquisa. Os Autoencoders são adequados para esta tarefa, pois podem ser treinados apenas com dados normais, eliminando a necessidade de coletar e rotular cenários de falha que podem ser raros e perigosos de obter (MOSIN et al., 2022). Embora as comparações diretas entre diferentes abordagens baseadas em Autoencoders possam não ser diretas, a

estratégia subjacente de aproveitar Autoencoders para detecção de falhas mostrou resultados promissores em várias aplicações (KARAPALIDOU *et al.*, 2023).

Existem diversos tipos de autoencoders, cada um adaptado para diferentes aplicações e tipos de dados:

- Autoencoders Subcompletos (Undercomplete): São autoencoders onde a dimensão da representação latente é menor que a da entrada. O objetivo é aprender a melhor função de codificação possível para reconstruir os dados de entrada a partir de um espaço latente de dimensionalidade reduzida (GOODFELLOW; BENGIO; COURVILLE, 2016).
- Autoencoders Regularizados: Para evitar que o autoencoder simplesmente aprenda a copiar a entrada para a saída, técnicas de regularização são empregadas (GOODFELLOW; BENGIO; COURVILLE, 2016).
- Autoencoders Esparsos (Sparse Autoencoders): Introduzem uma penalização na função de custo que força a ativação de neurônios a ser esparsa. Isso significa que, em qualquer amostra de entrada, apenas um pequeno número de neurônios é ativado, promovendo a aprendizagem de características significativas (BENGIO, 2009).
- Autoencoders Desnoisers (Denoising Autoencoders): São treinados para reconstruir a entrada original a partir de uma versão corrompida dela mesma. Isso obriga o modelo a aprender representações robustas que capturam a estrutura subjacente dos dados (MURPHY, 2012).
- Autoencoders Contrativos (Contractive Autoencoders): Introduzem uma penalização baseada na derivada da função de ativação em relação à entrada, incentivando o modelo a ser robusto a pequenas variações nos dados de entrada (GOODFELLOW; BENGIO; COURVILLE, 2016).
- Autoencoders Convolucionais: Utilizam camadas convolucionais em vez de totalmente conectadas, sendo particularmente eficazes para dados com estrutura espacial, como imagens. Eles capturam características locais e são mais eficientes em termos de parâmetros (GOODFELLOW; BENGIO; COURVILLE, 2016).

- Autoencoders Profundos (Deep Autoencoders): Compõem múltiplas camadas ocultas no codificador e no decodificador, permitindo a aprendizagem de representações hierárquicas e mais abstratas dos dados (MURPHY, 2012).
- Autoencoders Adversariais: Combinam autoencoders com redes adversariais generativas (GANs) para melhorar a qualidade da geração de dados e a aprendizagem de representações (GOODFELLOW; BENGIO; COURVILLE, 2016).
- Autoencoders Variacionais (VAEs): Os VAEs são uma abordagem probabilística que assume que os dados são gerados a partir de variáveis latentes sujeitas a uma distribuição de probabilidade. Eles aprendem tanto a codificar os dados quanto a modelar a distribuição latente, permitindo a geração de novos dados(PUTATUNDA, 2018).

O VAE estende a estrutura do autoencoder com uma abordagem probabilística, permitindo-lhe lidar com a incerteza dos dados. Como No autoencoder, o VAE emprega erro de reconstrução para detecção de falhas, mas oferece flexibilidade adicional devido as suas características generativas. Na versão variacional do AE, os dados de entrada são amostrados a partir de uma distribuição anterior parametrizada e o codificador e o decodificador são treinados em conjunto de modo que a saída minimize um erro de reconstrução no sentido da divergência de Kullback-Leibler entre o posterior paramétrico e o verdadeiro posterior (FINK et al., 2020).

Desta forma, o VAE recebe uma amostra prévia. O codificador comprime-o no espaço latente, então o decodificador obtém uma amostra da entrada do espaço latente e produz uma saída o mais próxima possível do codificador entrada (PUTATUNDA, 2018; FINK et al., 2020).

2.7.1 Fundamentos do autoencoder

No treinamento de autoencoders, a função de perda é um componente essencial que determina o quão bem o modelo está reproduzindo os dados de entrada na saída reconstruída. Ela mede a discrepância entre a entrada original e a saída gerada pelo autoencoder, orientando o ajuste dos pesos durante o processo de aprendizado (GOODFELLOW; BENGIO; COURVILLE, 2016). A escolha da função de

perda adequada é importante para garantir que o modelo capture efetivamente as características dos dados:

- Erro Quadrático Médio (MSE): Comum para dados contínuos, o MSE calcula a média dos quadrados das diferenças entre a entrada (x_i) e a saída reconstruída (\hat{x}_i), conforme apresentado na equação 04.

$$\left(\mathcal{L}_{MSE} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \right) \quad (04)$$

- Entropia Cruzada Binária (BCE): Usada para dados binários ou valores normalizados entre 0 e 1, a BCE mede a dissimilaridade entre as distribuições de probabilidade da entrada e da saída reconstruída, conforme apresentado na equação 05.

$$\left(\mathcal{L}_{BCE} = -\frac{1}{n} \sum_{n=1}^n [x_i \log(\hat{x}_i) + (1 - x_i) \log(1 - \hat{x}_i)] \right) \quad (05)$$

Os algoritmos de otimização desempenham um papel fundamental no treinamento de autoencoders, influenciando diretamente a velocidade com que o modelo converge para uma solução ótima e a qualidade dessa solução. A escolha adequada do otimizador pode acelerar o processo de aprendizado, melhorar a estabilidade durante o treinamento e aumentar a capacidade do modelo de generalizar para novos dados (GOODFELLOW; BENGIO; COURVILLE, 2016). Alguns dos otimizadores mais utilizados e suas principais características são apresentados a seguir:

- Gradiente Descendente Estocástico (SGD): Atualiza os pesos usando gradientes calculados em pequenos lotes de dados.
- Adam: Combina as vantagens do RMSProp e do Momentum, adaptando a taxa de aprendizado para cada parâmetro.
- RMSProp: Ajusta a taxa de aprendizado com base na média móvel dos gradientes quadráticos.

A preparação adequada dos dados é um passo fundamental no treinamento de autoencoders, influenciando diretamente a eficiência do aprendizado e a qualidade das representações latentes obtidas. Um conjunto de dados bem-preparado permite que o modelo aprenda de maneira mais eficaz, convergindo mais rapidamente e evitando problemas como *overfitting* ou *poor convergence* (MURPHY, 2012). Dois aspectos essenciais na preparação dos dados são:

- Normalização: Escalar os dados para uma faixa adequada (por exemplo, entre 0 e 1) é importante para melhorar a eficiência do treinamento. A normalização assegura que todas as características contribuam igualmente para o aprendizado, evitando que valores em escalas maiores dominem o processo de otimização (MURPHY, 2012).
- Divisão de Dados: Separar os dados em conjuntos de treinamento, validação e teste é essencial para avaliar o desempenho do modelo de forma objetiva. O conjunto de treinamento é usado para ajustar os pesos do modelo, o de validação auxilia na seleção de hiperparâmetros e na prevenção de *overfitting*, e o de teste fornece uma avaliação final da capacidade de generalização do autoencoder (MURPHY, 2012).

A seleção do tamanho do lote, ou batch size, é um aspecto fundamental no treinamento de autoencoders que influencia significativamente o desempenho e a eficiência do modelo. O tamanho do lote determina quantas amostras são utilizadas para calcular o gradiente antes de realizar a atualização dos pesos da rede neural. Essa escolha afeta não apenas a velocidade de treinamento, mas também a estabilidade da otimização e a capacidade de generalização do modelo. Compreender as implicações de utilizar pequenos ou grandes lotes é essencial para ajustar o processo de treinamento de acordo com os recursos disponíveis e os objetivos do modelo (GOODFELLOW; BENGIO; COURVILLE, 2016). Os efeitos de diferentes tamanhos de lote são:

- Pequenos Lotes: Apresentam maior ruído na estimativa do gradiente devido ao número reduzido de amostras, o que pode ajudar o modelo a escapar de mínimos locais e promover uma melhor generalização aos dados não vistos.

- Grandes Lotes: Oferecem uma estimativa mais estável e precisa do gradiente, pois incorporam mais amostras em cada atualização. No entanto, exigem mais memória computacional e podem levar o modelo a convergir para mínimos locais, possivelmente limitando sua capacidade de generalização.

O número de épocas é um parâmetro fundamental no treinamento de autoencoders, referindo-se ao número de vezes que o modelo passa por todo o conjunto de treinamento durante o processo de aprendizado. Cada época representa uma iteração completa sobre todos os dados disponíveis, permitindo que o modelo ajuste seus pesos e biases para minimizar a função de perda definida. Um número maior de épocas oferece ao modelo mais oportunidades para aprender os padrões e características intrínsecas dos dados, potencialmente melhorando sua capacidade de reconstrução e a qualidade das representações latentes. No entanto, é importante equilibrar esse aspecto, pois um número excessivo de épocas pode levar ao *overfitting*, onde o modelo se adapta tão intimamente aos dados de treinamento que perde a capacidade de generalizar para novos dados não vistos (GOODFELLOW; BENGIO; COURVILLE, 2016):

- Mais Épocas tem como vantagem: Permitir que o modelo aprenda melhor, refinando os pesos e capturando nuances nos dados. Desvantagem: Aumenta o risco de *overfitting*, comprometendo a capacidade de generalização do modelo.
- Menos Épocas tem como vantagem: Reduzir o tempo de treinamento e o risco de *overfitting*. Desvantagem: Pode resultar em um modelo subtreinado que não captura adequadamente os padrões dos dados (*underfitting*).

O embaralhamento dos dados é uma prática fundamental no treinamento de autoencoders e outros modelos de aprendizado de máquina. Consiste em aleatorizar a ordem das amostras no conjunto de treinamento antes de cada época. Essa técnica simples desempenha um papel importante na promoção da capacidade de generalização do modelo. Ao embaralhar os dados, evita-se que o modelo aprenda sequências ou padrões específicos inerentes à ordem original das amostras, o que poderia levar ao *overfitting* e comprometer o desempenho em dados não vistos (GOODFELLOW; BENGIO; COURVILLE, 2016).

A taxa de aprendizado é um hiperparâmetro fundamental no treinamento de autoencoders e redes neurais em geral. Ela controla o tamanho dos passos durante a atualização dos pesos do modelo, influenciando diretamente a velocidade de convergência e a estabilidade do processo de otimização. Uma taxa de aprendizado apropriada permite que o modelo aprenda eficientemente os padrões nos dados sem comprometer a convergência (MURPHY, 2012).

No entanto, a escolha desse valor exige um equilíbrio cuidadoso: taxas muito altas podem impedir a convergência, causando oscilações ou divergência no erro de treinamento, enquanto taxas muito baixas podem tornar o processo de treinamento excessivamente lento, prolongando o tempo necessário para o modelo atingir um desempenho satisfatório (GOODFELLOW; BENGIO; COURVILLE, 2016). A importância da taxa de aprendizado e suas implicações no treinamento eficaz de autoencoders são apresentadas a seguir:

- Taxas muito altas podem levar a oscilações no erro de treinamento ou até mesmo impedir a convergência, já que o modelo pode "pular" o mínimo da função de perda.
- Taxas muito baixas resultam em um treinamento lento, exigindo um número maior de épocas para alcançar um nível de erro aceitável, o que pode ser computacionalmente custoso.

Encontrar a taxa de aprendizado ideal é importante e pode envolver técnicas como o uso de otimizadores adaptativos (por exemplo, Adam, RMSProp) que ajustam automaticamente a taxa de aprendizado durante o treinamento, ou a implementação de agendas de aprendizado, onde a taxa é reduzida gradualmente ao longo das épocas (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.7.2 Aplicação de Autoencoders

A redução de dimensionalidade é uma das aplicações mais significativas dos autoencoders, permitindo simplificar conjuntos de dados complexos ao representar suas características essenciais em um espaço de dimensionalidade inferior. Ao aprender representações latentes compactas, os autoencoders capturam as estruturas intrínsecas dos dados, facilitando tarefas como visualização,

armazenamento eficiente e preparação de dados para algoritmos de aprendizado de máquina que são sensíveis à dimensionalidade. Diferentemente de métodos lineares tradicionais, como a Análise de Componentes Principais (PCA), os autoencoders são capazes de modelar relações não lineares entre as características, oferecendo uma representação mais rica e expressiva dos dados. A seguir, discutimos como os autoencoders são aplicados na redução de dimensionalidade e os benefícios que proporcionam em comparação com abordagens convencionais (VINCENT *et al.*, 2008).

A remoção de ruído dos dados, conhecida como desnoising, é outra aplicação fundamental dos autoencoders que melhora significativamente a qualidade dos dados para análises posteriores. Os autoencoders desnoisers são modelos treinados para reconstruir dados originais a partir de versões corrompidas, aprendendo a distinguir entre o sinal verdadeiro e o ruído presente nas entradas. Essa capacidade permite que o modelo capture características latentes essenciais, resultando em representações mais limpas e úteis dos dados (VINCENT *et al.*, 2008).

A detecção de anomalias é uma aplicação importante dos autoencoders em diversos campos, incluindo segurança da informação, monitoramento industrial e análise financeira. Ao serem treinados exclusivamente com dados que representam o comportamento "normal" de um sistema, os autoencoders aprendem a reconstruir essas entradas com alta precisão, minimizando o erro de reconstrução. Quando confrontados com dados anômalos ou que divergem dos padrões aprendidos, os autoencoders geralmente não conseguem reproduzi-los com a mesma eficácia, resultando em erros de reconstrução significativamente maiores. Essa característica permite que os autoencoders atuem como detectores de anomalias, identificando eventos ou padrões que diferem do comportamento esperado (AN; CHO, 2015).

A compressão de imagens é uma aplicação prática e relevante dos autoencoders, que aproveita sua capacidade de aprender representações latentes compactas e significativas dos dados. Os autoencoders podem ser utilizados para compressão de dados ao armazenar a representação latente em vez da imagem completa. Durante o treinamento, o modelo aprende a codificar as imagens de entrada em um espaço latente de dimensionalidade reduzida, preservando as características essenciais que permitem a reconstrução da imagem original. Ao armazenar apenas

essa representação latente, é possível reduzir significativamente o espaço necessário para armazenamento ou transmissão de imagens, o que é especialmente útil em sistemas com restrições de largura de banda ou capacidade de armazenamento limitada (THEIS *et al.*, 2017).

Na fase de decompressão, o decodificador do autoencoder reconstrói a imagem a partir da representação latente, permitindo recuperar uma versão aproximada da imagem original. Essa abordagem de compressão é especialmente vantajosa para dados de alta dimensionalidade, como imagens em alta resolução, onde métodos tradicionais de compressão podem não ser tão eficientes ou podem introduzir artefatos visuais indesejados (THEIS *et al.*, 2017).

A extração de características é uma aplicação fundamental dos autoencoders que tem um impacto significativo em diversas áreas do aprendizado de máquina. As representações latentes aprendidas pelo autoencoder, que são as codificações compactas dos dados de entrada, podem servir como características informativas para outras tarefas de aprendizado, como classificação, *clustering* e regressão. Ao transformar dados brutos em representações de alto nível que capturam as propriedades essenciais dos dados, os autoencoders permitem que algoritmos supervisionados ou não supervisionados operem de maneira mais eficiente e eficaz (ERHAN *et al.*, 2010).

Essa abordagem é especialmente útil quando se lida com dados de alta dimensionalidade ou complexidade, onde a seleção manual de características seria impraticável ou insuficiente. A utilização das representações latentes como características auxilia na redução de dimensionalidade, melhora a generalização dos modelos e pode levar a um desempenho superior nas tarefas subsequentes de aprendizado de máquina (ERHAN *et al.*, 2010).

2.7.3 Detalhes de Implementação

A arquitetura interna dos autoencoders é fundamental para determinar sua capacidade de capturar e representar as características essenciais dos dados de entrada. A construção de camadas envolve a seleção e a disposição das diferentes camadas neurais que compõem o modelo, influenciando diretamente a eficiência e a

eficácia do aprendizado (GOODFELLOW; BENGIO; COURVILLE, 2016). Dependendo do tipo de dados e dos objetivos específicos, diferentes tipos de camadas podem ser utilizadas para otimizar o desempenho do autoencoder:

- Camadas Densas (Fully Connected): Em camadas densas, cada neurônio está conectado a todos os neurônios da camada anterior e posterior. Essa conectividade total permite que o modelo aprenda padrões complexos e interdependentes em todo o conjunto de dados. As camadas densas são versáteis e amplamente utilizadas em autoencoders que lidam com dados tabulares ou onde a estrutura espacial não é primordial.
- Camadas Convolucionais: Projetadas para capturar características espaciais em dados estruturados, como imagens, as camadas convolucionais utilizam filtros que percorrem o input para extrair características locais, como bordas, texturas e formas. Ao preservar a estrutura espacial dos dados, essas camadas permitem que o autoencoder aprenda representações mais robustas e invariantes a transformações, tornando-as ideais para tarefas de processamento de imagens e visão computacional.
- Camadas de Pooling: As camadas de *pooling* reduzem a dimensionalidade espacial dos dados, mantendo as características mais importantes. Ao aplicar operações como *max pooling* ou *average pooling*, o modelo sintetiza regiões de ativação, o que ajuda a reduzir o número de parâmetros, diminuir o risco de *overfitting* e melhorar a eficiência computacional. Essas camadas são frequentemente utilizadas em conjunto com camadas convolucionais para construir autoencoders que equilibram detalhamento e eficiência.

A combinação estratégica dessas camadas permite a construção de autoencoders capazes de aprender representações latentes profundas e significativas dos dados. A escolha e a configuração das camadas devem ser alinhadas com a natureza dos dados e os objetivos do modelo, seja para redução de dimensionalidade, extração de características ou outras aplicações específicas (GOODFELLOW; BENGIO; COURVILLE, 2016).

As funções de ativação são componentes essenciais em autoencoders e redes neurais, responsáveis por introduzir não linearidade nos modelos. Essa não

linearidade permite que as redes neurais capturem relações complexas e padrões intrincados nos dados, indo além das limitações das transformações lineares (NAIR; HINTON, 2010). A escolha apropriada da função de ativação impacta diretamente a capacidade de aprendizado do modelo, a velocidade de convergência e o desempenho geral na tarefa proposta.

A ReLU (Rectified Linear Unit) é amplamente utilizada devido à sua simplicidade e eficiência computacional (NAIR; HINTON, 2010). Ela é definida pela equação 06:

$$(f(x) = \max(0, x)) \quad (06)$$

Essa função retorna zero para valores negativos de xx e mantém os valores positivos inalterados. A ReLU ajuda a mitigar o problema de desvanecimento de gradientes, permitindo que os modelos profundos sejam treinados de forma mais eficaz. Além disso, promove esparsidade nas ativações, o que pode melhorar a capacidade de generalização do modelo (NAIR; HINTON, 2010).

A função Sigmóide comprime a saída em um intervalo entre 0 e 1, sendo útil para modelos probabilísticos e para camadas de saída que representam probabilidades (BISHOP, 2006). Ela é definida pela equação 07:

$$(f(x) = \frac{1}{1 + e^{-x}}) \quad (07)$$

Embora tradicionalmente popular, a função sigmóide pode sofrer com o problema de desvanecimento de gradientes em redes profundas, pois suas derivadas tendem a valores muito pequenos para entradas muito positivas ou muito negativas. Isso pode dificultar o treinamento eficaz de modelos com muitas camadas (BISHOP, 2006).

A função Tanh (Tangente Hiperbólica) é semelhante à sigmóide, mas compressa a saída em um intervalo entre -1 e 1 (LECUN *et al.*, 2012). Ela é definida pela equação 08:

$$(f(x) = \tanh(x)) \quad (08)$$

Por ser uma versão escalonada e deslocada da sigmóide, a tanh também pode sofrer com o desvanecimento de gradientes, mas possui a vantagem de centralizar os dados em torno de zero. Essa centralização pode levar a uma convergência mais rápida durante o treinamento, já que as ativações podem ser positivas ou negativas, permitindo uma melhor dinâmica na atualização dos pesos (LECUN *et al.*, 2012).

A seleção da função de ativação adequada deve considerar a natureza dos dados, a profundidade da rede e o comportamento desejado para as ativações. Em muitos casos, uma combinação de diferentes funções de ativação é empregada em diferentes camadas para otimizar o desempenho do autoencoder (LECUN *et al.*, 2012).

2.7.4 Otimizadores

Os otimizadores são componentes essenciais no treinamento de autoencoders e outras redes neurais, pois controlam como os pesos do modelo são ajustados durante a minimização da função de perda. Diferentes otimizadores implementam variações nos métodos de atualização de pesos, influenciando a convergência, a estabilidade e o desempenho geral do modelo (SUTSKEVER *et al.*, 2013).

O Gradiente Descendente Estocástico (SGD) com Momentum é uma extensão do SGD padrão que acelera a convergência em direções relevantes, enquanto reduz oscilações em dimensões menos importantes. O momentum age como um amortecedor, acumulando gradientes passados e orientando o processo de atualização para seguir a direção média, em vez de reagir fortemente a gradientes ruidosos em cada iteração. Isso é particularmente útil em funções de perda com platôs ou regiões complexas, onde o SGD tradicional pode ser ineficiente. As vantagens do

SGD com Momentum são: reduzir oscilações durante a convergência e acelerar o treinamento em regiões de gradiente suave (SUTSKEVER *et al.*, 2013).

O Adagrad (Adaptive Gradient Algorithm) ajusta dinamicamente a taxa de aprendizado para cada parâmetro com base na frequência de suas atualizações. Parâmetros que recebem gradientes maiores têm sua taxa de aprendizado reduzida, enquanto aqueles com gradientes menores têm sua taxa aumentada. Essa adaptação torna o Adagrad adequado para problemas de alta dimensionalidade e esparsidade, onde diferentes características podem ter diferentes relevâncias (DUCHI; HAZAN; SINGER, 2011).

As vantagens do Adagrad são: adequação para problemas de esparsidade (por exemplo, NLP ou recomendação) e não exigência de ajuste manual constante da taxa de aprendizado. A limitação: pode levar a uma taxa de aprendizado excessivamente pequena ao longo do tempo, dificultando a convergência em etapas posteriores (DUCHI; HAZAN; SINGER, 2011).

Adamax e Nadam são variantes do Adam (Adaptive Moment Estimation), um dos otimizadores mais populares devido à sua eficácia em uma ampla gama de problemas:

- Adamax: Baseado no Adam, utiliza a norma L^∞ (máximo absoluto) em vez da norma L_2 para calcular atualizações. Isso o torna mais robusto a gradientes muito grandes, sendo eficaz em cenários onde os dados são ruidosos ou têm outliers (KINGMA; BA, 2015).
- Nadam (Nesterov-accelerated Adaptive Moment Estimation): Combina as vantagens do Adam com o método de gradiente acelerado de Nesterov, que ajusta os passos de atualização para incluir informações sobre a direção futura do gradiente. Isso pode melhorar a convergência em cenários complexos (RUDER, 2016).

A vantagem do Adamax é a Resistência a gradientes extremos, mantendo a estabilidade (KINGMA; BA, 2015). Ao passo que a vantagem do Nadam é acelerar a convergência, especialmente em problemas não convexos (RUDER, 2016).

2.7.5 Hiperparâmetros

O ajuste de hiperparâmetros é uma etapa essencial no treinamento de autoencoders e outros modelos de aprendizado de máquina. Os hiperparâmetros, como a taxa de aprendizado, o número de camadas, o tamanho do lote e o número de neurônios em cada camada, têm um impacto significativo no desempenho e na eficiência do modelo. Selecionar a configuração ideal desses parâmetros pode melhorar a precisão do modelo, acelerar o treinamento e evitar problemas como overfitting ou underfitting (BERGSTRA; BENGIO, 2012).

A validação cruzada (Cross-Validation) é uma técnica sistemática para avaliar o desempenho do modelo com diferentes configurações de hiperparâmetros. O conjunto de dados é dividido em várias partições (*folds*), e o modelo é treinado e validado em diferentes combinações dessas partições. Isso fornece uma estimativa robusta do desempenho do modelo em dados não vistos, ajudando a identificar configurações de hiperparâmetros que generalizam melhor. A validação cruzada reduz o viés de avaliação ao usar múltiplas partições e ajuda a identificar configurações que evitam *overfitting* (BERGSTRA; BENGIO, 2012).

A busca em grade (Grid Search) é uma abordagem exaustiva que testa todas as combinações possíveis de hiperparâmetros dentro de um espaço pré-definido. Para cada combinação, o modelo é treinado e avaliado, permitindo identificar a configuração que maximiza o desempenho (BERGSTRA; BENGIO, 2012).

A principal vantagem da busca em grade é que ela garante a exploração completa do espaço de busca, testando todas as combinações possíveis de hiperparâmetros dentro de um intervalo pré-definido. Isso a torna ideal para problemas em que o número de combinações é gerenciável, permitindo identificar com precisão a configuração que maximiza o desempenho do modelo. No entanto, sua principal limitação é o alto custo computacional, especialmente em espaços de hiperparâmetros grandes, onde o número de combinações cresce exponencialmente, tornando o processo impraticável em termos de tempo e recursos computacionais (BERGSTRA; BENGIO, 2012).

A busca aleatória (Random Search) explora o espaço de hiperparâmetros de forma estocástica, selecionando combinações de parâmetros de maneira aleatória

dentro de um intervalo pré-definido. Isso permite cobrir uma maior variedade de configurações em menos tempo (GOODFELLOW; BENGIO; COURVILLE, 2016).

A busca aleatória apresenta vantagens significativas em relação à busca em grade, especialmente em espaços de hiperparâmetros grandes. Ela é mais eficiente, pois foca na exploração de uma variedade maior de combinações, economizando tempo e recursos computacionais. Além disso, oferece uma maior probabilidade de encontrar boas configurações ao explorar regiões do espaço de busca que poderiam ser negligenciadas por métodos mais exaustivos. No entanto, sua principal limitação é não garantir a avaliação de todas as combinações possíveis, o que pode resultar na ausência de teste de configurações potencialmente ótimas (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.7.6 Regularização

A regularização é uma estratégia essencial para melhorar a capacidade de generalização dos autoencoders e prevenir o *overfitting*, especialmente em redes neurais complexas e profundas. As técnicas de regularização atuam impondo restrições ou modificações durante o treinamento do modelo, garantindo que ele não se ajuste excessivamente aos dados de treinamento (SRIVASTAVA et al., 2014).

O *dropout* é uma técnica que desativa aleatoriamente neurônios durante o treinamento, tornando suas ativações igual a zero em cada passo. Essa desativação aleatória força a rede a não depender de combinações específicas de neurônios, promovendo a aprendizagem de representações mais robustas e reduzindo a probabilidade de *overfitting*. Durante a inferência, todos os neurônios são reativados, e suas ativações são escaladas para refletir a média dos valores aprendidos (SRIVASTAVA et al., 2014).

As penalizações L1 e L2 adicionam termos à função de perda para controlar a magnitude dos pesos:

- L1 Regularization: Penaliza a soma dos valores absolutos dos pesos, incentivando a esparsidade no modelo, ou seja, muitos pesos tendem a ser reduzidos a zero (NG, 2004). Conforme apresentado na equação 09.

$$\left(\mathcal{L}1 \ loss = \lambda \sum_i |w_i| \right) \quad (09)$$

- L2 Regularization (ou Ridge): Penaliza a soma dos quadrados dos pesos, resultando em pesos menores, mas sem forçá-los a zero (NG, 2004). Conforme apresentado na equação 10.

$$\left(\mathcal{L}2 \ loss = \lambda \sum_i w_i^2 \right) \quad (10)$$

A regularização por penalização L1 e L2 oferece vantagens distintas e complementares no treinamento de modelos de aprendizado de máquina. A penalização L1 incentiva a esparsidade, promovendo a redução de pesos a zero, o que é particularmente útil para seleção de características relevantes em modelos de alta dimensionalidade. Já a penalização L2 contribui para a estabilidade do modelo ao evitar pesos extremamente grandes, reduzindo a sensibilidade a variações nos dados. Essas técnicas são frequentemente combinadas com outras estratégias de treinamento, como dropout e normalização, para melhorar a robustez do modelo e alcançar um equilíbrio eficaz entre a aprendizagem e a generalização (NG, 2004).

2.8 AUTOENCODERS LSTM

Os autoencoders LSTM (Long Short-Term Memory) são uma extensão dos autoencoders tradicionais que incorporam redes neurais recorrentes especializadas no processamento de dados sequenciais ou temporais. Enquanto os autoencoders convencionais são eficazes para dados estáticos, os autoencoders LSTM são projetados para capturar dependências temporais e padrões em sequências de dados, como séries temporais, texto e sinais biológicos (MALHOTRA, *et al.*, 2015).

A arquitetura básica de um autoencoder LSTM mantém a estrutura de codificador e decodificador, mas utiliza camadas LSTM em vez de camadas totalmente conectadas ou convolucionais. O codificador LSTM processa a sequência

de entrada, elemento por elemento, e aprende uma representação latente que captura as dependências temporais. O estado oculto final do codificador serve como a codificação latente da sequência completa (SUTSKEVER; VINYALS; LE, 2014; MALHOTRA, *et al.*, 2015).

O decodificador LSTM recebe a representação latente e tenta reconstruir a sequência original, novamente elemento por elemento. Muitas vezes, é utilizado um mecanismo de inicialização do estado oculto do decodificador com o estado final do codificador para facilitar a reconstrução (SUTSKEVER; VINYALS; LE, 2014; MALHOTRA, *et al.*, 2015).

O autoencoder LSTM tem funções de perda específicas para sequências, onde a função de perda tem de considerar a natureza temporal dos dados:

- Erro Quadrático Médio Sequencial: Calcula o MSE em cada passo temporal e faz a média ao longo da sequência. Onde (T) é o comprimento da sequência. Conforme apresentado na equação 11.

$$\left(\mathcal{L}_{seq} = \frac{1}{nT} \sum_{i=1}^n \sum_{t=1}^T (x_{i,t} - \hat{x}_{i,t})^2 \right) \quad (11)$$

- Entropia Cruzada Sequencial: Usada para sequências categóricas, como texto.

2.8.1 Treinamento de Autoencoders LSTM

O treinamento de autoencoders baseados em LSTM envolve diversas etapas importantes para garantir a eficácia do modelo. Entre essas etapas, destacam-se a preparação dos dados e as estratégias de treinamento empregadas (MALHOTRA, *et al.*, 2015). A preparação dos dados se dá antes de treinar o modelo, os dados precisam ser padronizados para garantir consistência e facilitar o aprendizado. Para isso, duas abordagens são comumente utilizadas:

- *Padding* e Truncamento: Como as sequências podem ter comprimentos variados, elas são ajustadas por meio de preenchimento ou truncamento, garantindo que todas tenham o mesmo tamanho.
- Normalização Temporal: As características dos dados podem variar ao longo do tempo, exigindo normalizações específicas para alinhar os valores e melhorar a estabilidade do treinamento.

Durante o treinamento do modelo, estratégias específicas são adotadas para otimizar o desempenho e reduzir discrepâncias entre as fases de treinamento e inferência:

- *Teacher Forcing*: Essa técnica acelera a convergência ao fornecer ao decodificador o valor real da sequência anterior como entrada, em vez de sua própria previsão.
- *Scheduled Sampling*: Uma abordagem híbrida que combina o uso de valores reais e previstos como entradas do decodificador, diminuindo a diferença entre os comportamentos observados durante o treinamento e a inferência.

Essas etapas e estratégias são fundamentais para garantir que os autoencoders LSTM sejam capazes de aprender e reproduzir padrões complexos em séries temporais de forma eficiente (SUTSKEVER; VINYALS; LE, 2014).

2.8.2 Aplicações de Autoencoders LSTM

Ao aprender os padrões normais de sequências temporais, o autoencoder LSTM é capaz de identificar anomalias quando o erro de reconstrução ultrapassa um determinado limiar. Essa característica torna o modelo uma ferramenta poderosa para monitorar sistemas e detectar desvios que indicam comportamentos inesperados ou falhas nos dados analisados (MALHOTRA, *et al.*, 2015).

Além da detecção de anomalias, o autoencoder LSTM pode ser aplicado em outras tarefas. Por exemplo, o estado latente aprendido pelo modelo pode ser utilizado para previsão de valores futuros em séries temporais, aproveitando a capacidade do LSTM de capturar dependências de longo prazo. Em tarefas de Processamento de Linguagem Natural (NLP), os autoencoders LSTM também podem ser empregados para aprendizagem não supervisionada de representações, contribuindo para

modelagem de linguagem e tradução automática. Essas aplicações destacam a versatilidade do modelo em diferentes contextos (MALHOTRA, *et al.*, 2015).

2.8.3 Detalhes de Implementação

A implementação de um modelo LSTM eficiente envolve decisões importantes relacionadas à construção das camadas. Camadas LSTM empilhadas são frequentemente utilizadas para aumentar a capacidade do modelo de capturar padrões complexos presentes nas séries temporais. Além disso, o uso de LSTMs bidirecionais permite que a sequência seja processada tanto no sentido direto quanto no reverso, possibilitando a captura de dependências temporais passadas e futuras de forma mais completa (MALHOTRA, *et al.*, 2015).

Outro aspecto fundamental está na configuração dos hiperparâmetros. O tamanho do estado oculto é importante, pois define a capacidade de memória do modelo e sua habilidade de armazenar informações ao longo do tempo. O número de camadas também desempenha um papel importante: enquanto camadas adicionais podem permitir a captura de padrões mais abstratos, elas também aumentam a complexidade do modelo, elevando o risco de *overfitting* caso não sejam aplicadas técnicas adequadas de regularização (MALHOTRA, *et al.*, 2015).

Por fim, a escolha das funções de ativação e dos otimizadores impacta diretamente o desempenho do modelo. As funções de ativação *tanh* e sigmóide, intrínsecas às células LSTM, regulam a entrada, saída e esquecimento de informações ao longo das sequências. Para o treinamento, otimizadores como Adam e RMSProp são populares. O Adam é amplamente utilizado devido à sua eficiência em problemas com muitos parâmetros, enquanto o RMSProp é particularmente adequado para RNNs, pois lida bem com os desafios de gradientes em sequências longas (MALHOTRA, *et al.*, 2015).

Os autoencoders LSTM expandem o alcance dos autoencoders para domínios onde a informação sequencial é fundamental. Sua capacidade de aprender e reconstruir sequências complexas os torna ferramentas valiosas em campos que lidam com dados temporais ou sequenciais (MALHOTRA, *et al.*, 2015).

3. REVISÃO DA LITERATURA

Neste capítulo, foi explorado as publicações acadêmicas recentes, traçando o panorama do uso de algoritmos de aprendizado de máquina no contexto de máquinas e equipamentos industriais, com ênfase especial na análise e detecção de falhas através de séries temporais binárias.

Para se realizar a revisão sistemática da literatura sobre aprendizado de máquina aplicada à análise e detecção de falhas em máquinas e equipamentos, foi seguido a metodologia *Transparent Reporting of Systematic Reviews and Meta-Analyses* (PRISMA) mais atual, que é uma estrutura para garantir a transparência e a completude em revisões sistemáticas (PAGE *et al.*, 2021). De forma que se procurou seguir suas diretrizes rigorosamente, especialmente no que diz respeito à documentação do processo de revisão, os detalhes da revisão se encontram nos apêndices I e II.

3.1 ALGORITMOS UTILIZADOS EM ANÁLISE E DETECÇÃO DE FALHAS EM MÁQUINAS E EQUIPAMENTOS INDUSTRIAIS

A análise dos 98 estudos selecionados pelo protocolo da revisão sistemática fornece uma visão da trajetória e do foco atual no campo de aprendizado de máquina em análise e detecção de falhas em máquinas e equipamentos. A inclinação das pesquisas mais recentes pode ser reflexo do rápido desenvolvimento tecnológico, bem como da crescente relevância e aplicação prática desses estudos nesta área.

A tendência dos algoritmos de aprendizado de máquina utilizados na análise e detecção de falhas em máquinas e equipamentos, revela uma evolução significativa ao longo do tempo, de 2008 a 2024.

- 2008 e 2011: Nestes primeiros anos, o destaque é para o uso de Modelos Ocultos de Markov, Hidden Markov Models (HMM), que são algoritmos eficazes para tarefas que envolvem sequências temporais. Isso indica um foco em métodos estatísticos clássicos.
- 2012: Observa-se uma transição com o uso de Máquinas de Vetores de Suporte, Support Vector Machines (SVM), um algoritmo poderoso para classificação e regressão em problemas de aprendizado supervisionado.

- 2015: A variedade de algoritmos aumenta, incluindo: Regressão Linear Múltipla, Multiple Linear Regression (MLR), Redes Neurais Artificiais, Artificial Neural Networks (ANN), e Máquinas de Aprendizado Extremo, Máquinas de Aprendizado Extremo (ELM), demonstrando um aumento na adoção de técnicas de aprendizado de máquina.
- 2017-2018: O foco muda para uma mistura de algoritmos tradicionais de aprendizado de máquina e aprendizado profundo, com a introdução de Redes Neurais Profundas, Deep Neural Networks (DNN), Naive Bayes (NB), e Árvores de Classificação e Regressão, Classification and Regression Trees (CART), além de continuar o uso de SVM.
- 2019-2020: Ocorre uma consolidação dos métodos de aprendizado profundo, com o uso frequente de Long Short-Term Memory (LSTM) e Random Forest (RF), evidenciando uma preferência por algoritmos que lidam bem com dados sequenciais e complexos.
- 2021-2022: Esses anos mostram a diversificação e especialização ainda maiores com algoritmos como XGBoost, Feedforward Neural Networks (FNN), Redes Neurais Convolucionais, Convolutional Neural Networks (CNN), e Redes Generativas Adversárias, Generative Adversarial Networks (GAN). Isso reflete a evolução e adaptação às necessidades crescentes de processamento de dados mais complexos.
- 2023-2024: Os últimos anos listados mostram uma continuação da tendência de aprendizado profundo com Autoencoder, Q-learning (um método de aprendizado por reforço), e Sistemas de Lógica Fuzzy de Tipo 2 (T2FLS). Isso sugere uma busca por soluções mais sofisticadas e adaptativas para desafios de detecção de falhas.

Em resumo, a tendência geral é uma progressão dos métodos estatísticos clássicos para técnicas mais avançadas de aprendizado de máquina e aprendizado profundo. Os algoritmos de aprendizado profundo são apresentados pelos estudos como uma solução capaz de revolucionar as práticas de manutenção preditiva em ambientes industriais (NAMUDURI *et al.*, 2020).

Embora soluções de aprendizado de máquina mais clássicas sejam discutidas no estudo de Padmanabh, *et al.* (2021) em um método para prever alarmes em

chillers, os autores analisam falhas através de dados de séries temporais, no entanto os dados não binários (que é o foco desta tese). Os autores descrevem como as informações sobre o progresso dos valores dos sensores ao longo do tempo, podem ser mais relevantes do que apenas os dados que acionam o alarme. Diversos algoritmos, como classificadores baseados SVM, regressão logística (LR) e KNN, são testados para construir modelos de previsão de alarmes com base em séries temporais de dados de sensores.

Técnicas diferentes também são utilizadas por Kwon e Kim (2020) que propõem um modelo de previsão de falhas utilizando seleção iterativa de características para ambientes de Internet Industrial das Coisas (IIoT), no entanto os dados utilizados no estudo não são séries temporais. A relevância entre cada característica e a falha no estudo é analisada usando RF. Uma nova característica é selecionada em cada iteração com base em sua importância e adicionada ao conjunto de características selecionadas. O modelo de previsão de falhas através de Máquina de Vetores de Suporte (SVM) é construído com base no conjunto de características selecionadas.

Seguindo a mesma linha, Li *et al.* (2020) analisa séries temporais de vibração provenientes de equipamentos mecânicos, como motores e rolamentos. Esses dados, não binários, são processados e codificados de forma a representar as características das falhas de maneira mais eficaz para a detecção por algoritmos de aprendizado de máquina. Segundo os autores, a abordagem proposta visa superar as limitações de métodos tradicionais de diagnóstico de falhas, que muitas vezes exigem cálculos complexos de vetores de características e podem não ser robustos o suficiente para lidar com uma ampla gama de perturbações externas.

O trabalho de Srivastava e Parida (2022) também segue a mesma metodologia para detecção de falhas, no entanto em dados que não são séries temporais. As técnicas principais utilizadas são SVM e Regressão de Processo Gaussiano (GPR), neste caso, a detecção de falhas também é tratada como um problema de classificação, onde o SVM é empregado para identificar o tipo de falha e a linha afetada. Por outro lado, a previsão de localização da falha é abordada com a técnica de GPR, que se baseia na teoria da probabilidade condicional.

Da mesma forma, as metodologias empregadas por Rocchetta, Mavroeidis e Petkovic (2022) incluem a combinação de modelos de aprendizado de máquina, como

SVM, com informações de estrutura do sistema e dados de séries temporais com informações de tempo até a falha. As técnicas utilizadas envolvem a geração de dados sintéticos para experimentos iniciais, treinamento de SVMs com margem suave e cálculo do número de vetores de suporte. Além disso, são obtidos parâmetros de confiabilidade e validade dos limites de erro por meio de estimativas numéricas.

Ao utilizar algoritmos como DT, *Adaptive Boosting* (AdaBoosting), Ridge Classifier (L2), LDA, SVM, RF, *Naive Bayes* (NB), KNN, *Gradient Boosting Machines* (GBM) e LR, o modelo seleciona os modelos básicos de melhor desempenho com base na precisão, área sob a curva (AUC) e pontuação F1. Através deste método de conjunto, o modelo alcança melhorias significativas em exatidão, precisão, recuperação e pontuação F1 em comparação com os métodos de empilhamento existentes (KIM; JEONG, 2022).

Segundo uma linha similar, o estudo apresentado por Kajmakovic *et al.* (2023) envolve a análise de discrepâncias nos sinais de relés redundantes, utilizando técnicas de aprendizado de máquina para identificar e isolar falhas. Os dados utilizados incluem sinais de discrepância entre os relés redundantes, bem como características estatísticas como assimetria, curtose e fator de forma, mas não são séries temporais. Além disso, o algoritmo realiza uma classificação binária por meio de algoritmos como DT, RF, KNN, NB e SVM.

Ericeira *et al.* (2020) utiliza técnicas como Redes Neurais Artificiais (ANN), especificamente o Perceptron Multicamadas (MLP), e RF para aprender padrões nos dados de ultrassom e generalizar para componentes não vistos em dados que não são séries temporais. Em resumo, a metodologia adotada neste estudo combina técnicas de sensoriamento ultrassônico, aprendizado de máquina e reconhecimento de padrões de sinal para desenvolver um modelo de detecção precoce de falhas em roletes de transportadores de correia, com foco na eficiência na classificação de roletes defeituosos e não defeituosos.

A pesquisa de Surantha e Gozali (2023a) apresenta uma metodologia abrangente que inclui coleta de dados, extração de características, desenvolvimento de modelos e avaliação de desempenho. Os dados utilizados contêm informações de telemetria em tempo real, registros de erros, histórico de manutenção e histórico de falhas de 100 máquinas com quatro componentes e sensores. Para a classificação

multiclasses de falhas de máquinas, são empregadas técnicas como LR, RF e ANN. Além disso, o artigo destaca o uso de Seleção de *feature* e métodos de reamostragem para lidar com desequilíbrios nos dados, visando melhorar a precisão dos modelos.

A pesquisa de Ma *et al* (2020) apresenta uma combinação de metodologias e técnicas para obter medições de pressão precisas sem sensores físicos, que difere da proposta desta tese que analisa diretamente os dados dos sensores físicos. O modelo é baseado em redes neurais *Long Short-Term Memory* (LSTM), um tipo de rede neural recorrente conhecida por sua capacidade de lidar com dados de séries temporais de forma eficaz.

A principal técnica empregada por Ma *et al* (2020) neste modelo é a Otimização Bayesiana (BO), que é usada para otimizar os hiper parâmetros do modelo LSTM. A BO ajuda no ajuste fino do modelo para melhorar seu desempenho na previsão de medições de pressão hidráulica. Além disso, o estudo avalia o desempenho do modelo proposto em comparação com outros modelos, como rede neural de retropropagação (BPNN), rede neural convolucional (CNN) e rede neural recorrente (RNN).

Embora o modelo Ma *et al* (2020) apresente resultados promissores para este caso de utilização específico, a sua generalização para outros tipos de máquinas ou equipamentos pode ser limitada. Pois o estudo menciona a necessidade de controlar os erros de previsão dentro de uma determinada faixa (por exemplo, dentro de 30 bar) para evitar mau funcionamento no sistema hidráulico da escavadeira. O limite específico para erros de previsão aceitáveis pode variar dependendo da aplicação.

A abordagem de aprendizado profundo de Azzouz *et al.* (2020) usando o classificador LSTM também mostrou resultados promissores em comparação com técnicas tradicionais de aprendizado de máquina para detecção de falhas em Redes de Sensores Sem Fio (RSSFs). No estudo, o LSTM demonstrou maior precisão e confiabilidade em tarefas de classificação de múltiplas falhas.

Especificamente, o classificador LSTM manteve uma precisão de detecção entre 80% e 90% mesmo com altas taxas de falhas, superando o SVM e outros classificadores tradicionais como RF, *Probabilistic Neural Network* (PNN) e MLP. Quando as taxas de falhas aumentaram, a precisão de todos os classificadores tendeu

a diminuir, mas o LSTM continuou sendo a técnica mais precisa segundo Azzouz et al. (2020), no entanto este estudo não analisa dados binários como nesta tese.

Além disso, em termos de Coeficientes de Correlação de Matthews (MCC), o LSTM ficou em segundo lugar depois do SVM, indicando sua eficácia na detecção correta de falhas na rede. LSTM e PNN também exibiram taxas de verdadeiros positivos (TPR) mais altas em comparação com outros classificadores, mostrando sua capacidade de detectar e classificar falhas com precisão de acordo com Azzouz et al. (2020).

No geral, a abordagem de aprendizado profundo usando o classificador LSTM provou ser uma técnica robusta e eficaz para detecção de falhas em RSSFs no estudo de Azzouz et al. (2020), superando os métodos tradicionais de aprendizado de máquina em termos de precisão, confiabilidade e métricas de desempenho.

Rahhal e Abualnadi (2020) demonstra que *Long Short-Term Memory Recurrent Neural Network* (LSTM-RNN) oferece várias vantagens sobre Vanilla-RNN no contexto de manutenção preditiva, em dados de séries temporais não binárias. Os autores resumem que o LSTM-RNN oferece maior precisão de previsão, melhor manuseio de sequências longas, menores erros de previsão e adequação para dispositivos críticos, tornando-o uma escolha valiosa para aplicações de manutenção preditiva onde a precisão e a confiabilidade são fatores-chave.

Outros autores que também confirmam essa vantagem são Ramu e Narayanan (2023), os autores afirmam que os modelos redes neurais recorrentes (RNN) e LSTM desempenham um papel importante na manutenção preditiva na agricultura, analisando diferentes características das culturas e iniciando atividades de manutenção para sensores associados às terras agrícolas. Esses modelos ajudam a prever falhas de sensores, reduzindo o tempo de inatividade e permitindo o monitoramento eficaz da saúde da colheita, mas também não analisam dados binários.

Ao considerar fatores como umidade, temperatura do solo, luminosidade e outras variáveis ambientais, os modelos RNN e LSTM podem fornecer previsões precisas para necessidades de manutenção em sistemas agrícolas. O uso de algoritmos de propagação direta e reversa com atributos de conjunto de dados

aprimora o ecossistema de previsão, permitindo a detecção precoce de falhas de sensores com base em dados treinados e melhorando a eficiência geral no manejo das culturas (RAMU; NARAYANAN, 2023).

Outra corrente de estudos também é percebida nos últimos anos, agora com algoritmos ainda mais avançados, o estudo apresentado por Mansouri e Vadera (2022) concentra-se no desenvolvimento de um modelo explicável e profundo para previsão de falhas usando sensores de vibração instalados em rolamentos industriais. As metodologias empregadas incluem o uso de técnicas de aprendizagem profunda, particularmente RNN, LSTM, Unidade Recorrente Fechada (GRU), Redes Neurais Convolucionais (CNN) e vários modelos híbridos que combinam essas arquiteturas.

Para melhorar a Interpretabilidade dos modelos de aprendizagem profunda, o estudo de Mansouri e Vadera (2022) incorpora técnicas de seleção de recursos por instância para identificar os recursos mais influentes para resultados de previsão de falhas. Um dos métodos desenvolvidos no estudo é o *Gumbel-Sigmoid eXplanator* (GSX), que serve como um seletor de recursos diferenciáveis e treináveis. Esta abordagem visa fornecer insights sobre o processo de tomada de decisão dos modelos de aprendizagem profunda e melhorar a transparência nas tarefas de previsão de falhas.

A pesquisa de Maurya, *et al.* (2021) descreve uma abordagem inovadora de Monitoramento Baseado em Condições (CBMT) em Condições Variáveis de Funcionamento de Máquinas usando Transferência de Conhecimento em Nível Baixo com Redes Neurais Profundas (DNN). A metodologia principal envolve a extração de características de baixo nível dos dados brutos das máquinas, treinamento de um modelo DNN com essas características e transferência de conhecimento para adaptar o modelo a diferentes domínios de dados.

Os tipos de dados abordados e analisados por Maurya, *et al.* (2021) são dados brutos de sensores coletados de máquinas em diferentes condições de funcionamento, com foco em identificar estados de saúde, tamanhos de falhas, velocidade do motor e carga na máquina, no entanto os dados não séries temporais. A abordagem proposta visa lidar com a variação desses dados em diferentes domínios de máquinas, garantindo que o desempenho no domínio-alvo não seja pior do que no

domínio de origem, através da transferência de conhecimento em nível baixo com DNN.

No estudo realizado por Wahid, Breslin e Intizar (2022), a estrutura híbrida CNN-LSTM foi destacada como uma solução tecnológica promissora para previsão de falhas de máquinas no contexto de Indústria 4.0, combinando CNN e LSTM. As CNNs são empregadas eficientemente para processar dados baseados em imagens em contextos de manutenção preditiva, sendo particularmente eficazes na extração de recursos de alto nível dos dados, uma funcionalidade importante para analisar dados de sensores em plantas e fábricas industriais.

Em paralelo, os LSTMs são utilizados na estrutura para explorar as relações entre diferentes variáveis em dados de série temporal não binária, aproveitando sua capacidade de memória. Essa funcionalidade é fundamental, pois os LSTMs são hábeis em capturar padrões de longo prazo em dados de séries temporais, tornando-os particularmente adequados para a previsão de falhas de máquinas baseadas em dados históricos de sensores (WAHID; BRESLIN; INTIZAR, 2022).

Adicionalmente, a estrutura híbrida CNN-LSTM incorpora conexões de salto, um elemento que desempenha um papel vital em melhorar o fluxo de informações e gradientes durante o treinamento do modelo. Essas conexões permitem um fluxo mais eficiente de informações entre as camadas, contribuindo para previsões mais acuradas e robustas (WAHID; BRESLIN; INTIZAR, 2022).

Ao sintetizar os pontos fortes das CNNs na extração de características e dos LSTMs na análise de dados de séries temporais, a estrutura Hybrid CNN-LSTM se estabelece como uma abordagem poderosa para a manutenção preditiva nos ambientes avançados da Indústria 4.0. A combinação dessas tecnologias representa um avanço significativo na capacidade de prever falhas de máquinas, destacando a inovação contínua no campo da manutenção preditiva (WAHID; BRESLIN; INTIZAR, 2022).

No estudo conduzido por Kim *et al.* (2024), foi destacado o papel fundamental dos modelos CNN na classificação de padrões de Descarga Parcial Resolvida em Fase (PRPD) em transformadores de resina fundida. A eficácia das CNNs nesse contexto deve-se à sua capacidade de extrair características significativas de dados

de imagem complexos e aprender efetivamente as características locais das imagens. As CNNs são utilizadas para o reconhecimento e classificação precisos de padrões de descarga parcial, essenciais para identificar defeitos específicos ou prevenir acidentes com antecedência.

Treinando o modelo de CNN com padrões de PRPD medidos por meio de defeitos artificiais e padrões de ruído de transformadores de resina fundida, o modelo aprende a diferenciar entre diferentes tipos de padrões de descarga associados a defeitos ou degradação. A capacidade do modelo CNN de analisar e classificar esses padrões contribui para a detecção precoce de riscos potenciais em transformadores de resina fundida, aprimorando as práticas de monitoramento e manutenção (KIM *et al.*, 2024).

Wang (2022) apresenta uma abordagem inovadora para prever o estado de sensores e medidores, focando na resolução do problema de classificação desbalanceada. A metodologia empregada envolve o uso de Redes Generativas Adversariais (GANs). Em termos de técnicas utilizadas, o trabalho explora a construção de funções objetivas relevantes para o modelo CatGAN-SP, as quais desempenham um papel fundamental na melhoria da precisão da predição do estado dos sensores. Além disso, são discutidas estratégias para otimizar o desempenho das redes geradoras e discriminadoras, garantindo um equilíbrio dinâmico durante o treinamento.

Embora o CatGAN-SP seja eficaz na abordagem do problema de classificação desbalanceada em sensores de fluxo ultrassônico, Wang (2022) não apresenta a aplicação para outros tipos de sensores e ambientes. Outra limitação potencial é a falta de discussão sobre o impacto de possíveis ruídos nos dados dos sensores e como o modelo CatGAN-SP lida com essas incertezas. A consideração de dados ruidosos e a avaliação da sensibilidade do modelo a essas variações poderiam fornecer insights adicionais sobre a confiabilidade e a precisão das previsões do estado dos sensores em ambientes do mundo real.

O módulo de detecção de falhas baseado em GAN no modelo de Wang *et al.* (2021) desempenha um papel importante na detecção de falhas em dados de séries temporais multivariadas. Ao contrário dos métodos tradicionais de aprendizagem supervisionada, o discriminador treinado em GAN no modelo CT-GAN aprende a

detectar dados falsos de maneira não supervisionada, tornando-o uma técnica atraente para detecção de falhas. Este módulo ajuda a capturar correlações entre múltiplas variáveis e dependências temporais dentro de cada variável, permitindo a identificação de falhas que possam indicar potenciais falhas no equipamento.

Züfle *et al* (2021) apresenta uma metodologia de Manutenção Preditiva que se destaca por sua abordagem abrangente e genérica para prever o tempo até a falha de máquinas industriais. Ela se baseia em técnicas de seleção de recursos e transformação, além de algoritmos como RF, *eXtreme Gradient Boosting* (XGBoost), *Feed-Forward Neural Network* (FNN) e *Automated Machine Learning* (AutoML) para previsão de tempo até a falha. Os resultados obtidos em um estudo de caso específico podem não se aplicar diretamente a outras máquinas ou contextos industriais, exigindo validação adicional em diferentes cenários.

No estudo de Steurewagen e Poel (2021) foi implementada uma abordagem de aprendizado de máquina supervisionado, com ênfase na previsão e diagnóstico de falhas em máquinas, utilizando um modelo XGBoost treinado com uma combinação de dados de sensores não binários e relatórios. O modelo XGBoost é uma técnica de *boosting* extremo de gradiente, conhecida por sua eficácia em problemas de previsão.

Na fase de diagnóstico, Steurewagen e Poel (2021) demonstram que os insights são enriquecidos com valores de *Shapley*, que são utilizados para fornecer uma visão mais precisa aos operadores da planta. Esses valores identificam as principais causas de situações de alto risco e podem ser empregados para oferecer suporte adicional aos operadores em inspeções e manutenções. Os tipos de dados utilizados incluem informações de sensores, relatórios de inspeção e dados de eventos, que são agregados e tratados para fornecer uma visão preditiva e diagnóstica das falhas nas máquinas.

Em uma abordagem metodológica distinta em seu estudo, Liu *et al.* (2023) descreve uma metodologia principal para o diagnóstico de falhas em sensores e atuadores de frenagem, que combina métodos baseados em modelos e estimativa de pressão baseada em dados. As técnicas específicas de análise e experimentação utilizadas incluem utilização de redes neurais como MLP, LSTM, GRU e Transformer para estimar a pressão do cilindro de roda.

Explorando uma metodologia diferente, Li *et al.* (2020) apresenta um método de diagnóstico de falhas baseado na codificação de séries temporais e na utilização de CNN. A metodologia proposta envolve a transformação de sinais de vibração unidimensionais em imagens bidimensionais para facilitar a detecção de falhas em equipamentos mecânicos. Além disso, são exploradas técnicas de codificação de séries temporais, como a transformação em imagens 2D, para melhorar a eficiência do processo de diagnóstico de falhas.

Yang *et al.* (2018) utiliza Sparse Autoencoder (SAE) para extração automática de características dos sinais, o que é importante para a detecção de falhas em sinais não lineares e não estacionários. Além disso, modifica a função de perda do SAE visando aprimorar a precisão do diagnóstico. Os dados utilizados são sinais de vibração originais coletados dos componentes de máquinas rotativas, os dados neste estudo também não são binários. A rede SAE, conforme apresentada pelos autores, tem potencial para extração automática de recursos dos sinais originais no domínio do tempo coletados pelos sensores. Isso ajuda a capturar informações relevantes dos sinais sem a necessidade de engenharia manual de recursos, o que pode ser demorado e subjetivo.

Ao pré-processar os sinais originais da série temporal e utilizar o modelo de rede de diagnóstico de falhas SAE, o método visa aprimorar a representação dos sinais de uma forma que destaque recursos importantes relacionados às condições de saúde e falhas do maquinário. Segundo Yang *et al.* (2018), modificar a função de perda da rede SAE pode levar a maior precisão e eficiência de diagnóstico. Esta modificação visa otimizar o processo de treinamento, acelerar a velocidade de convergência do modelo e evitar que ele fique preso em ótimos locais, levando, em última análise, a um melhor desempenho diagnóstico.

Os resultados experimentais do estudo de Yang *et al.* (2018) sugerem que o método proposto possui alta precisão e eficiência, juntamente com uma estrutura de modelo simples e forte capacidade de generalização. Isso indica que o método é adequado para diagnóstico de falhas em máquinas rotativas com alta precisão e eficiência, superando potencialmente os métodos rados de redes neurais artificiais (RNA) ao lidar com grandes dados amostrais.

O trabalho de Zhanhong *et al.* (2023) aborda o desenvolvimento de um framework de aprendizado supervisionado em séries temporais para prever falhas em chillers. Foram utilizadas técnicas de aprendizado profundo, como autoencoders, e modelos tradicionais de aprendizado de máquina. Os dados utilizados foram coletados por sensores instalados nos chillers e incluem aproximadamente 40 variáveis medidas, como temperaturas, pressões, taxas de fluxo, correntes de motores, entre outros, mas não analisa dados binários dos sensores.

No contexto do trabalho de Zhanhong *et al.* (2023), o autoencoder é utilizado para comprimir os dados de séries temporais das variáveis dos chillers em uma representação latente de menor dimensão, que captura as características mais importantes dos dados. Essa representação latente é então utilizada para reconstruir os dados originais, permitindo uma análise mais eficaz e a identificação de padrões relevantes para a previsão de falhas.

No estudo de Upasane *et al.* (2024), é mencionado que o sistema proposto em Type-2 Fuzzy Logic System (T2FLS) foi comparado com Stacked Autoencoders (SAEs). No contexto do artigo, os autoencoders foram utilizados como parte da comparação de desempenho do sistema proposto em relação a outras abordagens, como o Type-1 Fuzzy Logic System (T1FLS), Multi-Layer Perceptron (MLP) e Decision Trees (DT) em análise de dados de séries temporais de dados não binários.

De acordo com o artigo, o sistema proposto, baseado em lógica fuzzy do tipo 2 otimizado via big-bang big-crunch, obteve resultados superiores em comparação com outras técnicas avaliadas. Especificamente, o sistema proposto superou a precisão do Type-1 Fuzzy Logic System (T1FLS) em 8,9% e do Decision Trees (DT) em 529,2%, enquanto forneceu resultados comparáveis ao SAE.

Dix *et al.* (2021) detalha a utilização de autoencoders em sua abordagem de detecção de falhas em dados de séries temporais de equipamentos industriais. Os dados abordados e analisados são casos realistas de falhas em equipamentos de plantas industriais, simulados em um separador de processo, com o objetivo de avaliar a capacidade das arquiteturas de detectar e explicar as causas raiz das falhas. No contexto da detecção de falhas do estudo, os autoencoders são treinados para reconstruir dados normais, de modo que o erro de reconstrução seja minimizado.

Dense Autoencoders também são uma das arquiteturas de redes neurais avaliadas no estudo. Dix *et al.* (2021) utilizam para comprimir e reconstruir os dados de séries temporais de equipamentos industriais, sendo treinados para reconstruir os dados normais com o objetivo de identificar falhas com base no erro de reconstrução. Além dos Dense Autoencoders, o estudo também considera LSTM Autoencoders, que são autoencoders construídos com uma arquitetura LSTM. Esses modelos são capazes de capturar dependências de longo prazo em dados sequenciais, o que pode ser útil na detecção de falhas em séries temporais complexas.

Os autoencoders são avaliados no estudo de Dix *et al.* (2021) em termos de sua capacidade de detectar falhas em dados de séries temporais de equipamentos industriais, no entanto não são abordados dados binários no estudo. O desempenho dos autoencoders é medido com base na precisão da detecção de falhas e na capacidade de explicar as causas raiz das falhas identificadas.

Os Dense Autoencoders, no estudo de Dix *et al.* (2021), apresentaram o melhor desempenho na detecção de falhas em 20 casos de falha simulados. Eles foram capazes de identificar as falhas com maior precisão em comparação com as outras arquiteturas avaliadas. Além disso, os Dense Autoencoders demonstraram uma alta taxa de acurácia na explicação das causas raiz das falhas detectadas, atingindo uma precisão de 99% na identificação das causas raiz corretas.

Por outro lado, segundo Dix *et al.* (2021), os LSTM Autoencoders apresentaram um desempenho ligeiramente inferior na detecção de falhas em comparação com os Dense Autoencoders. Apesar de que os LSTM Autoencoders tenham sido mais eficazes na detecção de falhas, sua taxa de acurácia na explicação das causas raiz foi ligeiramente menor, atingindo uma precisão de 95% em comparação com os 99% dos Dense Autoencoders.

Portanto, os resultados indicam que, embora os LSTM Autoencoders tenham se saído um pouco melhor na detecção de falhas, os Dense Autoencoders foram mais eficazes na explicação das causas raiz das falhas identificadas. Ambas as arquiteturas de autoencoders foram capazes de detectar falhas com sucesso, mas com diferenças sutis em seus desempenhos na análise das causas raiz (DIX *et al.* 2021).

A tabela 01 facilita a identificação de padrões e tendências comuns, além de promover o reconhecimento de abordagens únicas e potencialmente inovadoras neste recorte de pesquisa.

Tabela 01 – Algoritmos utilizados na detecção de falhas e máquinas e equipamentos industriais.

Autores	Algoritmos	S. T.	S. T. B.	Aplicação
Ahn <i>et al.</i> (2021)	ML;	sim	não	Usina siderúrgica
Alfaro-Nango <i>et al.</i> (2022)	CNN;	sim	não	Aeronave
Arévalo, Rernentería e Schwung (2018)	DT; AdaBoost; NB;	não	não	Protótipo industrial
Azzouz <i>et al.</i> (2020)	LSTM; SVM; RF; PNN; MLP;	sim	não	Sensores IOT
Babayigit e Abubaker (2024)	RF; SVM; BNN;	sim	não	Motor elétrico
Baghbanpourasl, Kirchberger e Eitzinger (2021)	ML;	sim	não	Máquina de solda
Benetti <i>et al.</i> (2023)	mRMR; SVM; NB; KNNs; DT;	não	não	Contatos elétricos
Binding, Dykeman e Pang (2019)	LR; RF; XGBT;	sim	não	Impressora
Bykov, Voronov e Voronova (2019)	GB; KNN; SVM;	sim	não	Sistema hidráulico
Cajhuarina <i>et al.</i> (2024)	NARX; LSTM;	sim	não	Turbinas a gás
Choi, Sohn e Oh (2023)	ANN; CNN; LSTM;	sim	não	Sensores
Chomphu e Kjksirkul (2020)	ABD-LSTM; ANN; LR; RF; XGBoost; LightGBM;	sim	não	Sensores
Corallo <i>et al.</i> (2018)	K-Means; C-Means; ANN;	sim	não	Máquinas
Costache e Minzu (2012)	ANN;	não	não	Planta de tratamento de águas
Cózar, Puerta e Gámez (2017).	DBN;	sim	não	Dados de sensores
Dakhil, Ali e Abdulredah (2020)	KNN; DT; LD;	não	não	Motor
de Mansouri e Vadera (2022)	RNN; LSTM; GRU; CNN;	sim	não	Vibração de rolamentos
Diallo <i>et al.</i> (2021)	DCNN; DNN; ANN; FMLNM;	sim	não	Dados de rolamentos, sensores etc.
Diez-Olivan <i>et al.</i> (2018)	SVM; KNN;	sim	não	Motor diesel marítimo
Dix <i>et al.</i> (2021)	Dense Autoencoders; LSTM Autoencoders; LSTMs;	sim	não	Dados de sensores
Ericeira <i>et al.</i> (2020)	ANN; MLP; RF;	não	não	Esteira transportadora
Gugaratshan <i>et al.</i> (2023)	MLP; IF; RF; K-Means;	sim	não	Sensores
Hashemnia, Fan e Rocha (2021)	LSTM;	sim	não	Equipamentos industriais
Hassan, Sadek e Attia (2023)	Wavelet Analysis; CNN; SAE; LSTM;	sim	não	Ferramentas de usinagem
Hoong Ong, Niyato e Yuen (2020)	DQN;	sim	não	Aeronave
Huang <i>et al.</i> (2019)	BLSTM; FFNNs;	sim	não	Sensores
Jiao <i>et al.</i> (2022)	CNN;	sim	não	Rolamentos e caixa de engrenagens

Continuação Tabela 1 – Algoritmos utilizados na detecção de falhas e máquinas e equipamentos industriais.

Autores	Algoritmos	S. T.	S. T. B.	Aplicação
Kaewkorn <i>et al.</i> (2019)	FFT; SVM;	sim	não	Máquinas
Kajmakovic <i>et al.</i> (2023)	DT; RF; KNN; NB; SVM;	não	não	Sistemas de energia elétrica
Kanawaday e Sane (2017)	NB; SVM; CART; DNN;	sim	não	Máquina de corte
Kim e Jeong (2022)	SVM; RF; NB; KNN; GB; LR;	sim	não	Sistema hidráulico
Kim <i>et al.</i> (2024)	CNN; Grad-CAM;	não	não	Transformadores
Kokane e Bagavathi (2018)	ANN; LR;	sim	não	Suspenção automotiva
Kumar <i>et al.</i> (2008)	HMM;	sim	não	Saúde de uma broca ferramenta
Kwon e Kim (2020)	RF; SVM;	não	não	Dados de sensores
Lee <i>et al.</i> (2023)	HMM;	não	não	Sensores
Li <i>et al.</i> (2020),	CNN;	sim	não	Motores e rolamentos
Li <i>et al.</i> (2022)	GS; GATv2;	sim	não	Sensores
Liu <i>et al.</i> (2023)	MLP; LSTM; GRU; Transformer;	sim	não	Freio automotivo
Ma <i>et al</i> (2020)	LSTM;	sim	não	Escavadeira hidráulica
Maurya, <i>et al.</i> (2021)	DNN;	não	não	Rolamentos
Moreira <i>et al.</i> (2018)	CNN; MLP; SVM;	sim	não	Robô industrial
Mudita e Gupta (2021)	KNN; NB; SVM; ARIMA;	sim	não	Sensores IOT
Mujib e Djatna (2020)	RF;	não	não	Equipamentos industriais
Namuduri <i>et al.</i> (2020)	CNN; RNN; LSTM; DNN;	sim	não	Sensores
Natwsha e Gudetti (2021)	LPC; MFCC;	não	não	Máquinas industriais
Padmanabh, Al-Rubaie e Aljasmi (2022)	ANN;	sim	não	Chiller
Padmanabh, <i>et al.</i> (2021)	SVM; LR; KNN;	sim	não	Chiller
Patel <i>et al.</i> (2020)	ML;	sim	não	Equipamentos industriais
Patel <i>et al.</i> (2021)	CUSUM; GGM;	sim	não	Ativos industriais
Qin <i>et al.</i> (2023)	G-TELM;	sim	não	Máquinas industriais
Rahhal e Abualnadi (2020)	LSTM-RNN;	sim	não	Dispositivos IoT
Rajasekar (2023)	SVM; ANN; GB; RF; LR; ANN;	sim	não	Plataforma hidráulica
Ramu e Narayanan (2023)	RNN; LSTM;	sim	não	Sensores
Rocchetta, Mavroeidis e Petkovic (2022)	SVM;	sim	não	Dados de sensores

Continuação Tabela 1 – Algoritmos utilizados na detecção de falhas e máquinas e equipamentos industriais.

Autores	Algoritmos	S. T.	S. T. B.	Aplicação
Rodrigues e Zárate (2019)	AGNES; K-Means;	sim	não	Sytsbox
Sengottaiyan <i>et al.</i> (2022)	CNN; LSTM; SVM;	sim	não	Máquinas
Shehzad <i>et al.</i> (2023)	K-Means	sim	não	Motor elétrico
Silva e Souza (2021)	RNN;	sim	não	Gerador hidroelétrico
Silvestrin, Hoogendoorn e Koole (2019)	LSTM; TCN; RF; DT; KNN;	sim	não	Sistema hidráulico
Singh, Sahoo e Gadomí (2024)	ARMA; ARIMA; Markov Model; SVR; LSTM; GRU;	sim	não	Sensores
Srivastava e Parida (2020)	SVM; GPR;	não	não	Micro redes de corrente alternada
Srivastava e Parida (2022)	SVM; GPR;	sim	não	Micro redes de corrente alternada
Steurtewagen e Poel (2021)	XGBoost;	sim	não	Dados de sensores
Steurtewagen e Poel (2019)	RF; GB;	não	não	Compressor
Surantha e Gozali (2023a)	LR; RF; ANNN;	não	não	Equipamentos industriais
Surantha e Gozali (2023b)	ELM; WELM;	sim	não	Sensores
Tamilselvan, Wangm e Jayaraman (2012)	SVM;	não	não	Vibração de rolamentos
Tarik, Mnai e Jebari (2023)	SVM;	não	não	Dados de manutenção
Tobon-Mejia <i>et al.</i> (2011)	HMM;	sim	não	Rolamentos
Trivedi <i>et al.</i> (2019)	DT; SVM; RF; GB;	não	não	Ar-condicionado
Tsai <i>et al.</i> (2019)	BPCA;	sim	não	Sensores
Uhrich <i>et al.</i> (2023)	FFN; CNN; Transformer;	sim	não	Válvulas
Upasane <i>et al.</i> (2024)	T2FLS; MLP; SAE; DT;	sim	não	Bombas d'água
Uppal <i>et al.</i> (2023)	RF; DT; KNN; GNB;	não	não	Sensores IOT
Vaidyan e Tyagi (2022)	ML;	sim	não	Clp
Vivek e Kusuma (2023)	Q-learning;	sim	não	Dados de sensores
Vivek e Kusuma (2023)	LSTM; OS-ELM;	sim	não	Sensores
Wahid, Breslin, e Intizar (2022)	CNN; LSTM;	sim	não	Máquinas
Wang (2022)	GAN;	não	não	Medidor de fluxo ultrassônico
Wang <i>et al.</i> (2021)	GAN; CNN;	sim	não	Veículos
Wang <i>et al.</i> (2023)	HierCorrPool; GNNs; AConvLSTM;	sim	não	Aeronave
Wang, Wen e Lu (2015)	ELM; T-PLS;	sim	não	Equipamentos industriais

Continuação Tabela 1 – Algoritmos utilizados na detecção de falhas e máquinas e equipamentos industriais.

Autores	Algoritmos	S. T.	S. T. B.	Aplicação
Wang, Xian e Song (2023)	DNN;	sim	não	Dados de rolamentos, sensores etc.
Warriach, Ozcelebi e Lukkien (2015)	MLR; ANN;	sim	não	Baterias
Wu <i>et al.</i> (2018)	HMM;	sim	não	Motor
Xu <i>et al.</i> (2020)	TCNN; CNN;	sim	não	Rolamentos e bombas centrifugas
Xu <i>et al.</i> (2021)	Bi-LSTM;	sim	não	Rolamentos
YANG <i>et al</i> (2018)	SAE;	sim	não	Máquinas rotativas
Yu (2021)	BPNN;	não	não	Sistemas mecatrônicos
Yugapriya, Judeson e Jayanth (2022)	LR; DT; KNN; NB; RT;	não	não	Sistema hidráulico
Zhang <i>et al.</i> (2022)	FCN;	sim	não	Sistema hidráulico
Zhanhong <i>et al.</i> (2023)	AE;	sim	não	Chiller
Zhao (2023)	DBN;	não	não	Consumo de eletricidade
Zhao <i>et al.</i> (2018)	SVR; KNN; MLP; SAE-DNN; RNN; GRU; BiGRU;	sim	não	Caixa de engrenagens
Zhu <i>et al.</i> (2022)	CWT;	sim	não	Desgaste de ferramentas
Züfle <i>et al.</i> (2021)	XGBoost; FFNN;	sim	não	Prensa

Legenda: S. T. = Séries Temporais / S. T. B. = Séries Temporais Binárias

Fonte: Elaborado pelo autor.

A análise dos artigos capturados pela revisão sistemática da literatura revela não apenas as tendências atuais no campo de aprendizado de máquinas no contexto de máquinas e equipamentos industriais, mas também aponta para áreas menos exploradas, como séries temporais binárias. Com o objetivo de identificar similaridade nos tipos de dados analisados nos estudos captados, se fez necessária a condução de busca específica sobre o tipo de dado em que se prende analisar neste trabalho, a saber, séries temporais binárias.

3.1.1 Análise e detecção de padrões em séries temporais binárias

A análise e detecção de falhas em dados de séries temporais binárias apresenta desafios únicos devido à presença diferentes limites de classe normal. Várias técnicas existentes, como métodos estatísticos, métodos de aprendizagem estocástica, análise de sinais, técnicas de mineração de dados, algoritmos clássicos de aprendizagem de máquina ou métodos de aprendizagem profunda, foram desenvolvidas para lidar com esses desafios e detectar falhas de forma eficaz (SCHMIDL; WENIG; PAPENBROCK, 2022).

Dados de séries temporais binários são caracterizados por observações regulares de valores binários. Os métodos para análise de dados de séries temporais podem ser divididos em métodos baseados em regras, definindo limites ou regras baseadas no conhecimento do domínio e métodos baseados em aprendizado de máquina (DUFFIELD *et al.*, 2009; MA; HAN; ZHOU, 2023).

Embora as abordagens baseadas em regras sejam simples e interpretáveis, elas enfrentam problemas relacionados ao reconhecimento de padrões complexos. Algoritmos de aprendizado de máquina têm mostrado eficácia na detecção de falhas em dados de séries temporais, mas faltam pesquisas que abordem especificamente a detecção de falhas em dados de sensores binários (PRINCZ; SHALOO; EROL, 2024).

Como resultados, a revisão sistemática da literatura demonstra como as abordagens estão sendo adaptadas para analisar dados de séries temporais binárias para detecção de falhas, segmentação de padrões e automação de processos analíticos. Algumas das abordagens mais sofisticadas incluem:

- Métodos baseados em Autoencoders como os Variational Autoencoders (VAEs) ou Denoising Autoencoders, são utilizados para aprender representações latentes de dados de séries temporais binárias, facilitando a detecção de falhas (PRINCZ; SHALOO; EROL, 2024).
- Modelos de aprendizado profundo como redes recorrentes (RNNs), redes neurais convolucionais (CNNs) ou modelos baseados em transformers, têm sido aplicadas com sucesso na detecção de padrões complexos e na análise de séries temporais binárias para identificação de falhas (QUINN, 2022, NISHINO; MAEKAWA; HARA, 2023).
- Integração de métodos de aprendizado profundo com técnicas estatísticas avançadas combinar modelos de aprendizado profundo com métodos estatísticos como modelos *Auto-Regressive Integrated Moving Average* (ARIMA) ou modelos de espaço de estado pode melhorar a robustez e a interpretabilidade na análise de falhas em séries temporais binárias (HUNG *et al.*, 2008).

Estes métodos representam o estado da arte atual na análise de falhas em séries temporais binárias, utilizando tanto a capacidade de aprendizado complexo das redes neurais profundas quanto os fundamentos estatísticos para garantir resultados precisos e confiáveis na detecção e análise de falhas.

Munir *et al.*, 2019 categorizam métodos de detecção de falhas em séries temporais em: modelos probabilísticos; modelos estatísticos; modelos lineares; modelos baseados em proximidade e detecção de valores discrepantes em grandes dimensões. Em contrapartida, Braei e Wagner, 2020 divide todos os métodos de detecção de falhas em séries temporais em três categorias principais: métodos estatísticos; métodos clássicos de aprendizado de máquina e métodos utilizando redes neurais (aprendizado profundo).

Alguns estudos fundem a aprendizado de máquina e aprendizado profundo. Em contraste, a fronteira entre as abordagens estatísticas, transformações, análises matemáticas e aprendizado de máquina é vaga (PAPACHARALAMPOUS; TYRALIS; KOUTSOYIANNIS, 2019). Desta forma, esta pesquisa categorizou os 64 artigos identificados que analisam séries temporais binárias em duas categorias principais: abordagens estatísticas (que contempla os estudos sobre: modelos estatísticos,

transformações e análises matemáticas, algoritmos e métodos de otimização, métodos bayesianos e métodos e análises específicas) e Aprendizado de máquina e aprendizado profundo.

3.1.2 Abordagens estatísticas

Como abordagens estatísticas, destacam-se alguns modelos regressivos bem pesquisados como modelo autorregressivo (AR). AR é um modelo linear onde o valor atual (X_t) do processo estocástico (variável dependente) é baseado em um conjunto finito de valores anteriores (variáveis independentes) de comprimento (p) e um valor de erro (ϵ) (LU; FEI, 2011):

O modelo AR com um comprimento de janela anterior de (p) também é chamado de processo AR de ordem (p) ou AR(p). Os valores de erro (ϵ_t) são considerados não correlacionados e têm média constante igual a zero e variância constante (σ). Neste modelo, (ϵ) é usado para determinar a pontuação da falha. Os valores dos coeficientes a_1, \dots, a_p, c podem ser aproximados usando os dados de treinamento e resolvendo as equações lineares correspondentes com regressão de mínimos quadrados. Depois disso, (ϵ_t) para cada (X_t) pode ser calculado, o que representa a pontuação da falha. Portanto, a pontuação da falha é igual à diferença entre o valor previsto e observado (SUNG et al., 2020).

Outro modelo é a combinação de AR e do modelo de média móvel (MA), o modelo autorregressivo de média móvel (ARMA), que é frequentemente usado para séries temporais univariadas na prática (HUNG et al., 2008). Uma série temporal do modelo ARMA(p,q) depende das últimas (p) observações e (q) erros.

Os modelos ARMA usam menos variáveis na prática em comparação com AR e MA. No entanto, o principal desafio é selecionar valores apropriados para p e q . Quanto maiores forem esses dois valores, maior será a probabilidade de o modelo se ajustar demais, resultando em muitos falsos negativos no processo de detecção de falhas. Por outro lado, se os números escolhidos forem muito pequenos, o modelo ficará sub-ajustado e surgirão muitos falsos positivos, ou seja, os pontos de dados são detectados como falhas, embora não o sejam. Em ambos os casos, o modelo não consegue detectar as falhas corretamente (HUNG et al., 2008).

O modelo ARIMA é uma generalização do modelo ARMA. Além do parâmetro (p) e (q), também é definido por um parâmetro (d) que define o número de vezes que a série temporal é diferenciada (HUNG *et al.*, 2008). Após o ajuste no modelo ARIMA, as falhas são detectadas avaliando o desvio do ponto previsto para o observado (HUNG *et al.*, 2008).

3.1.3 Abordagens por algoritmos clássicos de aprendizado de máquina

Os algoritmos de aprendizado de máquina tentam detectar falhas em conjuntos de dados de séries temporais sem assumir um modelo gerativo específico. Eles se baseiam no fato de que não é necessário conhecer o processo subjacente dos dados para poder fazer previsões de séries temporais e detecção de falhas em séries temporais. Portanto, esses métodos estão bem avançados fora do campo da estatística (RIVEST; KOHAR, 2021).

K-Means é um algoritmo de cluster baseado em centroide que divide os dados em um número de clusters definido pelo usuário com base em uma medida de similaridade, como a distância euclidiana. Embora não seja normalmente usado para detecção de falhas, o K-means pode identificar valores discrepantes como pontos de dados que estão distantes do centroide do cluster. Sua simplicidade, eficiência e resultados intuitivos merecem consideração (BUDIARTO; PERMANASARI; FAUZIATI, 2019).

O *Clustering* espacial baseado em densidade de aplicativos com ruído (DBSCAN) é outro algoritmo baseado em densidade conhecido por sua capacidade para identificar clusters de formato arbitrário. Ao contrário dos algoritmos de *clustering* baseados em partição, como K-means, o DBSCAN não requer pré-especificação do número de clusters e é robusto ao ruído, permitindo a detecção de valores muito discrepantes de qualquer cluster. O método classifica os pontos de dados em três categorias diferentes: pontos centrais; pontos de fronteira e falhas (JAIN; BAJPAI; PAMULA, 2022).

Um dos modelos utilizados nos estudos é o algoritmo Isolation Forest (IF), que constrói florestas aleatórias para isolar eficientemente falhas. O algoritmo IF tem se mostrado promissor na detecção de falhas em dados de séries temporais binárias,

capitalizando a distinção das falhas em um ambiente de florestas aleatórias. O algoritmo constrói um conjunto de árvores de decisão, com cada árvore é treinada em um subconjunto de dados selecionado aleatoriamente. A IF apresenta alta eficiência, lida com grandes quantidades de dados e requer menos parâmetros, reduzindo o risco de sobre-ajuste (AHMED et al., 2019; CHEN et al., 2021; DI; HAOYUE; YUMING, 2022).

Assim, este método considera pontos com comprimentos de caminho mais curtos como candidatos com alta probabilidade de serem falhas. O processo de detecção de falhas utilizando Florestas de Isolamento é realizado geralmente em duas etapas: treinamento e avaliação (CHEN et al., 2021). Existem alguns métodos que estenderam o iForest para detectar falhas em séries temporais.

Uma abordagem principal para detectar falhas em dados univariados é analisar o conjunto de dados em sequências definidas pelo comprimento da janela (w). Em seguida, a pontuação de falhas em cada sequência é calculada, que é proporcional ao comprimento médio do caminho de uma instância. Usando o aprendizado supervisionado, o limite pode ser calculado no conjunto de treinamento e usados posteriormente no conjunto de teste(CHEN et al., 2021).

O Fator Outlier Local (LOF) é um algoritmo baseado em densidade que mede o desvio local da densidade de um ponto de dados em relação aos seus vizinhos. A noção de densidade pode ser calculada usando qualquer número de métricas de distância calculadas entre pontos de dados. Em LOF a métrica é usada para determinar a que distância os vizinhos mais próximos estão localizados do ponto de dados e quão próximos estão outros vizinhos desses pontos de dados. Este cálculo determina a densidade em torno dos pontos de dados selecionados e as densidades em torno de seus vizinhos mais próximos (KHARITONOV et al., 2022).

As diferenças entre essas densidades determinam a pontuação das falhas. Após as pontuações de falhas serem calculadas, os pontos de dados com as pontuações mais altas são classificados como falhas. O LOF identifica falhas no contexto local dos dados, tornando-o valioso para a detecção de falhas em conjuntos de dados onde as falhas estão próximas dos dados normais, mas distintas em suas vizinhanças locais (KHARITONOV et al., 2022).

Uma variação do LOF é um Fator Outlier Local Baseado em Cluster (CBLOF). Como o nome sugere, é um algoritmo baseado em cluster. A pontuação de falha neste algoritmo é calculada com base na distância até o centro do cluster em que o ponto de dados está localizado. Essa distância também é multiplicada pelo tamanho do cluster. Para a formação dos clusters propriamente ditos, qualquer algoritmo de clusterização pode ser utilizado dentro do CBLOF, incluindo KNN configurado para agrupamento (HE; XU; DENG, 2003; KHARITONOV *et al.*, 2022).

3.1.4 Abordagens por aprendizado profundo

Uma combinação de U-Net e CNN 1D, também é utilizada com sucesso para analisar uma série temporal binária que especifica as localizações temporais das repetições nos dados de aceleração. U-Net é um tipo específico de arquitetura de rede neural convolucional. As CNNs são modelos de redes neurais profundas frequentemente empregados em tarefas de processamento de imagens, incluindo segmentação e reconhecimento de padrões. No contexto do estudo analisado, as CNNs foram adaptadas para lidar com a detecção de repetições em dados de aceleração, mostrando que a abordagem proposta pode alcançar uma detecção precisa de repetições mesmo quando treinada apenas em dados compostos (NISHINO; MAEKAWA; HARA, 2023)

A máquina de vetores de suporte de classe única (OCSVM) é uma variante do SVM projetada para detecção de falhas. Ele aprende um limite de decisão em torno de dados normais em um espaço de recursos de alta dimensão e identifica falhas como pontos de dados fora desse limite. A configuração dos hiperparâmetros do modelo requer alta intensidade de tempo em alguns casos, por isso as estruturas AutoML tornaram-se cada vez mais importantes nos últimos anos (QIAO, Y.; WU, K.; JIN, 2023).

Quinn (2022) introduz o RNN-BOF, um modelo de Rede Neural Recorrente (RNN) multivariada global para previsão binária de resultados, que treina a partir e para uma população de séries temporais de pacientes para produzir avaliações de risco probabilísticas individuais. O estudo utilizou um esquema de treinamento com janela deslizante em um conjunto de dados do mundo real com 83 pacientes, onde a

série temporal binária principal representa a presença de eventos de agressão e as séries temporais de covariáveis representam características clínicas, demográficas e medidas psicométricas.

Gundersen *et al.* (2020) utilizou o Bayesian Convolutional Neural Network (BCNN) para classificar séries temporais binárias relacionadas a vazamentos de gases marinhos. O BCNN foi empregado para modelar a incerteza nas previsões de classificação, permitindo a geração de distribuições de probabilidade sobre as classes, facilitando a tomada de decisões informadas. Além disso, o BCNN foi explorado em um cenário de aprendizado de transferência, onde o modelo foi pré-treinado em dados de simulação e ajustado com dados limitados de medições *in loco*, demonstrando sua capacidade de generalização.

Os Spiking Neural Networks (SNNs) foram utilizados no estudo de Jang e Simeone (2022) para capturar eficiências semelhantes às do cérebro biológico em termos de codificação e processamento de informações. Os SNNs processam e comunicam informações por meio de sinais de disparo esparsos ao longo do tempo, em vez de números reais, em grafos de computação recorrentes. No estudo, foram exploradas redes neurais de disparo probabilísticas, que têm a capacidade de gerar múltiplos sinais de disparo independentes quando consultadas com o mesmo input. Isso permite robustecer decisões, quantificar incerteza durante a inferência e melhorar estimativas estatísticas durante o treinamento.

Princz, Shaloo e Erol (2024) destacam que AE exibiu alta precisão, ao avaliar diversos modelos. Os modelos diferiram na identificação correta de falhas, mas AE apresentou bom desempenho. O autoencoder é um método para detectar falhas e reduzir a dimensionalidade dos dados e projetá-los em um espaço inferior, ou seja, espaço latente, onde permanecem mais variáveis correlacionadas. A principal suposição sobre a distribuição dos dados é o fato de os dados normais e os dados falha serem significativamente diferentes neste espaço. Então, a projeção de volta ao espaço original mostrará diferenças significativas em alguns pontos de dados, que representam as instâncias de dados com falha. Isso torna o autoencoder apropriado para detecção de falhas (BRAEI; WAGNER, 2020).

Os autoencoders pertencem à família das redes neurais *feed-forward* e são otimizados para gerar as mesmas informações que foram inseridas na rede. O desafio

é que a primeira metade das camadas ocultas reduz a dimensão do conjunto de dados e a segunda metade aumenta a dimensão de volta ao valor original. Essas duas partes são nomeadas de acordo com parte de Codificação e Decodificação (ZHANHONG *et al.*, 2023).

A diversidade de aplicações de modelos para análise de séries temporais binárias demonstra a ênfase na inovação metodológica e refletem a evolução contínua do campo e sua relevância em múltiplos contextos. A tabela 02 apresenta as metodologias para análise de séries temporais binárias identificadas neste recorte de pesquisa.

Tabela 02 – Metodologias de análise de séries temporais binárias.

Autores	Cat.	T.	Metodologia	Objetivo do estudo
NANDI; KAFASHAN; CHING, 2017	ME	M	PPGLM;	Avaliar a controlabilidade de um modelo estatístico específico, o point-process generalized linear model (PPGLM).
KEDEM; FOKIANOS, 2005	ME	U	Regressão logística ; Regressão PROBit;	Analizar a relação entre uma série temporal binária e covariáveis estocásticas e dependentes do tempo
BODNER; CEULEMANS, 2023	ME	M	ConNEcT;	Desenvolvimento de um pacote de software em R chamado ConNEcT, projetado para construir redes de medidas de contingência com base em séries temporais binárias.
MOSCONI; SERI, 2006	ME	U	Modelo PROBit bivariado;	Modelagem de séries temporais binárias bivariadas para testar a não causalidade de Granger e a independência simultânea, utilizando um modelo PROBit bivariado dinâmico.
AMRO, 2013	ME	U	Modelo de Bernoulli; Cadeia de Markov de 2 estados;	Analizar a perda de pacotes em redes, segmentando os dados em intervalos de tempo específicos e aplicando modelos estatísticos Bernoulli e Cadeias de Markov, para entender e prever a perda de pacotes.
FOKIANOS; MOYSIADIS, 2017	ME	U	LR; Regressão PROBit; Modelos GARCH-Like;	Modelagem de séries temporais binárias impulsionadas por um processo latente. Investigação de questões de ergodicidade, estacionalidade e estimativa por máxima verossimilhança.
WANG; ZHANG, 2013	ME	U	Regressão Logística;	Analizar de forma detalhada o impacto da poluição do ar nas ausências escolares.
GAO; SHAHBABA; OMBAO, 2017	ME	U	Modelos estatísticos de efeitos mistos;	Desenvolvimento de um modelo de efeitos mistos para séries temporais binárias, onde um componente estocástico é representado por um processo Gaussiano.
MARMARELIS <i>et al.</i> , 2006	ME	U	Boolean-Volterra;	Modelagem de sistemas neurais com entradas e saídas de processo pontual que utiliza operadores booleanos modulo-2 para criar um modelo matemático denominado Boolean-Volterra.
TAGORE; SUTRADHAR, 2009	ME	U	Modelo de mistura linear para séries temporais binárias;	Investigar a modelagem da estrutura de correlação em dados de séries temporais binárias e comparar a eficácia de modelos lineares e não lineares na estimativa de parâmetros e previsão.
GAO; GILLEN; OMBAO, 2018	ME	U	Modelo Logístico Autorregressivo; Matriz de Informação de Fisher;	Investigar a modelagem de séries temporais binárias, que se concentram na análise de uma coorte de gestantes em relação à associação entre o nível de estresse auto-relatado e a frequência respiratória.
CHO; BROWN-SCHMIDT; LEE, 2018	ME	U	LMM; GLMM; AR-GLMM;	Análise estatística de dados de séries temporais binárias obtidos a partir de experimentos de rastreamento ocular utilizando a técnica de mundo visual.
PAUL; MAITY; MAITI, 2018	ME	U	MCMC; Marginal Likelihood; DIC;	Abordagem da análise bayesiana de séries temporais binárias, com foco na comparação de diferentes priors (normal, Student's t e Jeffreys prior) e métodos de modelagem com abordagens frequentistas.
LU, 2020	ME	U	Modelos estatísticos;	Modelagem de séries temporais binárias baseadas em parâmetros para lidar com dados binários, como os da competição de remo entre as universidades de Oxford e Cambridge.
EATOCK <i>et al.</i> , 2015	ME	U	Modelos estatísticos;	Avaliação de medidas de agrupamento de fibrilação atrial por meio de modelos estocásticos de recorrência de episódios e progressão da doença.

Continuação tabela 1 – Metodologias de análise de séries temporais binárias.

Autores	Cat.	T.	Metodologia	Objetivo do estudo
JENTSCH; REICHMANN, 2019	ME	U	NDARMA; gbARMA;	Introdução e análise de modelos estatísticos que possam lidar com a dependência serial presente nos dados do estudo.
SUNG <i>et al.</i> , 2020	ME	U	Modelo de Processo Gaussiano Generalizado para respostas binárias; AR;	O estudo explora a aplicação de um modelo de processo Gaussiano generalizado para dados binários, que incorpora uma função de média flexível para capturar diferentes estruturas de séries temporais e interações entre variáveis de controle.
NADIYA; WIDYANINGSIH; SARWINDA, 2021;	ME	U	RL; AR; Método de Verossimilhança Parcial;	Estimação de parâmetros para séries temporais binárias utilizando o método de verossimilhança parcial. Especificamente, o foco está na modelagem de séries temporais binárias.
HU <i>et al.</i> , 2021	ME	M	Abordagem Bayesiana; Metropolis-Hastings (MH);	Proposição de um modelo probabilístico de árvore para representar o caminho de propagação de eventos entre nós ou objetos em uma rede.
YARKAN; ARSLAN, 2007	ME	U	Modelos estatísticos;	O estudo explora a utilização de modelos estatísticos e algoritmos para analisar e prever a disponibilidade de espectro em diferentes bandas, com o objetivo de melhorar a eficiência e o uso do espectro disponível.
KLINGENBERG, 2008	ME	U	GLMMs; Monte Carlo EM Algorithm;	Modelagem estatística de séries temporais binárias com lacunas, focando em desenvolver modelos de regressão para lidar com a natureza binária dos dados e a presença de lacunas.
AMRO, 2013	ME	U	Modelos estatísticos;	Modelagem da perda de pacotes na rede EUMEDConnect, que conecta seis países árabes do Mediterrâneo para fins de pesquisa e educação.
PANG, 2010	ME	U	Bayesian Generalized Linear Multilevel Model AR(p); MCMC;	Modelagem estatística avançada de dados binários de séries temporais transversais (TSCS) com o objetivo de lidar com a heterogeneidade e a correlação serial presentes nos dados.
WU; CUI, 2014	ME	U	Regressão Logit; Análise de Regressão;	O objetivo principal é propor um modelo de regressão logit paramétrico que leve em consideração a dependência serial nos resultados das corridas, incorporando um processo latente auto correlacionado.
GORCIN <i>et al.</i> , 2011	ME	M	Modelo autorregressivo	Modelagem e previsão da ocupação espectral em ambientes de comunicação sem fio, com foco na eficiência do uso do espectro.
NYBERG, 2013	ME	U	Modelos PROBIT estáticos e dinâmicos	Analizar a previsibilidade dos mercados de ações em diferentes estados (urso e touro) e comparar o desempenho de modelos estáticos e dinâmicos na previsão desses estados de mercado.
CHOU; CHU, 2010	ME	U	DAR; INAR;	Analizar a dependência entre as duas séries temporais binárias e avaliar se é possível prever eventos futuros com base nas séries temporais passadas.
HUDECOVÁ, 2013	ME	U	Modelo Autorregressivo Binário (BAR);	Investigar a presença de mudanças na estrutura dos modelos autorregressivos binários e examinar como essas mudanças podem afetar a dependência dos valores binários anteriores na série temporal.
FOKIANOS; GOMBAY; HUSSEIN, 2014	ME	U	LR; AR(p); AR;	Desenvolver um procedimento estatístico para identificar mudanças nos coeficientes de um modelo de regressão logística aplicado a dados de séries temporais binárias.

Continuação tabela 1 – Metodologias de análise de séries temporais binárias.

Autores	Cat.	T.	Metodologia	Objetivo do estudo
FASANO <i>et al.</i> , 2021	ME	M	Modelos PROBit dinâmicos; Distribuições unificadas de skew-normais (SUN); MC;	Proposição de um método baseado em distribuições unificadas de skew-normais (sun) para calcular as distribuições de filtragem, predição e suavização de forma analítica em modelos de probit dinâmicos com variáveis de estado gaussianas.
DANIEL <i>et al.</i> , 2022	ME	M	Modelo de painel cruzado;	Examinar as relações entre estabilidade e dispersão nas estratégias de regulação emocional e os níveis de afeto e ansiedade, levando em consideração os efeitos autorregressivos e cruzados das avaliações.
LU; FEI, 2011	ME	M	Regressão Linear; Modelo AR de vetor não estacionário (TVPVAR);	O foco em extração de parâmetros de características dos sinais de fala e na aplicação de modelos estatísticos TVPVAR, para reconhecer e distinguir diferentes palestrantes com base nessas características.
SHANECHI <i>et al.</i> , 2013	ME	U	Modelos estatísticos;	Desenvolvimento de uma Interface Cérebro-Máquina (BMI) para o controle da supressão de explosões em coma médico.
OCZERETKO <i>et al.</i> , 2015	TAM	U	Transformada de Walsh-Hadamard; Transformada de Fourier;	Comparar a eficácia da transformada de Walsh-Hadamard com a transformada de Fourier na análise desses sinais, destacando a importância da sequência dominante normalizada como parâmetro diagnóstico.
SHU <i>et al.</i> , 2021	TAM	U	Ensemble Shapelet Transform Algorithm; Shapelet Transform (ST); Learning Time-series Shapelets (LS); Efficient Learning of Time-series Shapelets (EL); (Fast Shapelets (FS);	Aplicação do Short Isometric Shapelet Transform (SIST) para a classificação de séries temporais binárias. O objetivo principal do estudo é reduzir a complexidade temporal do algoritmo Ensemble Shapelet Transform, que é amplamente utilizado para a classificação de séries temporais.
DORI; PIEDRAHITA; LOUZOUN, 2019	TAM	M	Algoritmo de correlação parcial	Reconstrução de redes funcionais a partir de séries temporais binárias em um contexto de redes complexas.
HUNG <i>et al.</i> , 2008	TAM	U	ARMA; ANB;	Modelar séries temporais binárias com aplicação em experimentos de frequência de adesão. O estudo se concentra na análise de experimentos de adesão celular.
LO, 2005	TAM	U	Algoritmo de Mineração Sequencial Geral (GSP); Lax;	Melhorar a precisão das previsões em análises de séries temporais binárias através da inclusão do conceito de valor temporal.
SUNEHAG; SHAO; HUTTER, 2012	TAM	U	Context Tree Weighting (CTW); Krichevsky-Trofimov (KT);	Utilização de técnicas de compressão adaptativa e modelagem probabilística para lidar com dados não estacionários, destacando a importância da detecção de mudanças e outliers em séries temporais binárias.
GAO; KONTOYIANNIS; BIENENSTOCK, 2008	TAM	U	Context Tree Weighting (CTW);	Abordagem da estimativa da entropia de séries temporais binárias.
HU <i>et al.</i> , 2021	TAM	U	Short Isometric Shapelets Transform (SIST);	Aceleração da classificação de séries temporais binárias por meio do uso de Short Isometric Shapelets, que apresenta estratégias para reduzir a complexidade temporal e melhorar a eficiência da classificação.

Continuação tabela 1 – Metodologias de análise de séries temporais binárias.

Autores	Cat.	T.	Metodologia	Objetivo do estudo
AKSENTIJEVIC <i>et al.</i> , 2020	TAM	U	Complexidade de Kolmogorov (KC); Complexidade de Aksentijevic-Gibson (AG); Lempel-Ziv (LZA);	Abordagem da análise de séries temporais binárias utilizando medidas de complexidade, especificamente a complexidade de Kolmogorov (KC) e a complexidade de Aksentijevic-Gibson (AG).
MA, 2019	AMO	U	Algoritmo de expectativa-maximização (EM);	Reconstrução de estruturas de redes complexas a partir de séries temporais binárias.
LI <i>et al.</i> , 2023	AMO	U	Estimação de Máxima Verossimilhança (MLE); Algoritmo Expectation-Maximization (EM); Método de limiar; Algoritmo de identificação baseado em peso (WBI);	Proposição de uma matriz de probabilidade de influência para quantificar o peso das conexões entre os nós e utiliza métodos como Estimação de Máxima Verossimilhança (MLE) e o algoritmo Expectation-Maximization (EM) para obter essa matriz. Além disso, são propostos um método de limiar e um algoritmo de identificação baseado em peso para classificar a força das conexões na rede ponderada
RIVEST; KOHAR, 2021	AMO	U	Custo de Erro de Tempo Proposto (STE); Gradient Descent Algorithm;	Previsão de séries temporais binárias com ênfase na precisão temporal, ou seja, na capacidade de prever eventos em uma série temporal com alta precisão em termos de timing. O principal objetivo do estudo é propor e avaliar um novo custo de erro de tempo, denominado Squared Timing Error (STE).
BIGHAMIAN; SHANECHI, 2018	AMO	M	Expectation-Maximization (EM);	Identificação da dependência funcional entre os disparos neuronais e as redes de Potenciais de Campo Local (LFP) em diferentes escalas espaço-temporais durante o comportamento.
LIU <i>et al.</i> , 2022	ESP	M	Causalidade de Granger condicional; Cadeias de Markov de alta ordem;	Investiga a aplicação da causalidade de Granger condicional para inferir a topologia da rede a partir de séries temporais binárias de chegada de pacotes, sem a necessidade de decodificação dos pacotes.
VERSCHEURE <i>et al.</i> , 2006	ESP	U	Medidas de Similaridade Complementar; Agrupamento Progressivo; Modelos de Correlação de Dados Binários;	Identificação de pares de partes conversando em redes VoIP de forma anônima e criptografada. O objetivo principal do estudo é desenvolver uma metodologia eficaz para reconhecer as partes envolvidas em conversas de voz, mesmo quando essas conversas são protegidas por criptografia e os participantes desejam permanecer anônimos.
LEE; PARK, 2021	ESP	U	Proposição de modelo para caracterizar a autossimilaridade de séries temporais binárias;	Caracterização da autossimilaridade de séries temporais binárias por meio de um novo método que envolve a estimativa do perfil de entropia de similaridade de comprimento de ramo
LEE; PARK, 2021	ESP	U	Uso da função findpeaks do MATLAB;	Analizar as características de séries temporais binárias com base na entropia de similaridade de comprimento de ramificação (BLS).

Continuação tabela 1 – Metodologias de análise de séries temporais binárias.

Autores	Cat.	T.	Metodologia	Objetivo do estudo
LEE; PARK, 2022	ESP	U	Entropia de Similaridade de Comprimento de Ramo; Detecção de Anomalias; Differential Dynamic Time Warping (DTW); Discrete Fourier Transform (DFT) e Discrete Wavelet Transform (DWT);	Análise de características de picos e desníveis em perfis de entropia de similaridade de comprimento de ramo para séries temporais binárias. Os pesquisadores investigaram como picos e desníveis são formados e sua relação com as séries temporais, utilizando o conceito de entropia de similaridade de comprimento de ramo.
ROCA, 2004	ESP	U	GAM; Algoritmo de Pontuação Local Modificado; Modelo de Regressão Linear Parcial; Modelo de Regressão Linear Semi-Paramétrico; GLM;	Previsão de séries temporais binárias de concentração de dióxido de enxofre (SO ₂) utilizando o Modelo Aditivo Generalizado (GAM) com função de ligação desconhecida.
GOTO; TANIGUCHI, 2020	ESP	U	Proposição de método de análise discriminante baseada em séries temporais binárias;	Análise discriminante baseada em séries temporais binárias. Ele discute a classificação de séries temporais binárias derivadas de um processo latente, considerando um processo estacionário de mistura alfa elipsoidal e propõe um método de classificação com base nessas séries temporais binárias.
BRIDA.; GARRIDO, 2006	ESP	U	Análise de Séries Temporais Simbólicas (STSA); Modelo não linear proposto por Brida; Algoritmo de Otimização Genética;	Análise de dois regimes inflacionários em economias latino-americanas por meio de uma abordagem de séries temporais binárias. Os pesquisadores investigam a presença de dois regimes distintos de inflação nas economias da América Latina, utilizando dados mensais de taxas de inflação.
ZHU <i>et al.</i> , 2012	ESP	U	Modelo Caótico D-PL; Análise de Correlação;	Analizar as características do modelo caótico D-PL, sua capacidade de gerar sequências binárias caóticas com propriedades de aleatoriedade e ergodicidade, e sua eficácia na detecção de falhas em circuitos digitais.
SUE <i>et al.</i> , 2016	ML	U	MLP; Elman recurrent neural network (ERNN); support vector regression (SVR)	Previsão de informações de canal de controle durante intervalos de tempo em que não há atividade de transmissão ou recepção de dados do usuário.
GUNDERSEN <i>et al.</i> , 2020	ML	U	Rede Neural Convolucional Bayesiana (BCNN);	Desenvolver um modelo eficaz de classificação que possa distinguir entre situações de vazamento e não vazamento com base em séries temporais de medições obtidas de simulações hidrodinâmicas
JANG; SIMEONE, 2022	ML	U	PNN; GEM-SNN;	Abordagem da aplicação de algoritmos de aprendizado online em redes neurais de disparo probabilísticas.
MAHMOUD; LOTFI; LANGENSIEPEN, 2010	ML	U	NARX; MLP;	Abordagem da extração e previsão de padrões de ocupação em um ambiente inteligente habitado, com foco na monitorização de idosos que vivem de forma independente em suas próprias casas.

Continuação tabela 1 – Metodologias de análise de séries temporais binárias.

Autores	Cat.	T.	Metodologia	Objetivo do estudo
NISHINO; MAEKAWA; HARA, 2023	ML	M	CNN; U-Net; Binary Cross Entropy; Adam Optimizer;	Desenvolvimento de um método que possa identificar e rastrear as localizações temporais das repetições de ações humanas, como exercícios físicos, com base em dados de aceleração.
BAUTU; BAUTU; LUCHIAN, 2010	ML	U	GEP; Naive Bayes, SVM; MLP; DT; RF.	Abordagem da previsão da direção de mudanças nos índices de preços das ações, formulada como uma classificação binária.
PRINCZ; SHALOO; EROL, 2024	ML	U	Isolation Forest; Local Outlier Factor; Density-Based Spatial Clustering of Applications with Noise (DBSCAN); kMeans; AE; VAE; SVM; AutoML;	Detecção de anomalias em dados de séries temporais binárias em um contexto de monitoramento de condições em sistemas de produção industrial. O objetivo principal é utilizar técnicas de aprendizado de máquina não supervisionado para identificar padrões anômalos nos dados gerados pela ativação e desativação de sensores em máquinas e equipamentos industriais.
BAUTU <i>et al.</i> , 2009	ML	U	Hipernetworks evolutivas; SVM; RN; Regressão Logística:	Aplicar modelos de hipernetwork evolutiva para prever movimentos de preços em mercados de ações usando séries temporais binárias.
QUINN, 2022	ML	M	RNN; SVM; RF; Kernel Ridge Regressions; GBDT;	Previsão de eventos de agressão em pacientes hospitalizados, utilizando uma abordagem de previsão de resultados binários.

Legenda: Cat. = Categoria; ME = Modelos estatísticos; TAM = Transformações e análises estatísticas; AMO= Algoritmos e métodos de otimização; ESP = Métodos e análises específicas; ML = Aprendizado de máquina e aprendizado profundo; T. = Tipo de série; U = Série temporal univariada; M = Série temporal multivariada;

Fonte: Elaborado pelo autor.

3.2 ANÁLISE DE SÉRIES TEMPORAIS MULTIVARIADAS E APLICAÇÕES EM AMBIENTES INDUSTRIAIS

A revisão da literatura sobre séries temporais binárias revelou um total de 64 artigos que abordam o tema de forma geral. No entanto, ao analisar especificamente o uso de séries temporais binárias multivariadas, constatou-se que apenas 12 desses estudos tratavam de algum aspecto relacionado a essa categoria mais complexa. Isso demonstra uma lacuna significativa na literatura, uma vez que as séries temporais binárias multivariadas apresentam desafios adicionais, como a necessidade de modelar as interações entre múltiplas variáveis ao longo do tempo. A maioria dos artigos disponíveis foca em séries temporais univariadas, evidenciando a escassez de pesquisas que explorem as interdependências e correlações em dados binários multivariados, o que abre oportunidades para investigação mais profunda e desenvolvimento de novas abordagens.

Dentre os 52 artigos classificados como univariados, alguns mencionam a possibilidade de generalizar os seus modelos para um contexto multivariado. Embora o foco principal destes artigos seja em um modelo univariado ou bivariado para séries temporais binárias, os autores sugerem que extensões futuras do trabalho poderiam incluir a análise de dados multivariados, permitindo a consideração de múltiplas variáveis binárias ou contínuas simultaneamente, como o por exemplo o estudo de Mosconi e Seri (2006).

O estudo de Nandi, Kafashan e Ching (2017), classificado como multivariado, combina teoria de controle, modelos estatísticos e neurociência para explorar como padrões de disparo em redes neurais podem ser controlados e analisados, com um foco particular nas limitações e desafios dessa abordagem. Apesar do estudo abordar séries temporais multivariadas, o artigo discute as limitações do framework proposto, como a incapacidade de distinguir entre diferentes padrões que têm a mesma contagem de *spikes* e como as restrições de energia e controle afetam a viabilidade dos padrões. Um ponto importante mencionado no estudo é que o framework não diferencia entre diferentes padrões de disparo que têm a mesma contagem de *spikes*. Isso pode levar a uma classificação excessivamente ampla de padrões como viáveis, mesmo que eles possam ter dinâmicas muito diferentes.

O artigo menciona que padrões que deveriam ser classificados de maneira diferente podem ser mal classificados devido a essas limitações. Em resumo, a análise reconhece que, embora o método forneça uma estrutura útil para avaliar a controlabilidade de padrões de disparo, ele tem limitações significativas, especialmente em cenários do mundo real onde as condições não são ideais (NANDI; KAFASHAN; CHING, 2017). Diante dessas limitações, o uso dessa técnica não se mostra vantajoso para a análise de falhas em máquinas ferramenta, já que a incapacidade de distinguir entre diferentes padrões pode comprometer a precisão necessária para identificar falhas específicas em sistemas industriais complexos.

Bodner e Ceulemans (2022) abordam o desenvolvimento de um pacote de software em R chamado ConNEcT, projetado para construir redes de medidas de contingência com base em séries temporais binárias. O software oferece sete medidas de contingência diferentes e permite investigar associações simultâneas e sequências temporais entre variáveis. O artigo menciona que o ConNEcT é projetado para lidar com pares de séries temporais binárias, permitindo a investigação de associações simultâneas e sequências temporais.

No entanto, devido à sua limitação em lidar apenas com pares de séries temporais binárias, essa ferramenta não é adequada para a análise de falhas em máquinas ferramenta, onde a interação entre múltiplos componentes e variáveis é essencial para capturar a complexidade dos sistemas. A análise de pares de variáveis pode ser insuficiente para entender completamente as interdependências e padrões em sistemas multivariados, comprometendo a precisão e a eficácia na detecção de falhas. (BODNER; CEULEMANS, 2022).

Bighamian e Shanel (2018) abordam a análise de séries temporais multivariadas, modelando a atividade de disparo de múltiplos neurônios (representados como processos pontuais) em relação a várias características de Potenciais de Campo Local (LFP) em uma rede. Essa abordagem permite capturar as dependências funcionais entre os disparos neuronais e as características LFP em um contexto multiescalar, considerando a interação entre diferentes canais de LFP e a atividade de disparo de neurônios em um único modelo.

No entanto, os autores destacam que a modelagem dessas dependências condicionais apresenta desafios computacionais significativos, devido ao grande número de pares de neurônios e características de LFP a serem analisados, o que pode dificultar a detecção de dependências. Além disso, as diferenças inerentes entre os dados binários dos disparos neuronais e os LFPs, que são dados contínuos, complicam a modelagem e a análise estatística, afetando a precisão da estimativa das dependências funcionais. Diante dessas limitações, o uso dessa abordagem para análise de falhas em máquinas ferramenta não é vantajoso, pois a complexidade computacional e a divergência entre tipos de dados podem comprometer a acuracidade e a eficiência da detecção de falhas em sistemas industriais complexos (BIGHAMIAN; SHANECH, 2018).

Nishino, Maekawa e Hara (2023) utilizaram dados de aceleração coletados por sensores em três eixos (x, y e z), representando uma série temporal multivariada, para analisar movimentos complexos e detectar repetições de ações humanas. A abordagem proposta envolve a criação de uma Matriz de Autossimilaridade (SSM) a partir dessas séries, utilizada como entrada para uma rede neural, permitindo identificar de forma eficaz as localizações temporais das repetições. Apesar dos resultados promissores, o método apresenta limitações para aplicação em máquinas-ferramentas devido à análise restrita a apenas três canais e à variabilidade dos dados, como posição do sensor e natureza das atividades, que podem afetar a precisão. Além disso, as repetições nem sempre são capturadas uniformemente entre os canais, reduzindo a eficácia do modelo em certos contextos.

Hu *et al.* (2021) abordam a detecção bayesiana de padrões de propagação de eventos a partir de séries temporais binárias multivariadas. Os pesquisadores propuseram um modelo probabilístico de árvore para representar o caminho de propagação de eventos entre nós ou objetos em uma rede. Eles desenvolveram uma abordagem bayesiana completa, utilizando um algoritmo Metropolis Hastings (MH), para detectar esses caminhos de propagação de eventos a partir dos dados observados. O objetivo principal do estudo era identificar e caracterizar como os eventos se propagam entre os nós ao longo do tempo, permitindo a inferência dos padrões de propagação de eventos em uma estrutura de árvore probabilística. A pesquisa demonstrou a eficácia do método proposto em detectar com precisão os

caminhos de propagação de eventos, mesmo em cenários com múltiplas estruturas de árvore.

Entretanto, apesar de sua eficácia na identificação de padrões de propagação, a abordagem de Hu *et al.*, (2021) pode não ser adequada para análise de falhas em máquinas ferramenta. A complexidade associada à construção e análise de múltiplas estruturas de árvore e a necessidade de um modelo probabilístico sofisticado podem limitar sua aplicabilidade em ambientes industriais, onde a precisão e a simplicidade operacional são essenciais. Além disso, a modelagem de propagação de eventos entre nós em uma rede pode não capturar com clareza as interações específicas e detalhadas que ocorrem em sistemas multivariados de máquinas ferramenta (MA *et al.*, 2022).

O estudo de Gorcim *et al.*, (2011) aborda a modelagem e previsão da ocupação espectral em ambientes de comunicação sem fio, com foco na eficiência do uso do espectro. Os autores mencionam que o modelo proposto é capaz de lidar com múltiplas bandas de frequência simultaneamente, considerando a dependência entre diferentes canais de frequência ao modelar a ocupação espectral. Essa abordagem permite que o modelo capture as interações entre as variáveis de ocupação espectral em diferentes frequências, caracterizando assim uma análise de séries temporais multivariadas.

Além disso, a análise comparativa da previsão de ocupação espectral em diferentes locais também sugere que múltiplas variáveis (como a ocupação em diferentes frequências e locais) estão sendo consideradas no processo de modelagem e previsão, reforçando a natureza multivariada da análise (GORCIM *et al.*, 2011).

No entanto, embora o modelo proposto seja eficaz para prever a ocupação espectral em ambientes de comunicação sem fio, sua aplicação para análise de falhas em máquinas-ferramenta não seria vantajosa. A modelagem de espectro se concentra principalmente na eficiência de frequência e na dependência entre canais, o que não se alinha diretamente com as dinâmicas e interações complexas entre componentes mecânicos de máquinas. Além disso, a previsão de ocupação espectral não lida com o comportamento físico ou mecânico das falhas em sistemas industriais, tornando-o inadequado para capturar a variabilidade e os padrões que ocorrem em séries

temporais binárias de máquinas-ferramenta (WANG *et al.*, 2016; XU; GUO; SALEH, 2021).

O trabalho de Fasano et al. (2021) analisa séries temporais multivariadas utilizando um modelo dinâmico multivariado de Probit, onde as observações são representadas por vetores de dados binários, que podem incluir múltiplas variáveis observadas ao longo do tempo. O estudo discute a modelagem conjunta dessas variáveis binárias, utilizando distribuições unificadas de skew normais (SUN) para calcular as distribuições de filtragem, predição e suavização. A abordagem multivariada é relevante em contextos como a análise de dados financeiros, onde múltiplos fatores podem influenciar as direções de abertura de um índice de mercado, como o CAC40 (FASANO *et al.*, 2021).

No entanto, apesar de sua relevância para a análise de interdependências em séries temporais binárias multivariadas, o uso desse modelo na análise de falhas em máquinas ferramenta apresenta limitações significativas. A abordagem de probit dinâmico e a utilização de distribuições skew normais são mais adequadas para contextos em que as relações entre as variáveis são de natureza estatística ou econômica, como no caso de dados financeiros. Em sistemas industriais, como máquinas-ferramenta, as interações entre os componentes podem envolver fatores físicos e mecânicos complexos, que podem não ser adequadamente capturados por esse tipo de modelagem estatística. Além disso, a alta complexidade computacional envolvida nas distribuições SUN pode ser uma desvantagem em aplicações industriais que exigem rapidez e precisão na detecção de falhas. (QIN, 2012; UDOH; UDOM; OHAEGBUNEM, 2021).

Daniel *et al.* (2022) analisam séries temporais multivariadas no contexto das transições entre diferentes estratégias de regulação emocional ao longo do tempo. Os pesquisadores utilizam métricas de estabilidade e dispersão para entender como as mudanças nas estratégias de regulação emocional se relacionam com as avaliações subsequentes de afeto e ansiedade. A abordagem metodológica envolve a modelagem de dados de séries temporais coletados em múltiplos momentos, permitindo a análise das interações e transições entre as variáveis de interesse (estratégias de regulação emocional, afeto e ansiedade) ao longo do tempo. Essa análise é realizada em um modelo de painel cruzado.

Embora a abordagem de Daniel et al. (2022) seja eficaz para compreender transições entre variáveis emocionais ao longo do tempo, essa metodologia não é diretamente aplicável ao contexto de análise de falhas em máquinas ferramenta. A natureza das variáveis analisadas, como estratégias de regulação emocional, afeto e ansiedade, envolve dados psicológicos e subjetivos, que possuem características e dinâmicas muito diferentes das interações físicas e mecânicas observadas em máquinas. Além disso, o uso de modelos de painel cruzado, embora útil para entender mudanças comportamentais, pode não capturar adequadamente a complexidade e as interdependências entre os componentes de um sistema mecânico, o que é essencial na detecção de falhas em ambientes industriais (KRISHNAN, R.; VIRANI, 2018; VELIBEYOGLU; NOH; POZZI, 2019).

Lu e Fei (2011) analisam séries temporais multivariadas utilizando o modelo de vetor AR não estacionário (TVPVAR) para tratar e analisar séries temporais que envolvem múltiplas variáveis. O estudo menciona que, após a amostragem de um processo aleatório multidimensional, são obtidas séries temporais vetoriais, que são analisadas para extrair características relevantes para o reconhecimento de palestrantes. A abordagem multivariada permite uma análise mais rica e complexa dos sinais de fala, considerando a interdependência entre diferentes características do sinal.

Todavia, embora a abordagem TVPVAR seja eficaz no reconhecimento de padrões de fala e na análise de sinais acústicos, sua aplicação na análise de falhas em máquinas ferramenta apresenta limitações significativas. O modelo foi projetado para lidar com características específicas de sinais de fala, que possuem dinâmicas e padrões próprios, muito diferentes das variáveis físicas e mecânicas que compõem os sistemas de máquinas-ferramenta. A complexidade dos sinais analisados nesse contexto pode não ser compatível com a necessidade de identificar falhas em tempo real em um ambiente industrial, onde as variáveis envolvidas podem ser mais dependentes de fatores como desgaste mecânico e variações no desempenho do equipamento (CHEN et al., 2013; LUO; BADDOUR; LIANG, 2017).

O estudo de Quinn et al. (2022) apresenta o modelo RNN-BOF (Recurrent Neural Network for Binary Outcome Forecasting), projetado para análise de séries temporais multivariadas em contextos clínicos, utilizando dados de características

clínicas, demográficas e psicométricas. O modelo adota uma abordagem global, aprendendo a partir de uma população de pacientes para prever eventos de agressão. Embora eficaz em cenários clínicos, sua aplicação em análise de falhas em máquinas ferramenta é limitada devido às diferenças nas interações físicas e mecânicas, além da necessidade de abordagens mais individualizadas para sistemas industriais, onde padrões de falhas são frequentemente específicos a cada máquina ou componente.

Liu *et al.* (2022) utilizam a causalidade de Granger condicional para inferir a topologia de uma rede, considerando a relação entre múltiplas séries temporais que representam diferentes nós na rede. A abordagem se baseia na ideia de que a comunicação entre os nós pode ser modelada como um processo de séries temporais, onde a presença ou ausência de eventos (como a transmissão de pacotes) em um nó pode influenciar a atividade em outros nós. Além disso, explora o uso de cadeias de Markov de alta ordem para modelar o comportamento de comunicação entre os nós, especialmente em cenários com amostras pequenas, visando melhorar a precisão da detecção de topologia.

Apesar desta abordagem ser inovadora e eficaz para inferir a topologia de redes sem colaboração, ela apresenta limitações quando aplicada à análise de falhas em máquinas ferramenta. A causalidade de Granger condicional é útil para modelar interações entre nós em redes de comunicação, mas em sistemas industriais, como máquinas ferramenta, as interações físicas e mecânicas entre componentes são muito mais complexas do que as dependências temporais observadas na transmissão de pacotes (BORITH *et al.*, 2020; SAEZ *et al.*, 2020).

Além disso, o foco do estudo em redes não colaborativas e a ausência de decodificação de pacotes não se alinha diretamente com a necessidade de monitoramento contínuo e análise detalhada de dados operacionais em ambientes industriais. Essas diferenças tornam o método menos adequado para capturar as interdependências mecânicas e os padrões de falha em sistemas de máquinas ferramenta (BORITH *et al.*, 2020; SAEZ *et al.*, 2020).

Dori, Piedrahita e Louzoun (2019) tem um foco principal em séries temporais binárias univariadas, a abordagem proposta para a reconstrução de redes funcionais envolve a consideração de múltiplos nós (ou variáveis) simultaneamente, o que caracteriza uma análise multivariada. O objetivo é inferir a conectividade funcional entre esses

múltiplos nós com base em suas interações ao longo do tempo, utilizando um algoritmo de correlação parcial modificado para identificar as relações entre eles.

Ainda que a abordagem seja eficaz para reconstruir redes funcionais a partir de séries temporais binárias e inferir conectividades entre variáveis, ela apresenta limitações significativas quando aplicada à análise de falhas em máquinas ferramenta. A modelagem da conectividade entre nós com base em séries temporais binárias, usando correlação parcial, pode ser adequada em cenários onde os relacionamentos entre variáveis são relativamente simples. No entanto, sistemas industriais como máquinas ferramenta envolvem interações mecânicas e físicas complexas que podem não ser completamente capturadas por esse tipo de análise estatística. Além disso, a dependência de séries univariadas limita a capacidade de explorar interdependências mais profundas e dinâmicas entre múltiplos componentes, que são essenciais para uma detecção de falhas precisa em máquinas industriais. (BLOWERS; IRIBARNE; SCOTT, 2009; HASHEMI; AHMED; KHAN, 2015).

A partir da revisão da literatura realizada, foi possível identificar diversas abordagens para a análise de séries temporais multivariadas, incluindo métodos baseados em redes neurais, modelos probabilísticos e algoritmos de inferência de topologia. No entanto, nenhum dos métodos encontrados na literatura se mostrou diretamente aplicável aos dados coletados de sensores de máquinas ferramenta. As especificidades desses sistemas industriais, como a complexidade das interações mecânicas e físicas entre os componentes, não são adequadamente capturadas pelas técnicas existentes. Portanto, torna-se evidente a necessidade de desenvolver uma abordagem específica que considere as características particulares dos dados de séries temporais binárias provenientes de sensores industriais, a fim de melhorar a detecção de falhas e otimizar a confiabilidade dos sistemas.

4. AMBIENTE DE DESENVOLVIMENTO E ENSAIO DO MODELO PROPOSTO

O objetivo deste capítulo é apresentar as principais características do ambiente de ensaio utilizado para implementar o modelo detalhado no Capítulo 6. Neste capítulo, será apresentado o aparato experimental desenvolvido para esta tese, que inclui tanto a interface de coleta de dados quanto o dispositivo criado especificamente para simular os sinais e o comportamento de um *magazine* de um centro de usinagem vertical (Um *magazine* em um centro de usinagem é sistema utilizado para armazenar e organizar as ferramentas de corte que serão utilizadas durante o processo de usinagem). Ele permite que as ferramentas sejam trocadas automaticamente pelo equipamento de acordo com a necessidade de cada etapa do trabalho, otimizando o processo, conforme apresentado na Figura 04).

Figura 04 – Centro de Usinagem.



Fonte: Adaptado de Neway (2023).

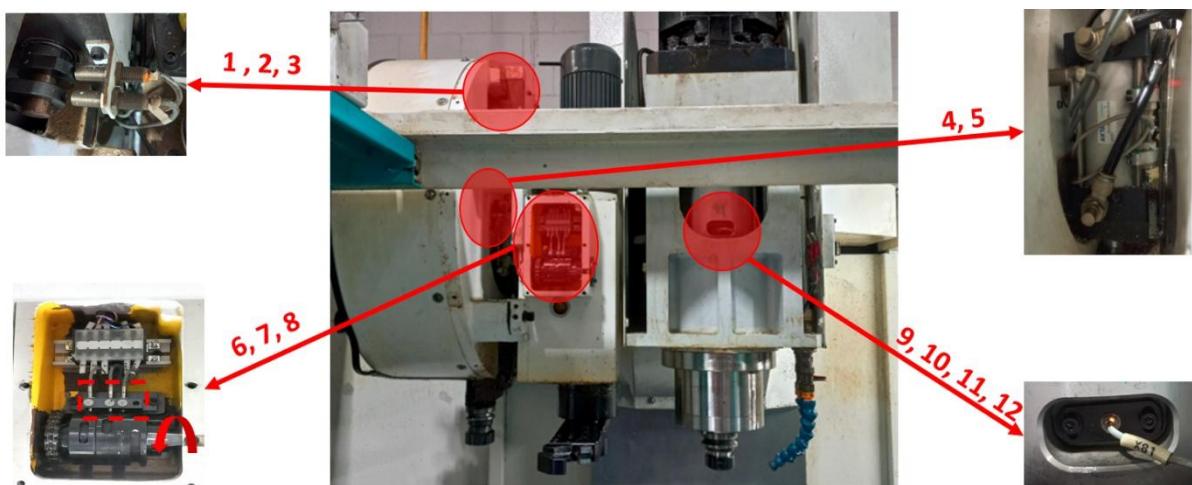
Os elementos, desenvolvidos no contexto desta pesquisa, desempenham um papel fundamental no processo de ensaio e validação do modelo proposto, permitindo a coleta de dados simulados e reais de forma eficiente e estruturada para a análise de falhas. O ambiente de ensaio foi elaborado combinando o conhecimento adquirido na academia e no mundo profissional, uma sinergia que contribui para os setores de manutenção da indústria e da academia ao demonstrar a funcionalidade de diversas técnicas abordadas em conjunto.

4.1 CARACTERÍSTICAS PRINCIPAIS DO AMBIENTE DE ENSAIO

Os conjuntos de dados foram extraídos via uma interface desenvolvida para realização desta pesquisa com a finalidade de coletar a base de dados brutos em máquinas ferramenta tais como Centros de Usinagem e Tornos CNC. Apesar da solução proposta neste trabalho ser idealizada para um cenário de mais 50 máquinas ferramentas diferentes, para fins de detalhamento dos dados e da aplicação do modelo, foi selecionado um Centro de Usinagem vertical para análise das falhas. Na Figura 05 são ilustrados os posicionamentos dos sensores do *magazine* deste Centro de Usinagem vertical, equipado com *magazine* para 24 ferramentas, e trocador tipo braço, do qual foram coletados os dados.

O sistema de troca de ferramentas deste centro de usinagem é composto por 10 sensores indutivos do tipo PNP, Positive-Negative-Positive (PNP), que operam com configuração positiva de acionamento e 24 Volts de tensão em corrente contínua (Vcc), 02 sensores magnéticos PNP tensão 24Vcc, caixa de redução do *magazine*, motor do *magazine*, caixa engrenada do braço trocador, motor do braço trocador, *drawbar* e *spindle*.

Figura 05 – Ilustração dos posicionamentos dos sensores do *magazine*.



LEGENDA

- 1 Sensor fase A •2 Sensor fase B •3 Sensor referência •4 Sensor pote acima •5 Sensor pote abaixo
- 6 Sensor pos. inicial •7 Sensor braço 90° •8 Sensor braço 180° •9 Sensor ferramenta presa
- 10 Sensor ferramenta solta •11 Sensor sem ferramenta •12 Sensor orientação

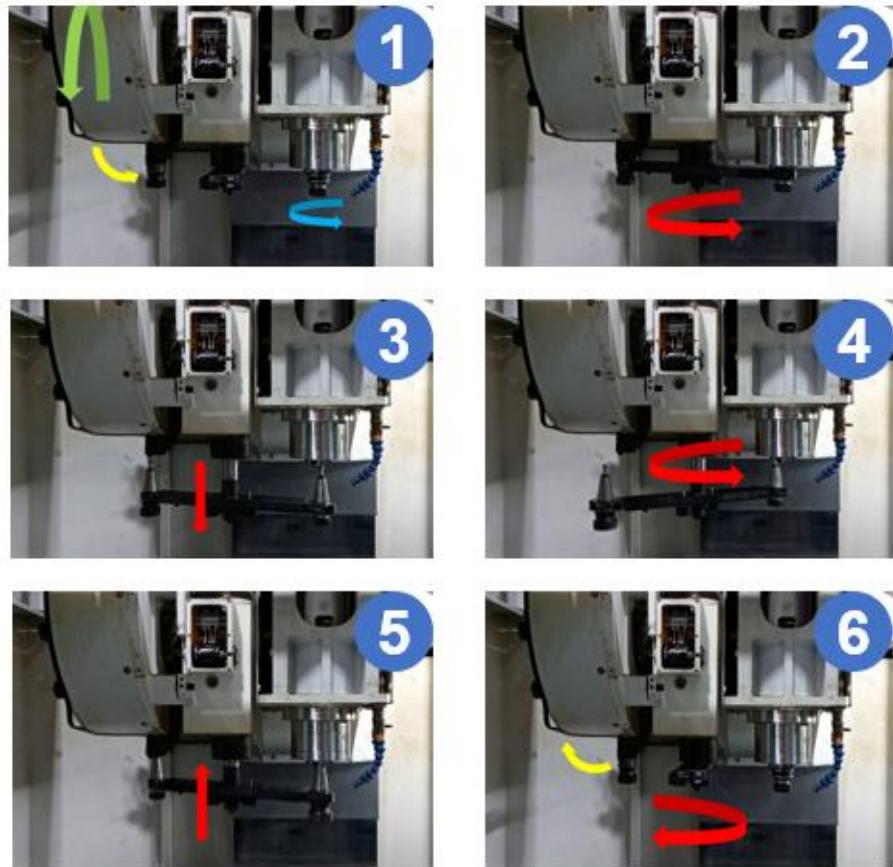
Fonte: Elaborado pelo autor.

Os dados são gerados a partir da atuação dos sensores nos componentes responsáveis por cada etapa do ciclo de troca de ferramenta do centro de usinagem. Este conjunto de dados pode trazer informações sobre a eficiência, vida útil, funcionamento do equipamento e contém informações sobre os eventos que ocorrem ao longo da operação do equipamento. Como as falhas do equipamento precisam ser identificadas e solucionadas evitando a parada dos processos de produção, aplicando-se abordagens analíticas nestes dados é possível encontrar resultados para tomadas de decisões estratégicas como no planejamento da parada do equipamento para manutenção, redução no número de falhas, redução no estoque de peças de reposição, entre outros (SEZER *et al.*, 2018).

Para realização de um ciclo de troca de ferramenta, o conjunto do *magazine* e braço trocador realiza 6 etapas, a saber:

- Etapa 1: movimentação simultaneamente do *magazine* de ferramentas para posicionar a ferramenta selecionada enquanto o *Spindle* realiza um movimento de giro para orientação do vão de encaixe no grau correto para o braço engatar na ferramenta presente no *Spindle*, em seguida o pote do *magazine* desce para posição de troca de ferramenta, deixando a ferramenta selecionada em posição.
- Etapa 2: giro de 90° do braço do trocador para engatar na ferramenta e subsequente liberação da ferramenta pelo *Spindle*.
- Etapa 3: descida do braço trocador para retirada das ferramentas do pote e do *Spindle* simultaneamente.
- Etapa 4: giro de 180° do braço do trocador para inverter a posição da ferramenta atual com a da próxima ferramenta.
- Etapa 5: subida do braço para encaixar a nova ferramenta no *Spindle* e de forma simultânea a ferramenta anterior no pote do *magazine*.
- Etapa 6: aguardo de confirmação de fixação da ferramenta, giro de 90° do braço trocador para retornar à posição inicial e subida do pote para guarda a ferramenta no *magazine*. Todas as etapas são ilustradas na Figura 06.

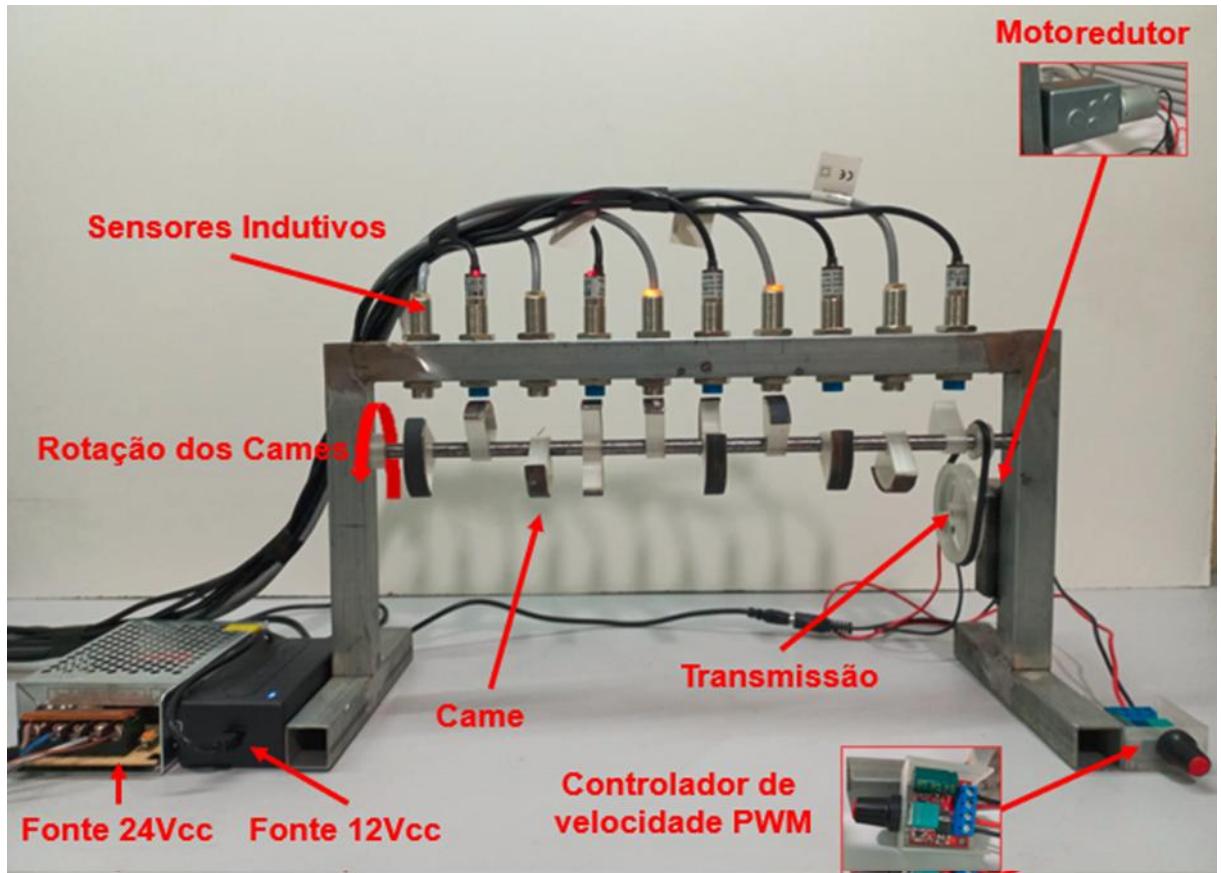
Figura 06 – Ilustração das etapas de funcionamento de um trocador de ferramentas.



Fonte: Elaborado pelo autor.

Para avaliação da metodologia proposta neste trabalho, foi desenvolvido um dispositivo para simulação dos sinais e comportamento de um *magazine* de um centro de usinagem vertical a fim de experimentação e com o objetivo de coletar o máximo de informações possível para criar atributos que ajudarão a construir algoritmos de aprendizado de máquina conforme demonstrado na Figura 07. Cabe ressaltar que o dispositivo foi especialmente projetado para coletar dados e simular falhas com a finalidade de treinar o modelo com dados de falha, visto que, se falhas forem colocadas propositalmente no *magazine* do centro de usinagem, pode-se ocorrer danos ao equipamento.

Figura 07 – Dispositivo desenvolvido para simular o funcionamento de um *magazine*.



Fonte: Elaborado pelo autor.

O dispositivo consiste em uma estrutura metálica que suporta 10 sensores indutivos e um eixo com cames (componente com um perfil ou superfície curva que impõe um deslocamento por contato ao elemento seguidor (BÖGELSACK *et al.*, 1983)), que é tracionado por um moto redutor de 12Vcc, 1.5 Amperes (A), 20 rotações por minutos (rpm) e com torque de 4 quilograma força por centímetro (Kgf/cm) controlado por uma placa com modulação por largura de pulso (PWM) que trabalha em uma faixa de tensão 5 - 35Vcc, 5A, 90 Watt (W) e frequência de até 20 quilo Hertz (khz). O funcionamento do dispositivo se dá pela rotação dos cames metálicos que ativam os sensores ao passarem por baixo da área de detecção do sensor, este movimento síncrono simula o movimento dos atuadores de uma máquina ferramenta e seu sistema de sensoreamento. A proposta deste dispositivo é proporcionar experimentos com dados que simulam o *magazine* de uma máquina em funcionamento, sob diversas circunstâncias, e dados que simulam situações de falha

de um ou mais componentes. A tabela 03 apresenta as especificações do aparato experimental.

Tabela 03 – Especificações do dispositivo

Item	Qtd.	Especificações
Motoredutor	1	12Vcc 1.5A 20rpm 4Kgf/cm
Sensor indutivo	10	PNP 24Vcc NA 3 fios
Fonte chaveada	1	24Vcc 5A
Fonte chaveada	1	12Vcc 3A
Placa controladora	1	PWM 5 - 35Vcc 5A 90W 20khz

Fonte: Elaborado pelo autor.

Os conjuntos de dados gerados de forma artificial neste dispositivo são úteis no campo de aprendizado de máquina para experimentação em aplicações de manutenção preditiva e fornecem uma coleção de dados que simulam dados históricos de funcionamento de um *magazine* de uma máquina ferramenta.

4.2 COLETA DE DADOS

Para possibilitar a coleta de dados sem interferências, foram realizados estudos, pesquisas e modificações de componentes já existentes de forma a atender a proposta de forma funcional e robusta. O sistema é composto basicamente por 6 componentes físicos, nos quais ajustes foram realizados para que possam operar coletivamente na coleta de dados em tempo real. Esta seção descreve o desenvolvimento de cada componente e os testes controlados realizados para validá-los.

O sistema proposto deve coletar os dados, fazer o envio dos dados e armazená-los em um sistema de arquivos. Os dados são compostos pelos sinais de sensores digitais que monitoram os atuadores de máquinas ferramentas. Estes dados são lidos a partir do momento em que se inicia o dispositivo e faz a transmissão a uma taxa de 9600 bits por segundos (bps). O componente escolhido para orquestrar a coleta de dados foi o Arduino Mega 2560. A tabela 04 apresenta as especificações da placa.

Tabela 04 – Especificações do Arduino Mega

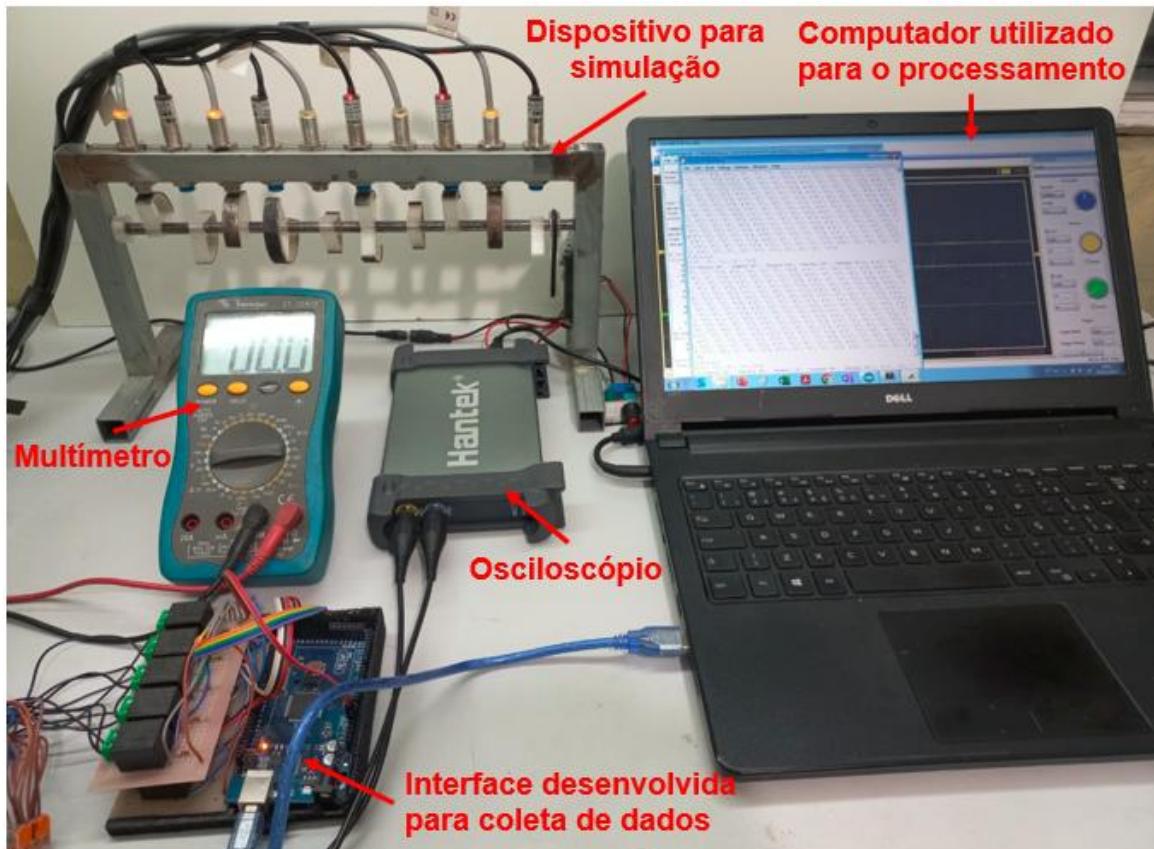
Arduino Mega ADK	Especificações
Microcontrolador	ATmega2560
Velocidade de <i>clock</i>	16MHz
Corrente continua por entrada	40mA
Corrente continua por pino 3,3V	50mA
Pinos analógicos de entrada	16
Pinos de entradas digital	54
Tensão de alimentação	7 a 12V
Tensão de operação	5V

Fonte: Disponível em: <http://arduino.cc/en/Main/ArduinoBoardMegaADK>.

Para garantir a integração eficiente de todos os componentes, foram realizados testes rigorosos de comunicação e sincronização entre os sensores e o Arduino, visando minimizar a perda de dados e garantir a precisão na coleta. Além disso, foi implementado um protocolo de comunicação serial que permite a transmissão contínua dos dados dos sensores sem sobrecarregar o sistema. Esses dados são então armazenados em tempo real em um sistema de arquivos, preparado para receber grandes volumes de informações. Cada componente foi ajustado para atender às demandas específicas do ambiente experimental, assegurando que o sistema funcione de forma estável e robusta durante os ensaios e testes de validação do modelo proposto.

Para a coleta de dados foi montado o aparato experimental em uma bancada de testes a qual é formada por um computador utilizado para o processamento, a linguagem de programação Python, Osciloscópio Digital Hantek, modelo HT6022BE, com largura de banda de 20 Mega Hertz (MHz) conectado à porta universal, *Universal Serial Bus (USB)* do computador, um multímetro Minipa ET-2042D, e o dispositivo de interface para coleta dos dados, conforme apresentado na Figura 08.

Figura 08 – Aparato experimental desenvolvido para coletar dados.



Fonte: Elaborado pelo autor.

Além dos componentes mencionados, o sistema também incluiu duas fontes de alimentação independentes, reguladas para garantir a estabilidade elétrica dos dispositivos conectados, evitando interferências que pudessem comprometer a precisão dos dados coletados. Todos os equipamentos foram devidamente calibrados e configurados para garantir medições precisas dos sinais elétricos provenientes dos sensores. A escolha do osciloscópio e do multímetro foi fundamental para monitorar e validar o comportamento dos sinais durante os testes, garantindo que o aparato experimental fosse capaz de capturar tanto as condições normais quanto as condições de falha simuladas no sistema.

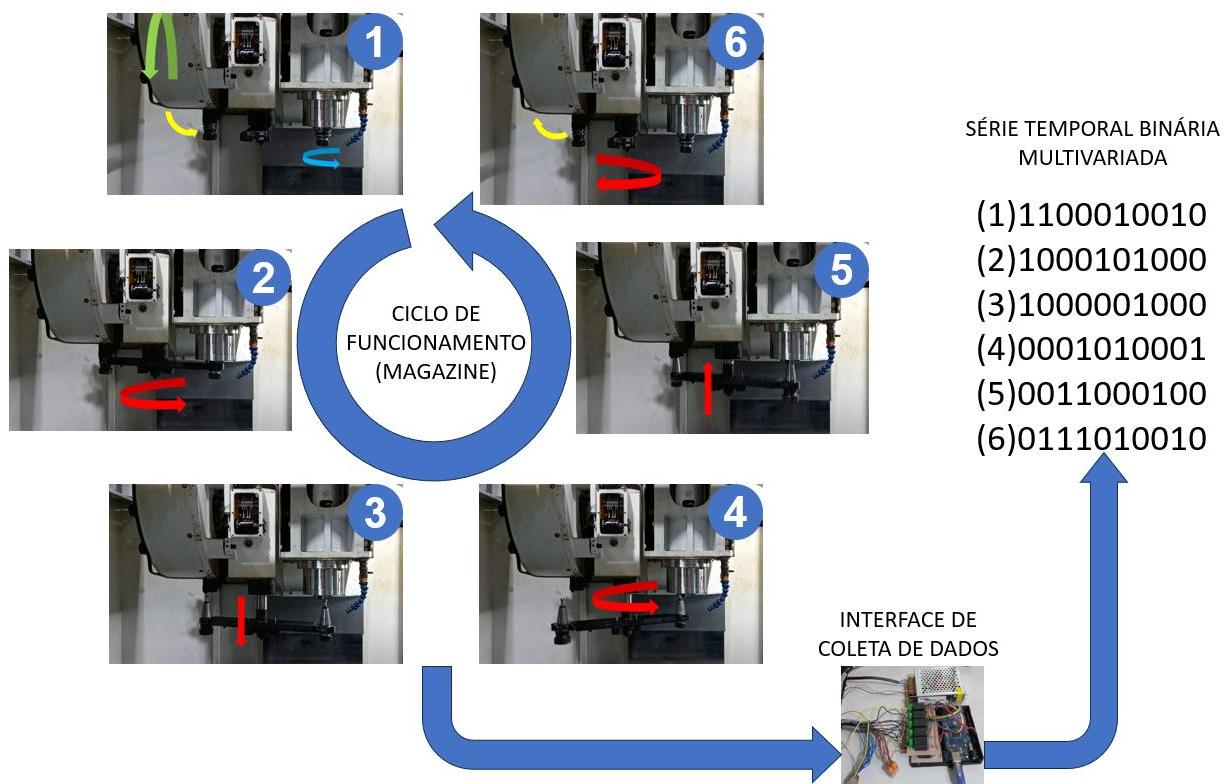
4.3 DETALHAMENTO DOS CONJUNTOS DE DADOS

Para se obter conjuntos de dados que incluísse tanto a condição normal de funcionamento quanto as condições de falha do equipamento, foi utilizado um aparato

experimental desenvolvido para simular o funcionamento de um *magazine* de ferramentas de um centro de usinagem vertical. Esse aparato permitiu a coleta de dados artificiais, representando tanto o estado normal quanto os cenários de falhas simuladas.

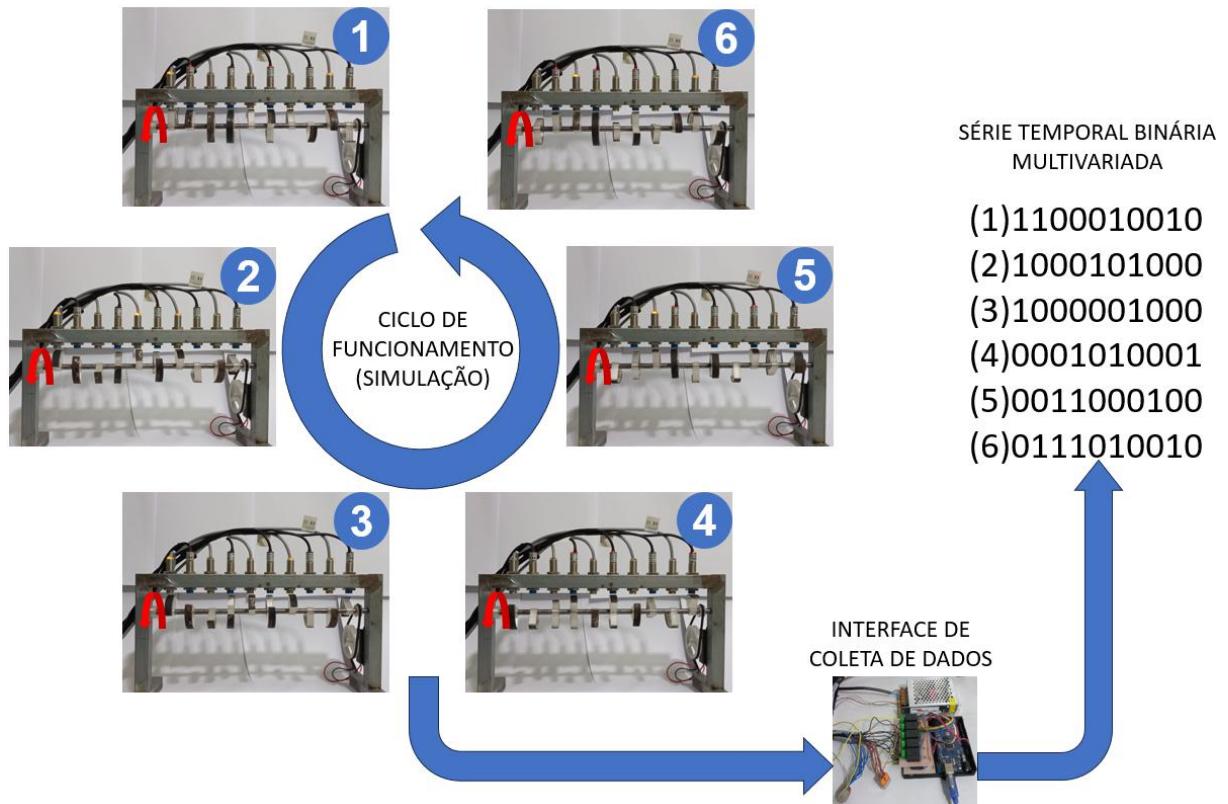
Os dados gerados a partir do próprio *magazine* ou do aparato experimental são compostos por valores binários, onde o valor 1 representa o instante em que o sensor está ligado e o valor zero representa o instante em que o sensor está desligado. Estes dados constituem uma série temporal binária multivariada que contém 10 canais relacionados entre si pelo mesmo instante de tempo (t). A Figura 09 ilustra o processo de coleta de dados reais do *magazine* de um centro de usinagem vertical e a Figura 10 ilustra a coleta de dados artificiais no aparato experimental.

Figura 09 – Ilustração da coleta de dados em uma máquina ferramenta



Fonte: Elaborado pelo autor.

Figura 10 – Ilustração da coleta de dados no aparato experimental



Fonte: Elaborado pelo autor.

Com o aparato experimental, foi possível gerar dados semelhantes aos reais do funcionamento do *magazine* de ferramentas e simular falhas idênticas às que ele apresenta. Assim, foram criados conjuntos de dados com falhas típicas de um *magazine* de ferramentas, incluindo também dados de seu funcionamento em condições normais. A caracterização dos dados normais e os dados com falha é apresentado a seguir:

- Máquina em condição normal de funcionamento (N): o comportamento da máquina em condição normal de funcionamento é dado através da ativação / desativação de cada um dos sensores em seus respectivos exatos momentos de atuação.
- Máquina em condição de falha: o comportamento da máquina em condição de falha é composto pela diferença ou falta de atuação de um ou mais sensores em relação a condição normal de funcionamento. As falhas são representadas nos dados por: atraso na atuação de um ou mais sensores; não atuação de um ou mais sensores; atuação

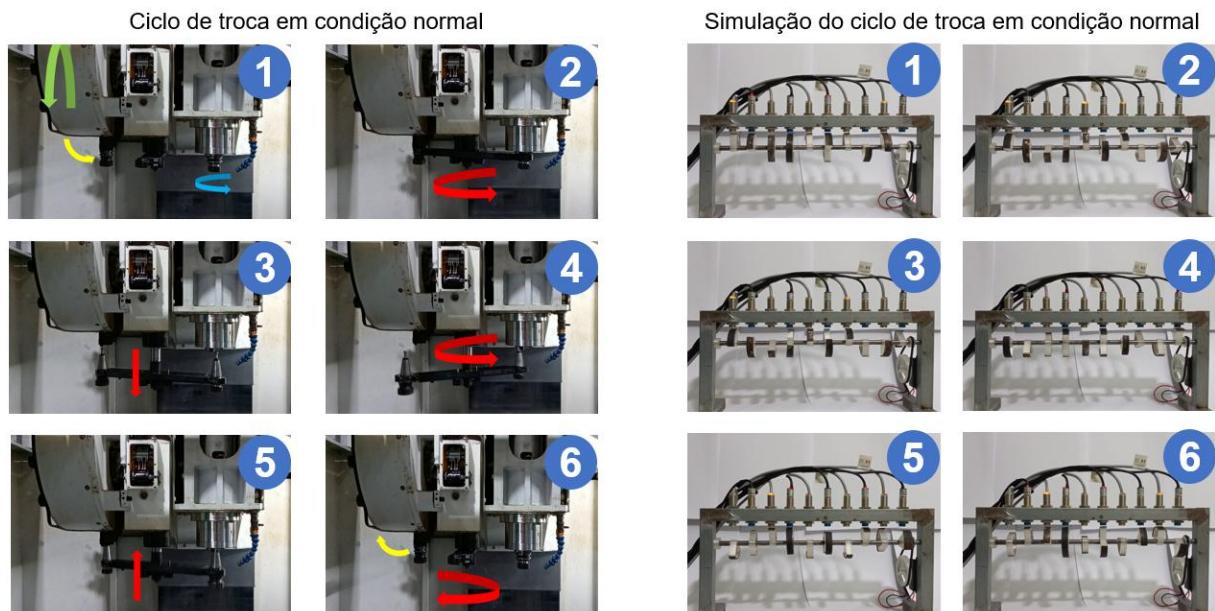
esporádica de um ou mais sensores; atuação por um período maior ou contínuo de um ou mais sensores, em relação aos demais sensores.

De forma que esses conjuntos de dados refletem o comportamento da máquina em seu estado de funcionamento normal e as variações observadas quando ocorrem falhas.

4.4 COLETA DE DADOS NO APARATO EXPERIMENTAL

Para se realizar a coleta dos dados, primeiramente foi definido no experimento a posição do *cames* e sensores para simular o *magazine* em condição normal de funcionamento, ao qual resultou em uma série temporal binária multivariada de condição normal de funcionamento da máquina (dados sem falha), conforme apresentado na Figura 11.

Figura 11 – equivalência do funcionamento mecânico entre *magazine* e aparato experimental em condição normal.

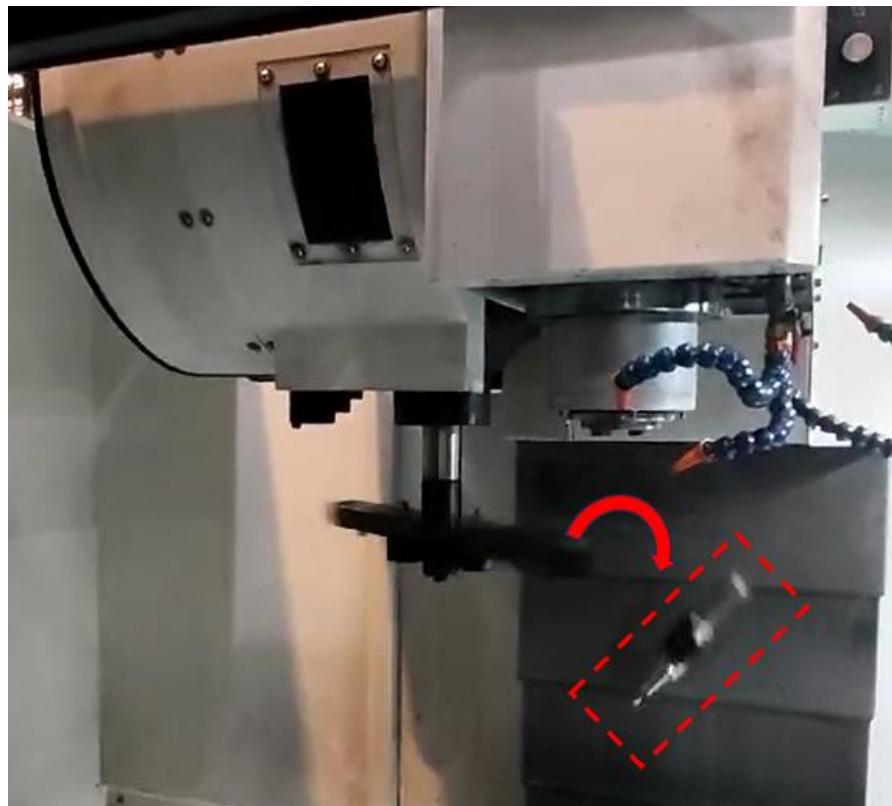


Fonte: Elaborado pelo autor.

Em seguida foi ajustada a posição do *cames* do sensor 5 a fim de que houvesse uma adiantamento do instante de atuação deste sensor em relação ao funcionamento em condição normal do *magazine*. Este adiantamento do sensor

representa uma anomalia mecânica muito comum em *magazines*, onde o braço do trocador avança mesmo sem a completa liberação da fixação da ferramenta no *spindle*, fazendo com que a ferramenta de usinagem caia durante o giro do *magazine*, configurando assim uma falha, conforme apresentado na Figura 12.

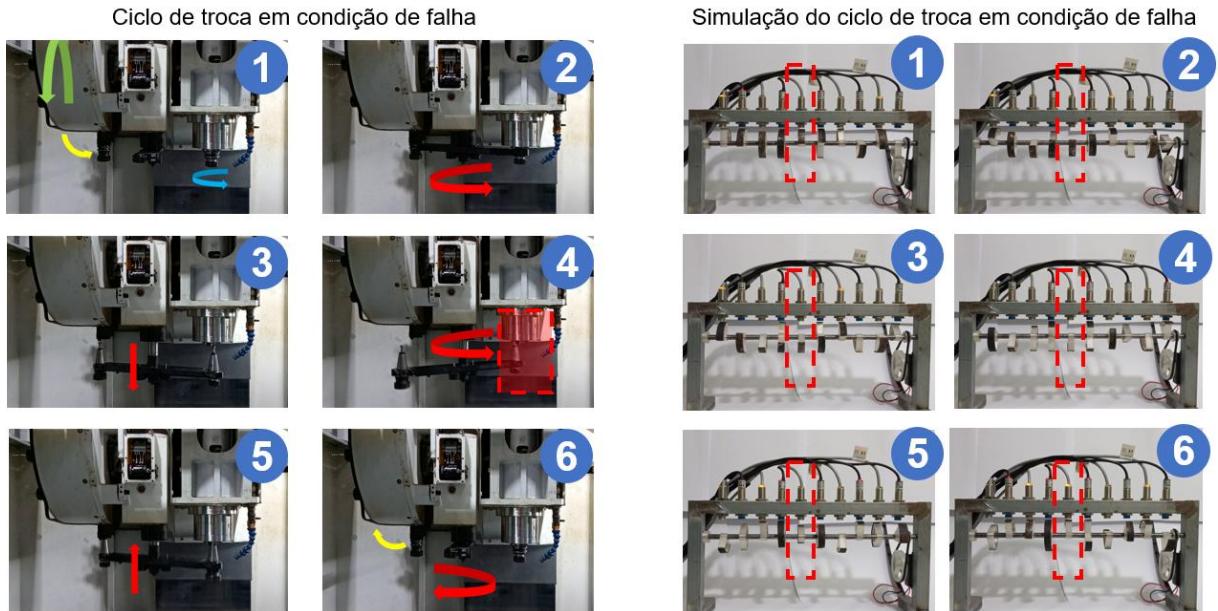
Figura 12 – Queda de ferramenta durante falha na liberação da fixação no *spindle*.



Fonte: Elaborado pelo autor.

Essa mesma situação foi simulada no aparato experimental para replicar a falha descrita. O ajuste do *cames* do sensor 5 foi reproduzido no sistema de teste, permitindo observar como o comportamento incorreto do sensor gera uma série temporal diferente da série temporal de condição normal de funcionamento. A simulação permitiu verificar a antecipação da atuação do sensor em relação à condição normal, refletindo a falha mecânica comum no processo de troca de ferramentas. Os resultados dessa simulação são apresentados na Figura 13, destacando o comportamento anômalo do sensor e seu efeito sobre o sistema.

Figura 13 – Simulação de falha no aparato experimental.



Fonte: Elaborado pelo autor.

A seguir, é apresentada a Figura 14 que ilustra a comparação entre as séries temporais coletadas nas condições de funcionamento normal e com falha. Essa comparação visual permite identificar claramente a diferença no sinal do sensor, destacando a antecipação de atuação de determinado sensor na condição de falha. A análise dessas séries temporais é fundamental para validar a eficácia do modelo de detecção de falhas, permitindo verificar como as anomalias impactam o comportamento geral do sistema.

Figura 14 – Simulação de falha no aparato experimental.

SÉRIE TEMPORAL BINÁRIA MULTIVARIADA (DADOS NORMAIS)	SÉRIE TEMPORAL BINÁRIA MULTIVARIADA (DADOS COM FALHA)
(1)1100010010	(1)1100010010
(2)1000101000	(2)1000001000
(3)1000001000	(3)1000001000
(4)0001010001	(4)0001010001
(5)0011000100	(5)0011000100
(6)0111010010	(6)0111110010

Fonte: Elaborado pelo autor.

A fim de se demonstrar a estruturação dos dados coletos, a tabela 05 apresenta os primeiros 35 registros de um conjunto de dados resultante de um experimento. Esses registros correspondem aos sinais coletados dos sensores durante o funcionamento do aparato experimental.

Tabela 05 – Primeiros 30 registros resultantes de um experimento.

Dados
1,0,1,1,0,0,0,1,0,0
1,0,1,0,0,0,0,1,0,0
1,0,1,0,0,0,0,1,0,0
1,0,1,0,0,0,0,1,0,0
1,0,1,1,0,0,0,1,0,0
1,0,1,1,1,1,1,1,0,0
1,1,1,1,1,1,1,1,0,0
1,1,1,1,1,1,1,1,0,0
1,1,1,1,1,1,1,1,0,0
1,1,1,1,1,1,1,1,0,0
1,1,1,1,0,1,1,1,0,0
1,1,1,1,0,1,1,1,0,0
1,1,1,1,0,1,0,1,0,0
1,1,1,1,0,1,0,1,0,0
1,1,1,1,0,1,0,1,0,0
1,1,1,1,0,1,0,1,0,0
1,1,1,1,0,1,0,1,0,0
1,1,1,1,0,1,0,1,0,0
1,1,1,1,0,1,0,1,0,0
1,1,1,1,0,1,0,1,0,0
1,1,1,1,1,1,0,1,0,0
1,1,1,1,1,1,1,0,1,0,0
1,1,1,1,1,1,1,0,1,0,0
1,1,1,1,1,1,1,0,1,0,0
1,1,1,1,1,1,1,0,1,0,0
1,0,1,1,0,1,0,1,0,0
1,0,1,1,0,1,0,1,0,0
1,0,1,1,0,0,0,1,0,0
1,0,1,0,0,0,0,1,0,0
1,0,1,0,0,0,0,1,0,0
1,0,1,1,1,1,1,1,0,0
1,1,1,1,1,1,1,1,0,0
1,1,1,1,1,1,1,1,0,0
1,1,1,1,1,1,1,1,0,0

Fonte: Elaborado pelo autor.

4.5 DELIMITAÇÃO DO ESCOPO DOS DADOS

Embora o aparato experimental tenha sido projetado para se assemelhar ao funcionamento de um *magazine* de ferramentas, ele possui flexibilidade suficiente para ser configurado para simular diversas sequências de dados. Essa característica torna o aparato aplicável não apenas a *magazines* de centros de usinagem, mas também a uma ampla gama de máquinas-ferramentas, como tornos CNC, mandrilhadoras, afiadoras, entre outras. Dessa forma, a proposta desta tese não se limita a uma aplicação específica, mas visa atender a diferentes tipos de equipamentos industriais, proporcionando uma solução robusta e versátil para a análise e detecção de falhas em diferentes contextos de manufatura.

O principal objetivo do aparato experimental é gerar sequências de dados nas quais falhas podem ser inseridas propositalmente, permitindo testar os modelos de detecção em uma variedade de situações. Essa abordagem possibilita a criação de cenários controlados para avaliar a eficácia dos modelos propostos. Como aplicação prática, a ideia é treinar um modelo específico para cada tipo de máquina em condições perfeitas de funcionamento e, posteriormente, utilizar esses modelos durante atendimentos de assistência técnica. Isso permitirá otimizar os atendimentos, identificando falhas de forma mais eficiente e precisa, agilizando a manutenção e reduzindo o tempo de inatividade das máquinas.

Cabe ressaltar que as máquinas ferramentas são equipadas com um CNC, responsável pelo controle preciso da movimentação dos eixos da máquina, e um CLP, que opera com uma programação, geralmente desenvolvida em linguagem LADDER. Essa programação tem como função monitorar e avaliar se a ativação e desativação dos sinais dos sensores ocorrem dentro de uma margem de tolerância aceitável, conforme o momento esperado de cada evento. Essa margem, que pode variar de acordo com a função e sensibilidade de cada sensor, é definida em milissegundos ou segundos, dependendo do processo controlado.

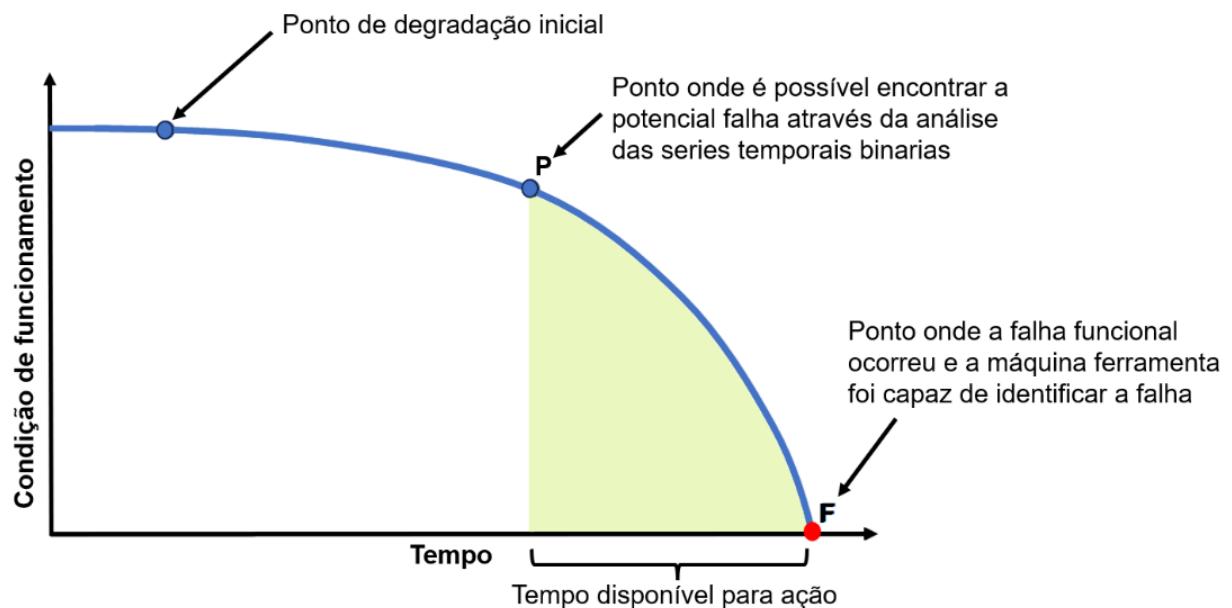
Essa lógica programada permite que o algoritmo do CLP detecte falhas na máquina ferramenta apenas no momento exato em que elas acontecem, acionando respostas rápidas para prevenir danos maiores. No entanto, esse monitoramento reativo não detecta a degradação gradual do sistema. A análise das séries temporais binárias, geradas pelos sensores, oferece uma abordagem mais preventiva, capaz de

identificar padrões que sugerem o início de uma falha antes que ela provoque uma parada inesperada.

Além disso, essa análise permite a detecção de falhas esporádicas, que podem não ser percebidas pelo monitoramento tradicional, e facilita a localização exata do componente defeituoso, sem a necessidade de desmontar toda a máquina. Isso é possível porque os modelos de detecção serão previamente treinados com dados da máquina em condição normal de funcionamento, permitindo uma comparação precisa entre o comportamento esperado e o estado atual do equipamento.

Com isso, tanto a degradação gradual quanto falhas intermitentes podem ser detectadas e corrigidas com antecedência, otimizando a manutenção preventiva e evitando paradas inesperadas ou desmontagens desnecessárias. A Figura 15 ilustra como a análise dessas séries temporais pode identificar falhas emergentes e esporádicas que o CLP, de forma tradicional, só detectaria no momento da falha.

Figura 15 – Início da degradação do sistema.



Fonte: Adaptado de Upasane *et al.* (2024).

5. MATERIAIS E MÉTODOS

Este capítulo apresenta um método de análise e detecção de falhas que emprega autoencoder. Ele também descreve as principais etapas desta estrutura, incluindo uma técnica autoral para análise de séries temporais binárias multivariada, que se baseia na análise de séries temporais binárias que contêm múltiplas variáveis de forma simultânea através da fusão destas variáveis em uma série temporal binária univariada.

5.1 CARATERIZAÇÃO METODOLÓGICA

A metodologia de pesquisa apresentada neste trabalho está estruturada como um estudo descritivo na literatura com o objetivo de avaliar e obter uma compreensão mais detalhada da aplicação do aprendizado de máquina para analisar e detectar falhas de máquinas e equipamentos. Dada a sua investigação experimental e natureza quantitativa, este trabalho também pode ser descrito como uma pesquisa aplicada, uma vez que existe uma relação direta entre os dados experimentais recolhidos e os resultados numéricos produzidos por cada modelo implementado e testado durante o estudo.

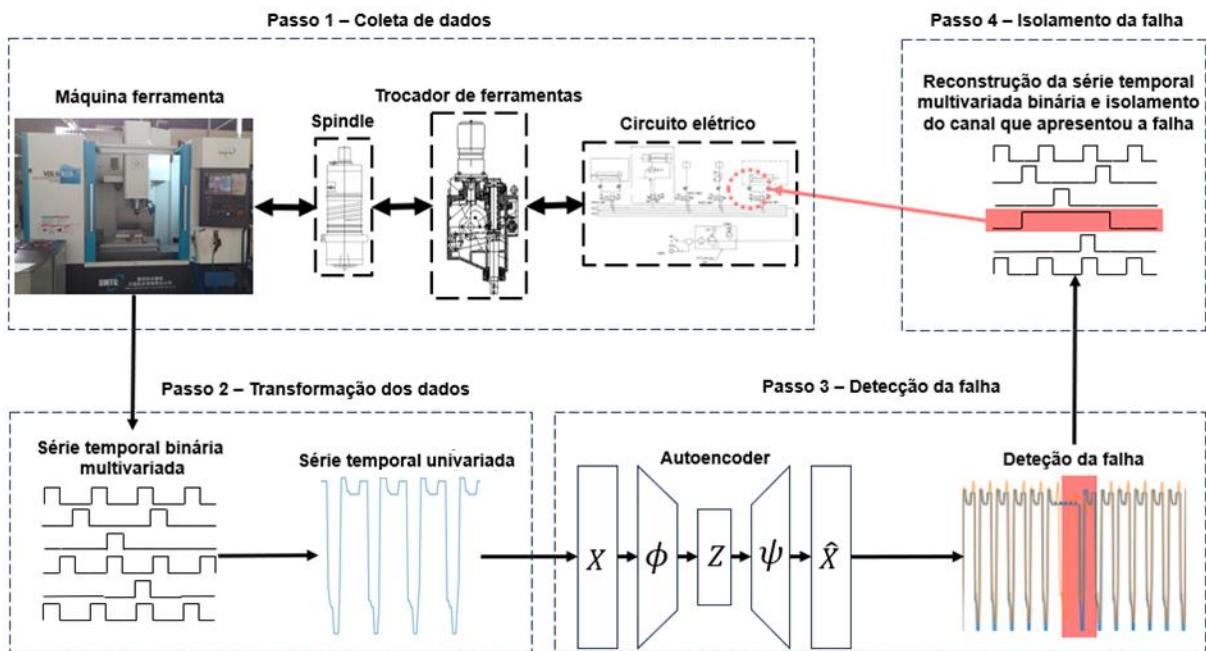
5.2 DETECÇÃO DE FALHAS

A detecção de falhas tem como objetivo identificar problemas nas fases iniciais, antes que possam causar falhas catastróficas, garantindo a segurança e eficiência operacional, além de minimizar os custos de manutenção. Anomalias, ou desvios notáveis em relação ao comportamento esperado dos dados, podem indicar o início de uma falha potencial. Para alcançar resultados eficazes, é essencial selecionar o método adequado de detecção de falhas, levando em consideração as propriedades específicas dos dados inspecionados (BRAEI; WAGNER, 2020).

A detecção de falhas em séries temporais assume que os pontos de dados não são completamente independentes, mas presume-se que os pontos de dados mais recentes na sequência influenciam as marcações de tempo seguintes. Depois disso, os valores da sequência mudam suavemente ou mostrar um padrão regular. Desta forma, mudanças repentinas na sequência podem ser consideradas uma falha (AGGARWAL, 2016).

A abordagem proposta neste trabalho utiliza um autoencoder profundo, treinado com dados de monitoramento da máquina em condições operacionais normais, para identificar falhas. Através da análise do erro de reconstrução, que ocorre quando o modelo tenta reproduzir os dados de entrada, é possível detectar desvios significativos que indicam potenciais problemas. Ele se concentra em detectar e isolar possíveis divergências nas medições de monitoramento dos dados, que podem indicar uma falha, em relação ao padrão esperado dos dados e identificar exatamente o sinal, ou sinais que estão apresentando anomalias. Essa extração é feita usando uma função univariada que representa a análise multivariada simultânea para cada *timestamp* da série temporal. Os passos propostos para aplicação prática da solução apresentada neste trabalho são apresentados na Figura 16.

Figura 16 – Abordagem proposta para análise e detecção de falhas em máquinas ferramenta.



Fonte: Elaborado pelo autor.

Os passos fundamentais para a análise e detecção de falhas, conforme ilustrado na Figura 16, serão detalhados nos subcapítulos 5.2.1 a 5.2.5. Primeiramente, será abordado a coleta de dados, seguida pela transformação dos dados, que prepara a informação para a etapa de detecção da falha com AE. Por fim, discute-se o processo de isolamento da falha, que completa a sequência de procedimentos descritos.

Com o objetivo de avaliar a eficácia do AE, também foram criadas diferentes arquiteturas de redes neurais para detecção de anomalias em séries temporais binárias. Foram desenvolvidos e treinados modelos de RNN e CNN. Através da comparação do desempenho desses modelos no conjunto de dados coletado por meio do aparato experimental, buscou-se identificar qual arquitetura apresenta maior precisão e robustez na identificação de padrões anômalos em séries temporais binária.

5.2.1 Passo 1 - Coleta dos dados

O procedimento de coleta de dados foi realizado utilizando o dispositivo de coleta desenvolvido especificamente para esta tese, descrito no Capítulo 4. Esse dispositivo foi projetado para capturar dados do estado de sensores durante os experimentos conduzidos no aparato experimental e para coletar os dados das máquinas ferramenta na aplicação prática em campo, garantindo a precisão e a qualidade necessárias para as etapas subsequentes. No Capítulo 4 são descritas as condições e configurações utilizadas para garantir a representatividade e validade das informações coletadas.

Devido ao desenvolvimento de uma interface dedicada à coleta de dados e ao acesso a muitas máquinas ferramentas, além da criação de um aparato experimental para gerar dados artificiais, a pesquisa conta com um volume abundante de dados. Isso permitiu evitar a tradicional divisão dos dados em porcentagens específicas para treinamento e teste. Em vez disso, optou-se por um enfoque no qual o modelo autoencoder foi treinado exclusivamente com dados coletados em condições normais, sem falhas, para aprender o comportamento típico da máquina. Posteriormente, os testes foram realizados utilizando dados que continham falhas, permitindo avaliar a capacidade do modelo de detectar desvios e identificar problemas com base no aumento do erro de reconstrução.

A tabela 5 apresentada anteriormente no capítulo 4.5, demonstrou como são as séries temporais binárias multivariadas que foram coletadas e serão analisadas ao longo dos próximos capítulos. Essas séries representam os sinais provenientes de múltiplos sensores instalados nas máquinas ferramentas, registrando o estado de

ativação ou desativação de cada sensor em função do tempo. Cada sensor gera uma sequência binária que reflete eventos discretos, como a ocorrência de uma determinada condição ou a mudança de estado em componentes específicos da máquina. A análise dessas séries é fundamental para identificar padrões que possam indicar o início de falhas. A visualização da tabela 5 facilitara a compreensão da estrutura dos dados e do processo de monitoramento adotado na pesquisa.

Diversos modelos de autoencoder foram treinados durante o desenvolvimento deste trabalho, com uma grande quantidade de dados, abrangendo tanto cenários reais quanto simulados. Para os dados reais, cada modelo foi treinado com aproximadamente 100.000 registros (tamanho do número de registros das séries temporais coletadas) de máquinas em condição normal de funcionamento, específicos para cada tipo de máquina. Além disso, os modelos também foram treinados com cerca de 50.000 registros de dados artificiais gerados em diferentes configurações do aparato experimental. Esse volume expressivo de dados permitiu testar os modelos em uma variedade de condições, com o objetivo de avaliar se os autoencoders conseguem detectar falhas de forma eficaz em séries temporais distintas e generalizar seu desempenho para diferentes situações e tipos de dados.

Para a aplicação prática dos modelos, o plano é coletar dados operacionais de cada tipo de máquina disponível enquanto estão em condição normal, sem falhas. Com esses dados, será treinado um modelo de autoencoder específico para cada tipo de máquina, de forma que o modelo aprenda o comportamento típico e esperado do equipamento. Posteriormente, os modelos serão utilizados para analisar os dados coletados durante o funcionamento das máquinas, identificando eventuais falhas com base em desvios em relação ao padrão aprendido. Essa abordagem permitirá a detecção precoce de problemas e otimizará a manutenção, minimizando paradas inesperadas e custos operacionais.

Para a elaboração deste capítulo e o detalhamento de cada etapa envolvida, optou-se por coletar séries temporais artificiais menores, com aproximadamente 1200 registros, geradas pelo aparato experimental. Essa escolha visa facilitar a visualização dos dados quando as séries temporais são plotadas, permitindo uma análise mais clara e detalhada das sequências de eventos e dos comportamentos dos sensores.

Dessa forma, é possível ilustrar de maneira precisa as etapas do processo e os resultados obtidos.

5.2.2 Passo 2 - Transformação dos dados

Não foi utilizada a técnica de normalização dos dados, pois as séries temporais analisadas já eram compostas por valores binários (0 e 1). Como esses dados representam estados discretos de ativação e desativação dos sensores, a normalização não se faz necessária, uma vez que os valores já estão padronizados e possuem a mesma escala. Além disso, o cuidado com a qualidade dos dados foi garantido durante a etapa de desenvolvimento da interface de coleta, assegurando que os dados capturados fossem precisos e representassem fielmente o comportamento das máquinas monitoradas.

Para a transformação dos dados, realizasse a fusão da sequência de números binários individuais em um único número para cada registro da série temporal. Esta fusão consiste em concatenar os valores binários na ordem em que aparecem, formando uma sequência contínua. Por exemplo, dada a sequência binária original de um registro ($X = (1,1,1,1,0,1,1,1,0,0)$) cada elemento representa um dígito binário em uma posição específica. Ao juntar esses dígitos sem separadores, obtém-se o número ($X = (1111011100)$), que representa a sequência original como um único valor binário contínuo. A representação matemática dessa concatenação pode ser vista através da fórmula (12):

$$\left(Y = \sum_{i=1}^n x_i \cdot 10^{n-i} \right) \quad (12)$$

Onde (n) é o comprimento da sequência binária, (x_i) é o valor binário na posição (i), com (i) variando de 1 a (n). Assim, o registro binário (X) se torna ($Y = (1111011100)$), que representa a concatenação dos dígitos binários individuais em uma única sequência continua.

Essa técnica proposta permite compactar a informação de uma série de eventos binários discretos em uma única representação, facilitando a análise e o processamento dos dados. Desta forma é feita a fusão da série temporal binária multivariada em uma série temporal univariada. A tabela 06 apresenta os primeiros 20 registros de uma série temporal binária multivariada aplicando-se a técnica de fusão para uma série temporal binária univariada. Está é a forma em que os dados são apresentados aos modelos.

Tabela 06 – Primeiros 20 registros resultantes de uma série temporal binária aplicando-se a da técnica de fusão.

Série Temporal Binária Multivariada	→	Fusão para Série Temporal Binária Univariada
1,0,1,1,0,0,0,1,0,0	→	1011000100
1,0,1,0,0,0,0,1,0,0	→	1010000100
1,0,1,0,0,0,0,1,0,0	→	1010000100
1,0,1,0,0,0,0,1,0,0	→	1010000100
1,0,1,1,0,0,0,1,0,0	→	1011000100
1,0,1,1,1,1,1,1,0,0	→	1011111100
1,1,1,1,1,1,1,1,0,0	→	1111111100
1,1,1,1,1,1,1,1,0,0	→	1111111100
1,1,1,1,1,1,1,1,0,0	→	1111111100
1,1,1,1,0,1,1,1,0,0	→	1111011100
1,1,1,1,0,1,1,1,0,0	→	1111011100
1,1,1,1,0,1,1,1,0,0	→	1111010100
1,1,1,1,0,1,1,1,0,0	→	1111010100
1,1,1,1,0,1,1,1,0,0	→	1111010100
1,1,1,1,0,1,1,1,0,0	→	1111010100
1,1,1,1,0,1,1,1,0,0	→	1111010100
1,1,1,1,1,1,1,1,0,0	→	1111111100

Fonte: Elaborado pelo autor.

5.2.3 Passo 3 - Detecção da falha

Em uma série temporal binária coletada da máquina, ou do aparato experimental, pode ocorrer a situação em que um mesmo registro binário, como (1011111100), pode representar uma falha ou não, dependendo do contexto e do

momento em que essa sequência aparece nos dados. Esse cenário torna a análise com autoencoder mais complexa, pois o modelo precisa ser capaz de reconhecer não apenas os padrões individuais, mas também o contexto temporal em que ocorrem para identificar corretamente as falhas. Essa ambiguidade exige uma abordagem mais sofisticada para distinguir comportamentos normais de anomalias, aumentando o desafio da detecção de falhas com precisão.

Para lidar com a complexidade de uma série temporal binária onde uma mesma sequência pode ou não representar uma falha, uma solução incorporar neste trabalho, foi a de utilização de janelas de tempo. Em vez de analisar os registros de forma isolada, o modelo foi treinado para considerar janelas de tempo compostas por múltiplos registros consecutivos. Essa abordagem permite ao autoencoder, ou a outros modelos testados, capturar padrões e comportamentos temporais subjacentes nos dados, indo além da simples análise de valores binários individuais.

Por exemplo, em vez de analisar apenas um valor de registro (1011111100), o modelo analisa uma sequência maior que inclui vários registros antes e depois, fornecendo contexto para entender se a sequência indica uma falha. O processo para criar estas janelas de tempo na série temporal univariada, pode ser representada matematicamente usando uma matriz que agrupa os dados em subsequências de comprimento fixo. De forma que temos a série temporal univariada ($Y = (y_1, y_2, \dots, y_N)$) onde (N) é o número total de registros na série.

Desta forma, definiu-se que (w) seja o tamanho da janela, em outras palavras, o número de registros que serão incluídos em cada subsequência. E assim a matriz de janelas será formada por subsequências (W_i), onde cada janela (W_i) contém (w) elementos consecutivos da série temporal original. A quantidade total de janelas será ($N-w+1$), devido as janelas se deslocam uma posição de cada vez, tanto nos dados coletados das máquinas ferramentas, quanto nos dados artificiais gerados no aparato experimental.

Matematicamente pode-se representar as janelas como uma matriz (W), onde cada linha (W_i) é uma janela de comprimento (w). Conforme apresentado na equação 13:

$$W = \begin{pmatrix} y_1 & y_2 & \cdots & y_w \\ y_3 & y_3 & \cdots & y_{w+1} \\ y_3 & y_4 & \cdots & y_{w+2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-w+1} & y_{N-w+2} & \cdots & y_N \end{pmatrix} \quad (13)$$

Onde cada linha ($W_i = (y_i, y_{i+1}, \dots, y_{i+w-1})$) representa uma janela de comprimento (w). E a matriz (W) terá ($N-w+1$) linhas, correspondendo ao número total de janelas possíveis, e (w) colunas, correspondendo ao tamanho de cada janela. Essa representação transforma a série temporal original em uma matriz de janelas sobrepostas, permitindo que os modelos de aprendizado de máquina utilizem o contexto temporal local para análise.

Para identificar o comprimento (w) da janela e os intervalos em que cada janela se inicia e termina em uma série temporal referente ao funcionamento de uma máquina ferramenta que trabalha em ciclos idênticos, foi utilizado abordagens da matemática de correlação. Como os ciclos são idênticos e se repetem por todo o comprimento da série temporal, tanto na máquina ferramenta quanto no aparato experimental, pode-se identificar o tamanho da janela observando os padrões que se repetem na sequência de dados.

Para isso, foi utilizado a técnica de autocorrelação para encontrar o período dos ciclos. A autocorrelação mede a similaridade entre a série temporal e uma versão deslocada de si mesma, conforme apresentado na equação 14. Os picos na função de autocorrelação indicam os pontos em que os ciclos se repetem.

$$R(K) = \sum_{i=1}^{N-k} (y_i - \bar{y})(y_{i+k} - \bar{y}) \quad (14)$$

Onde ($R(K)$) é o valor da autocorrelação para um deslocamento (k), (N) é o comprimento total da série temporal, (y_i) são os valores na série temporal univariada

e (\bar{y}) é a média da série. O valor de (k) onde ($R(K)$) apresenta picos recorrentes indica o período dos ciclos. Esse valor será o comprimento da janela (w). Depois de determinar o tamanho da janela (w), podemos segmentar a série temporal em janelas consecutivas. Cada janela (W_i) se inicia em ($y_{(i-1)w+1}$) e termina em (y_{iw}), onde (i) representa o índice da janela e varia de 1 a ($|N/w|$) (número de ciclos completos).

Os modelos autoencoders foram treinados utilizando a matriz (W), que representa as janelas extraídas da série temporal. Cada linha da matriz (W) corresponde a uma janela de dados consecutivos, capturando o comportamento da máquina ou do aparato experimental em ciclos repetitivos. Essas janelas permitem que o modelo aprenda os padrões típicos de funcionamento ao analisar os dados em segmentos que mantêm a continuidade temporal. Dessa forma, o autoencoder pode identificar desvios em relação ao padrão normal quando é testado com novos dados, indicando possíveis falhas no sistema.

Cabe ressaltar novamente que durante o treinamento, o autoencoder foi alimentado exclusivamente com dados normais, coletados em condições operacionais sem falhas. Essa abordagem permitiu que o modelo aprendesse o comportamento padrão da máquina, sem interferências causadas por falhas. Com isso, o modelo foi ajustado para minimizar o erro de reconstrução ao reproduzir janelas de dados normais, criando uma base sólida para detectar desvios significativos quando exposto a dados contendo falhas.

Cabe ressaltar novamente também que para fins de detalhamento do modelo e visualização dos resultados, foram utilizados nos exemplos a seguir, dados com aproximadamente 1.300 registros, o que permitiu plotar as séries temporais de maneira mais clara e identificar as falhas. Essa escolha facilitou o acompanhamento das janelas de dados e a observação dos padrões de erro de reconstrução gerados pelo autoencoder. Embora o treinamento completo tenha sido realizado com um volume muito maior de dados normais, o uso de um subconjunto reduzido permitiu ilustrar o comportamento do modelo e a sua capacidade de detectar anomalias de forma precisa e compreensível.

Para o desenvolvimento da solução para deteção de falhas foram utilizados Autoencoders LSTM para incorporar informação adicional sobre o estado operacional

da máquina durante o treinamento, melhorando a capacidade do modelo de distinguir entre falhas e variações normais no comportamento. Com esta abordagem, percebeu-se que a análise de séries temporais binárias se tornou mais robusta e adaptada para identificar falhas de maneira precisa, mesmo diante do desafio de uma mesma sequência binária pode assumir significados diferentes dependendo do contexto.

Para treinar o autoencoder na matriz (W) , onde (W) representa as janelas extraídas da série temporal, observou-se que a matriz $(W \in \mathbb{R}^{(N-w+1) \times w})$, onde cada linha (W_i) , é uma janela de tamanho (w) extraída da série temporal. De forma que o objetivo é treinar o autoencoder para aprender uma representação latente (z) que compacta os dados, permitindo a reconstrução das janelas de forma aproximada.

O autoencoder consiste em duas partes principais, o *encoder* que mapeia a entrada (W_i) para uma distribuição latente $(q(z|W_i))$, e o *decoder* que mapeia a variável latente (z) de volta ao espaço de entrada para reconstruir (\hat{W}_i) . Para cada janela (W_i) , o *encoder* mapeia a entrada para dois vetores: vetor de médias $(\mu \in \mathbb{R}^d)$, que representa o centro da distribuição latente e o vetor de desvios padrão $(\sigma \in \mathbb{R}^d)$, que representa a dispersão da distribuição latente. Esses vetores são obtidos através de funções (f_μ) e (f_σ) que são implementadas como redes neurais profundas, conforme apresentado nas equações 15 e 16. Na equação 16, o logaritmo do desvio padrão é utilizado para garantir que (σ) seja sempre positiva.

$$(\mu = f_\mu(W_i)) \quad (15)$$

$$(\log \sigma = f_\sigma(W_i)) \quad (16)$$

Para obter uma amostra (z) da distribuição latente $(q(z|W_i))$ utilizou-se a reparametrização de (z) , conforme equação 17.

$$(z = u + \sigma \odot \epsilon) \quad (17)$$

Onde ($\epsilon \sim \mathcal{N}(0,1)$) é uma variável aleatória amostrada de uma distribuição normal padrão, e (\odot) denota a multiplicação elemento a elemento.

O decodificador mapeia a variável latente (z) de volta ao espaço original para reconstruir a janela (\widehat{W}_i), conforme apresentado na equação 18.

$$\left(\widehat{W}_i = g(z) \right) \quad (18)$$

Onde (g) é uma função de decodificação implementada por uma rede neural.

O treinamento do autoencoder utiliza uma função de custo composta por duas partes: Erro de Reconstrução e Divergência KL. O erro de reconstrução mede a diferença entre a janela original (W_i) e a reconstruída (\widehat{W}_i), usando o erro quadrático médio (MSE), conforme apresentado na Figura 19.

$$\left(L_{recons} = \frac{1}{w} \sum_{j=1}^w (W_{ij} - \widehat{W}_{ij})^2 \right) \quad (19)$$

Divergência KL (Kullback-Leibler) regulariza a distribuição latente para que seja próxima de uma distribuição normal padrão ($\mathcal{N}(0,1)$), conforme apresentado na equação 20.

$$\left(L_{KL} = -\frac{1}{2} \sum_{k=1}^d (1 + \log(\sigma_k^2) - \mu_k^2 - \sigma_k^2) \right) \quad (20)$$

A função de custo total é a soma ponderada das duas componentes, conforme apresentado na equação 21.

$$(L_{total} = L_{recons} + \beta L_{KL}) \quad (21)$$

Onde (β) é um fator de ponderação que controla a importância da divergência (KL).

Os pesos das redes neurais que implementam (f_μ) , (f_σ) e (g) são atualizados através de gradientes descendentes, minimizando a função de custo (L_{total}) em toda a matriz de janelas (W).

Esse processo continua iterativamente até que o autoencoder seja capaz de reconstruir janelas (W_i) de dados normais com baixo erro, aprendendo assim a representação latente dos dados normais e estando preparado para detectar desvios em dados com falhas.

As falhas são detectadas com base no erro de reconstrução gerado pelo autoencoder, que foi treinado para aprender os padrões normais de funcionamento da máquina ou aparato experimental. Durante o treinamento, o modelo é ajustado para reproduzir com precisão os dados normais, resultando em um erro de reconstrução relativamente baixo para essas condições. Quando os dados coletados durante o funcionamento real da máquina ou no aparato experimental apresentam um comportamento diferente do padrão normal (como em casos de falhas), o erro de reconstrução tende a ser significativamente maior.

O autoencoder é treinado utilizando apenas dados que representam o funcionamento normal da máquina, sem falhas. O objetivo é aprender uma representação compacta e a reconstrução precisa dos dados normais, minimizando o erro de reconstrução. O erro de reconstrução (E_i) para cada janela (W_i) é calculado como a diferença entre os dados de entrada (W_i) e os dados reconstruídos (\widehat{W}_i). Com os dados normais, é possível calcular uma distribuição de erros de reconstrução e estabelecer um limiar de falha (θ), que separa os comportamentos normais dos anômalos. O limiar é definido com base em um percentil superior da distribuição dos erros de reconstrução nos dados normais (no caso, 95º percentil foi utilizado).

Após a etapa de treino, o autoencoder é utilizado para processar novos dados coletados. Para cada janela (W_i) o modelo gera uma reconstrução (\widehat{W}_i), e o erro de reconstrução (E_i) é calculado. O erro de reconstrução mede o quanto o comportamento observado difere do comportamento normal aprendido pelo modelo.

Uma falha é detectada se o erro de reconstrução (E_i) para uma janela (W_i) exceder o limiar de falha (θ), conforme apresentado na equação 22.

(Falha detectada se $E_i > \theta$)

(22)

Quando o erro ultrapassa o limiar, significa que o comportamento dos dados atuais é anômalo em comparação com os dados normais aprendidos pelo autoencoder. Isso indica a presença de uma potencial falha ou anomalia na máquina. Uma vez que as falhas são detectadas, elas podem ser analisadas para identificar a causa raiz e tomar ações corretivas. Em sistemas de manutenção preditiva, essa detecção precoce de falhas permite que intervenções sejam realizadas antes que os problemas se agravem. Portanto, a detecção de falhas com autoencoders é baseada na identificação de desvios significativos no erro de reconstrução, o que sugere que o sistema monitorado está se comportando de maneira diferente do normal.

5.2.4 Passo 4 - Isolamento da falha

Para isolar a falha detectada nos dados após a análise com autoencoder, é necessário primeiramente identificar quais janelas da série temporal ou quais variáveis específicas estão associadas ao aumento do erro de reconstrução. Como a série temporal não tem identificação explícita de tempo, mas apenas as sequências de registro dentro de cada janela, a identificação de onde a falha ocorre na série original é feita através do índice das janelas e dos registros relativos dentro dessas janelas. Cada janela (W_i) corresponde a um segmento consecutivo da série original. Embora não haja uma referência temporal explícita, o índice (i) da janela indica a ordem sequencial em que os registros ocorrem. Se o tamanho da janela é (w), a janela (W_i) cobre os registros da posição $((i - 1)w + 1)$ até a posição (iw) na série original.

Se uma janela (W_i) apresenta um erro de reconstrução acima do limiar de falha, é possível afirmar que uma falha ocorre em algum lugar dentro dessa janela. Para aumentar a precisão, a análise dos erros de reconstrução para cada variável (j) na janela pode ajudar a isolar os registros específicos que mais contribuíram para a falha. A localização da falha é feita com base na ordem dos dados, a janela (W_i) é mapeada para um intervalo de registros, e o erro de reconstrução é associado a essa sequência. Essa abordagem permite localizar e isolar falhas na sequência dos dados com base nos índices das janelas e registros dentro delas.

Desta forma, a serie temporal binária original ($Y = [y_1, y_2, \dots, y_N]$), onde (N) é o número total de registros, (w) número total de registros em cada janela e ($M = [N/w]$) sendo o número total de janelas na série temporal. Para cada janela (W_i), com (i) variando de 1 a (M), os registros das janelas são definidos de acordo com a equação 23.

$$(W_i = [y_{(i-1)w+1}, y_{(i-1)w+2}, \dots, y_{iw}]) \quad (23)$$

Portanto a janela (W_i) corresponde ao intervalo de registros da posição $((i-1)w + 1)$ até a posição (iw) na série original. Como os dados são analisados em janelas de tempo (W_i), a primeira etapa é identificar quais janelas específicas têm erros de reconstrução (E_i) que excedem o limiar (θ). Isso indicará as regiões temporais onde as falhas foram detectadas.

Dado que uma janela (W_i) tem um erro de reconstrução (E_i), a identificação das janelas com falhas pode ser calculada através da equação 24.

$$\left(E_i = \frac{1}{w} \sum_{j=1}^w (W_{ij} - \hat{W}_{ij})^2 \right) \quad (24)$$

Onde (w) é o tamanho da janela, (W_{ij}) e (\hat{W}_{ij}) são, respectivamente, os valores originais e reconstruídos para cada variável (j) na janela (W_i), calcula-se o erro individual para o valor na posição (j).

A falha é detectada se o erro de reconstrução (E_i) exceder o limiar (θ), conforme apresentado na equação 25. Se essa condição for satisfeita, considera-se que a janela (W_i) contém uma falha.

$$(E_i > \theta) \quad (25)$$

Se uma falha é detectada na janela (W_i), isto é, ($E_i > \theta$), então a falha está localizada no intervalo de registros correspondente, equação 26.

$$(y_{(i-1)w+1}, \dots, y_{iw}) \quad (26)$$

Para identificar mais precisamente a localização da falha dentro dessa janela, analisa-se os erros individuais (E_{ij}), equação 27.

$$\left(E_{ij} = (W_{ij} - \hat{W}_{ij})^2 \right) \quad (27)$$

Os valores (E_{ij}) mais altos indicam quais registros específicos ($y_{(i-1)w+j}$) são mais prováveis de estar associados à falha.

Portanto, a falha é identificada em uma ou mais janelas (W_i) e a localização é dada pelo intervalo de registros que compõem essa janela ($y_{(i-1)w+1}, \dots, y_{iw}$). Dentro desse intervalo, os índices (j) com maiores (E_{ij}) indicam os registros específicos que mais contribuem para a falha.

A partir daí, sabendo-se quais registros foram identificados com falha, é possível comparar cada registro identificado com falha com registros de janelas de dados normais. Essa comparação ajuda a entender o quanto o comportamento do registro com falha difere dos padrões observados durante o funcionamento normal. Para essa comparação, foram selecionadas várias janelas de dados normais que foram registradas em condições normais de funcionamento da máquina. Essas janelas escolhidas, são um trecho do conjunto de treinamento utilizado no treinamento do autoencoder.

Ao se comparar cada registro identificado com falha com várias janelas de dados normais, se fornece uma visão mais robusta e confiável, permitindo avaliar se o registro com falha se desvia consistentemente do padrão normal ou se as diferenças são apenas variações esperadas dentro do comportamento típico.

De forma que, quando o modelo detecta janelas com falha ($W_{falha,1}, W_{falha,2}, \dots, W_{falha,v}$), onde (v) é o número de janelas que ultrapassaram o limiar de falha, essas janelas com falha serão comparadas com um conjunto de

janelas normais ($W_{normal,1}, W_{normal,2}, \dots, W_{normal,u}$), onde (u) representa o número de janelas normais disponíveis.

Para dados binários, a comparação entre janelas de dados com falha e janelas normais pode identificar quais bits divergem do comportamento esperado. A partir daí, compara-se todas as janelas com falhas, registro por registro, em relação às janelas normais e identifica-se exatamente quais bits apresentam falha.

Para cada janela com falha ($W_{falha,i}$) e para cada registro (j), compara-se os bits com os valores correspondentes nas janelas normais ($W_{normal,u}$), onde (u), representa as diferentes janelas normais usadas para a comparação.

A fim de ilustrar de forma escrita o cálculo da diferença bit a bit entre cada registro (j) da janela com falha ($W_{falha,i}$) e os registros correspondentes em cada janela normal ($W_{normal,u}$) foi utilizado a operação XOR, que verifica se os bits são diferentes e os classifica como 1 ou iguais e os classifica como 0.

A comparação entre os bits é representada com a equação 28.

$$(\Delta_{bit,iju} = W_{falha,ij} \oplus W_{normal,uj}) \quad (28)$$

Onde ($W_{falha,ij}$) é a representação binária do registro (j) na janela com falha ($W_{falha,i}$) enquanto ($W_{normal,uj}$) é a representação binária do registro (j) na janela normal ($W_{normal,i}$) e (\oplus) denota a operação XOR, que retorna 1 se os bits são diferentes e 0 se são iguais.

Para determinar se um bit específico apresenta uma falha, calcula-se a soma das diferenças em todas as janelas normais usadas para comparação, conforme apresentado na equação 29.

$$\left(\Delta_{bit,ij} = \frac{1}{u} \sum_{u=1}^u \Delta_{bit,iju} \right) \quad (29)$$

Se ($\Delta_{bit,ij} > 0$) indica que o bit (j) do registro (i) na janela com falha difere dos padrões normais e, portanto, pode ser considerado um bit com falha. Com base nos

valores de $(\Delta_{bit,ij})$, é possível criar uma matriz que indica quais bits em cada registro são diferentes em relação aos padrões normais. Os bits que mais frequentemente apresentam diferenças podem ser considerados responsáveis pela falha.

Por exemplo, se a janela com falha (W_{falha}) possui o registro binário:

$$(W_{falha,1} = 1011010111)$$

E as janelas normais possuem os seguintes registros binários correspondentes:

$$(W_{normal,1,1} = 1011110111), (W_{normal,1,2} = 1011010111), (W_{normal,1,3} = 1011110111)$$

Comparando bit a bit com XOR, obtemos as diferenças:

$$(\Delta_{bit,1,1} = 0000100000), \quad (\Delta_{bit,1,2} = 0000000000), \quad (\Delta_{bit,1,3} = 0000100000)$$

Calculando a média das diferenças:

$$\left(\Delta_{bit,1} = \frac{1}{3} (0000100000 + 0000000000 + 0000100000) = 0000100000 \right)$$

Nesse exemplo, o resultado indica que o bit na posição 5 difere das janelas normais, indicando que ele estar relacionado à falha, ou em outras palavras, o sensor 5 apresentou falha. O processo identifica exatamente quais bits são diferentes entre os registros com falha e os registros normais, isolando os bits que apresentam comportamento anômalo.

5.2.5 Desempenho

A taxa de deteção, ou *Recall*, mede a capacidade do modelo de identificar corretamente todos os exemplos positivos, em outras palavras, as falhas verdadeiras, equação 30. As abreviações representam: positivos verdadeiros (TP), falso positivos (FP), negativos verdadeiros (TN) e falso negativos (FN).

$$\left(Recall = \frac{TP}{TP + FN} \right) \quad (30)$$

A taxa de precisão, ou *Precision*, mede a proporção de exemplos classificados como positivos que realmente são positivos, equação 31.

$$\left(\text{Precision} = \frac{TP}{TP + FP} \right) \quad (31)$$

Precisão global, ou *Accuracy*, mede a proporção de previsões corretas (tanto positivas quanto negativas) em relação ao total de exemplos, equação 32

$$\left(\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \right) \quad (32)$$

O *F1-Score* é uma métrica que combina a precisão e a taxa de detecção em uma única métrica que leva em conta tanto os falsos positivos quanto os falsos negativos. É calculado como a média harmônica entre precisão e *recall*, equação 33.

$$\left(\text{F1Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (33)$$

6. RESULTADOS

Neste capítulo, serão apresentados os experimentos realizados para o processo de análise de séries temporais e detecção de falhas utilizando autoencoder. Inicialmente, discute a identificação do tamanho de ciclos em séries temporais utilizando autocorrelação, fundamental para determinar a janela de observação dos dados. Em seguida, foi descrito o processo de treinamento do modelo, detalhando a arquitetura do autoencoder e a preparação dos dados. Este capítulo apresenta também o processo de detecção de falhas, onde o desempenho do modelo foi avaliado quanto à sua capacidade de identificar anomalias nos dados. Por fim, abordou as métricas de desempenho utilizadas para mensurar a eficácia do modelo na tarefa de detecção de falhas.

6.1 IDENTIFICAÇÃO DO TAMANHO DE CICLOS DE SÉRIES TEMPORAIS UTILIZANDO AUTOCORRELAÇÃO

O objetivo dos experimentos realizados neste capítulo foi ajustar o limiar da autocorrelação de forma a detectar corretamente os padrões cíclicos nos dados. Para identificar corretamente o tamanho dos ciclos repetidos nos dados da máquina, foi utilizada a técnica de autocorrelação, equação 10. A autocorrelação é uma ferramenta útil para detectar padrões cíclicos em séries temporais, pois calcula a semelhança da série com versões defasadas de si mesma (KHAN; SIDDIQI; LEE, 2013).

A série temporal foi analisada utilizando a autocorrelação, e múltiplos valores de limiar foram testados para identificar o primeiro pico significativo após o *lag* zero (deslocamento da série). Este pico indica a repetição do ciclo, e a distância entre o *lag* zero e o primeiro pico representa o tamanho do ciclo. Para determinar o limiar ideal, foram realizados diversos experimentos com valores de limiar variando de 0.1 a 1.5. Cada valor de limiar foi testado para verificar se o ciclo identificado correspondia ao ciclo real repetido pela série temporal.

Os experimentos L1 a L73 sobre limiar geraram os seguintes resultados, considerando séries artificiais geradas no aparato experimental com diferentes tamanhos de ciclo (27, 54, e 97 registros por ciclo) e séries reais coletadas de máquinas ferramentas (95, 109, e 221 registros por ciclo):

Limiar = 0.1 a 0.7: Para todos os valores de limiar entre 0.1 e 0.7, o ciclo detectado foi de tamanho 1 em todas as séries testadas, indicando que a autocorrelação identificou apenas padrões muito curtos, insuficientes para detectar o ciclo real.

Limiar = 0.8: Com um limiar de 0.8, o tamanho de ciclo detectado foi ainda de 1 para a maioria das séries. No entanto, a série artificial com 97 registros teve o tamanho de ciclo corretamente identificados como 97, respectivamente.

Limiar = 0.9: Esse limiar apresentou uma identificação correta dos ciclos para todas as séries artificiais e reais. Os ciclos detectados foram de 27, 54, e 97 registros para as séries artificiais, e 95, 109, e 221 registros para as séries reais, correspondendo aos tamanhos de ciclo esperados.

Limiar = 1.0: Com um limiar de 1.0, os ciclos detectados variaram significativamente, apresentando tamanhos maiores do que os ciclos reais em algumas séries. Por exemplo, foram detectados ciclos de 540, 54 e 22898 registros para as séries artificiais e de 4845, 6540 e 221 registros para as séries reais.

Limiar = 1.1 a 1.5: Para limiares mais altos, os ciclos detectados foram significativamente maiores, indicando que os valores de autocorrelação só alcançaram esses limiares em *lags* muito elevados. Por exemplo, foram detectados ciclos de 1294, 4860 e 55872 registros para as séries artificiais, e 5700, 52315 e 21216 registros para as séries reais.

A identificação do limiar ideal de autocorrelação foi importante para a correta segmentação dos dados em janelas. O processo iterativo de ajuste do limiar permitiu uma identificação precisa do ciclo repetido, servindo de base para as próximas etapas de análise de falhas. Os experimentos confirmaram que o valor de limiar de 0.9 foi o mais adequado para os conjuntos de dados, sendo utilizado nos demais experimentos desta tese. O ajuste dos limiares permitiu identificar corretamente os ciclos esperados em diferentes condições, garantindo maior precisão na análise dos dados.

A Tabela 07 apresenta um resumo dos valores de limiar testados e os respectivos tamanhos de ciclo identificados para cada série. Esta tabela ilustra como a variação dos limiares influencia a detecção dos ciclos, com o limiar de 0.9 resultando na detecção do ciclo correto para todas as séries.

Tabela 07 – Valores de limiar testados para identificação de tamanho de ciclos.

Limiar	S.T.a. 27r.	S.T.a. 54r.	S.T.a. 97r.	S.T.r. 95r.	S.T.r.109r.	S.T.r.221r.
0.1	1	1	1	1	1	1
0.3	1	1	1	1	1	1
0.5	1	1	1	1	1	1
0.7	1	1	1	1	1	1
0.8	1	1	97	1	1	1
0.9	27	54	97	95	109	221
1.0	540	54	22989	4845	6540	221
1.1	1294	4860	55872	5700	52315	21216
1.3	1295	4860	55872	5700	52320	21216
1.5	1295	4860	55872	5700	52320	21216

Legenda: S.T.a. 27r. = Série temporal artificial com 27 registros por ciclo; S.T.a. 54r. = Série temporal artificial com 54 registros por ciclo; S.T.a. 97r. = Série artificial com 97 registros por ciclo; S.T.r. 95r. = Série temporal real com 95 registros por ciclo; S.T.r.109r. = Série temporal real com 109 registros por ciclo; S.T.r.221r. = Série temporal real com 221 registros por ciclo.

Fonte: Elaborado pelo autor.

6.2 TREINAMENTO DO MODELO

O treinamento do autoencoder é uma etapa importante para seu sucesso, pois o desempenho do modelo pode ser fortemente influenciado pela qualidade e pelas características dos dados de treinamento (LAGOGIANNIS *et al.*, 2023). Um aspecto importante a considerar ao treinar autoencoders é o uso de dados normais, ou sem falhas (BANK *et al.*, 2020).

Tradicionalmente, os autoencoders são treinados com dados que não contêm falhas, pois o objetivo é que o modelo aprenda uma representação compactada dos dados de entrada que possa ser reconstruída com precisão. Essa abordagem permite que o autoencoder capture os padrões e estruturas subjacentes dos dados normais, os quais podem ser utilizados para detectar anomalias durante a fase de teste ou inferência (DELISE, 2023).

O conjunto de dados escolhido para o treinamento do autoencoder foi nomeado como (DADOS NORMAIS ARTIFICIAIS 27x51840). Este conjunto de dados contém apenas dados de funcionamento normal de uma máquina ferramenta, gerados

artificialmente no aparato experimental e compostos por 27 registros por ciclo de funcionamento da máquina, tamanho total de 51840 registros.

Uma série de experimentos foi conduzida com o objetivo de avaliar a eficácia de diferentes tipos e ajustes de autoencoders na detecção de falhas em séries temporais binárias. Diversas arquiteturas de autoencoders, com variações em camadas, neurônios e funções de ativação, foram treinadas e testadas no conjunto de dados. Os resultados obtidos permitiram identificar a configuração mais adequada para a tarefa em questão.

A tabela 08 apresenta os parâmetros utilizados no primeiro experimento, T01, conduzido para a análise de detecção de falhas por meio de um autoencoder. Neste experimento, o modelo foi treinado utilizando a matriz de janelas extraída dos dados, sem a utilização de um conjunto de validação. O objetivo deste teste inicial foi avaliar a capacidade do autoencoder de reconstruir os padrões presentes nos dados. A seguir estão descritos os parâmetros estruturais do modelo, como o número de neurônios em cada camada, a função de ativação utilizada, bem como os detalhes do processo de treinamento.

Tabela 08 – Parâmetros utilizados no experimento T01.

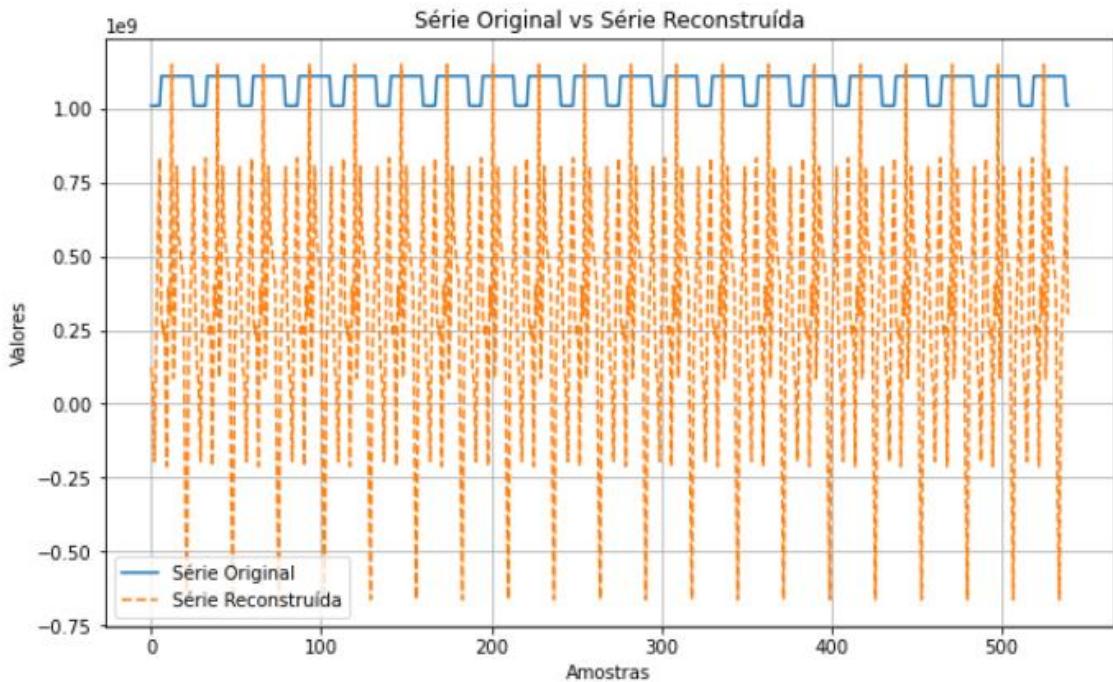
Parâmetro	Valor
Tamanho das Janelas	27
Camadas Ocultas (Encoder)	16 neurônios (ReLU), 8 neurônios (ReLU)
Camadas Ocultas (Decoder)	16 neurônios (ReLU)
Camada de Saída	27 neurônios (ativação linear)
Otimizador	adam
Função de Custo	mean_squared_error
Épocas	50
Tamanho do Lote	32
Dados de Treinamento	matriz_janelas
Embaralhamento	Sim (shuffle=True)

Fonte: Elaborado pelo autor.

O experimento T01 utilizou um autoencoder básico para aprender a reconstruir os dados em matriz de janelas e utilizou uma arquitetura simples com duas camadas de codificação (16 e 8 neurônios) e duas camadas de decodificação simétricas. O uso do otimizador adam e da função de perda de erro quadrático médio são escolhas padrão para este tipo de tarefa de reconstrução. O resultado da reconstrução dos

dados é apresentado na Figura 16, com a finalidade de facilitar a visualização da série optou-se por plotar apenas as primeiras 600 amostras.

Figura 16 – Resultado da reconstrução da série no experimento T01.



Fonte: Elaborado pelo autor.

A plotagem da série reconstruída (Figura 17) observou-se que a série original, linha azul, segue um padrão claro e repetitivo, como esperado para dados binários dos ciclos de funcionamento da máquina. A série reconstruída, linha laranja tracejada, reconstruída pelo autoencoder oscilou muito em relação à série original. Em vez de capturar os valores binários (0 ou 1), o modelo reconstruiu valores entre aproximadamente -0.5 e 0.75, e não seguiu o padrão binário da série original.

A reconstrução do autoencoder ficou longe dos valores esperados (0 ou 1) e apresentou uma oscilação que não reflete o comportamento binário da série original. Neste experimento o autoencoder não conseguiu capturar corretamente a amplitude dos dados binários, indicando que a arquitetura utilizada não estava adequada para capturar o padrão simples de alternância entre 0 e 1.

Para o experimento T02 foi selecionada uma função de ativação mais adequada para dados binários, a função de ativação *sigmoid* foi implementada, pois ela limita os valores de saída no intervalo [0, 1]. Mas a reconstrução também oscilou

muito. Para o experimento T03 foi aumentado o número de neurônios tanto no *encoder* quanto no *decoder*, preservando a ativação *sigmoid* na saída para reconstrução de dados binários. No entanto também não se obteve êxito.

Por se trabalhar com dados binários, optou-se no experimento T04 usar a função de perda binária, *binary crossentropy*, na tentativa de ajudar o modelo a aprender melhor a alternância entre 0 e 1. Conforme apresentado na tabela 09.

Tabela 09 – Parâmetros utilizados no experimento T04.

Parâmetro	Valor/Configuração
Tamanho das Janelas	27
Camadas Ocultas (Encoder)	32 neurônios (ReLU), 16 neurônios (ReLU), 8 neurônios (ReLU)
Camadas Ocultas (Decoder)	16 neurônios (ReLU), 32 neurônios (ReLU)
Camada de Saída	27 neurônios (ativação sigmoid)
Função de Custo	<i>binary_crossentropy</i>
Otimizador	<i>adam</i>
Épocas	50
Tamanho do Lote	32
Dados de Treinamento	<i>matriz_janelas</i>
Embaralhamento	Sim (<i>shuffle=True</i>)

Fonte: Elaborado pelo autor.

Mesmo com a mudança da ativação do decoder para *sigmoid*, e utilização da função de perda binária, *binary crossentropy*, a reconstrução não foi feita corretamente e função de perda ainda não estava adequada para capturar as diferenças entre os valores binários. Tentou-se então aumentar o número de neurônios nas camadas intermediárias (*encoder* e *decoder*) e adicionar mais camadas nos experimentos T05 a T11,

Como resultados dos experimentos T05 a T11, a série original continuou a seguir o padrão binário esperado, alternando entre dois valores dentro do intervalo (0, 1). indicando que os dados estavam dentro do intervalo adequado para trabalhar com a função *sigmoid*. Quanto a série reconstruída pelo autoencoder continuou oscilando, mas em uma amplitude menor. Ela apresentou flutuações e não estava capturando com precisão o padrão binário da série original. Isso sugeriu que, apesar

das melhorias, o autoencoder ainda estava com dificuldade em reconstruir corretamente a série binária.

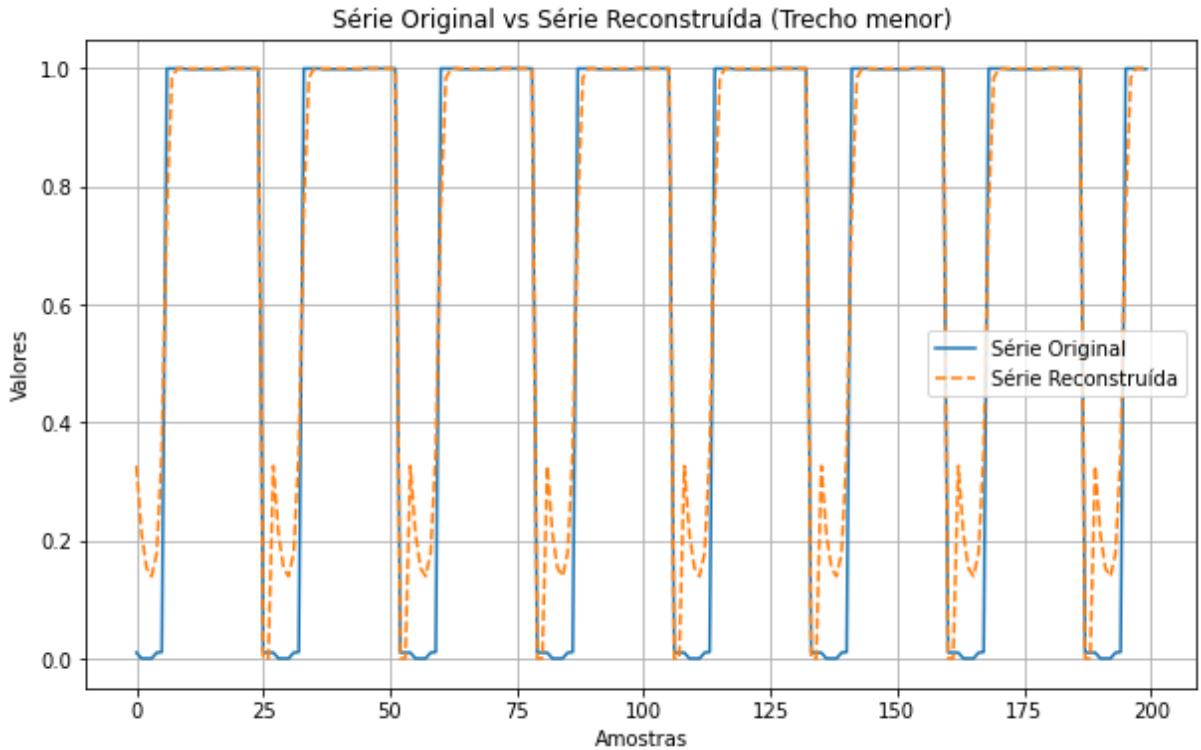
Nos experimentos T12 – T23 foi tentado adicionar mais neurônios ou camadas ao modelo. Para permitir ao autoencoder capturar mais informações sobre os padrões nos dados. Tentou-se reduzir a taxa de aprendizado do otimizador com a finalidade de se descobrir se com uma taxa de aprendizado menor permitiria que o modelo ajuste os pesos de forma mais precisa durante o treinamento, o que poderia melhorar a reconstrução. E tentou-se aumentar o número de épocas pode permitir que o modelo ajustasse melhor seus parâmetros.

Notou-se que a reconstrução da série binária continuava falhando, com a série reconstruída tendo uma oscilação muito pequena e não capturando a alternância clara entre 0 e 1 presente na série original. Desta forma, cogitou-se que apesar de aumentar a capacidade do modelo (mais camadas e neurônios), o autoencoder ainda poderia não estar aprendendo os padrões de forma adequada.

Como os dados binários possuem padrões simples, analisou-se que poderia exigir uma abordagem mais específica, como uma estrutura de autoencoder especializada em séries temporais, como autoencoder LSTM, que são mais adequados para capturar padrões temporais (DAMAVANDI *et al.*, 2019).

Nos experimentos T24 – T41 com autoencoders LSTM o resultado foi muito positivo em comparação com as tentativas anteriores. A reconstrução passou a ser visivelmente mais próxima da série original. Embora a série reconstruída segue de perto a série original, existiam pequenas oscilações, especialmente nos valores intermediários entre as transições de 0 para 1 e de 1 para 0. Com a finalidade de facilitar a visualização da série optou-se por plotar apenas as primeiras 200 amostras. Conforme apresentado na Figura17.

Figura 17 – Resultado da reconstrução da série no experimento T41.



Fonte: Elaborado pelo autor.

Nos experimentos T42 – T67 foi adicionado *dropout* ao modelo para regularização, e ajudar a prevenir *overfitting* e melhorar a capacidade do modelo de generalizar, reduzindo a oscilação. A capacidade do modelo foi aumentada ao duplicar o número de neurônios LSTM para 128 unidades. O número de épocas foi aumentado para 150, permitindo mais tempo de treinamento e ajuste dos pesos do modelo. Conforme apresentado na tabela 10.

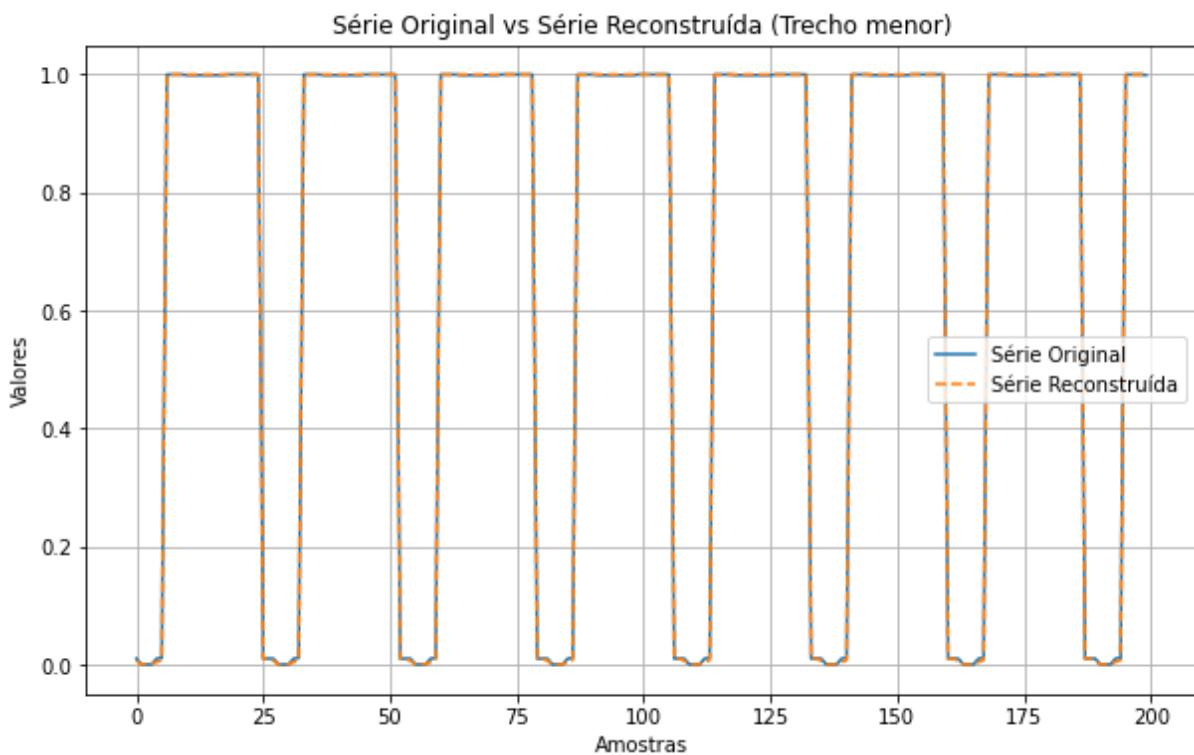
Tabela 10 – Parâmetros utilizados no experimento T67.

Parâmetro	Valor/Configuração
Tamanho das Janelas	27
Encoder	LSTM com 128 unidades (ativação ReLU)
Regularização	Dropout com taxa de 0.2
Camada de Repetição	RepeatVector (repetição de 27 vezes)
Decoder	LSTM com 128 unidades, return_sequences=True, ativação ReLU
Regularização (Decoder)	Dropout com taxa de 0.2
Saída do Decoder	TimeDistributed(Dense(1, ativação 'sigmoid'))
Função de Custo	binary_crossentropy
Otimizador	Adam (taxa de aprendizado = 0.0001)
Épocas	150 (aumento no número de épocas)
Tamanho do Lote	32
Dados de Treinamento	matriz_janelas
Embaralhamento	Sim (shuffle=True)

Fonte: Elaborado pelo autor.

No experimento T67, foi utilizada uma arquitetura LSTM Autoencoder com 128 unidades tanto no *encoder* quanto no *decoder*, além de *dropout* para regularização, com uma taxa de 0.2, para evitar *overfitting*. A taxa de aprendizado foi mantida baixa e o número de épocas foi aumentado para 150, permitindo um treinamento mais prolongado e regularizado. Obteve-se o seguinte resultado, conforme apresentado na Figura 18. Com a finalidade de facilitar a visualização da série optou-se por plotar apenas as primeiras 200 amostras.

Figura 18 – Resultado da reconstrução da série no experimento T67.



Fonte: Elaborado pelo autor.

O resultado do experimento T67 apresentado na Figura 19 demonstra que as duas séries estão muito próximas uma da outra, Série Original versus Série Reconstruída, com as transições entre 0 e 1 sendo capturadas corretamente. Isso é um indicativo de que o modelo conseguiu aprender o padrão temporal binário da série de forma eficiente. Observando-se o gráfico, as transições bruscas de 0 para 1 e de 1 para 0 são bem capturadas pelo modelo, visto terem sido um dos maiores desafios nas tentativas anteriores.

Diferente dos gráficos anteriores, não há mais oscilações ou desvios significativos entre as séries original e reconstruída. Isso sugere que o aumento da capacidade do modelo, mais unidades LSTM, e o uso de *dropout* para regularização ajudaram a eliminar as pequenas discrepâncias. O LSTM Autoencoder foi bem-sucedido em capturar e reconstruir os padrões binários da série temporal, e os ajustes que são o aumento de neurônios LSTM, adição de *dropout*, e aumento no número de épocas, resultaram em uma reconstrução muito precisa. Desta forma o modelo foi considerado pronto para uso em aplicações de deteção de falhas.

6.3 DETEÇÃO DE FALHAS

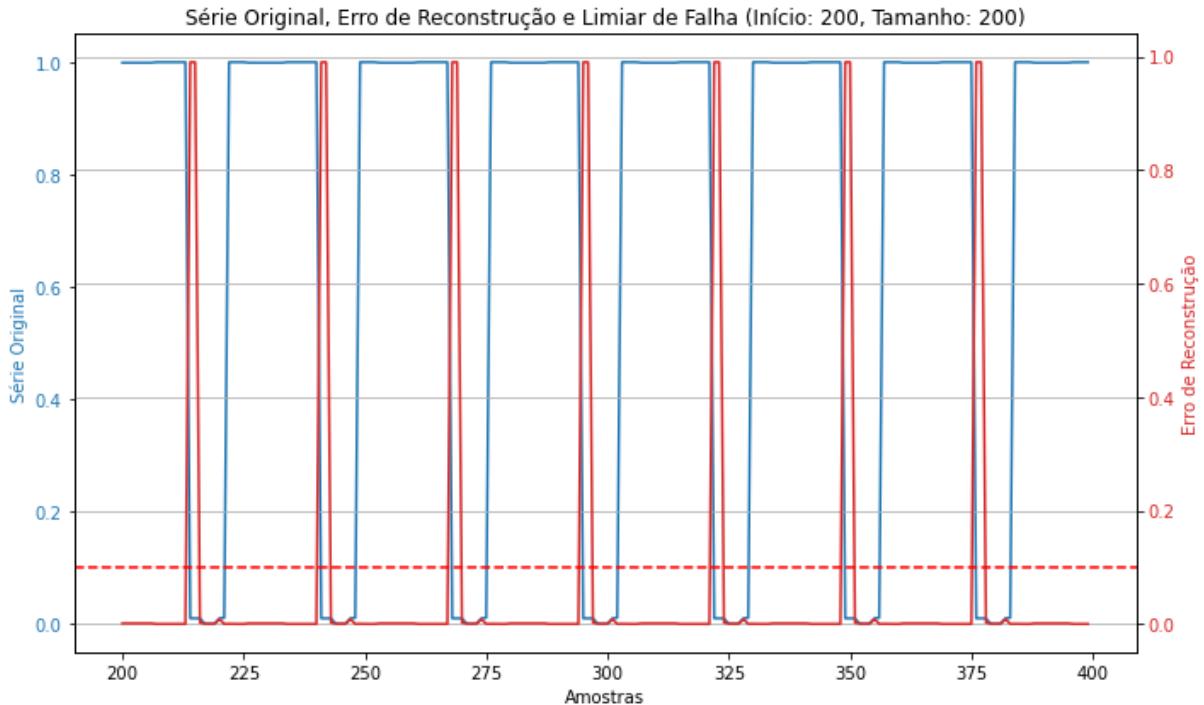
Com o modelo autoencoder salvo, pode-se usá-lo para detectar falhas em novos dados. A ideia nesta etapa é carregar o modelo salvo, passar os novos dados através dele, e comparar a reconstrução dos dados com os dados originais. Falhas são identificadas quando o erro de reconstrução, a diferença entre os dados originais e os dados reconstruídos, for maior que um certo limiar.

Para realização do experimento 1 de deteção de falhas, o modelo treinado e os novos dados são carregados, janelas são criadas com o mesmo tamanho de ciclo usado durante o treinamento, os novos dados são passados pelo autoencoder e a reconstrução é calculada.

Tem-se o erro de reconstrução como a diferença absoluta entre os dados originais e a reconstrução e define-se um limiar, no experimento D01 foi definido 0.1 para identificar as falhas. Em seguida o erro de reconstrução, a série original e o limiar de falha são plotados e as falhas destacadas. Também se apresentam os índices das amostras onde falhas foram detectadas.

O experimento 1 foi realizado com um conjunto de dados artificiais composto por 27 registros por ciclos, contendo em torno de 1200 registro no conjunto em que falhas foram colocadas propositalmente entre os registros 200 e 400. A Figura 19 apresenta a plotagem do resultado do experimento D01.

Figura 19 – Resultado da reconstrução da série no experimento D01.

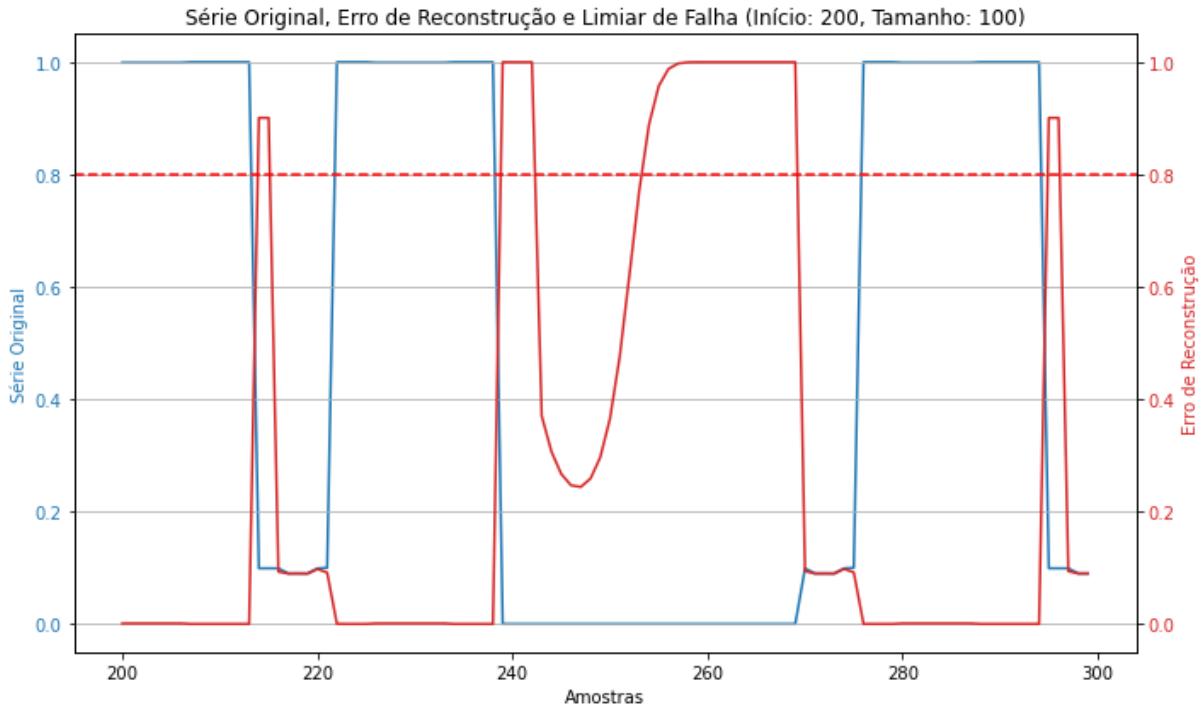


Fonte: Elaborado pelo autor.

Notou-se neste experimento que o tamanho reduzido do conjunto de dados influenciou na deteção da série original e as falhas não foram identificadas. No entanto percebeu-se que o erro de reconstrução também estava plotado de forma coerente com a série temporal. Ele seguiu um padrão muito baixo, o que indica que o autoencoder estava conseguindo reconstruir bem os dados. Há pequenos picos nos momentos de transição entre 0 e 1, mas, em geral, o erro permanece bem controlado.

Para o experimento D02, o limiar de falha foi ajustado para 0.8 e foi utilizado um conjunto de dados artificiais composto por 27 registros por ciclos, contendo em torno de 50000 registro no conjunto em que falhas foram colocadas propositalmente entre os registros 200 e 400. A Figura 20 apresenta a plotagem do resultado do experimento D02.

Figura 20 – Resultado da reconstrução da série no experimento D02.



Fonte: Elaborado pelo autor.

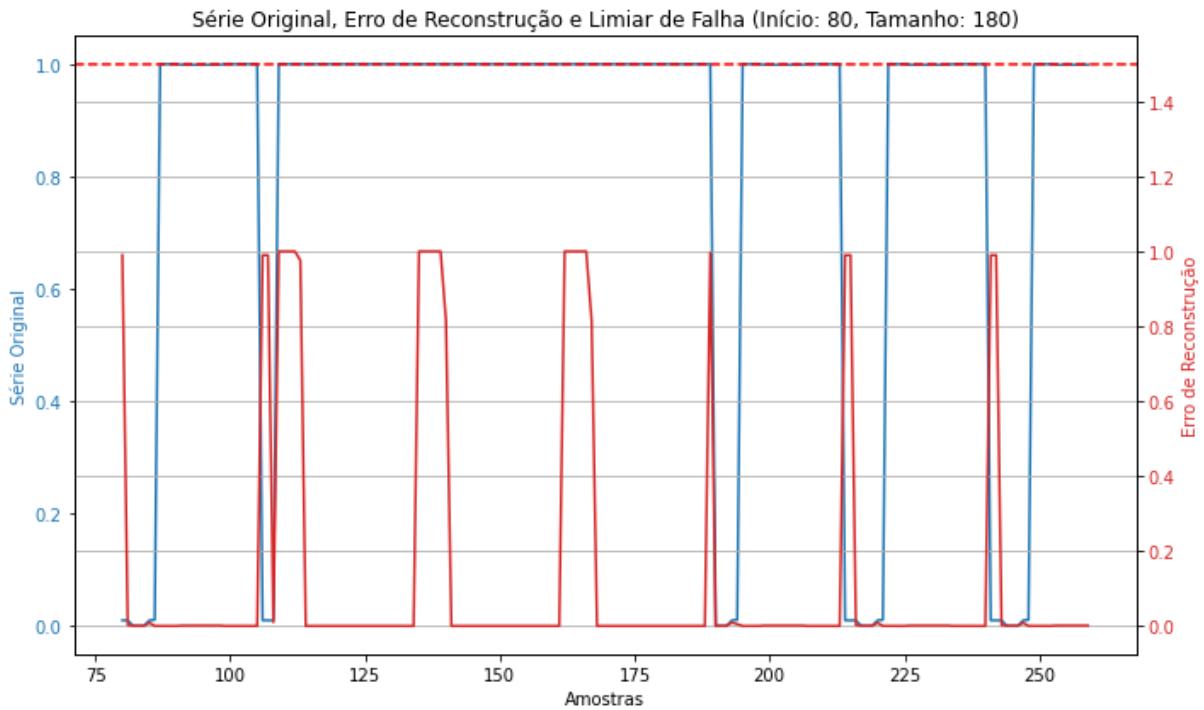
Analizando-se o novo gráfico que o experimento D02 forneceu, pode-se identificar alguns pontos importantes para consideração, tais como: a série original foi plotada corretamente, linha azul, o erro de reconstrução, linha vermelha continua, aumentou significativamente em algumas regiões, chegando a valores acima de 0.8. Isso sugeriu que o autoencoder não estava conseguindo reconstruir corretamente os dados.

O erro de reconstrução ultrapassou o limiar de falha, linha vermelha tracejada, em várias partes do gráfico do experimento D02. Isso indicou que o modelo estava detectando anomalias ou falhas nesses pontos, já que o erro estava muito maior do que o esperado.

A partir do experimento D03 foi utilizado um conjunto de dados artificiais composto por 27 registros por ciclos, contendo em torno de 50000 registro no conjunto em que falhas foram colocadas propositalmente entre os registros 110 e 190, as quais foram verificadas manualmente a fim de se ter certeza de quais eram os dados com falhas e quais eram suas localizações exatas. A Figura 21 apresenta a plotagem do resultado do experimento D03, utilizando um valor de limiar de falha de 1.5 a fim de

se observar se falhas seriam identificadas com valores de limiar mais, no entanto nenhuma falha foi identificada.

Figura 21 – Resultado da reconstrução da série no experimento 3.



Fonte: Elaborado pelo autor.

Analizando-se o gráfico do experimento D03, notou-se que o comportamento estava alinhado com o esperado, dado que as falhas foram inseridas entre os registros 110 e 190. Apesar do erro de reconstrução, linha vermelha, ser significativamente elevado entre os registros 110 e 190, confirmava que o autoencoder está detectando as falhas inseridas nesse intervalo. Isso estava de acordo com a expectativa, uma vez que falhas foram artificialmente inseridas nesses registros.

Fora do intervalo onde as falhas foram inseridas, o erro de reconstrução permaneceu baixo, indicando que o autoencoder reconheceu essas partes dos dados como normais. O gráfico do experimento D03 mostrou claramente que o autoencoder estava funcionando corretamente para detectar falhas. O modelo conseguiu capturar com precisão as anomalias inseridas entre os registros 110 e 190, o que é indicado pelos picos elevados no erro de reconstrução.

No experimento D04 o limiar de falha foi reduzido para 0.1, e o autoencoder passou a detectar mais falhas além do intervalo entre 110 e 190. Este resultado do

experimento D04 foi esperado visto que quando o limiar é ajustado para ser mais sensível, resulta na detecção de pequenas discrepâncias que, antes, não eram consideradas como falhas. O modelo detectou falhas em amostras como 80, 106-109, 214-215, e 241-242, que estão fora do intervalo (110-190). Isso sugeriu que, com o limiar mais baixo, o modelo estava capturando pequenos desvios que não podem ser considerados falhas.

Nos experimentos D05 a D29 tentou-se ajustar o limiar para um valor intermediário, entre 0.1 e 1.5, para reduzir a quantidade de falsas detecções sem perder a sensibilidade às falhas reais, visto o autoencoder está funcionando corretamente, e a sensibilidade da detecção de falhas poder ser controlada ajustando o limiar.

Como resultado dos experimentos D05 a D29, notou-se que o comportamento do modelo não estava mudando entre os limiares de 0.11 até 0.99, com a mesma lista de falhas sendo detectada. No entanto, ao testar com o limiar 1.0, o modelo não detectou nenhuma falha. Isso sugeriu que o erro de reconstrução está em uma faixa específica, onde apenas um ajuste maior do limiar causava uma diferença visível.

A partir daí, com os experimentos D30 a D42 passou-se a testar valores de ajuste de limiar até se identificar que o ajuste do limiar para 0.999 funcionou. Isso confirmou que o erro de reconstrução estava concentrado em uma faixa específica, logo abaixo de 1.0, e que um limiar mais alto era necessário para isolar as falhas relevantes. Em resumo, os experimentos com o ajuste do limiar demonstraram:

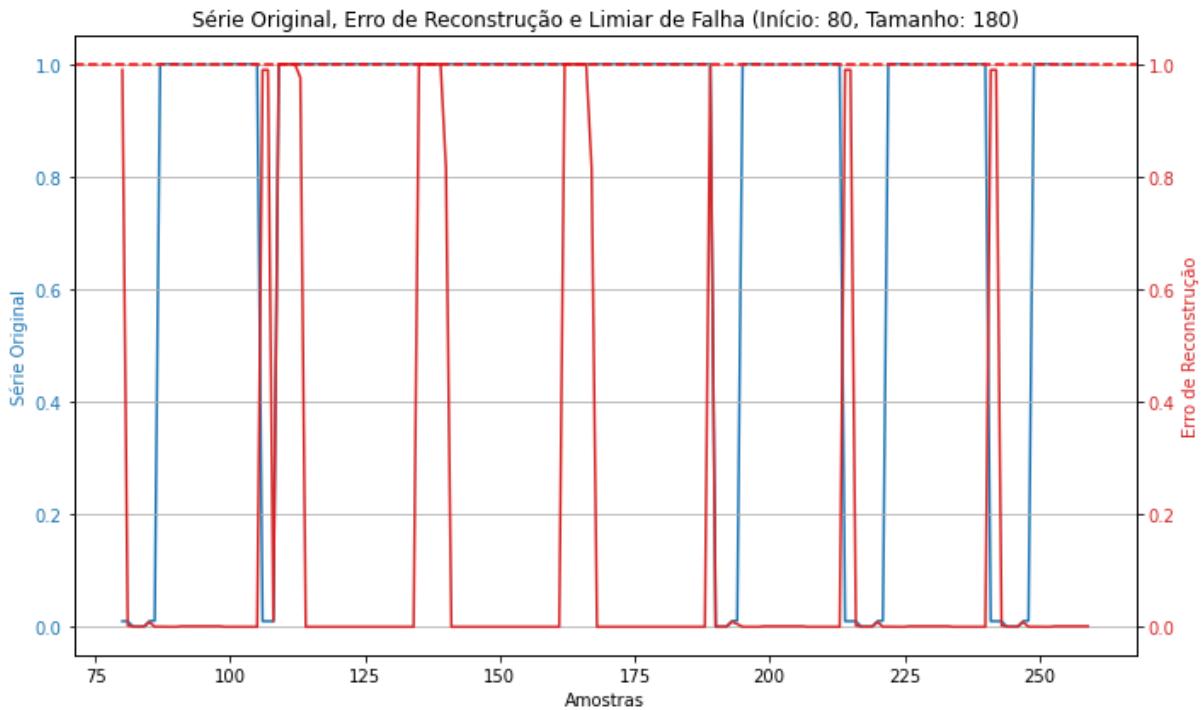
- Limiar entre 0.11 e 0.99: O erro de reconstrução ficou consistentemente acima de 0.1, mas não o suficiente para variar com esses limiares intermediários.
- Limiar 1.0: O erro de reconstrução ficou todo abaixo de 1.0, o que fez com que nenhuma falha fosse detectada.
- Limiar 0.999: Esse valor foi ajustado de forma que o autoencoder detectasse corretamente as falhas sem incluir falsos positivos, conforme apresentado nas Figuras 22 e 23.

Figura 22 – Resultado da detecção de falhas nas amostras no experimento D42.

```
In [197]: runfile('C:/Users/Intermaquinas/sem título20.py', wdir='C:/Users/Intermaquinas')
60/60 1s 6ms/step
Falhas detectadas nas amostras (no intervalo): [109 110 111 112 135 136 137 138 139 162 163 164 165 166]
```

Fonte: Elaborado pelo autor.

Figura 23 – Resultado da reconstrução da série no experimento D42.



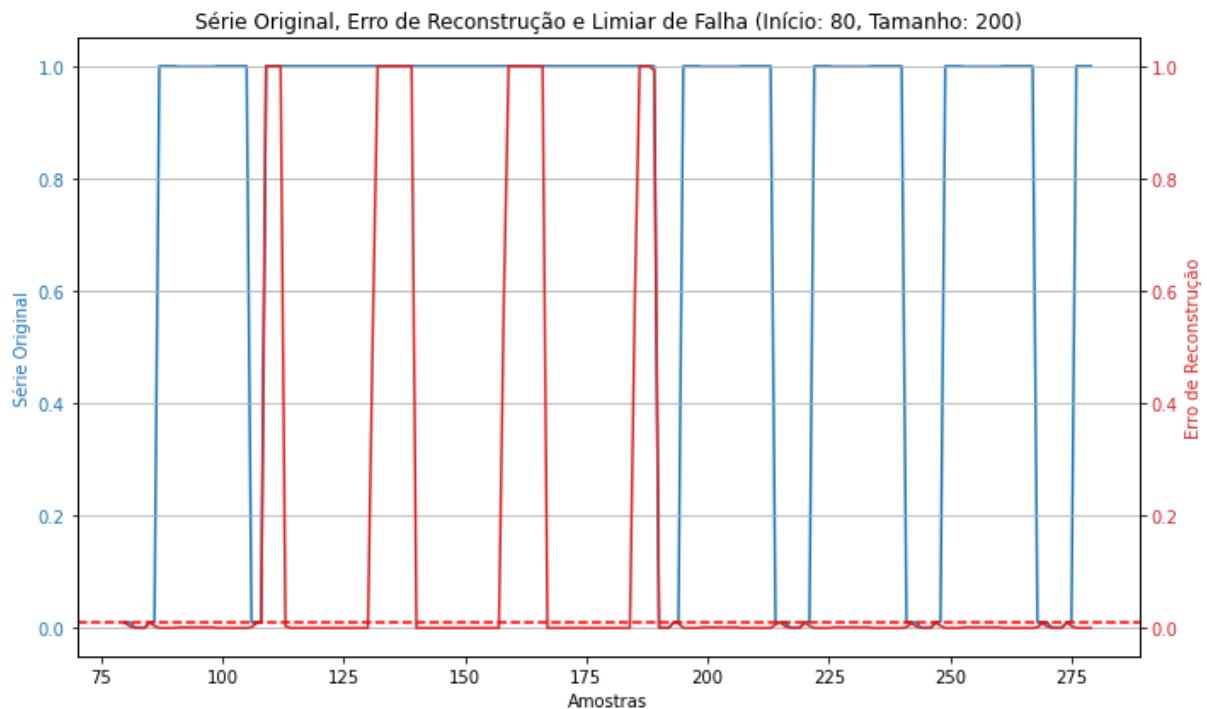
Fonte: Elaborado pelo autor.

Após a série inicial de experimentos para definir o limiar de detecção, foi realizada uma nova rodada de testes visando encontrar um equilíbrio mais adequado entre as métricas de sensibilidade e precisão do modelo. Nesses novos experimentos, D43 – D65, foram testados diferentes valores de limiar com o objetivo de ajustar o modelo de forma a capturar anomalias reais sem aumentar demasiadamente o número de falsos positivos. Ao longo dessa análise, observou-se que, embora o limiar de 0,999 fornecesse uma boa precisão, ele deixava de capturar algumas pequenas discrepâncias que poderiam ser relevantes para uma detecção de falhas mais sensível.

Por fim, após várias iterações e refinamentos nos experimentos, chegou-se ao limiar de 0,01 como o valor mais adequado. Esse ajuste final demonstrou um melhor equilíbrio entre as métricas, permitindo ao modelo identificar falhas com maior sensibilidade e abrangência, sem comprometer a estabilidade geral na classificação

de amostras normais. Destaca-se a definição de limites como uma limitação do trabalho. A Figura 24 apresenta a plotagem do trecho da série no experimento D65.

Figura 24 – Resultado da reconstrução da série no experimento D65.



Fonte: Elaborado pelo autor.

6.3.1 Localização de falhas

Conforme descrito no capítulo 5.2.4, o modelo de autoencoder foi ajustado para detectar falhas com base no erro de reconstrução das janelas de dados. No entanto, para aprimorar o processo de detecção e possibilitar uma análise mais detalhada, foi implementado um isolamento de falhas dentro de cada janela onde uma anomalia foi detectada.

Esse processo de isolamento permite localizar exatamente quais bits dentro de uma janela apresentam falhas, ao comparar o erro de reconstrução ponto a ponto com o limiar de falha previamente ajustado. Ao realizar essa análise detalhada, foi possível detectar com precisão os pontos onde o erro de reconstrução foi maior que o limiar, indicando uma falha no funcionamento.

Com essa abordagem, não apenas foi possível detectar as janelas que continham anomalias, mas também identificar quais bits específicos estavam defeituosos. Isso é particularmente importante no contexto de uma máquina-ferramenta, onde cada bit de dados está relacionado ao funcionamento de um sensor ou parte crítica da máquina.

Através da identificação desses bits com falhas, é possível determinar quais sensores ou componentes específicos da máquina-ferramenta apresentaram comportamento anômalo, facilitando a intervenção técnica para manutenção corretiva. Esse nível de detalhamento aprimora o processo de manutenção preditiva, permitindo que os defeitos sejam corrigidos antes que resultem em falhas maiores ou paradas inesperadas da máquina. A Figura 25 apresenta a saída do modelo identificando as janelas e bits com falhas, aplicando-se a metodologias descrita no capítulo 5.2.4

Figura 25 – Saída do modelo AE identificando os sensores específicos que falham no experimento D65 do capítulo 6.3.

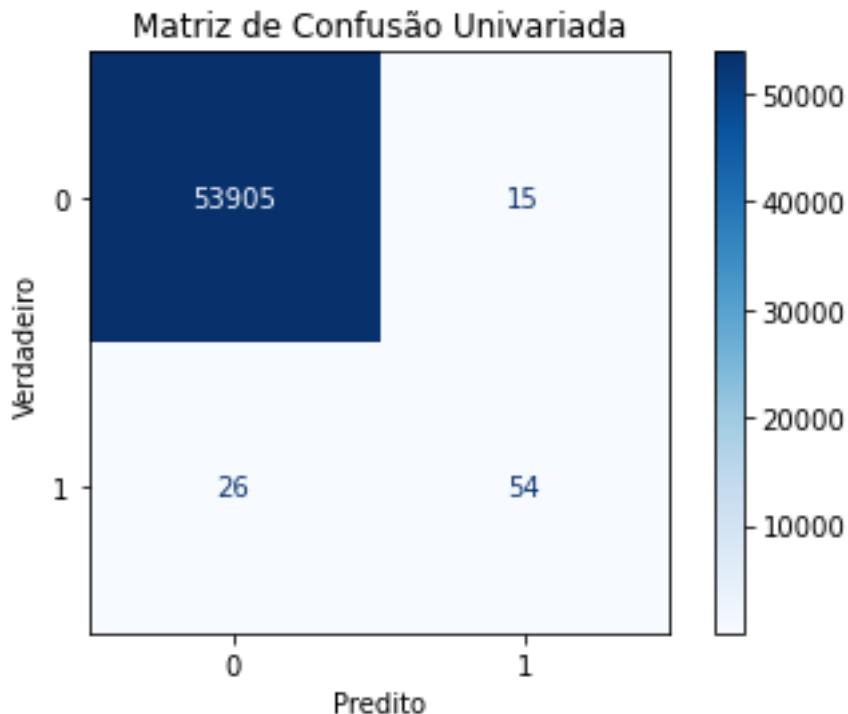
```
In [200]: runfile('C:/Users/Intermaquinas/detectar e isolar falhas AS LSTM OK final.py', wdir='C:/Users/Intermaquinas')
60/60 1s 6ms/step
Falhas detectadas nas amostras: [109 110 111 112 135 136 137 138 139 162 163 164 165 166]
Falhas detectadas na janela 4 (amostras 108-135): Bits com falha: [1 2 3 4]
Falhas detectadas na janela 5 (amostras 135-162): Bits com falha: [0 1 2 3 4]
Falhas detectadas na janela 6 (amostras 162-189): Bits com falha: [0 1 2 3 4]
```

Fonte: Elaborado pelo autor.

6.4 MÉTRICAS DE DESEMPENHO

A matriz de confusão fornece uma representação detalhada dos resultados das previsões do modelo em relação aos rótulos verdadeiros, organizando-os em uma tabela que mostra o número de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. A partir dessa matriz, é possível calcular várias métricas de desempenho, como precisão, *recall* e *F1-score*, que ajudam a entender a eficácia do modelo em identificar corretamente as classes. Além disso, a matriz de confusão permite visualizar onde o modelo está cometendo erros, facilitando a identificação de classes que podem estar mal representadas ou onde o modelo pode estar se confundindo (CAO; LIU; HUANG, 2024). A Figura 26 apresentam a matriz de confusão para análise de desempenho do modelo.

Figura 26 – Matriz de confusão do modelo AE.



Fonte: Elaborado pelo autor.

Analizando-se a matriz de confusão do experimento D65 do capítulo 6.3., pode-se observar que:

- Verdadeiros Negativos (TN): O modelo classificou corretamente 53.905 casos como não falha. Isso é um excelente resultado, indicando que a maioria dos casos normais foi corretamente identificada.
- Falsos Positivos (FP): Apenas 15 caso foi incorretamente classificado como falha quando, na verdade, não era uma falha. Isso é um ótimo resultado, já que um baixo número de falsos positivos significa que o modelo raramente gera falsos alarmes.
- Falsos Negativos (FN): O modelo não detectou 26 casos de falhas reais, classificando-os incorretamente como não falha. Esse é um ponto que poderá ser melhorado no modelo, já que idealmente, deseja-se minimizar os falsos negativos, especialmente em sistemas de manutenção preditiva.

- Verdadeiros Positivos (TP): 54 falhas reais foram corretamente identificadas como falhas. O modelo indicou corretamente várias falhas, o que indica uma capacidade de detecção razoável para anomalias.

Em resumo o modelo teve um excelente desempenho em detectar e classificar corretamente os casos normais, alto número de verdadeiros negativos e pouco falso positivo. No entanto o modelo está perdendo várias falhas reais (alto número de falsos negativos) e identificando um número relativamente baixo de falhas corretamente. Isso sugere que o modelo pode estar sub-detectando falhas, possivelmente porque o limiar ainda não está perfeitamente ajustado para capturar anomalias mais sutis e serão necessários mais experimentos.

O modelo também foi avaliado quanto a taxa de detecção que se refere à capacidade do modelo em identificar corretamente as anomalias presentes no conjunto de dados, enquanto a precisão indica a proporção de detecções corretas em relação ao total de detecções realizadas pelo modelo. A taxa de detecção é um parâmetro importante para avaliar a eficácia dos modelos na identificação de eventos anômalos, pois uma taxa alta sugere que o modelo é eficiente em reconhecer falhas ou comportamentos fora do padrão. Por outro lado, a precisão é fundamental para entender o grau de confiabilidade das detecções realizadas pelo modelo, minimizando o número de falsos positivos (FIORENTINI; LOSA, 2020).

Essas métricas fornecem uma visão abrangente sobre a eficácia e a confiabilidade de cada modelo na detecção de anomalias. Modelos com alta taxa de detecção e precisão são preferidos, pois garantem que a maioria das anomalias é identificada corretamente, reduzindo o risco de falhas não detectadas e melhorando a eficiência dos processos de monitoramento e manutenção. O F1-score é uma média harmônica de precisão e *recall*, oferece uma métrica única que captura o desempenho geral do algoritmo (BLAGEC et al., 2020). Conforme apresentado na tabela 11.

Tabela 11 – Métricas de desempenho dos modelos.

Métrica	Precision	Recall	F1-score
False	1.00	1.00	1.00
True	0.78	0.68	0.72
Accuracy			1.00

Fonte: Elaborado pelo autor.

Analizando as métricas de desempenho do modelo, pode-se detalhar o desempenho do modelo com base nas principais métricas: precisão, sensibilidade, F1-score e suporte:

- A precisão para a classe não falha foi 1.00, o que significa que o modelo não cometeu praticamente nenhum erro ao classificar os dados normais, sem falha. Em outras palavras, praticamente todas as amostras que o modelo classificou como não falha estavam corretas.
- O *recall* para a classe não falha também foi 1.00, o que significa que o modelo identificou corretamente todos os exemplos normais.
- O F1-score para a classe não falha foi 1.00, que é uma média harmônica entre precisão e *recall*. Isso reflete o excelente desempenho do modelo para essa classe.
- A precisão para a classe falha foi 0.78, o que significa que, das falhas detectadas, 78% estavam corretas.
- O *recall* para a classe falha foi 0.68, o que significa que o modelo detectou 68% das falhas reais. Isso reflete a dificuldade do modelo em identificar todas as falhas.
- O F1-score para a classe falha foi 0.72, o que reflete o equilíbrio entre precisão e *recall*.

Em resumo o desempenho do modelo tem um desempenho excelente na detecção da classe dominante, ou seja, os casos normais não falha. Isso é indicado pela precisão, *recall* e F1-score de 1.00 para essa classe. O *recall* de 0.68 para a classe falha indica que o modelo está falhando um pouco em detectar a maioria das falhas reais. Mesmo com uma precisão alta de 0.78 para as falhas detectadas, ele deixa de identificar algumas falhas.

Como foram utilizados dados artificiais nestes experimentos com poucas falhas para se verificar a sensibilidade do modelo em detectar falhas esporádicas geraram impacto na desproporção de classes. A disparidade no número de exemplos entre as classes falha e não falha está claramente influenciando o desempenho. Como há muito mais exemplos da classe não falha, o modelo aprende a priorizar a detecção dessa classe, o que afeta negativamente a sua capacidade de detectar falhas.

6.5 TESTES COMPARATIVOS ENTRE AUTOENCODER PARA SÉRIES TEMPORAIS UNIVARIADAS E MULTIVARIADAS

Para determinar a abordagem mais eficaz na detecção de falhas em dados binários de séries temporais, foram realizados testes comparativos entre duas versões de AE. A primeira versão utilizou um modelo univariado, no qual todas as variáveis foram combinadas em uma única sequência temporal, utilizando-se a técnica proposta nesta tese, enquanto a segunda versão utilizou um modelo multivariado, preservando as variáveis originais como canais separados. Esses testes permitiram uma análise detalhada das vantagens e limitações de cada abordagem, tanto em termos de precisão na detecção de falhas quanto de complexidade computacional e desempenho.

O objetivo principal dos testes foi avaliar o desempenho de um Autoencoder univariado, que utiliza uma única sequência temporal resultante da combinação de todas as variáveis, em comparação com um autoencoder multivariado, que mantém as variáveis separadas, permitindo a captura de interações entre elas. Os testes focaram em quatro aspectos principais:

- Precisão na detecção de falhas em dados binários.
- Tempo de treinamento de cada modelo.
- Capacidade de generalização para novos conjuntos de dados.
- Complexidade computacional associada ao treinamento e à inferência.

Para ambos os testes, foi utilizada uma mesma base de dados contendo séries temporais binárias extraída do aparato experimental. O mesmo conjunto de dados foi transformado em duas versões:

- Versão univariada: Foi aplicado a técnica de fusão em uma série temporal binária univariada.
- Versão multivariada: As variáveis foram mantidas separadas, com cada série temporal representando um canal distinto.

Os dados utilizados para os testes incluíam tanto versão com períodos de operação normal para realizar os testes, quanto versão com períodos com 80 falhas simuladas, permitindo avaliar a eficácia dos modelos na detecção de anomalias. Na primeira versão do teste, foi criado um autoencoder univariado, treinado com uma

única sequência de dados, resultante da combinação das variáveis originais. Na segunda versão, foi treinado um autoencoder multivariado para analisar as variáveis separadamente. O treinamento foi realizado com os parâmetros descritos na tabela 12:

Tabela 12 – Parâmetros utilizados nos modelos.

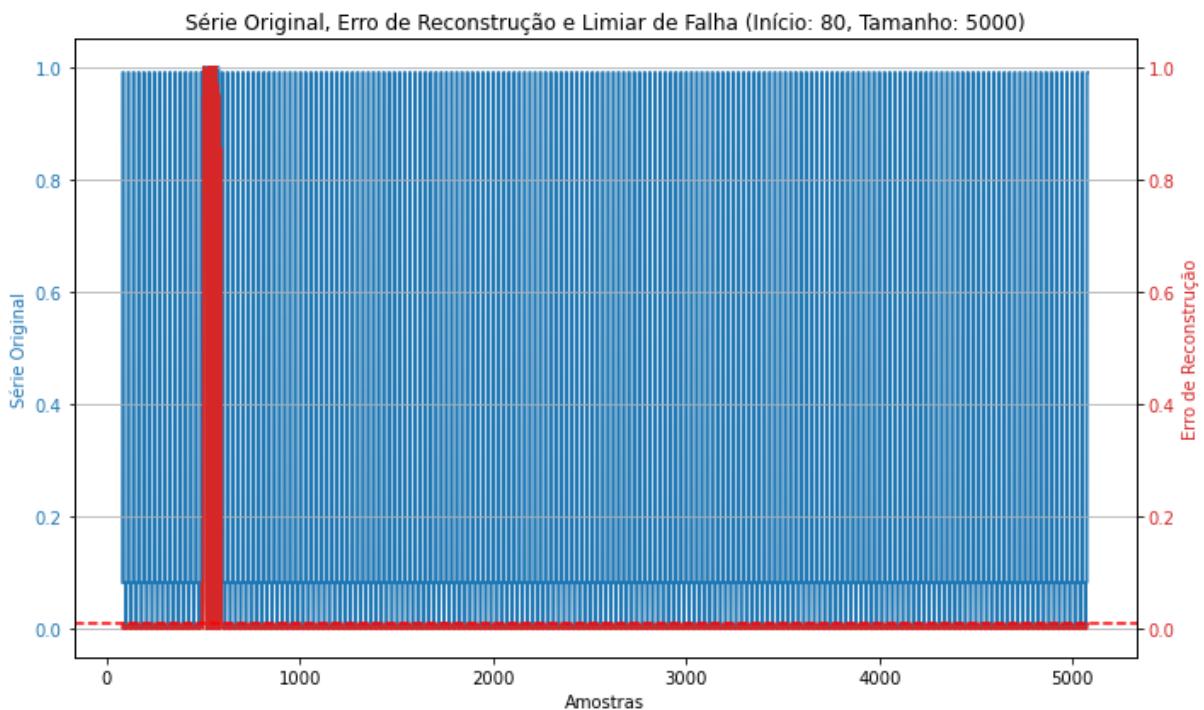
Aspecto	Autoencoder Univariado	Autoencoder Multivariado
Estrutura do Encoder	LSTM, sem sequência	LSTM, sem sequência
Encoder	128	128
Regularização (Dropout)	0.2 (encoder e decoder)	Não aplicado
Estrutura do Decoder	LSTM, com sequência	LSTM, com sequência
Decoder	128	128
Camada de Saída	TimeDistributed(Dense)	TimeDistributed(Dense)
Função de Ativação	Relu, Sigmoid na saída	Relu
Optimizer	Adam	Adam
Função de Perda	Binary Crossentropy	MSE
Número de Épocas	150	150
Tempo por Época (Aprox.)	1s	90s
Dimensiona. dos Dados	Uma variável	Várias variáveis

Fonte: Elaborado pelo autor.

Para comparar as duas abordagens, foi medida a taxa de detecção de falhas, que indica o percentual de falhas corretamente identificadas, além de falsos positivos, que ocorrem quando o modelo detecta uma falha inexistente. Avaliou-se o tempo de treinamento e inferência, que considera o tempo total necessário para treinar cada modelo e realizar as previsões nos dados de teste. Por fim, a adequação para técnicas de binarização e aplicação em sistemas embarcados e tempo real.

A abordagem univariada, com *recall* de 0.68 e precisão de 0.78, ofereceu um equilíbrio entre detecção de falhas e controle de falsos positivos. O tempo de execução é muito curto, ideal para sistemas embarcados e aplicações de tempo real, pois requer menor poder computacional. A estrutura univariada permite facilidade de binarização e otimização, tornando-a ideal para aplicações de baixo consumo e sistemas com limitação de recursos. A Figura 27 apresenta a plotagem da série temporal univariada e a detecção das falhas entre os registros 500 a 580 através do erro de reconstrução.

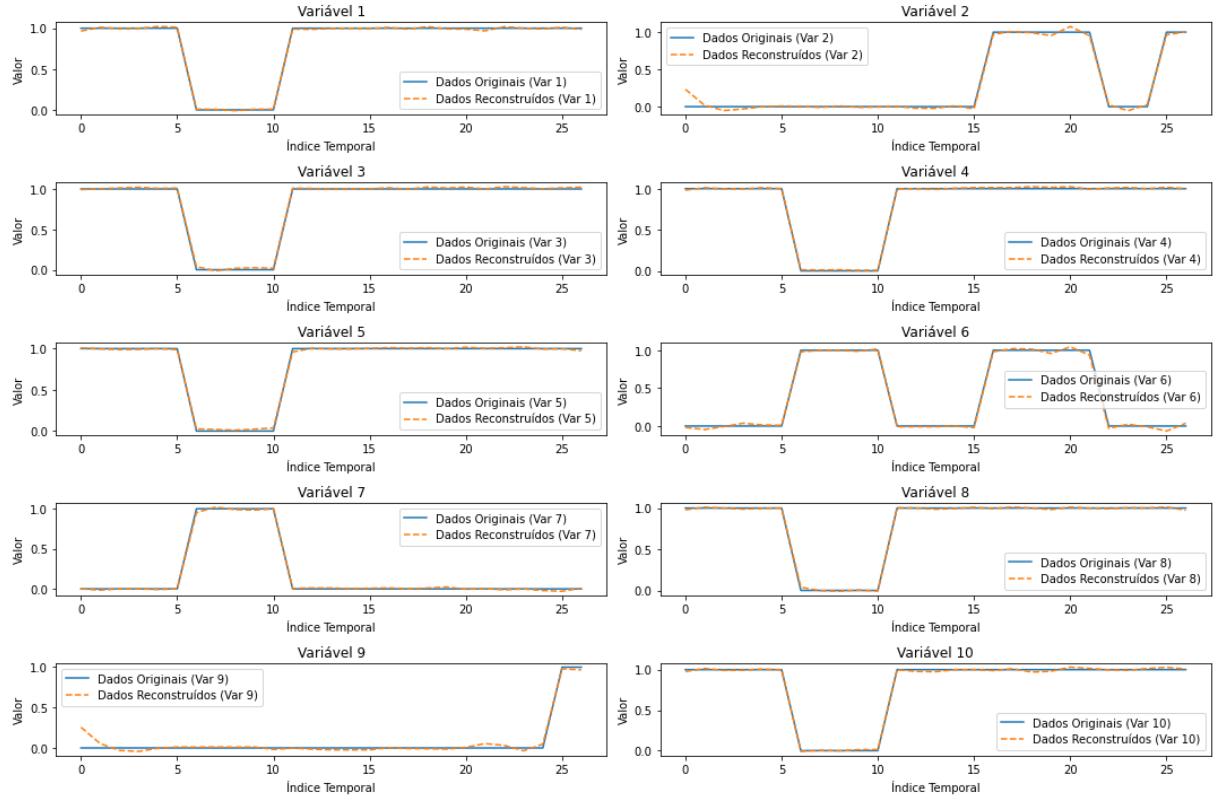
Figura 27 – Deteção de falha no modelo univariado.



Fonte: Elaborado pelo autor.

A abordagem multivariada apresentou excelente desempenho em detecção de falhas, com *recall* de 0.99 e um F1-Score elevado 0.85. No entanto, o tempo de execução é significativamente mais longo do que o modelo univariado, tornando-o menos adequado para aplicações em tempo real ou sistemas embarcados. A Figura 28 apresenta a plotagem de uma janela de dados de cada variável no modelo multivariado. Sua estrutura complexa e demanda computacional tornam a binarização e otimização mais desafiadoras, dificultando sua aplicação em dispositivos com restrições de recursos, como Raspberry Pi.

Figura 28 – Plotagem de uma janela de dados de cada variável do modelo multivariado.



Fonte: Elaborado pelo autor.

Na questão de detecção de falhas, os testes demonstraram que o modelo multivariado apresenta uma vantagem em *recall* 0.99, capturando quase todas as falhas, o que poderia ser uma vantagem em aplicações críticas onde a perda de falhas pode ter consequências sérias. O modelo univariado, embora tenha apresentado um menor *recall* 0.68, ainda oferece um bom desempenho em detecção com uma taxa mais baixa de falsos positivos, evitando alarmes desnecessários, que acarretaria horas de máquina parada para desmontagem e confirmação da falha.

Quanto ao tempo de execução, o modelo univariado se destaca pelo tempo de execução rápido (1 segundo), adequado para aplicações em tempo real e sistemas embarcados que necessitam de respostas imediatas (FANG *et al.*, 2017). O modelo multivariado, com tempo de execução de 11 segundos, é mais indicado para sistemas que podem tolerar maior latência e dispõem de maior poder computacional. Os resultados foram obtidos em um computador com a seguinte configuração: processador Intel® Core™ i5-13450HX de 10 núcleos, com cache de 20MB e

frequência máxima de até 4.6GHz, executando o sistema operacional Windows 11 Pro. A máquina também conta com uma GPU NVIDIA® GeForce® RTX™ 3050, com 6GB de memória GDDR6, e 16GB de memória RAM DDR5 (2x8GB) com frequência de 4800MT/s.

A Tabela 13 apresenta uma comparação detalhada entre as abordagens univariada e multivariada para detecção de falhas, ambas utilizando o mesmo limiar de falha de 0,01. A análise inclui acurácia, precisão, *recall*, F1-Score e falsos positivos, além de considerar o tempo de execução e a adequação para sistemas embarcados. Cada abordagem é avaliada quanto à sua capacidade de atender a diferentes requisitos de implementação, como aplicações em tempo real e binarização, evidenciando as forças e limitações de cada uma.

Tabela 13 – Comparação das Métricas e Adequação.

Métrica	Univariado	Multivariado	Observações
Tempo de Execução	1s (12ms/step)	11s (6ms/step)	O modelo univariado é muito mais rápido, ideal para aplicações em tempo real e embarcadas.
Acurácia	1.00	0.9995	Ambas as abordagens apresentam alta acurácia, mas o univariado é ligeiramente superior.
Precisão para Falhas	0.78	0.75	O modelo univariado tem leve vantagem em precisão, com menos falsos positivos.
<i>Recall</i> para Falhas	0.68	0.99	O multivariado é superior em <i>recall</i> , identificando praticamente todas as falhas.
F1-Score para Falhas	0.72	0.85	O multivariado apresenta melhor F1-Score, refletindo o equilíbrio entre <i>recall</i> e precisão.
Falsos Positivos	15	27	A abordagem univariada possui menos falsos positivos, o que pode reduzir falsos alarmes.

Fonte: Elaborado pelo autor.

No quesito aplicação, a abordagem univariada é mais vantajosa para aplicações embarcadas, pois é mais fácil de otimizar e binarizar, ocupando menos memória e processamento. A abordagem multivariada pode ser um desafio para sistemas embarcados devido à complexidade do modelo e maior tempo de execução, o que dificulta a implementação em tempo real (LAZAROU, 2020).

Conforme apresentado na Tabela 12, o tempo de treinamento das abordagens univariada e multivariada diferiu consideravelmente também. A abordagem univariada completou 150 épocas com uma média de 1 segundo por época, totalizando

aproximadamente 2,5 minutos. Em contraste, a abordagem multivariada que também foi treinada por 150 épocas, obteve uma média de 90 segundos por época, resultando em um tempo total estimado de cerca de 225 minutos, ou aproximadamente 3 horas e 45 minutos. Esses tempos refletem a diferença de complexidade entre os modelos, com o multivariado exigindo um treinamento mais longo devido ao maior número de variáveis processadas simultaneamente.

Foram realizados 5 experimentos adicionais para explorar a capacidade de generalização das abordagens univariada e multivariada ao serem aplicadas a novos conjuntos de dados. Observou-se que o modelo univariado apresentou uma ligeira vantagem em generalização, especialmente em métricas de acurácia e precisão. Em comparação, o modelo multivariado mostrou-se menos consistente ao longo dos experimentos, mantendo uma precisão e *recall* levemente inferiores, o que indica uma sensibilidade maior a variações nos dados de entrada. A abordagem univariada, além de menor complexidade, demonstrou um maior potencial de estabilidade em diferentes cenários de teste, tornando-a uma escolha favorável para ambientes onde a consistência e o desempenho previsível são importantes. A tabela 14 apresenta o resultado dos experimentos.

Tabela 14 – Capacidade de generalização das abordagens.

Exp.	Acurácia Uni.	Acurácia Multi.	Precisão Uni.	Precisão Multi.	Recall Uni.	Recall Multi.	F1-Score Uni.	F1-Score Multi.
Exp. 1	0,9997	0,9992	0,79	0,76	0,68	0,94	0,73	0,84
Exp. 2	0,9998	0,9991	0,8	0,75	0,67	0,92	0,73	0,83
Exp. 3	0,9996	0,9993	0,81	0,77	0,69	0,95	0,75	0,85
Exp. 4	0,9995	0,9991	0,8	0,76	0,68	0,96	0,74	0,85
Exp. 5	0,9999	0,9992	0,79	0,75	0,67	0,99	0,73	0,85

Legenda: Exp. = Experimento; Uni. = Univariado; Mult.= Multivariado.

Fonte: Elaborado pelo autor.

Desta forma, conclui-se que, com os experimentos realizados testando os modelos, que a abordagem univariada é melhor para sistemas embarcados, aplicações de tempo real, e casos em que é importante minimizar falsos positivos e simplificar a implementação. É mais fácil de binarizar e otimizar para dispositivos de baixo consumo. Em contrapartida, a abordagem multivariada é mais adequada para sistemas onde a detecção máxima de falhas é uma prioridade e onde o maior tempo de processamento é aceitável. Sua alta capacidade de detecção *recall* de 0.99 a torna

ideal para ambientes com maior poder computacional e necessidade de máxima cobertura de falhas.

Sendo assim, a escolha depende, portanto, das necessidades específicas de cada aplicação. Assim, para alta detecção de falhas, o multivariado é ideal, para simplicidade e eficiência em sistemas embarcados, o univariado é preferível. Como a solução proposta nesta tese foi idealizada para ser utilizada em cenários de manutenção de máquinas industriais no próprio local de operação e está vinculada ao uso da interface de coleta de dados desenvolvida, a abordagem univariada foi tida como a mais vantajosa.

Esta escolha visa avançar para a área de binarização e otimização de modelos em trabalhos futuros para reduzir ainda mais o consumo de recursos e melhorar o desempenho em dispositivos de baixo consumo.

A binarização de modelos para tecnologia embarcada em pequenos dispositivos, é um fator importante a ser considerado ao comparar a análise de falhas usando séries temporais univariadas e multivariadas. A binarização refere-se a técnicas que tornam os modelos mais leves e eficientes para rodarem em dispositivos com recursos limitados, como memória, capacidade de processamento e energia (GRØNBECH *et al.*, 2020).

Ao rodar um Autoencoder em dispositivos de baixo consumo é importante pensar na complexidade do modelo, eficiência computacional e uso de memória (KUMAR, 2022). Comparando as abordagens univariadas e multivariadas, a binarização do modelo traz algumas implicações específicas.

Um modelo univariado é naturalmente mais simples, com uma arquitetura menos complexa, o que facilita a binarização. Isso significa que o modelo pode ser facilmente reduzido em tamanho, tornando-se adequado para dispositivos com recursos limitados (COWAN *et al.*, 2020). Como o AE univariado processa apenas uma variável por vez, o consumo de memória e poder de processamento é significativamente menor.

6.6 EXPERIMENTOS COM ABORDAGEM ESTATÍSTICA

O modelo ARIMA é uma das técnicas estatísticas mais utilizadas para análise de séries temporais, especialmente para previsões em dados com padrão temporal (HUNG *et al.*, 2008). A montagem do modelo ARIMA para detecção de falhas foi realizada ajustando os parâmetros (p), (d) e (q) de forma a capturar o comportamento normal dos dados, no intuito de que desvios significativos indicassem falhas. Para definir esses parâmetros, foram utilizados métodos como o gráfico Autocorrelation Function (ACF) e Partial Autocorrelation Function (PACF), que indicaram as ordens apropriadas de autorregressão e média móvel. Em seguida, vários ajustes finos foram realizados para tentar otimizar a precisão do modelo. A Figura 29 apresenta um trecho do algoritmo para detecção de falhas.

Figura 29 – Trecho do algoritmo utilizado para detecção de falha com o modelo ARIMA.

```
# Detectar falhas usando o modelo salvo
else:
    def detectar_falhas_arima(matriz_janelas, limiar=0.01):
        anomalias = []
        erros = []

        for i, janela in enumerate(matriz_janelas):

            modelo_path = os.path.join(save_dir, f'arima_model_janela_{i}.joblib')

            if os.path.exists(modelo_path):
                modelo = joblib.load(modelo_path)
                previsoes = modelo.predict(start=0, end=len(janela)-1)

                # Calcula o erro MSE
                mse = np.mean((janela - previsoes) ** 2)
                erros.append(mse)

                # Identificar as falhas
                if mse > limiar:
                    anomalias.append(True)
                else:
                    anomalias.append(False)
            else:
                print(f"Modelo para janela {i} não encontrado. Treine o modelo primeiro.")
                anomalias.append(None)

        return erros, anomalias

# Executar a detecção de falhas
erros, anomalias = detectar_falhas_arima(matriz_janelas)

# Exibe os resultados
print("Erros:", erros)
print("Anomalias:", anomalias)
```

Fonte: Elaborado pelo autor.

Durante o processo de desenvolvimento e ajuste do modelo ARIMA, realizou-se uma série de testes para avaliar sua capacidade de identificar falhas na série

temporal. Primeiramente foram testados diferentes limiares de erro para classificar uma observação como falha. Esses limiares variaram de 0,01 até 0,2, na tentativa de identificar um valor que conseguisse distinguir com precisão entre dados normais e falhas.

Para tentar melhorar o aprendizado do modelo e aumentar sua capacidade de identificar falhas, foram realizados ajustes no conjunto de dados aumentando-se a proporção de dados de falha durante o treinamento do modelo. Dessa forma, o modelo teria mais contato com padrões de falhas e, teoricamente, uma capacidade aprimorada de reconhecer esses desvios. A hipótese era que a exposição a mais exemplos de falhas ajudaria o ARIMA a identificar anomalias com maior precisão (HUNG *et al.*, 2008).

Mesmo assim, apesar das diversas tentativas para ajustar o modelo ARIMA e dos experimentos com diferentes limiares e proporções de dados de falhas, o desempenho do modelo na detecção de falhas foi insatisfatório. Independentemente do limiar utilizado, o modelo mostrou-se incapaz de identificar corretamente as anomalias. A precisão, a revocação e a pontuação F1 ficaram em valores muito baixos, indicando que o modelo classifica a maioria dos pontos provavelmente como normais, falhando em detectar as falhas presentes na série temporal.

Como provável causa para o baixo desempenho, acredita-se ser a natureza dos dados. Visto o modelo ARIMA ser adequado para séries temporais lineares e estacionárias, e os dados de falhas apresentarem padrões complexos que talvez não possam ser capturados apenas com autorregressão e médias móveis. Também se notou que mesmo com limiares mais baixos, o ARIMA mostrou-se incapaz de identificar pequenas variações como falhas. Isso pode ocorrer porque o modelo é projetado para previsões suavizadas e tende a ignorar variações mínimas (HUNG *et al.*, 2008).

Observa-se que o modelo ARIMA acompanha parcialmente o comportamento da série original, especialmente nos momentos iniciais, onde ocorre um rápido aumento nos valores. No entanto, o modelo apresenta dificuldades em seguir mudanças bruscas e acentuadas na série, como a queda significativa logo após o sexto ponto da amostra. Em vez de capturar essa variação repentina, o ARIMA gera previsões suavizadas, não respondendo de forma precisa à variação dos dados reais.

Além disso, como apresentado na Figura, após a queda no valor, o modelo continua a prever valores mais altos, demonstrando uma resposta lenta ao novo comportamento da série.

Apesar das tentativas com diferentes ajustes, o modelo ARIMA mostrou-se inadequado para a tarefa de detecção de falhas nestes conjuntos de dados. Esse resultado sugere que modelos mais robustos, como autoencoder pode ser mais eficaz para detectar padrões de falhas complexos e não-lineares.

Foram realizados também experimentos utilizando Modelos de Markov Ocultos (HMM) como uma abordagem estatísticas alternativa para detecção de falhas na série temporal binária univariada. O objetivo foi explorar as capacidades do HMM em capturar as transições de estados e as possíveis falhas baseando-se na sequência observada de valores binários na série. A Figura 30 apresenta um trecho do algoritmo para detecção de falhas.

Figura 30 – Trecho do algoritmo utilizado para detecção de falha com o modelo HMM.

```
# Detectar falhas usando o modelo salvo
else:
    def detectar_falhas_hmm(matriz_janelas, limiar=0.01):
        anomalias = []
        erros = []

        for i, janela in enumerate(matriz_janelas):

            modelo_path = os.path.join(save_dir, f'hmm_model_janela_{i}.joblib')

            if os.path.exists(modelo_path):
                modelo = joblib.load(modelo_path)
                janela_reshape = janela.reshape(-1, 1) # HMM requer entrada 2D
                log_prob = modelo.score(janela_reshape)

                # Considerar o log-probabilidade como métrica de erro
                erros.append(log_prob)

                # Identificar as falhas com base no limiar
                if log_prob < limiar: # log_prob negativo indica menor probabilidade
                    anomalias.append(True)
                else:
                    anomalias.append(False)
            else:
                print(f"Modelo para janela {i} não encontrado. Treine o modelo primeiro.")
                anomalias.append(None)

        return erros, anomalias

# Executar a detecção de falhas
erros, anomalias = detectar_falhas_hmm(matriz_janelas)

# Exibe os resultados
print("Erros (log-probabilidades):", erros)
print("Anomalias:", anomalias)
```

Fonte: Elaborado pelo autor.

Para os experimentos, o HMM foi configurado com diferentes parâmetros de estados ocultos, variando entre estados para capturar padrões normais e de falha. Vários limiares de detecção foram testados, abrangendo valores entre 0,01 e 0,2, buscando identificar um equilíbrio entre detecção e minimização de falsos positivos. Além disso, os dados foram reorganizados para que o modelo tivesse maior exposição a segmentos com falhas, na tentativa de melhorar a sensibilidade do HMM para detectar anomalias e a abordagem foi configurada para capturar anomalias com base na log-verossimilhança das observações, onde valores anômalos indicariam possíveis falhas.

Os resultados revelaram que o HMM não obteve um desempenho satisfatório. As log-probabilidades observadas mostraram uma variação limitada, e o modelo identificou poucas falhas, resultando em uma precisão e revocação de 0,0, bem como um F1-Score também de 0,0. Esses valores indicam que o modelo teve dificuldades em identificar padrões de falha presentes nos dados.

Desconfia-se que a natureza binária dos dados pode ter limitado a capacidade do HMM de distinguir entre padrões normais e de falha. Além disso, o modelo parece ter ajustado bem os padrões de normalidade, mas com baixa sensibilidade para captar mudanças abruptas associadas a falhas. Por fim, mesmo ao alterar os dados para expor o modelo a mais casos de falha, os resultados mantiveram-se abaixo do esperado, indicando que o HMM pode não ser a abordagem ideal para este tipo de série temporal binária (LEE *et al.*, 2023).

Observa-se que o HMM segue uma estrutura bastante simplificada, mantendo os estados em valores constantes para a maior parte da janela. O comportamento apresentado pode indicar que o HMM teve dificuldade em capturar a transição dos estados de forma precisa ao longo da janela. Em vez de responder às variações na série original, o modelo parece ter achatado os estados, possivelmente devido à natureza binária dos dados e à falta de complexidade nos padrões que o modelo conseguiu identificar (LEE *et al.*, 2023).

Esse resultado reforça a conclusão dos experimentos com o HMM, que pode não ser sensível o suficiente para capturar as mudanças que indicariam falhas na série temporal, especialmente se elas ocorrerem de maneira súbita ou em intervalos específicos (LEE *et al.*, 2023). Os experimentos realizados neste capítulo destacam

os desafios em se aplicar modelos estatísticos em séries temporais binárias coletadas das máquinas ferramentas para detecção de falhas.

6.7 EXPERIMENTOS COMPARATIVOS ENTRE AE, CNN E RNN NA DETEÇÃO DE FALHAS

Este capítulo apresenta os experimentos realizados para avaliar a eficácia do Autoencoder LSTM utilizado na detecção de falhas em séries temporais binárias. Foram conduzidos testes com diferentes arquiteturas de redes neurais, incluindo Autoencoder LSTM, CNN + LSTM e RNN simples. Os modelos foram treinados com um conjunto de dados contendo falhas artificiais inseridas em pontos estratégicos, permitindo uma análise detalhada de sua capacidade de generalização e identificação de padrões anômalos. Durante os experimentos, foram ajustados hiperparâmetros essenciais, como taxa de aprendizado, número de neurônios por camada e técnicas de regularização, para garantir um desempenho otimizado. A partir dos resultados obtidos, são apresentadas as vantagens e limitações de cada abordagem, bem como as métricas utilizadas para a comparação, destacando as características que tornam cada modelo mais ou menos adequado para aplicações em detecção de falhas.

6.7.1 Metodologia dos experimentos

Os experimentos foram conduzidos utilizando uma série temporal binária extraída através da interface de coleta de dados. O conjunto de dados compreende 54.000 registros, sendo que foram inseridas falhas artificiais entre os registros 500 e 590 para permitir a avaliação dos modelos.

Neste estudo, foram comparadas três arquiteturas distintas para a detecção de falhas, cada uma com abordagens e características específicas. O Autoencoder LSTM foi utilizado para reconstrução da série temporal, aproveitando a capacidade das redes LSTM de capturar dependências temporais de longo prazo. A abordagem CNN + LSTM combinou a extração de padrões locais através de convoluções com a modelagem sequencial da LSTM, buscando um equilíbrio entre identificação de padrões locais e aprendizado de sequências temporais. Por fim, o modelo RNN simples foi testado para verificar a eficiência de uma rede recorrente convencional na detecção de falhas, utilizando a estrutura de aprendizado sequencial sem a

complexidade adicional das LSTMs. Esses modelos foram treinados e avaliados de forma comparativa para determinar qual técnica apresentava o melhor desempenho na tarefa de identificação de anomalias nas séries temporais binárias.

Os modelos foram treinados com parâmetros específicos para garantir um desempenho eficiente na detecção de falhas. Cada arquitetura foi ajustada com diferentes configurações de camadas, taxas de aprendizado e técnicas de regularização, visando melhorar a capacidade de generalização e minimizar *overfitting*. Além disso, foram utilizadas diferentes funções de perda apropriadas para cada abordagem, permitindo a comparação justa entre os modelos, sendo eles:

- **Autoencoder LSTM:**

Camadas LSTM: 128 neurônios

Dropout: 0.2

Função de perda: Binary Crossentropy

Otimizador: Adam (*learning rate* = 0.0001)

Épocas: 150

Batch size: 32

- **CNN + LSTM:**

Camadas Conv1D: 64 filtros, *kernel size* = 3

Camadas LSTM: 64 neurônios

Dropout: 0.2

Função de perda: Binary Crossentropy

Otimizador: Adam (*learning rate* = 0.0001)

Épocas: 150

Batch size: 32

- **RNN Simples:**

Camadas SimpleRNN: 128 e 64 neurônios

Dropout: 0.2

Função de perda: Mean Squared Error

*Otimizador: Adam (*learning rate* = 0.0001)*

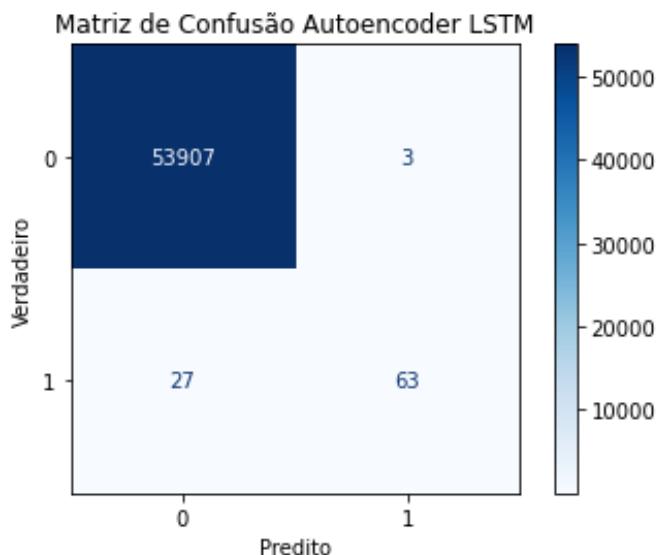
Épocas: 150

Batch size: 32

6.7.2 Resultados dos experimentos

A análise da matriz de confusão obtida para o modelo Autoencoder LSTM revelou sua eficácia na reconstrução da série temporal e detecção de falhas, apresentando um desempenho superior em relação a outras abordagens. Esse modelo demonstrou alta precisão na identificação de padrões normais e anômalos, minimizando a ocorrência de falsos positivos e falsos negativos. A estrutura baseada em LSTM permitiu capturar as dependências temporais de longo prazo, contribuindo para uma reconstrução mais fiel dos dados. Essa capacidade de aprendizado temporal faz do Autoencoder LSTM uma solução robusta para aplicações de monitoramento preditivo e manutenção preventiva. A Figura 31 apresenta a matriz de confusão obtida para o modelo Autoencoder LSTM.

Figura 31 – Matriz de confusão obtida para o modelo Autoencoder LSTM.

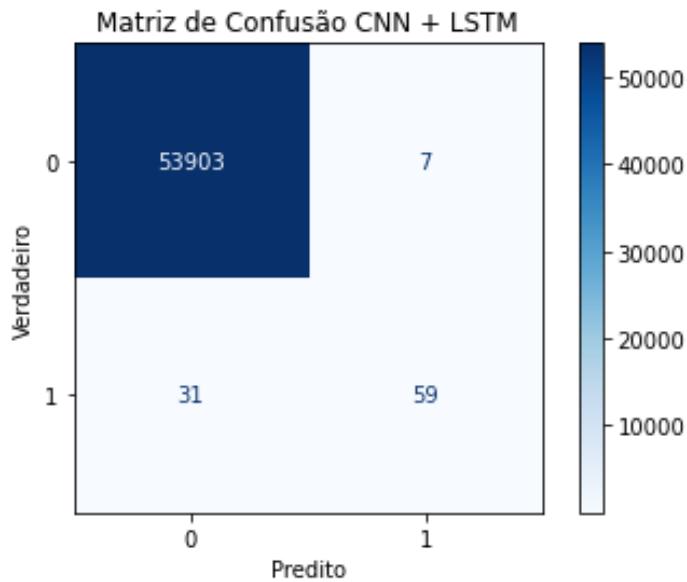


Fonte: Elaborado pelo autor.

O relatório de classificação apresentou uma visão detalhada do desempenho do modelo Autoencoder LSTM na detecção de falhas. Com uma precisão de 95%, o modelo demonstrou uma capacidade notável de identificar corretamente os eventos de falha, minimizando falsos positivos. O recall de 70% indica que o modelo conseguiu recuperar a maioria das falhas presentes na série temporal, garantindo uma cobertura ampla na detecção de falhas. O F1-score de 81% reflete um equilíbrio adequado entre precisão e *recall*, tornando essa abordagem eficaz para aplicações de monitoramento preditivo. Esses resultados reforçam a robustez do Autoencoder LSTM na identificação de padrões temporais e na prevenção de falhas em sistemas industriais.

A Figura 32 apresenta a matriz de confusão obtida para o modelo CNN + LSTM, ilustrando o desempenho do modelo na classificação de falhas. Essa matriz permite uma visualização clara da quantidade de previsões corretas e incorretas, evidenciando o número de falsos positivos e falsos negativos.

Figura 32 – Matriz de confusão obtida para o modelo CNN + LSTM.

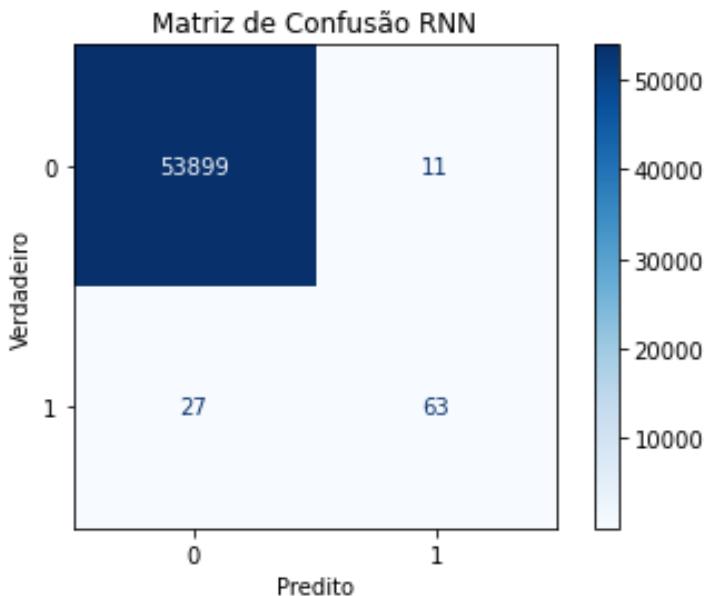


Fonte: Elaborado pelo autor.

O modelo obteve uma precisão de 89%, demonstrando sua capacidade de prever corretamente as falhas quando estas ocorrem. Entretanto, com um *recall* de 66%, ele deixou de identificar algumas falhas reais, o que impacta sua eficácia. O F1-score de 76% reflete um equilíbrio entre precisão e *recall*.

A Figura 33 apresenta a matriz de confusão obtida para o modelo RNN, destacando sua capacidade de classificação de falhas. Essa matriz auxilia na compreensão do desempenho do modelo em relação à identificação de falhas.

Figura 33 – Matriz de confusão obtida para o modelo RNN.



Fonte: Elaborado pelo autor.

O modelo obteve uma precisão de 85%, demonstrando uma boa capacidade de prever corretamente as falhas, mas ainda apresentando alguns falsos positivos. Com um *recall* de 70%, ele conseguiu recuperar a maioria das falhas reais, garantindo uma boa cobertura na identificação de anomalias. O F1-score de 77% reflete um equilíbrio entre precisão e *recall*.

6.7.3 Conclusões dos experimentos

No contexto da manutenção preditiva, a detecção de falsos positivos tem um impacto significativo na eficiência operacional. Uma menor taxa de falsos positivos significa que menos componentes da máquina precisarão ser desmontados para verificação, reduzindo o tempo de inatividade e os custos associados à manutenção desnecessária. Isso é um fator importante, pois evita interrupções não planejadas e otimiza a utilização de recursos. Em contrapartida, embora falsos negativos possam resultar na não detecção de uma falha real, as estratégias de monitoramento contínuo

podem mitigar esse risco ao longo do tempo. Portanto, minimizar falsos positivos pode ser um critério mais relevante do que a simples redução de falsos negativos em aplicações industriais de manutenção.

A análise dos resultados demonstra que o Autoencoder LSTM apresentou o melhor desempenho, especialmente em termos de precisão (95%) e F1-score (81%). Esse modelo conseguiu manter uma baixa taxa de falsos positivos, garantindo uma boa detecção de falhas sem gerar muitos alarmes falsos, conforme apresentado na Tabela 15.

Tabela 15 – Comparativo entre Autoencoder LSTM, CNN + LSTM e RNN.

Modelo	Precisão (Falha)	Recall (Falha)	F1-Score (Falha)	Falsos Positivos	Falsos Negativos
Autoencoder LSTM	95%	70%	81%	3	27
CNN + LSTM	89%	66%	76%	7	31
RNN	85%	70%	77%	11	27

Fonte: Elaborado pelo autor.

O RNN simples apresentou desempenho similar ao Autoencoder LSTM no *recall* (70%), mas com uma precisão menor (85%), indicando mais falsos positivos. Isso pode ser resultado da incapacidade do modelo de capturar padrões temporais tão bem quanto a abordagem com LSTM. O CNN + LSTM teve o pior *recall* (66%), sugerindo que a abordagem convolucional teve dificuldades em detectar falhas menores na série.

Os experimentos demonstram que o Autoencoder LSTM é a abordagem mais eficaz para detecção de falhas, proporcionando um melhor balanço entre precisão e *recall*. A abordagem RNN simples também é viável, mas pode gerar mais falsos positivos. Por outro lado, a combinação CNN + LSTM teve dificuldades na detecção de falhas, em comparação com os outros modelos.

7. DISCUSSÕES

Neste capítulo, será realizada uma análise comparativa entre o estudo recente publicado por Princz; Shaloo; Erol (2024), que aborda a detecção de falhas em séries temporais binárias coletadas em equipamentos, e as técnicas desenvolvidas nesta tese. Também será analisado o artigo de Pereira et al. (2023), que, embora não trate de séries temporais binárias, utiliza uma abordagem de análise individual de dados de sensores. O objetivo é avaliar as abordagens, metodologias e conclusões desses trabalhos para identificar semelhanças, diferenças e contribuições únicas.

Além disso, serão apresentados os principais autores da área, suas contribuições e os limites alcançados em seus estudos, contextualizando como esta tese avança nesse campo. Por fim, será discutida a diferença entre a análise direta de séries temporais multivariadas capturadas pela interface de coleta e a técnica de transformação proposta nesta pesquisa.

7.1 ANÁLISE COMPARATIVA COM ESTUDOS RECENTES

Enquanto o método de Princz; Shaloo; Erol, 2024 descreve um processo detalhado de preparação de dados para transformar informações coletadas de sensores em uma série temporal binária e calcular as medias de tempo de ativação dos sensores para identificar as falhas a partir destas informações, esta tese se destaca por fornecer uma abordagem alternativa através da criação de uma interface para coleta dados diretamente de todos os sensores simultaneamente, conforme apresentado no Capítulo 4.2. Esta abordagem oferece várias vantagens significativas sobre o processo descrito por Princz; Shaloo; Erol, 2024.

Primeiramente, a coleta simultânea dos dados por meio de uma interface integrada elimina a necessidade de etapas complexas de pré-processamento e limpeza, como a filtragem de colunas, a eliminação de entradas duplicadas e a transposição de dados. Ao capturar os dados em tempo real, a interface assegura que todas as informações dos sensores são registradas de maneira precisa e completa desde o início, reduzindo a probabilidade de perda de dados e minimizando a necessidade de ajustes posteriores (BARCELÓ-ORDINAS et al., 2018).

Além disso, a coleta simultânea permite que os dados sejam armazenados em seu formato original, íntegro e em seus respectivos *timestamps*, evitando possíveis

distorções que podem ocorrer durante as transformações (CUZZOCREA, 2020). Esta abordagem não só melhora a precisão dos dados, mas também simplifica o fluxo de trabalho, permitindo uma análise mais direta e eficiente.

A criação da interface para coleta de dados em tempo real também pode aumentar a velocidade e a eficiência da detecção de anomalias. Com dados disponíveis de forma imediata e contínua, é possível implementar algoritmos de detecção de falhas que operam em tempo real, fornecendo alertas mais rápidos e precisos sobre qualquer irregularidade no sistema. Isso é especialmente importante em ambientes industriais, onde a rápida identificação e resposta a falhas podem significar a diferença entre evitar ou enfrentar graves problemas operacionais.

Por fim, a interface de coleta reduz a complexidade e o custo associado ao processamento de dados. Sem a necessidade de realizar diversas etapas de preparação e limpeza, os recursos podem ser alocados de maneira mais eficiente para outras áreas, como a melhoria contínua do sistema e a análise avançada dos dados.

Portanto, ao adotar uma abordagem que permite a coleta simultânea dos dados de todos os sensores, é possível não só evitar as etapas adicionais e propensas a erros do pré-processamento, mas também melhorar a precisão dos dados e a eficiência na detecção de falhas, proporcionando uma solução mais robusta e eficaz para a análise e detecção de falhas em máquinas ferramentas.

Embora Princz, Shaloo e Erol (2024) descreva um método abrangente e detalhado para a preparação e análise de dados de sensores para a detecção de falhas, há argumentos sólidos para a superioridade do método proposto neste estudo para análise direta de séries temporais binárias para esse objetivo. A abordagem descrita, que envolve a normalização das durações dos sensores e a divisão dos dados em conjuntos de treinamento e teste, pode introduzir complexidades e potenciais fontes de imprecisão que podem ser evitadas por meio da análise direta das séries temporais binárias (DING et al., 2020; OH; HAN; JEONG, 2020)

Primeiramente, a análise direta das séries temporais binárias preserva a integridade e a granularidade dos dados originais, evitando a perda de informações que pode ocorrer durante o pré-processamento e a normalização. Ao trabalhar com dados binários brutos, é possível capturar nuances e padrões específicos que podem

ser ocultados ou distorcidos pela normalização e transformação dos dados. Esses detalhes podem ser importantes para identificar anomalias e falhas com maior precisão (WANG, X.; WANG, C. 2020; OH; HAN; JEONG, 2020).

Além disso, a divisão dos dados em conjuntos de treinamento e teste, como descrito no método mencionado, pode criar um ambiente artificial onde a capacidade de generalização dos modelos é comprometida. Em contraste, a análise direta das séries temporais binárias permite uma abordagem contínua e holística, onde todos os dados são considerados simultaneamente. Isso pode melhorar a detecção de falhas ao fornecer uma visão completa e ininterrupta do comportamento dos sensores ao longo do tempo (BAEK et al., 2021; DING et al., 2020).

Além disso, enquanto a automatização da *pipeline* de aprendizado de máquina pode ser eficiente, a análise direta das séries temporais binárias pode fornecer insights mais granulares e específicos sobre o comportamento dos sensores. Isso é especialmente relevante para a detecção de falhas, onde a capacidade de identificar padrões sutis e evolutivos ao longo do tempo é importante. A abordagem direta pode, portanto, oferecer uma vantagem significativa na identificação precoce de falhas e na melhoria da manutenção preditiva (BARBARIOL et al., 2023).

Em suma, embora o método descrito seja abrangente e detalhado, a análise direta das séries temporais binárias oferece uma abordagem que pode preservar a integridade dos dados e proporcionar uma detecção de falhas mais precisa e eficiente, ao evitar os potenciais problemas associados ao pré-processamento e à divisão dos dados.

Ainda que Princz, Shaloo e Erol (2024) mencione a análise de dados de sensores em ambientes industriais, ele não se concentra explicitamente em séries temporais multivariadas, mas sim em séries temporais univariadas, onde cada sensor pode ser considerado como uma série temporal separada. No entanto, a abordagem de detecção de anomalias mencionada pode ser aplicada a dados de múltiplos sensores, segundo os autores, o que poderia ser interpretado como uma análise de séries temporais multivariadas, mas não detalha uma análise explícita de séries temporais multivariadas.

Portanto, o artigo de Princz, Shaloo e Erol (2024) envolve dados de múltiplos sensores, mas a análise em si é centrada em séries temporais binárias de cada sensor individualmente, e não em uma análise formal de séries temporais multivariadas.

Em uma abordagem semelhante, o artigo de Pereira *et al.* (2023) apresenta uma solução inovadora para otimização da manutenção preventiva em sistemas reparáveis, utilizando um modelo que considera a imperfeição das intervenções de manutenção e um fator de redução de idade variável. A metodologia propõe flexibilizar os planos de manutenção, levando em conta o nível de intervenção necessário para cada atividade, determinado pelo número de tarefas realizadas, tempo de execução e itens substituídos, permitindo uma manutenção personalizada para minimizar os custos totais. A aplicação do modelo é demonstrada em trocadores de calor de uma usina hidrelétrica, afetados por incrustações de mexilhões dourados, onde o método mostrou eficácia na redução de falhas e custos de manutenção.

No contexto de uso de autoencoders para detectar falhas em sistemas complexos com múltiplos sensores e canais, a abordagem descrita no artigo de Pereira *et al.* (2023) se assemelha a abordagem de Princz, Shaloo e Erol (2024), onde se cria um autoencoder específico para cada canal, e apresenta uma limitação significativa. Esse método ignora a interdependência potencial entre os diferentes sensores. Algumas falhas só podem ser detectadas com precisão se analisarmos variações simultâneas em múltiplos canais; por exemplo, desvios em três sensores específicos podem indicar uma anomalia que não seria identificada analisando cada canal isoladamente.

A criação de um modelo individual para cada canal significa que os desvios multivariados, aqueles que dependem da correlação entre sensores, podem passar despercebidos. Além disso, como cada modelo autoencoder tem uma precisão independente, a soma dos resultados desses modelos distintos pode gerar incertezas, reduzindo a precisão global da detecção de falhas. Para mitigar esse problema, propõem-se a solução apresentada nesta tese, onde a criação de um modelo único a utilização da técnica de fusão de dados permite a análise integrada dos diferentes sensores, capturando tanto as falhas isoladas quanto as que dependem de múltiplos canais.

Ao contrário de dos estudos de Pereira *et al.* (2023) e Princz, Shaloo e Erol (2024), o estudo de Nishino, Maekawa e Hara (2023), analisam séries temporais multivariadas em contextos específicos, como a detecção de movimentos humanos. Apesar desta tese explora séries temporais binárias multivariadas em um cenário completamente diferente, voltado para a detecção de falhas em máquinas e equipamentos industriais.

Embora o trabalho de Nishino apresente técnicas relevantes, como a criação de Matrizes de Autossimilaridade (SSM), o estudo de Nishino utiliza apenas três variáveis (eixos x, y e z) para capturar dados de aceleração, esta tese analisa conjuntos de dados mais complexos, com um total de 10 variáveis, permitindo uma representação mais abrangente e detalhada dos fenômenos envolvidos, a aplicação proposta nesta tese exige abordagens adaptadas às características das séries temporais binárias e aos desafios inerentes ao ambiente industrial, como a complexidade mecânica e operacional. A seguir, serão discutidas as contribuições e limitações do estudo de Nishino, Maekawa e Hara (2023) em relação aos objetivos desta pesquisa.

Nishino, Maekawa e Hara (2023) utilizam dados de aceleração coletados de sensores, que podem ser considerados como uma série temporal multivariada, no entanto com apenas 3 canais. Isso se deve ao fato de que os dados de aceleração são capturados em múltiplos eixos (por exemplo, eixos x, y e z), resultando em várias dimensões de dados ao longo do tempo. Cada eixo fornece uma série temporal separada, e a combinação dessas séries permite a análise de movimentos complexos e a detecção de repetições de ações humanas.

A abordagem proposta no estudo de Nishino, Maekawa e Hara (2023) envolve a criação de uma Matriz de Autossimilaridade (SSM) a partir dessas séries temporais multivariadas, que é então utilizada como entrada para a rede neural. A SSM ajuda a representar as características repetitivas nos dados de aceleração, permitindo que a rede neural identifique as localizações temporais das repetições de forma eficaz.

Apesar da técnica utilizada por Nishino, Maekawa e Hara (2023) obterem resultados mais próximos dos desejados para a solução do problema proposto nesta tese, o fato de analisar apenas 3 canais e dos autores mencionarem algumas limitações relacionadas ao método proposto, limitam o seu uso neste trabalho.

Nishino, Maekawa e Hara (2023) mencionam que a variabilidade nos dados, como a posição do sensor e a natureza das atividades realizadas, pode afetar a precisão do modelo. A variabilidade nas ações humanas e nas condições de movimento pode dificultar a detecção precisa de repetições em diferentes contextos, condições estas que são presentes nas máquinas ferramentas.

Outra limitação observada no estudo é que as repetições não são sempre capturadas de maneira uniforme em todos os canais da série temporal multivariada. Por exemplo, a Matriz de Autossimilaridade para o eixo z pode não capturar bem as repetições, o que pode impactar a eficácia do modelo (NISHINO; MAEKAWA; HARA, 2023).

Enquanto a tese apresentada foca na análise de séries temporais binárias multivariadas para a detecção de falhas em máquinas e equipamentos industriais, o trabalho de Gundersen et al. (2020) aborda a classificação de séries temporais binárias em um contexto diferente, relacionado a vazamentos de gases marinhos. Uma diferença importante é que a abordagem de Gundersen utiliza o Bayesian Convolutional Neural Network (BCNN) para modelar a incerteza nas previsões e gerar distribuições de probabilidade, o que apoia decisões mais informadas. Por outro lado, a tese enfatiza a simplicidade e eficiência computacional ao trabalhar com um conjunto maior de variáveis e explorar características específicas das séries binárias no contexto industrial. Além disso, enquanto Gundersen et al. (2020) exploram o aprendizado por transferência, pré-treinando o modelo com dados simulados e ajustando-o com medições *in loco*, esta tese propõe uma abordagem mais integrada ao cenário industrial, com técnicas adaptadas às características mecânicas e operacionais dos equipamentos analisados. Essa comparação evidencia como diferentes contextos de aplicação levam a escolhas distintas de metodologias e prioridades no desenvolvimento de soluções.

O estudo de Quinn *et al.* (2022) oferece uma abordagem interessante para a análise de séries temporais multivariadas, utilizando o modelo Recurrent Neural Network for Binary Outcome Forecasting (RNN-BOF) em um contexto clínico. No entanto, ao compararmos os resultados obtidos neste estudo com os apresentados nesta tese, algumas diferenças importantes emergem. Enquanto o RNN-BOF é projetado para capturar relações temporais e interações entre variáveis clínicas e

psicológicas em populações de pacientes, a abordagem desta tese é voltada para a análise de séries temporais binárias multivariadas no contexto industrial, com foco na detecção de falhas em máquinas e equipamentos. A seguir, serão discutidas as principais contribuições e limitações do trabalho de Quinn *et al.* (2022), evidenciando as distinções de aplicação, metodologia e resultados em relação à pesquisa aqui apresentada.

O estudo de Quinn *et al.* (2022) analisa séries temporais multivariadas utilizando o modelo Recurrent Neural Network for Binary Outcome Forecasting (RNN-BOF), projetado para lidar com dados de séries temporais que incluem múltiplas variáveis. O modelo utiliza uma abordagem global de previsão de séries temporais, onde a série temporal binária principal representa a presença de eventos de agressão, enquanto as séries de covariáveis incluem características clínicas, demográficas e medidas psicométricas.

O estudo enfatiza que, ao contrário de abordagens tradicionais que tratam cada paciente de forma isolada, o RNN-BOF treina um único modelo global que aprende a partir de dados de uma população de pacientes, permitindo capturar relações temporais e interações entre diferentes variáveis ao longo do tempo (QUINN *et al.*, 2022).

Apesar da eficácia do modelo RNN-BOF para previsões em contextos clínicos, seu uso na análise de falhas em máquinas ferramenta apresenta limitações. O foco do modelo está na previsão de eventos relacionados a condições clínicas e psicológicas, que possuem características muito diferentes das interações físicas e mecânicas encontradas em máquinas ferramenta. Além disso, o treinamento de um único modelo global a partir de uma população pode não ser aplicável a sistemas industriais, onde cada máquina ou componente pode apresentar padrões de falha únicos e específicos, exigindo abordagens mais individualizadas. A dependência de covariáveis clínicas e demográficas também torna o modelo inadequado para capturar a complexidade das variáveis mecânicas e operacionais envolvidas em sistemas industriais. (TV *et al.*, 2018; WU; HUANG; SUTHERLAND, 2020).

7.2 AVANÇO PROPORCIONADO PELO MÉTODO DESENVOLVIDO

O método teórico desenvolvido nesta tese representa um avanço significativo na análise de séries temporais binárias multivariadas, abordando lacunas importantes da literatura. Enquanto métodos convencionais frequentemente se concentram na detecção de anomalias gerais, o modelo proposto não apenas identifica falhas, mas também é capaz de localizá-las temporalmente, oferecendo insights precisos sobre quando e onde ocorrem os desvios nos dados. Esse avanço teórico é sustentado por três pilares principais:

1. **Transformação e Representação de Dados Binários:** O método introduz uma técnica inovadora para transformar séries temporais binárias multivariadas em um formato que permite capturar padrões latentes de maneira eficiente. Essa abordagem supera limitações de métodos tradicionais, que frequentemente ignoram as interações complexas entre variáveis binárias ao longo do tempo.
2. **Capacidade de Generalização:** Utilizando técnicas avançadas de aprendizado profundo, o modelo demonstra uma capacidade superior de generalização, identificando padrões normais e detectando falhas em contextos industriais. A robustez do método permite sua aplicação a diferentes máquinas e condições operacionais, tornando-o altamente versátil.
3. **Detecção e Localização de Falhas:** Um diferencial importante deste trabalho é a capacidade de localizar falhas temporalmente, permitindo identificar com precisão os pontos específicos nos dados em que as anomalias ocorrem. Essa funcionalidade é essencial para diagnósticos detalhados e ações corretivas direcionadas, permitindo maior precisão na identificação de falhas esporádicas que poderiam passar despercebidas por métodos tradicionais e promovendo maior eficiência na manutenção preditiva.

Esse avanço teórico não apenas amplia o conhecimento em análise de séries temporais binárias, mas também estabelece uma base sólida para futuras investigações.

7.3 APLICABILIDADE E EFICÁCIA PRÁTICA DAS FERRAMENTAS E ABORDAGENS EM CENÁRIOS REAIS

As ferramentas e abordagens desenvolvidas nesta tese foram projetadas para atender às necessidades práticas de ambientes industriais, demonstrando alta aplicabilidade e eficácia em cenários reais. A seguir, destacam-se os principais aspectos que reforçam essa aplicabilidade e eficiência prática:

Capacidade de operar com dados reais de sensores industriais: As ferramentas foram desenvolvidas para lidar diretamente com séries temporais binárias multivariadas, provenientes de sensores instalados em máquinas ferramenta. Essa característica elimina a necessidade de etapas complexas de pré-processamento ou ajustes manuais nos dados, tornando o modelo aplicável a cenários reais onde os dados são continuamente gerados.

Detecção antecipada e localização de falhas: O modelo demonstrou eficácia prática ao detectar falhas de forma antecipada, mesmo em situações onde falhas esporádicas poderiam passar despercebidas por sistemas tradicionais. Além disso, a capacidade de localizar temporalmente as falhas nos dados permite diagnósticos rápidos e precisos, reduzindo o tempo de inatividade das máquinas e maximizando a eficiência operacional.

Facilidade de integração com sistemas de manutenção preditiva (pdm): A abordagem proposta pode ser integrada a sistemas de manutenção preditiva já existentes, ampliando sua funcionalidade. A identificação de padrões anômalos e a sinalização de falhas críticas contribuem diretamente para a tomada de decisões informadas e para a redução de custos associados a manutenções reativas ou desnecessárias.

Testes em cenários reais e robustez ao longo do tempo: A interface desenvolvida para coletar os dados e os dados coletados em condições reais reforçam a aplicabilidade do método. Além disso, a abordagem foi projetada para lidar com a variabilidade dos dados industriais, garantindo robustez e confiabilidade mesmo em cenários dinâmicos e sujeitos a mudanças operacionais.

Contribuição para a digitalização industrial: O trabalho apoia a transformação digital de processos industriais, ao fornecer uma solução baseada em aprendizado de

máquina capaz de monitorar máquinas ferramenta de forma autônoma. Isso alinha o trabalho às iniciativas da Indústria 4.0, onde a inteligência nos processos é um fator determinante para a competitividade.

Esses fatores demonstram que as ferramentas desenvolvidas não são apenas teoricamente relevantes, mas também apresentam um impacto direto e significativo em cenários reais, oferecendo soluções práticas para desafios comuns enfrentados por indústrias que buscam maior confiabilidade e eficiência operacional.

8. CONCLUSÕES

Este trabalho teve como objetivo desenvolver uma abordagem avançada de fusão de dados e autoencoder para a detecção de falhas em máquinas ferramenta, focando na análise de séries temporais binárias. A importância desse tema está diretamente relacionada ao impacto que falhas nesses equipamentos têm sobre o tempo de operação e a eficiência dos processos produtivos na indústria.

Embora as máquinas ferramentas sejam equipadas com sistemas sofisticados, como CNC e CLP, a lógica de monitoramento tradicional utilizada por esses sistemas é reativa, respondendo apenas no momento exato da falha. Este estudo busca avançar para uma abordagem mais proativa, que permita identificar a degradação do sistema antes que falhas críticas ocorram, trazendo benefícios à manutenção preditiva.

Os principais resultados deste trabalho demonstraram que o uso de autoencoders em conjunto com a transformação de séries temporais binárias multivariadas pode ser uma estratégia eficaz para detectar falhas em máquinas ferramenta de forma antecipada. A análise dessas séries temporais, provenientes dos sensores instalados nas máquinas, permitiu identificar padrões que indicam falhas. Além disso, os modelos desenvolvidos foram capazes de detectar falhas esporádicas, que poderiam passar despercebidas pelo monitoramento tradicional baseado em limites de tolerância, como é o caso dos sistemas de CNC e CLP.

Ao treinar os modelos com dados de máquinas em condições normais de operação, foi possível alcançar uma precisão elevada na comparação entre o comportamento esperado e o comportamento atual da máquina, permitindo uma detecção eficiente de anomalias. A validação do modelo com dados reais, incluindo falhas propositalmente inseridas, mostrou que a técnica proposta é robusta e eficaz em cenários práticos. Como resultado, essa abordagem se mostrou capaz de melhorar significativamente a manutenção preditiva, otimizando a identificação e a resolução de falhas, sem a necessidade de desmontagens desnecessárias do equipamento.

Este trabalho traz contribuições teóricas e práticas importantes para o campo da detecção de falhas em máquinas ferramenta. No aspecto teórico, a principal contribuição reside na técnica autoral desenvolvida para análise de séries temporais binárias multivariada, que se baseia na análise de séries temporais binárias que contêm múltiplas variáveis de forma simultânea através da fusão destas variáveis em uma série temporal binária univariada.

Ao integrar esta técnica aos dados de sensores e aplicar técnicas de aprendizado de máquina avançadas, foi possível desenvolver uma metodologia capaz de explorar representações latentes de múltiplos canais de sinal simultaneamente, melhorando a precisão das estimativas de falhas.

No campo prático, este estudo contribui diretamente para o avanço da manutenção preditiva na indústria para máquinas ferramenta. A metodologia proposta não apenas detecta falhas de forma mais antecipada e precisa, como também oferece benefícios concretos, como a redução de custos de manutenção, menor tempo de inatividade, aumento da produção e maior segurança para os operadores. A aplicação dessa técnica pode otimizar o gerenciamento de recursos e melhorar a eficiência operacional, impactando positivamente o ciclo de vida das máquinas-ferramenta.

Apesar dos avanços e contribuições deste trabalho, é importante reconhecer algumas limitações que podem impactar os resultados e a generalização das soluções propostas. Essas limitações são inerentes tanto aos dados utilizados quanto às metodologias aplicadas, e é fundamental considerá-las ao avaliar o alcance e aplicabilidade das conclusões desta pesquisa.

Uma das principais limitações está relacionada à qualidade e à quantidade dos dados utilizados para o treinamento dos modelos. A coleta de dados realizada in loco, durante a instalação e manutenção das máquinas, pode não garantir uniformidade em termos de consistência e completude, principalmente devido a erros de operação ou à falta de treinamento adequado dos usuários responsáveis pelo manuseio da interface de coleta de dados. Essas falhas humanas podem resultar em dados incompletos ou com ruídos, afetando negativamente a acurácia do modelo na detecção de falhas, o que compromete sua eficácia em cenários reais de manutenção preditiva.

Essas limitações, embora presentes, não diminuem as contribuições deste estudo, mas servem como ponto de reflexão para melhorias futuras e o aprimoramento de metodologias em pesquisas subsequentes.

Com base nos resultados e limitações deste trabalho, é possível identificar diversas oportunidades para expandir e aprimorar as metodologias aplicadas. Uma das principais recomendações para trabalhos futuros é a inclusão da análise do grau de falha em séries temporais binárias. No presente estudo, o foco esteve na detecção binária da presença ou ausência de falhas, apesar da solução apresentada identificar a degradação gradual do sistema, não considera a intensidade ou a progressão das mesmas. Explorar o conceito de grau de falha permitiria uma análise mais detalhada, fornecendo insights mais ricos sobre a severidade e a evolução das falhas ao longo do tempo (BRAEI; WAGNER, 2020).

Incorporar essa dimensão seria especialmente relevante em contextos de manutenção preditiva, onde a identificação precoce da gravidade de uma falha pode orientar melhor as ações corretivas. Modelos que integrem diferentes níveis de severidade poderiam não apenas detectar falhas, mas também prever o impacto dessas falhas no desempenho da máquina, ajudando a priorizar intervenções de manutenção.

Essas recomendações abrem caminhos promissores para o desenvolvimento de sistemas de manutenção preditiva mais sofisticados, capazes de detectar e qualificar falhas com maior precisão e relevância, oferecendo suporte decisivo à tomada de decisões em ambientes industriais.

9. REFERÊNCIA BIBLIOGRÁFICAS

- ABNT. **NBR 5462 Confiabilidade e manutenibilidade**. Rio de Janeiro: Associação Brasileira de Normas Técnicas. 1994.
- AGGARWAL. C. C. 2017. **Outlier Analysis**. Springer International Publishing, 2017.
- AHMAD, F.; ADNAN, M.; AMIN, A. A.; ADNAN, M. A comprehensive review of fault diagnosis and fault-tolerant control techniques for modular multi-level converters. **SAGE Publishing**, v. 105, a. 3, 2022.
- AHMAD, R.; KAMARUDDIN, S. An overview of time-based and condition-based maintenance in industrial application. **Computers & Industrial Engineering**, v. 63, p. 135-149, 2012.
- AHMADZADEH, F.; LUNDBERG, J. Remaining useful life estimation: review. **International Journal of System Assurance Engineering and Management**, v. 5, p. 461–474, 2014.
- AHMED, S.; LEE, Y.; HYUN, S. -H.; KOO, I. Unsupervised Machine Learning-Based Detection of Covert Data Integrity Assault in Smart Grid Networks Utilizing Isolation Forest. **IEEE Transactions on Information Forensics and Security**, v. 14, a. 10, p. 2765-2777, 2019.
- AHN, H.; BALASUBRAMANIAN, R.; CHAHAR, R.; GUPTA, A.; OLIVAR, S.; SAHU, A. Building a Scalable Intelligent System to Advise Predictive Maintenance Operations in a Steel Mill. **AISTech 2021**, p. 1156-1163, 2021.
- AIWINA, H.; SHENG, Z.; ANDY, C. C. T.; JOSEPH, M. Rotating machinery prognostics: State of the art, challenges and opportunities. **Mechanical Systems and Signal**, v. 23, a. 3, p. 724–739, 2009.
- AKSENTIJEVIC, A.; MIHAILOVIĆ, D. T.; KAPOR, D.; CRVENKOVIĆ, S.; NIKOLIC-DJORIĆ, E.; MIHAILOVIĆ, A. Complementarity of information obtained by Kolmogorov and Aksentijevic–Gibson complexities in the analysis of binary time series. **Chaos, Solitons & Fractals**, v. 130, 2020.
- ALFARO-NANGO, A. J.; ESCOBAR-GÓMEZ, E. N.; CHANDOMÍ-CASTELLANOS, E.; VELÁZQUEZ-TRUJILLO, S.; HERNANDEZ-DE-LEÓN H. R.; BLANCO-GONZÁLEZ,

L. M. Predictive Maintenance Algorithm Based on Machine Learning for Industrial Asset. 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), Istanbul, Turkey, p. 1489-1494, 2022.

ALI, H.; CHEN, D.; HARRINGTON, M.; SALAZAR, N.; AMEEDI, M. A.; KHAN, A. F.; BUTT, A. R.; CHO, J. -H. A Survey on Attacks and Their Countermeasures in Deep Learning: Applications in Deep Neural Networks, Federated, Transfer, and Deep Reinforcement Learning. **IEEE Access**, v. 11, p. 120095-120130, 2023.

ALKHAMMASH, E. H.; KAMEL, A. F.; AL-FATTAH, S. M.; ELSHEWEY, A. M. Optimized multivariate adaptive regression splines for predicting crude oil demand in Saudi arabia. **Discrete Dynamics in Nature and Society**, v. 12, 2022.

ALLOGHANI, M.; AL-JUMEILY, D.; MUSTAFINA, J.; HUSSAIN, A.; ALJAAF, A. J. **A systematic review on supervised and unsupervised machine learning algorithms for data Science**. In: BERRY, M.; MOHAMED, A.; YAP, B. **Supervised and Unsupervised Learning for Data Science**, p. 3-21, Charm: Springer, 2020.

ALPAYDIN, E. Introduction tom machine learning. 2^a. ed. Cambridge: MIT press 2010.

ALQUDAH, M.; KEZUNOVIC, M.; OBRADOVIC, Z. Automated Power System Fault Prediction and Precursor Discovery Using Multi-Modal Data. **IEEE Access**, v. 11, p. 7283-7296, 2023.

AMRO, I. Dynamic Estimation Algorithm for Markovian Model for Packet Loss. 2013 8th EUROSIM Congress on Modelling and Simulation, Cardiff, UK, p. 610-615, 2013.

AMRO, I. Markovian Model for Packet Loss in EUMEDConnect Network. 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), p.137-141, 2013.

AN, J.; CHO, S. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. **Special Lecture on IE**, v. 2, a. 1, p. 1-18, 2015.

ANAS, N. M.; YUSOF, M. Y. M.; AZIZ, M. Z. A. Machine Tool Condition Monitoring System: A Review on Feasible Solution. 2022 International Conference on Artificial Intelligence of Things (ICAIoT), Istanbul, Turkey, p. 1-4, 2022.

ARAÚJO, C. L. D. O.; SANTOS, T. J. N.; GALVÃO, H. M. Decision-Making by Data, An Application of Data Science with Cognitive Ergonomics. **International journal of advances in management and economics**, v. 7, a. 4, 2018.

ARENA, S.; MANCA, G.; MURRU, S.; ORRÙ, P. F.; PERNA, R.; RECUPERO, D. R. Data Science Application for Failure Data Management and Failure Prediction in the Oil and Gas Industry: A Case Study. **Applied Sciences**. v. 12, a. 20, p.10617, 2022.

ARÉVALO, F.; RERNENTERÍA, J.; SCHWUNG, A. Fault Detection Assessment Architectures based on Classification Methods and Information Fusion. 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, p. 1343-1350, 2018.

ARIYALURAN HABEEB, R. A.; NASARUDDIN, F.; GANI, A.; TARGIO HASHEM, I. A.; AHMED, E.; IMRAN, M. Real-time big data processing for anomaly detection: A Survey, **International Journal Information Management**, v. 45, p. 289–307, 2019. AZZOUZ, I.; BOUSSAID, B.; ZOUINKHI, A.; ABDELKRIM, M. N. Multi-faults classification in WSN: A deep learning approach. 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, p. 343-348, 2020.

BABAYIGIT, B.; ABUBAKER, M.; Industrial Internet of Things: A Review of Improvements Over Traditional SCADA Systems for Industrial Automation. **IEEE Systems Journal**, v. 18, a. 1, p. 120-133, 2024.

BAEK, J.; ALHINDI, T. J.; JEONG, Y.; JEONG, M. K.; SEO, S.; KANG, J.; SHIM, W.; HEO, Y. Real-time fire detection system based on dynamic time warping of multichannel sensor networks. **Fire Safety Journal**, v. 123, p. 103364-103364, 2021.

BAGHBANPOURASL, A.; KIRCHBERGER, D.; EITZINGER, C. Failure prediction through a model-driven machine learning method. 2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT), Rome, Italy, p. 527-531, 2021.

BANK, D.; KOENIGSTEIN, N.; GIRYES, R. Autoencoders. **Cornell University**, 2020.

BANZI, M. **Getting Started with Arduino**. Sebastopol: O'Reilly, 2011.

BARBARIOL, T.; MASIERO, D.; FELTRESI, E.; SUSTO, G. A. Time series Forecasting to detect anomalous behaviors in Multiphase Flow Meters. Cornell University, 2023.

BARCELÓ-ORDINAS, J. M.; GARCÍA-VIDAL, J.; DOUDOU, M.; RODRIGO, S.; CEREZO-LLAVERO, A. Calibrating low-cost air quality sensors using multiple arrays of sensors. 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, p. 1-6, 2018.

BAUTU, E.; BAUTU, A.; LUCHIAN, H.; EVOLVING Gene Expression Programming Classifiers for Ensemble Prediction of Movements on the Stock Market. 2010 International Conference on Complex, Intelligent and Software Intensive Systems, Krakow, Poland, p. 108-115, 2010.

BAUTU, E.; KIM, S.; BAUTU, A.; LUCHIAN, H.; ZHANG, B. -T. Evolving hypernetwork models of binary time series for forecasting price movements on stock markets. 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, p. 166-173, 2009.

BEGHI, A.; BRIGNOLI, R.; CECCHINATO, L.; MENEGAZZO, G.; RAMPAZZO, M.; SIMMINI, F. Data-driven fault detection and diagnosis for hvac water chillers. **Control Engineering Practice**, v. 53, p. 79–91, 2016.

Benetti, D.; Dureck, E. H.; Dreyer, U. J.; Pipa, D. R.; Cardozo da Silva, J. C. Feature Extraction and Selection for Identifying Faults in Contactors Using Fiber Bragg Grating. **IEEE Sensors Journal**, v. 23, a. 17, p. 20357-20367, 2023.

BENGIO, Y. Learning Deep Architectures for AI. Foundations and Trends in Machine Learning, v. 2, a. 1, p.1-127, 2009.

BERGSTRA, J.; BENGIO, Y. Random Search for Hyper-Parameter Optimization. **Journal of Machine Learning Research**, v. 13, a. 1, p. 281-305, 2012.

BIGGIO, L.; KASTANIS, I. Prognostics and Health Management of Industrial Assets: Current Progress and Road Ahead, **Frontiers in Artificial Intelligence**, v. 3, 2020.

BIGHAMIAN R.; SHANECHI, M. M. Estimation of Functional Dependence in High-Dimensional Spike-Field Activity. 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, p. 2635-2638, 2018.

BINDING, A.; DYKEMAN, N.; PANG, S. Machine Learning Predictive Maintenance on Data in the Wild. 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, p. 507-512, 2019.

- BISHOP, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.
- BISWAL, S; SABAREESH, G. R. Design and development of a wind turbine test rig for condition monitoring studies, 2015 International Conference on Industrial Instrumentation and Control (ICIC), Pune, India, p. 891-896, 2015.
- BLAGEC, K.; DORFFNER, G.; MORADI, M.; SAMWALD, M. A critical analysis of metrics used for measuring progress in artificial intelligence. Cornell University, 2020.
- BLOWERS, M.; IRIBARNE, J.; SCOTT, G. M. Multivariable analysis, correlation, and prediction. SPIE Defense, Security, and Sensing, 2009, Orlando, Florida, United States, v. 7347, 2009.
- BODNER, N.; CEULEMANS, E. ConNEcT: An R package to build contingency measure-based networks on binary time series. **Behavior Research Methods**, v. 55, p. 301–326, 2023.
- BÖGELSACK, G.; GIERSE, F. J.; ORAVSKÝ, V.; PRENTIS, J. M.; ROSSI, A. Terminology for the theory of machines and mechanisms. **Mechanism and Machine Theory**, v. 18, n. 6, p. 397-408, 1983.
- BORITH, T.; BAKHIT, S.; NASRIDINOV, A.; YOO, K. Prediction of Machine Inactivation Status Using Statistical Feature Extraction and Machine Learning. **Multidisciplinary Digital Publishing Institute**, v. 10, a. 21, p. 7413-7413, 2020.
- BOZKURT, E.; ORTEGA, A. Non-Negative Kernel Graphs for Time-Varying Signals Using Visibility Graphs. 2022 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, p. 1781-1785, 2022.
- BRAEI, W.; WAGNER. S. Anomaly Detection in Univariate Time-series: A Survey on the Stateof- the-Art. **ArXiv**, 2020.
- BREIMAN, L. Random Forests. **Machine Learning**, v. 45, p. 5–32, 2001.
- BREIMAN, L. Statistical Modeling: The Two Cultures. **Statistical Science**, 2001.
- BRIDA, J. G.; GARRIDO, N. Exploring Two Inflationary Regimes in Latin-American Economies: A Binary Time Series Analysis. **International Journal of Modern Physics C**, v. 17, a. 3, p. 343-356, 2006.

BUDIARTO, E.; PERMANASARI, A.; FAUZIATI, S. Unsupervised Anomaly Detection Using K-Means, Local Outlier Factor and One Class SVM. 5th International Conference on Science and Technology (ICST), p. 1-5. 2019.

BUSCEMA, M.; SACCO, P. L. Digging deeper on deep learning: A computational ecology approach. **Behavioral and Brain Sciences**, v. 40, a. 256, 2017.

BUTTE, S.; PRASHANTH, A. R.; PATIL, S. Machine Learning Based Predictive Maintenance Strategy: A Super Learning Approach with Deep Neural Networks. **2018 IEEE Workshop on Microelectronics and Electron Devices (WMED)**, Boise, ID, USA, p. 1-5, 2018.

BYKOV, A. D.; VORONOV, V. I.; VORONOVA, L. I.; Machine Learning Methods Applying for Hydraulic System States Classification. 2019 Systems of Signals Generating and Processing in the Field of on-Board Communications, Moscow, Russia, p. 1-4, 2019.

CAJAHUARINGA, A.; PALACIOS, R. A.; MAURICIO VILLANUEVA, J. M.; MORALES-VILLANUEVA, A.; MACHUCA, J.; CONTRERAS, J.; RODRÍGUEZ BAUTISTA, K. Uncertainty Evaluation of a Gas Turbine Model Based on a Nonlinear Autoregressive Exogenous Model and Monte Carlo Dropout. **Sensors**, v. 24, p. 465, 2024.

CAMPOS, V.; SASTRE, F.; YAGÜES, M.; MÍRIAM BELLVER, GIRÓ-I-NIETO, X.; TORRES, J. Distributed training strategies for a computer vision deep learning algorithm on a distributed GPU cluster. **Procedia Computer Science**, v. 108, p. 315-324, 2017.

CAO, S., LIU, A., & HUANG, C. (2024, January 1). Designing for Appropriate Reliance: The Roles of AI Uncertainty Presentation, Initial User Decision, and User Demographics in AI-Assisted Decision-Making. Cornell University.

CARVALHO, T. P.; SOARES, F. A.; VITA, R.; FRANCISCO, R. D.; BASTO, J. P.; ALCALÁ, S. G. A systematic literature review of machine learning methods applied to predictive maintenance. **Computers & Industrial Engineering**, p. 137, 2019.

CAVALCANTE, M. A.; TAVOLARO, C. R.C.; MOLISANI, E. Física com Arduino para iniciantes. **Revista Brasileira de Ensino de Física**. v.33, n.4, p. 4503, 2011.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection. **Association for Computing Machinery**, v. 41, a. 3, p. 1-58, 2009.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. **Computers & Electrical Engineering**, v. 40, a. 1, p. 16 – 28, 2014.

CHEN, C. -B.; WANG, Y.; FU, X.; YANG, H. Recurrence Network Analysis of Histopathological Images for the Detection of Invasive Ductal Carcinoma in Breast Cancer. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 20, a. 5, p. 3234–3244, 2023.

CHEN, G.; ZHANG, L.; ARINEZ, J.; BILLER, S. Energy-Efficient Production Systems Through Schedule-Based Operations. **IEEE Transactions on Automation Science and Engineering**, v. 10, a. 1, p. 27-37, 2013.

CHEN, K. S.; LIU, C. C.; CHEN, C. H. Fuzzy evaluation of process quality with process yield index. **Mathematics**, v. 10, p. 2514, 2022.

CHEN, K.; FENG, M.; WIRJANTO, T. S. (2023, January 1). Multivariate Time Series Anomaly Detection via Dynamic Graph Forecasting. Cornell University, 2023.

CHEN, Y.; ZHAO, Z.; WU, H.; CHEN, X.; XIAO, Q.; YU, Y. Fault Anomaly Detection of Synchronous Machine Winding Based on Isolation Forest and Impulse Frequency Response Analysis. **Measurement**, p. 188, 2021.

CHENG, Y.; ARAPIN, A.; ZHANG, Z.; ZHANG, Q.; LI, H.; FEAMSTER, N.; JIANG, J. GRACE: Loss-Resilient Real-Time Video through Neural Codecs. **Cornell University**. 2023.

CHENG, Z.; SUN, H.; TAKEUCHI, M.; KATTO, J. Deep convolutional autoencoder-based lossy image compression, Picture Coding Symposium (PCS), IEEE 2018, p. 253–257, 2018.

CHO, S. J.; BROWN-SCHMIDT, S.; LEE, W. Y. Autoregressive Generalized Linear Mixed Effect Models with Crossed Random Effects: An Application to Intensive Binary Time Series Eye-Tracking Data. **Psychometrika**, v. 83, a. 3,p. 751-771, 2018.

CHOI, N.-H.; SOHN, J.W.; OH, J.-S. Defect Detection Model Using CNN and Image Augmentation for Seat Foaming Process. **Mathematics**, v. 11, p. 4894, 2023.

CHOI, R. Y.; COYNER, A. S.; KALPATHY-CRAMER, J.; CHIANG, M. F.; PETER CAMPBELL, J. Introduction to machine learning, neural networks, and deep learning. **Translational Vision Science and Technology**, v. 9, n. 2, p. 14, 2020.

CHOMPHU, W.; KIJSIRIKUL, B. 2020. Wellhead Compressor Failure Prediction Using Attention-based Bidirectional LSTMs with Data Reduction Techniques. In Proceedings of the 2020 4th International Conference on Compute and Data Analysis (ICCDA '20). Association for Computing Machinery, New York, NY, USA, p. 16–22, 2020.

CHOU, C.; CHU, C. -S. Testing independence of two autocorrelated binary time series. **Statistics & Probability Letters**, v. 80, a. 1, p. 69-75, 2010.

CINAR, Z. M.; ABDUSSALAM, N. A.; ZEESHAN, Q.; KORHAN, O.; ASMAEL, M.; SAFAEI, B. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. **Sustainability**, v. 12, a. 19, p. 8211, 2020.

CINAR, Z. M.; ZEESHAN, Q.; SOLYALI, D.; KORHAN, O. **Simulation of factory 4.0: a review. in industrial engineering in the digital disruption era**. Cham, Switzerland, Springer, 2019.

CORALLO, A.; CRESPINO, A.; DIBICCARI, C.; LAZOI, M.; LEZZI, M.; Processing Big Data in Streaming for Fault Prediction: An Industrial Application. 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Las Palmas de Gran Canaria, Spain, p. 730-736, 2018.

COSTACHE, M. -C.; MINZU, V. Multi-agent systems in industrial fault diagnosis. 2012 16th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 2012, pp. 1-6, 2012.

COWAN, M.; MOREAU, T.; CHEN, T.; CEZE, L. Automating Generation of Low Precision Deep Learning Operators. **Cornell University**, 2018.

CÓZAR, J.; PUERTA, J.; GÁMEZ, J. An Application of Dynamic Bayesian Networks to Condition Monitoring and Fault Prediction in a Sensored System: a Case Study. **International Journal of Computational Intelligence Systems**, v. 10, p. 176, 2017. .

CUI, Y.; KOPPOL, P.; ADMONI, H.; NIEKUM, S. SIMMONS, R.; STEINFELD, A.; FITZGERALD, T. Understanding the Relationship between Interactions and Outcomes

in Human-in-the-Loop Machine Learning. **Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence IJCAI-21**, p. 4382-4391, 2021.

CUZZOCREA, A. Scalable Distributed Pivot Analysis over Massive Big Data: Models, Paradigms, New Advancements, 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy, p. 696-700, 2020.

CUZZOCREA, A.; MARTINELLI F.; MERCALDO, F. Detecting and Analyzing Anomalies Across Historical Data Changes: A Data-Driven Approach. 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), Volos, Greece, p. 832-837, 2018.

DAKHIL, A. F.; ALI, W. M.; ABDULREDAH, A. A. Predicting Prior Engine Failure with Classification Algorithms and web-based IoT Sensors. 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), Bangladesh, p. 1-6, 2020.

DAMAVANDI, H. G.; SHAH, R.; STAMPOULIS, D.; WEI, Y.; BOSCOVIC, D.; SABO, J. L. Accurate Prediction of Streamflow Using Long Short-Term Memory Network: A Case Study in the Brazos River Basin in Texas. International journal of environmental science and development, v.10, a.10, p. 294-300, 2019.

DANIEL, K.; & MOULDER, R.; TEACHMAN, B.; BOKER, S. Stability and spread: A novel method for quantifying transitions within multivariate binary time series data. **Behavior Research Methods**, v. 55, a.5, 2022.

DAS, S.; DATTA, S.; CHAUDHURI, B. B.; Handling data irregularities in classification: Foundations, trends, and future challenges, **Pattern Recognition**, v. 81, p. 674-693, 2018.

DELISE, T. Deep Semi-Supervised Anomaly Detection for Finding Fraud in the Futures Market. **Cornell University**, 2023.

DI, W.; HAOYUE, L.; YUMING, L. 2022. Intelligent weight generation algorithm based on binary isolation tree. **Engineering Applications of Artificial Intelligence**, p. 109, 2022.

DIALLO, M. S.; MOKEDEM, S. A.; BRAUD, A.; FREY, G.; Lachiche, N. Identifying Benchmarks for Failure Prediction in Industry 4.0. **Informatics**, v. 8, p. 68, 2021.

DIEZ-OLIVAN, A.; PAGAN, J. A.; KHOA, N. L. D.; SANZ, R.; SIERRA, B. Kernel-based support vector machines for automated health status assessment in monitoring sensor data. **Internacional journal of advanced manufacturing technology**, v. 95, p. 327–340, 2018.

DING, X.; WANG, H.; SU, J.; WANG, C. Misplaced Subsequences Repairing with Application to Multivariate Industrial Time Series Data. Cornell University, 2020.

DING, Z.; FEI, M. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. **IFAC Proceedings Volumes**, v. 46, a. 20, p. 12–17, 2013.

DIX, M.; CHOUHAN, A.; GANGULY, S.; PRADHAN, S.; SARASWAT, D.; AGRAWAL, S.; PRABHUNE, A. Anomaly detection in the time-series data of industrial plants using neural network architectures. 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, United Kingdom, p. 222-228, 2021.

DIZAJI, K. G.; HERK, A.; DIZAJI, G.; HERANDI, A.; DENG, C.; CAI, W.; HUANG, H. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in: Proceedings of the IEEE international conference on computer vision, p. 5736–5745, 2017.

DMTG. **VDLS-850 Operation manual**. Dailan, China, 2020.

DÔRES, S. N.; ALVES, L.; RUIZ, D. D.; BARROS, R. C. A meta-learning framework for algorithm recommendation in software fault prediction. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16). **Association for Computing Machinery**, New York, NY, USA, p. 1486–1491, 2016.

DORI, N.; PIEDRAHITA, P.; LOUZOUN, Y. Two stage approach to functional network reconstruction for binary time-series. **The European Physical Journal B**, v. 92, a. 45, 2019.

DUCHANAY, C. A.; CALVO, H.; MORENO-ARMENDÁRIZ, M. A. ASAMS: An adaptive sequential sampling and automatic model selection for artificial intelligence surrogate modeling. **Sensors**, v. 20, a. 18, p. 5332, 2020.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. **Journal of Machine Learning Research**, 12, 2121-2159, 2011.

DUFFIELD, N.; HAFFNER, P.; KRISHNAMURTHY, B.; RINGBERG, H. Rule-Based Anomaly Detection on IP Flows, IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 2009, p. 424-432, 2009.

EATOCK, J.; LIN, Y. T.; CHANG, E.; GALLA, T.; CLAYTON, R. Assessing Measures of Atrial Fibrillation Clustering via Stochastic Models of Episode Recurrence and Disease Progression, **Computing in Cardiology**, v. 42, p. 265-268, 2015.

ENZBERG, S. V.; NASKOS, A.; METAXA, I. N.; KÖCHLING D.; KÜHN. A. Implementation and transfer of predictive analytics for smart maintenance: a case study. **Frontiers of Computer Science**, v. 2, 2020.

ERHAN, D.; BENGIO, Y.; COURVILLE, A.; VINCENT, P. (2010). Why Does Unsupervised Pre-training Help Deep Learning? **Journal of Machine Learning Research**, v.11, p. 625-660, 2010.

ERICEIRA, D. R.; ROCHA, F.; BIANCHI, A. G. C.; PESSIN, G. Early Failure Detection of Belt Conveyor Idlers by Means of Ultrasonic Sensing. 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, p. 1-8, 2020.

FANG, S.; FEI, Y.; XU, Z.; TSAO, Y. Learning Transportation Modes From Smartphone Sensors Based on Deep Neural Network. **IEEE Sensors**, p. 17, a. 18, p. 6111-6118, 2017.

FASANO, A.; REBAUDO, G.; DURANTE, D.; PETRONE, S. A closed-form filter for binary time series. **Statistics and Computing**, v. 31, 2021.

FINK, O.; WANG, Q.; SVENSEN, M.; DERSIN, P.; LEE, W.; DUCOFFE, M. Potential, Challenges and Future Directions for Deep Learning in Prognostics and Health Management Applications. **Engineering Applications of Artificial Intelligence**, v. 92, a. 6, p. 103678, 2020.

FIORENTINI, N.; LOSA, M. Long-Term-Based Road Blackspot Screening Procedures by Machine Learning Algorithms. **Sustainability** , v.12, p.5972, 2020.

- FOKIANOS, K.; GOMBAY, E.; HUSSEIN, A. Retrospective change detection for binary time series. **Journal of Statistical Planning and Inference**, v. 145, p. 102–112, 2014.
- FOKIANOS, K.; MOYSIADIS, T. Binary time series models driven by a latent process, **Econometrics and Statistics**, v. 2, p. 117-130, 2017.
- FORMENTIN, S.; VAN HEUSDEN, K.; KARIMI, A. Model-based and data-driven model-reference control: A comparative analysis, 2013 European Control Conference (ECC), Zurich, Switzerland, p. 1410-1415, 2013.
- FORTUIN, V.; RÄTSCH, G.; MANDT, S. Multivariate Time Series Imputation with Variational Autoencoders. Cornell University, 2019.
- FRONTONI, E.; POLLINI, R.; RUSSO, P.; ZINGARETTI, P.; CERRI, P. Hdomo: Smart sensor integration for an active and independent longevity of the elderly. **Sensors**, v. 17, a. 11, p. 2610, 2017.
- FUMEO, E.; ONETO, L.; ANGUITA, D. Condition Based Maintenance in Railway Transportation Systems Based on Big Data Streaming Analysis, **Procedia Computer Science**, v. 53, p. 437-446, 2015.
- GAO, X.; GILLEN, D.; OMBAO, H. Fisher information matrix of binary time series. **METRON**, v. 76, p. 287–304, 2018.
- GAO, X.; SHAHBABA, B.; OMBAO, H. Modeling Binary Time Series Using Gaussian Processes with Application to Predicting Sleep States. **Journal of Classification**, v.35, 2017.
- GAO, Y.; KONTOYIANNIS, I.; BIENENSTOCK, E. Estimating the Entropy of Binary Time Series: Methodology, Some Theory and a Simulation Study. **Entropy**, v. 10, p. 71-99, 2008.
- GIDON, J. S.; BORAH, J.; SAHOO, S.; MAJUMDAR, S.; FUJITA, M. Bidirectional LSTM Model for Accurate and Real-Time Landslide Detection: A Case Study in Mawiongrim, Meghalaya, India. **IEEE Internet of Things Journal**, v. 11, a. 3, p. 3792-3800, 2024.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. MIT Press, 2016.

GORCIN, A.; CELEBI, H.; QARAQE, K. A.; ARSLAN, H. An autoregressive approach for spectrum occupancy modeling and prediction based on synchronous measurements. 2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, Toronto, ON, Canada, p. 705-709, 2011.

GOTO, Y.; TANIGUCHI, M. Discriminant analysis based on binary time series. **Metrika**, p. 83, 2020.

GRØNBECH, C. H.; VORDING, M. F.; TIMSHEL, P.; SØNDERBY, C. K.; PERS, T. H.; WINTHER, O. scVAE: variational auto-encoders for single-cell gene expression data, **Bioinformatics**, v. 36, a. 16, p. 4415–4422, 2020.

GUERRA, A.; GADHIYA, V.; SRISURIN, P. Crash prediction on road segments using machine learning methods. **Asean Engineering journal**, v. 12, a. 3, 2022.

GUGARATSHAN, G.; BARTHLOW, D.; LINGENFELSER, D.; THUMATI, B. Machine Learning-Driven RAM Analysis Using Multi-variate Sensor Data. 2023 Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, p. 1-6, 2023.

GUNDERSEN, K.; ALENDAL, G.; OLEYNIK, A.; BLASER, N. Binary Time Series Classification with Bayesian Convolutional Neural Networks When Monitoring for Marine Gas Discharges. **Algorithms**, v. 13, p. 145, 2020.

GUO, G.; WANG, H.; BELL, D.; BI, Y. KNN Model-Based Approach in Classification. 2004.

GUPTA, C.; GILL, P. Machine learning techniques and extreme learning machine for early breast cancer prediction, **International Journal of Innovative Technology and Exploring Engineering**, v. 9, a. 4, p. 163-167, 2020.

HANSDA, R. CHAKRABORTY, S. Influence of Nodes in Complex Networks. 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, p. 1-6, 2023.

HAO, W.; FAN, J. M. Review on Evaluation Criteria of Machine Learning Based on Big Data. **Journal of Physics: Conference Series**, v. 5, 2020.

HASHEMI, S. J.; AHMED, S.; KHAN, F. Correlation and Dependency in Multivariate Process Risk Assessment. **IFAC-PapersOnLine**, v. 48, a. 21, p. 1339-1344, 2015.

HASHEMIAN, H. M.; BEAN, W. C. State-of-the-art predictive maintenance techniques. **IEEE Transactions on Instrumentation and measurement**, v. 60, a. 10, p. 3480–3492, 2011.

HASHEMIA, N.; FAN, Y.; ROCHA, N. Using Machine Learning to Predict and Avoid Malfunctions: A Revolutionary Concept for Condition-Based Asset Performance Management (APM). 2021 IEEE PES Innovative Smart Grid Technologies - Asia (ISGT Asia), Brisbane, Australia, p. 1-8, 2021.

HASSAN, M.; SADEK, A.; ATTIA, H. A Real-Time Deep Machine Learning Approach for Sudden Tool Failure Prediction and Prevention in Machining Processes. **Sensors**, v. 23, p. 3894, 2023.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference and Prediction**. New York: Springer, 2008.

HE, Z.; XU, X.; DENG, S. Discovering cluster-based local outliers. **Pattern Recognition Letters**, v. 24, p. 1641–1650, 2003.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. **Science**, v. 313, a. 5786, p. 504-507, 2006.

HO, T. K. Random decision forests. **Proceedings of 3rd internation conference on document analysis and recognition, IEEE**, v.1, p. 278-282, 1995.

HOONG ONG, K. S.; NIYATO, D.; YUEN, C. Predictive Maintenance for Edge-Based Sensor Networks: A Deep Reinforcement Learning Approach. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, p. 1-6, 2020.

HU, J.; CHEN, Z.; LIN, W.; FAN, X. Bayesian detection of event spreading pattern from multivariate binary time series. **Communications in Statistics - Simulation and Computation**, v. 52, p. 1-38, 2021.

HU, W.; YAO, Y.; LYU, S.; LI, J.; CHEN, H. Short isometric shapelet transform for binary time series classification. **Knowledge and Information Systems**, v. 63, p. 2023–2051, 2021.

HUANG, C. -G.; LI, X. -Y.; HUANG, H. -Z.; LI, Y. -F. Fault prognosis of Engineered Systems: A Deep Learning Perspective, 2019 Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, pp. 1-7, 2019.

HUDECOVÁ, S. Structural changes in autoregressive models for binary time series. **Journal of Statistical Planning and Inference**, v. 143, a. 10, p. 1744-1752, 2013.

HUNG, Y.; ZARNITSYNA, V.; ZHANG, Y.; ZHU, C.; WU, C. Binary Time Series Modeling With Application to Adhesion Frequency Experiments. **Journal of the American Statistical Association**, v. 103, p.1248-1259, 2008.

ILIOPOULOS, A.; VIOLOS, J.; DIOU, C.; VARLAMIS, I. Detection of Anomalies in Multivariate Time Series Using Ensemble Techniques. Cornell University, 2023.

Iqbal, A.; Raza, M. S.; Ibrahim, M.; Baz, A.; Alhakami, H.; Saeed, M. A. An improved approach for finding rough set based dynamic reducts. **IEEE Access**, v. 8, p. 173008-173023, 2020.

JAIN, P.; BAJPAI, M.; PAMULA, R. A Modified DBSCAN Algorithm for Anomaly Detection in Time-series Data with Seasonality. **The International Arab Journal of Information Technology**, 2022.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning**. New York: Springer New York, 2014.

JANG, H.; SIMEONE, O. Multisample Online Learning for Probabilistic Spiking Neural Networks. **IEEE Transactions on Neural Networks and Learning Systems**, v. 33, a. 5, p. 2034-2044, 2022.

JANG, H.; SIMEONE, O. Training Dynamic Exponential Family Models with Causal and Lateral Dependencies for Generalized Neuromorphic Computing. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, p. 3382-3386, 2019.

JARDINE, A. K. S.; LIN, D.; BANJEVIC, D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. **Mechanical Systems and Signal Processing**, v. 20, a. 7, p. 1483-1510, 2006.

JATANA, N.; PURI, S.; AHUJA, M.; KATHURIA, I.; GOSAIN, D. A survey and comparison of relational and non-relational database. **International Journal of Engineering Research & Technology**, v. 1, a. 6, p. 1–5, 2012.

JENTSCH, C.; REICHMANN, L. Generalized Binary Time Series Models. **Econometrics**, v. 7, a. 4, p. 47, 2019.

JIA, B.; MENG, Q.; CHEN, Y.; YANG, H. Prediction of Membrane Protein Amphiphilic Helix Based on Horizontal Visibility Graph and Graph Convolution Network. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 20, a. 6, p. 3567-3574, 2023.

JIANG, T. Z.: 'Research on multi-target fusion algorithm for distributed detection', **Journal China Academy of Electronics and Information Technology**, v. 5, a. 6, p. 594–598, 2010.

JIAO, J.; LIANG, K.; DING, C.; LIN, J. Towards Prediction Constraints: A Novel Domain Adaptation Method for Machine Fault Diagnosis. **IEEE Transactions on Industrial Informatics**, v. 18, a. 10, p. 7198-7207, 2022.

JIN, G.; JIN, S.; WANG, D.; PIAO, C. D.; YIN, F. Study on the reliability allocation of machining centers based on data envelopment analysis method. **Proceedings of the SPIE**, v. 12744, p. 9, 2023.

KAEWKORN, S.; JOOCHIM, C.; PRASERTPRASASNA, S.; LEARTRUSSAMEEJIT, C.; KUHATAPARUKS, H.; KUNAPINUN, A. Notifying problems of a machine by using Machine Learning. 2019 Research, Invention, and Innovation Congress (RI2C), Bangkok, Thailand, p. 1-6, 2019.

KAJMAKOVIC, A.; PESTANA, J.; DIWOLD, K.; RÖMER, K. Fault Diagnosis and Isolation Prediction for Redundant Relays Based on Discrepancy Analysis. In *2023 13th International Conference on Power, Energy and Electrical Engineering, CPEEE*, p. 286-295, 2023.

KANAWADAY, A.; SANE, A. Machine learning for predictive maintenance of industrial machines using IoT sensor data. 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, p. 87-90, 2017.

KANE, A P.; KORE, A. S.; KHANDALE, A. N.; NIGADE, S. S.; JOSHI, P P. Predictive Maintenance using Machine Learning. **Cornell University**, 2022.

KARAPALIDOU, E.; ALEXANDRIS, N.; ANTONIOU, E.; VOLOGIANNIDIS, S.; KALOMIROS, J.; VARSAMIS, D. Implementation of a Sequence-to-Sequence Stacked Sparse Long Short-Term Memory Autoencoder for Anomaly Detection on Multivariate Timeseries Data of Industrial Blower Ball Bearing Units. **Multidisciplinary Digital Publishing Institute**, p. 23 a.14, 2023.

KAUFMAN, E.; IAREMENKO, A. Anomaly Detection for Fraud in Cryptocurrency Time Series. Cornell University, 2022.

KAŹMIERCZAK, S. Prediction of the Costs of Forwarding Contracts with Machine Learning Methods. **Proceedings of the 17th conference on computer science and intelligence systems**, p. 413-416, 2022.

KEDEM, B.; FOKIANOS, K. **Regression Models for Binary Time Series**. In: Modeling Uncertainty, p.185-199. 2005.

KHAN, A. M.; SIDDIQI, M. H.; LEE, S. -W. Exploratory Data Analysis of Acceleration Signals to Select Light-Weight and Accurate Features for Real-Time Activity Recognition on Smartphones. **Sensors**, v. 13, p. 13099-13122, 2013.

KHARITONOV, A.; NAHHAS, A.; POHL, M.; TUROWSKI, K. Comparative analysis of machine learning models for anomaly detection in manufacturing. **Procedia Computer Science**, v. 200, p. 1288–1297, 2022,

KHOSRAVINIA, P. Enhancing road safety through accurate detection of hazardous driving behaviors with graph convolutional recurrent networks. **ResearchGate**, 2023.

KIM, H. -S.; JUNG, J.; HWANG, R.; PARK, S. -C.; LEE, S. -J.; TAE, G.; LEE, B. -W. Classification of PRPD pattern in cast-resin transformers using CNN and Implementation of Explainable AI (XAI) with Grad-CAM. **IEEE Access**, p. 1, 2024.

KIM, I.; KIM, W. Development and validation of a data-driven fault detection and diagnosis system for chillers using machine learning algorithms, **Energies**, v. 14, a. 7, p. 1945, 2021.

- KIM, K.; AND JEONG, J. Multi-layer Stacking Ensemble for Fault Detection Classification in Hydraulic System. 2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC), Crete, Greece, p. 341-346, 2022.
- KINGMA, D. P.; BA, J. L. Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), 2015.
- KINGMA, D. P.; WELLING, M. Auto-Encoding Variational Bayes. **arXiv**, 2014.
- KIV, A.; BRYUKHANOV, A.; SOLOVIEV, V.; BIELINSKYI, A.; KAVETSKYY, T.; DYACHOK, D.; DONCHEV, I.; LUKASHIN, V. Complex Network Methods for Plastic Deformation Dynamics in Metals. **Dynamics**, v. 3, p. 34-59, 2023.
- KLINGENBERG, B. Regression models for binary time series with gaps, **Computational Statistics & Data Analysis**, v. 52, a. 8, p. 4076-4090, 2008.
- KOHYARNEJADFARD, I.; ALOISE, D.; DAGENAIS, M. R.; SHAKERI, M. A Framework for Detecting System Performance Anomalies Using Tracing Data Analysis. **Entropy**, v. 23, p. 1011, 2021.
- KOKANE, P.; BAGAVATHI SIVAKUMAR, P. Online Model for Suspension Faults Diagnostics Using IoT and Analytics. **Advances in Intelligent Systems and Computing**, p. 145–154, 2018.
- KORDNOORI, S.; KOBARI, M. S.; MOSTAFAEI, H. Predict the spread of COVID-19 in Iran with a SEIR model. **IPTEK**, v. 32, a. 1, 2021.
- KRISHNAN, R.; VIRANI, S. A Systems Approach towards Developing a Diagnostic System for Complex Robots. Wiley, INCOSE International Symposium, v. 28, a. 1, p. 1682-1690, 2018.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. **Commun. Association for Computing Machinery**, v. 60, a. 6, p. 84–90, 2017.
- KUHN, M. The caret Package. **Journal of Statistical Software**. v. 28, 2012.
- KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. New York: Springer, 2013.
- KUHN, M.; JOHNSON, K. **Feature Engineering and Selection: A Practical Approach for Predictive Models**, 1 ed., Chapman and Hall/CRC, 2019.

Kumar, A. Vision Transformer Compression with Structured Pruning and Low Rank Approximation. **Cornell University**, 2022.

KUMAR, A.; TSENG, F.; GUO, Y.; CHINNAM, R. B. Hidden-Markov model based sequential clustering for autonomous diagnostics. 2008 IEEE International Joint Conference on Neural Networks. IEEE World Congress on Computational Intelligence, Hong Kong, China, p. 3345-3351, 2008.

KUMAR, S.; SINHA, S.; KOJIMA, T.; YOSHIDA, H.; Development of parameter based fault detection and diagnosis technique for energy efficient building management system. **Energy Conversion and Management**, v. 42 p. 7, p. 833–854, 2001.

KUNFT, A.; KATSIFODIMOS, A.; SCHELTER, S.; BREß, S.; RABL, T.; MARKL, V. An intermediate representation for optimizing machine learning pipelines. **Proceedings of the VLDB Endowment**, v. 12, a. 11, p. 1553-1567, 2019.

KUNZER, B. & BERGES, M. & DUBRAWSKI, A. The Digital Twin Landscape at the Crossroads of Predictive Maintenance, **Machine Learning and Physics Based Modeling**, 2022.

KWON, J. H.; KIM, E. J. Failure Prediction Model Using Iterative Feature Selection for Industrial Internet of Things. **Symmetry**, v. 12, p. 454, 2020.

LAGOGIANNIS, I.; MEISSEN, F.; KAISSIS, G.; RUECKERT, D. Unsupervised Pathology Detection: A Deep Dive Into the State of the Art. **Cornell University**, 2023.

LAZAROU, C. Autoencoding Generative Adversarial Networks. **Cornell University**, 2010.

LECUN, Y.; BOTTOU, L.; ORR, G. B.; MÜLLER, K. R. Efficient BackProp. In **Neural Networks: Tricks of the Trade**, p. 9-50, Springer, 2012.

LEE, S. -H.; PARK, C. -M. Characteristics of Peak and Cliff in Branch Length Similarity Entropy Profiles for Binary Time-Series and Their Application. **IEEE Access**, v. 10, p. 107259-107268, 2022.

LEE, S. -H.; PARK, C. -M. Park, A New Measure to Characterize the Self-Similarity of Binary Time Series and its Application. **IEEE Access**, v. 9, p. 73799-73807, 2021.

LEE, S.; LIM, J.; KIM, J.; IM, S.; SON S.; WON, S. CIS FBM Classification Using Machine Learning Algorithm and Specific Defect Prediction Analysis. 2023 34th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), Saratoga Springs, NY, USA, p. 1-4, 2023.

LEE, S.; LIM, J.; KIM, J.; IM, S.; SON, S.; WON, S. CIS FBM Classification Using Machine Learning Algorithm and Specific Defect Prediction Analysis. 2023 34th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), Saratoga Springs, NY, USA, p. 1-4, 2023.

LEE, S.-H.; PARK, C.-M. Novel Features for Binary Time Series Based on Branch Length Similarity Entropy. **Entropy**, v. 23, p. 480, 2021.

LEVER, J.; KRZYWINSKI, M.; ALTMAN, N. Model selection and overfitting. **Nature Portfolio**, v. 13, a. 9, p. 703-704, 2016.

LI, C.; KWOH, C. K.; LI, X.; MO, L.; YAN, R. Rotating Machinery Fault Diagnosis Based on Multi-sensor Information Fusion Using Graph Attention Network. 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, Singapore, p. 678-683, 2022.

LI, C.; MO, L.; YAN, R. Rolling Bearing Fault Diagnosis Based on Horizontal Visibility Graph and Graph Neural Networks. 2020 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD), Xi'an, China, p. 275-279, 2020.

LI, C.; XIONG, J.; ZHU, X.; ZHANG Q.; WANG, S. Fault Diagnosis Method Based on Encoding Time Series and Convolutional Neural Network. **IEEE Access**, v. 8, p. 165232-165246, 2020.

LI, D.; HU, G.; SPANOS, C. J. A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis, **Energy Build**, v. 128, p. 519–529, 2016.

LI, G.; HU, Y.; LIU, J.; FANG, X.; KANG, J. Review on fault detection and diagnosis feature engineering in building heating, ventilation, air conditioning and refrigeration systems. **IEEE Access**, v. 9, p. 2153–2187, 2021.

LI, J.; IZAKIAN, H.; PEDRYCZ, W.; JAMAL, I. Clustering-based anomaly detection in multivariate time series data, **Applied Soft Computing**, v.100, p.106919, 2021.

LI, X.; LIN, L.; LIU, Q.; KONG, Z.; DING, L. Topology Identification of Weighted Networks Via Binary Time Series From Propagation Dynamics. **IEEE Transactions on Computational Social Systems**, v. 10, a. 1, p. 24-35, 2023.

LIANG, H.; SUN, X.; SUN, Y.; GAO, Y. Text feature extraction based on deep learning: a review. **EURASIP Journal on Wireless Communications and Networking**, v. 211, 2017.

LIN Y. -B.; LIN, Y. -W. SensorTalk: Extending the Life for Redundant Electrical Conductivity Sensor. **IEEE Internet of Things Journal**, v. 9, a. 17, p. 16619-16630, 2022.

LIU, Y.; CHEN, Z.; WEI, L.; WANG X.; LI, L. Braking Sensor and Actuator Fault Diagnosis With Combined Model-Based and Data-Driven Pressure Estimation Methods, **IEEE Transactions on Industrial Electronics**, v. 70, a. 11, p. 11639-11648, 2023.

LIU, Y.; WANG, Y.; ZHANG, J. **New Machine Learning Algorithm: Random Forest**; IN: LIU, B. MA, M.; CHANG, J. Information Computing and Applications. Berlin, Heidelberg. Springer, p. 246-252, 2012.

LIU, Z.; WANG, W.; DING, G.; WU, Q.; WANG, X. Topology Sensing of Non-Collaborative Wireless Networks With Conditional Granger Causality. **IEEE Transactions on Network Science and Engineering**, v. 9, a. 3, p. 1501-1515, 2022.

LO, S. Binary prediction based on weighted sequential mining method. The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), Compiegne, France, p. 755-761, 2005.

LOK, L. K.; HAMEED, V. A.; RANA, M. E.; Hybrid machine learning approach for anomaly detection. **Indonesian journal of electrical engineering and computer science**, v. 27, a. 2, 2022.

LU, X.; FEI, W. Speaker recognition through nonstationary vector AR model. 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Shanghai, China, p. 462-466, 2011.

- LU, Y. A simple parameter-driven binary time series model. **Journal of Forecasting**, v. 39, a. 2, p. 187-199. 2020.
- LUO, M.; YAN, H. C.; HU, B.; ZHOU, J. H.; PANG, C. K. A data-driven two-stage maintenance framework for degradation prediction in semiconductor manufacturing industries. **Computers & Industrial Engineering**, v. 85, p. 414-422, 2015.
- LUO, W.; HU, T.; YE, Y.; ZHANG, C.; WEI, Y. A hybrid predictive maintenance approach for CNC machine tool driven by Digital Twin. **Robotics and computer-integrated manufacturing**, v. 65, p. 101974, 2020.
- LUO, Y.; BADDOUR, N.; LIANG, M. Effects of gear center distance variation on time varying mesh stiffness of a spur gear pair. **Engineering Failure Analysis**, v. 75, p. 37-53, 2017.
- MA, C.; CHEN, H.; LAI, Y.; ZHANG, H. Statistical inference approach to structural reconstruction of complex networks from binary time series. **Physical Review E**, v. 97, 2019.
- MA, M.; HAN, L.; ZHOU, C. BTAD: A binary transformer deep neural network model for anomaly detection in multivariate time series data', **Advanced Engineering Informatics**, v. 56, p. 101949, 2023.
- MA, M.; REN, A.; TYLER, C. W.; KARANDIKAR, J.; GÓMEZ, M. A.; SHI, T.; SCHMITZ, T. L. Integration of discrete-event dynamics and machining dynamics for machine tool: modeling, analysis and algorithms. **Cornell University**, 2022.
- MA, W.; TAN, L.; FENG, H.; MA, S.; CAO, D.; YIN, C. A Data-Driven LSTM Soft Sensor Model Based on Bayesian Optimization for Hydraulic Pressure Measurement of Excavator. **IEEE Sensors Journal**, v. 23, a, 21, p. 25749–25759, 2023.
- MAHARA, T.; JOSEPHINE, V. L. H.; SRINIVASAN, R.; PRAKASH, P.; ALGARNI, A. D.; VERMA, O. P. Deep vs. Shallow: A Comparative Study of Machine Learning and Deep Learning Approaches for Fake Health News Detection. **IEEE Access**, v. 11, p. 79330-79340, 2023.
- MAHMOUD, S. M.; LOTFI, A.; LANGENSIEPEN, C. Occupancy Pattern Extraction and Prediction in an Inhabited Intelligent Environment Using NARX Networks. 2010 Sixth

International Conference on Intelligent Environments, Kuala Lumpur, Malaysia, p. 58-63, 2010.

MALHOTRA, P.; VIG, L.; SHROFF, G.; AGARWAL, P. Long Short Term Memory Networks for Anomaly Detection in Time Series. Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2015), 2015.

MANSOURI, T.; VADERA, S. A Deep Explainable Model for Fault Prediction Using IoT Sensors, **IEEE Access**, v. 10, p. 66933-66942, 2022.

MARMARELIS, V. Z.; ZANOS, T. P.; COURELLIS, S. H.; BERGER, T. W. Boolean modeling of neural systems with point-process inputs and outputs. Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), p. 2114-21147. 2006.

MARTINS, E.; GALEGAL, N. V. Detecção de fraudes no segmento de crédito financeiro utilizando aprendizado de máquina: uma revisão da literatura. **eTECH**, v.15, a. 3, 2022.

MATETIĆ, I.; ŠTAJDUHAR, I.; WOLF, I.; LJUBIC, S. A review of data-driven approaches and techniques for fault detection and diagnosis in hvac systems. **Sensors**, v. 23, a.1, 2023.

MAUBRAY, J. **Reliability-Centered Maintenance**. Oxford: Butterworth Heinemann, 1997.

MAURYA, S.; SINGH, V.; VERMA N. K.; MECHEFSKE, C. K. Condition-Based Monitoring in Variable Machine Running Conditions Using Low-Level Knowledge Transfer With DNN. **IEEE Transactions on Automation Science and Engineering**, v. 18, a. 4, pp. 1983-1997, 2021.

MCKINNEY, W. Pandas: a Foundational Python Library for Data Analysis and Statistics. **Python High Performance Science Computer**. 2011.

MEESAD, P.; BOONMATHAM, S. A combination of machine learning based natural language processing with technical analysis for stock trading. **Indonesian journal of electrical engineering and computer science**, v. 30, a. 1, 2023.

MITCHELL, T. M. **Machine Learning**. Portland: McGraw-Hill, 1997.

MOBLEY, R. K. **An introduction to predictive maintenance.** Butterworth: Heinemann, 2002.

MOGHADDAM, D. E. P.; MUGULI, A.; AAZHANG, B. A Novel Cardiac Arrhythmia Classification Method Using Visibility Graphs and Graph Convolutional Network, 2023 57th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 2023, p. 866-870, 2020.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning.** Cambridge: MIT press, 2018.

MOLNAR, C. **Interpretable machine learning: A guide for making black box models explainable.** 2^a. ed. Morrisville: Lulu 2022.

MONTAVON, G.; SAMEK, W.; MÜLLER, K.; Methods for interpreting and understanding deep neural networks. **Digital Signal Processing**, v. 73, p. 1-15, 2018.

MOREIRA, G. R.; LAHR, G. J. G.; BOAVENTURA, T.; SAVAZZI, J. O.; CAURIN, G. A. P. Online prediction of threading task failure using Convolutional Neural Networks. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, p. 2056-2061, 2018.

MOSCONI, R.; SERI, R. Non-causality in bivariate binary time series. **Journal of Econometrics**, p. 132, 2006.

MOSIN, V.; STARON, M.; TARAKANOV, Y.; DURISIC, D. Comparing autoencoder-based approaches for anomaly detection in highway driving scenario images. **Springer Nature**, p. 4, a.12, 2022.

MUDITA; GUPTA, D. Prediction of Sensor Faults and Outliers in IoT Devices, 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, p. 1-5, 2021.

MUJIB, A.; DJATNA, T. Ensemble Learning for Predictive Maintenance on Wafer Stick Machine Using IoT Sensor Data. 2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA), Bogor, Indonesia, p. 1-5, 2020.

MU-QIN, T.; YU-FEI, L.; XIU-FEN, L. Research on Intelligent Diagnosis for Early Faults in Three-Phase Equipment. 2009 International Conference on New Trends in Information and Service Science, Beijing, China, p. 162-165, 2009.

MURPHY, K. P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012.

NADIYA, I.; WIDYANINGSIH, Y.; SARWINDA, D. Parameter estimation for binary time series using partial likelihood. **Journal of Physics: Conference Series**, p.1725, 2021.

NAIR, V.; HINTON, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10), p. 807–814, 2010.

NAMUDURI, S.; NARAYANAN, B.; DAVULURU, V.; BURTON, L.; Bhansali, S. Review - Deep Learning Methods for Sensor Based Predictive Maintenance and Future Perspectives for Electrochemical Sensors. **Journal of The Electrochemical Society**, v. 167, 2020.

NANDI, A.; KAFASHAN, M.; CHING, S. Control Analysis and Design for Statistical Models of Spiking Networks. **IEEE Transactions on Control of Network Systems**, p. 1, 2017.

NASA. **RCM Guide - Reliability-Centered Maintenance Guide**. [S.I.]: National Aeronautics and Space Administration, 2008.

NATEKIN, A.; KNOLL, A. Gradient Boosting Machines, A Tutorial. **Frontiers in neurorobotics**, v. 7, a. 21, 2013.

NATWSHA, B. V.; GUDDETI, R. M. R. Fog-Based Intelligent Machine Malfunction Monitoring System for Industry 4.0. **IEEE Transactions on Industrial Informatics**. v. 17, a. 12, p. 7923-7932, 2021.

NEWAY; NEWAY CNC EQUIPMENT (SUZHOU) CO. LTD. Vertical machine centers VM Séries. Suzhou, China, 2023.

NG, A. Y. Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance. Proceedings of the 21st International Conference on Machine Learning (ICML-04), 2004.

NGUYEN, K. T. P.; MEDJAHER, K. A new dynamic predictive maintenance framework using deep learning for failure prognostics, **Reliability Engineering & System Safety**, v. 188, p. 251-262, 2019.

NGUYEN, T. D.; NGUYEN, T.; NGUYEN, T. H.; NGUYEN, K.; LE NGUYEN, P. Joint Optimization of Charging Location and Time for Network Lifetime Extension in WRSNs. **IEEE Transactions on Green Communications and Networking**, v. 6, a. 2, p. 1186-1197, 2022.

NIE, B.; XU, J.; ALTER, J.; CHEN, H.; SMIRNI, E. Mining Multivariate Discrete Event Sequences for Knowledge Discovery and Anomaly Detection, p. 552-563, 2020.

NISHINO, Y.; MAEKAWA, T.; HARA, T. Detecting Repetitive Human Actions by Neural Networks Trained on Composite Data Only. 2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Atlanta, GA, USA, p. 188-193, 2023.

NOVIA, R.; PRASETYOWATI, S. S.; SIBARONI, Y. Identify user behavior based on the type of tweet on twitter platform using gaussian mixture model clustering. **Journal of computer system and informatics**, v. 3, a. 4, 2022.

NOWLAN, F. S.; HOWARD F. H. **Reliability-Centered Maintenance**. Department of Defense, Washington, D.C. 1978.

NYBERG, H. Predicting bear and bull stock markets with dynamic binary time series models. **Journal of Banking & Finance**, v. 37, a. 9, p. 3351-3363, 2013.

OCZERETKO, E.; BOROWSKA, M.; BRZOZOWSKA, E.; PAWLINSKI, B.; BORUSIEWICZ, A. GAJEWSKI, Z. Walsh-Hadamard spectral analysis of signals representing bioelectrical activity of the reproductive tract in pigs. 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE), Belgrade, Serbia, p. 1-5, 2015.

OEHMCKE, S.; ZIELINSKI, O.; KRAMER, O. Event Detection in Marine Time Series Data. KI 2015: Advances in Artificial Intelligence, p. 279-286, 2015.

OH, C.; HAN, S.; JEONG, J. Time-Series Data Augmentation based on Interpolation. **Procedia Computer Science**, v. 175, p. 64-71, 2020.

ORRÙ, P. F.; ZOCCHEDDU, A.; SASSU, L.; MATTIA, C.; COZZA, R.; ARENA, S. Machine learning approach using mlp and svm algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. **Sustainability**, v. 12, p. 4776, 2020.

ÖZTÜRK, E. Development of a hybrid model for large-scale plant RUL prediction based on data and physical models. 15th World congress on computational mechanics, 2022.

PADMANABH, K.; AL-RUBAIE, A.; ALJASMI, A. A. A. A. Health Estimation and Fault Prediction of the Sensors of a HVAC System. 2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS), BALI, Indonesia, p. 87-92, 2022.

PADMANABH, K.; AL-RUBAIE, A.; DAVIES, J.; CLARKE, S. S.; ALJASMI, A. A. A. A. Fault Prediction in HVAC Chillers by Analysis of Internal System Dynamics. 2021 International Conference on Smart Applications, Communications and Networking (SmartNets), Glasgow, United Kingdom, p. 1-6, 2021.

PAGE, M. J.; MCKENZIE J. E.; BOSSUYT, P. M.; BOUTRON, I.; HOFFMANN, T. C.; MULROW, C. D.; SHAMSEER, L.; TETZLAFF, J. M.; AKL, E. A.; BRENNAN, S. E.; CHOU, R.; GLANVILLE, J.; GRIMSHAW, J. M.; HRÓBJARTSSON, A.; LALU, M. M.; LI, T.; LODER, E. W.; MAYO-WILSON, E.; MCDONALD, S. MCGUINNESS, L. A.; STEWART, L. A.; THOMAS, J.; TRICCO, A. C.; WELCH, V. A.; WHITING, P.; MOHER, D. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. **International Journal of Surgery** v. 88, p. 105906, 2021.

PAN, X.; LIU, Y.; FAN, M.; XIANG, S.; ZHOU, H.; CHEN, G. A multi-level representation learning method for the classification with emerging new classes on power event monitoring data. **Journal of Physics: Conference Series**, p. 2232, 2022.

PANCHAL, S. M.; UMA, S. V. Assessing effectiveness of exercised variants of machine learning techniques. **International journal of innovative technology and exploring engineering**, v. 9, a. 4, p. 3259-3267, 2020.

PANG X. Modeling Heterogeneity and Serial Correlation in Binary Time-Series Cross-sectional Data: A Bayesian Multilevel Model with AR(p) Errors. **Political Analysis**, v. 18, a. 4, p.470-498, 2010.

PAOLANTI, M.; ROMEO, L.; FELICETTI, A.; MANCINI, A.; FRONTONI, E.; LONCARSKI, J.; Machine Learning approach for Predictive Maintenance in Industry 4.0, 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), Oulu, Finland, p. 1-6, 2018.

PAPACHARALAMPOUS, G.; TYRALIS, H.; KOUTSOYIANNIS, D. Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes. **Stochastic Environmental Research and Risk Assessment**, v. 33, a. 2, p. 481–514, 2019.

PATEL, D.; NGUYEN, L. M.; RANGAMANI, A.; SHRIVASTAVA, S.; KALAGNANAM, J. ChieF: A Change Pattern based Interpretable Failure Analyzer. 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, p. 1978-1985, 2018.

PATEL, D.; ZHOU, N.; SHRIVASTAVA, S.; KALAGNANAM, J. Doctor for Machines: A Failure Pattern Analysis Solution for Industry 4.0. 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, p. 1614-1623, 2020.

PAUL, E.; MAITY, A.; MAITI, R. Bayesian comparative study on binary time series. **Journal of Statistical Computation and Simulation**, v. 88, p. 1-16, 2018.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: machine learning in python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PEREIRA, F. H.; MELANI, A. H. D. A.; KASHIWAGI, F. N.; ROSA, T. G. D.; SANTOS, U. S. D.; SOUZA, G. F. M. D. Imperfect Preventive Maintenance Optimization with Variable Age Reduction Factor and Independent Intervention Level. **Applied. Sciences**, v. 13, p. 10210, 2023.

PETERS, Ralph W. **Reliable Maintenance Planning, Estimating, and Scheduling**. [S.I.]: Gulf Professional Publishing, 2015.

PRINCZ, G.; SHALOO, M.; EROL, S. Anomaly Detection in Binary Time Series Data: An unsupervised Machine Learning Approach for Condition Monitoring, **Procedia Computer Science**, v. 232, p. 1065-1078, 2024.

PUTATUNDA, S. Care2Vec: A Deep learning approach for the classification of self-care problems in physically disabled children. **Neural Computer & application**, 2018.

QIAO, Y.; WU, K.; JIN, P. Efficient Anomaly Detection for High-Dimensional Sensing Data With One-Class Support Vector Machine. **IEEE Transactions on Knowledge and Data Engineering**, v. 35, a. 1, p. 404–417, 2023.

QIN, A.; MAO, H.; ZHONG, J.; HUANG, Z.; LI, X. Generalized Transfer Extreme Learning Machine for Unsupervised Cross-Domain Fault Diagnosis With Small and Imbalanced Samples. **IEEE Sensors Journal**, v. 23, a. 14, p. 15831-15843, 2023.

QIN, S. J. Survey on data-driven industrial process monitoring and diagnosis. **Annual Reviews in Control**, v. 36, a. 2, p. 220-234, 2012.

QIU, L.; CHINCHILLI, V. M.; LIN, L. Interpretable Deep Representation Learning from Temporal Multi-view Data. Cornell University, 2020.

QIU, S.; CUI, X.; PING, Z.; SHAN, N.; LI, Z.; BAO, X.; XU, X. Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review. **Sensors**, v. 23, p. 1305, 2023.

QUINN, A.; SIMMONS, M.; SPIVAK, B.; BERGMEIR, C. RNN-BOF: A Multivariate Global Recurrent Neural Network for Binary Outcome Forecasting of Inpatient Aggression. 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, p. 1-8, 2022.

RAHHAL, J. S.; ABUALNADI, D. IOT Based Predictive Maintenance Using LSTM RNN Estimator. 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Istanbul, Turkey, p. 1-5, 2020.

RAJASEKAR, S. Prediction of Component Level Degradation in a Hydraulic Rig using Machine Learning Methods. 2023 15th International Conference on Developments in eSystems Engineering (DeSE), Baghdad & Anbar, Iraq, p. 351-356, 2023.

- RAMASSO, E.; GOURIVEAU, R. Prognostics in switching systems: Evidential markovian classification of real-time neuro-fuzzy predictions. 2010 Prognostics and System Health Management Conference, Macao, China, p. 1-10, 2010.
- RAMU, K.; NARAYANAN, K. Predictive maintenance using RNN and LSTM models. **Journal of Theoretical and Applied Information Technology**, v. 101, a. 11, 2023.
- RASCHKA, S. **Python Machine Learning**. 2^a. ed. Birmingham: Packt Publishing 2017.
- RISH, I. **An Empirical Study of the Naïve Bayes Classifier**. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence. v. 3, 2001.
- RIVEST, F.; KOHAR, R. A New Timing Error Cost Function for Binary Time Series Prediction. **IEEE Transactions on Neural Networks and Learning Systems**, v. 31, a. 1, p. 174-185, 2020.
- ROCA, J.; GONZÁLEZ-MANTEIGA, W.; FEBRERO-BANDE, M.; PRADA-SÁNCHEZ, J.; CADARSO-SUÁREZ, C. Predicting binary time series of SO₂ using generalized additive models with unknown link function. **Environmetrics**, v. 15, p. 729 – 742, 2004.
- ROCCHETTA, R.; GAO, Q.; MAVROEIDIS, D.; PETKOVIC, M. A robust model selection framework for fault detection and system health monitoring with limited failure examples: Heterogeneous data fusion and formal sensitivity bounds, **Engineering Applications of Artificial Intelligence**, v. 114, 2022.
- RODRIGUES M. W.; ZÁRATE, L. E. Time Series Analysis Using Synthetic Data for Monitoring the Temporal Behavior of Sensor Signals. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, p. 453-458, 2019.
- ROKACH, L.; MAIMON, O. **Data Mining with Decision Trees: Theory and Applications**. World Scientific Publishing Co., Inc., Edge, New Jersey, 2008.
- RONDEAU, V.; BERHANE, K.; THOMAS D. C. A three-level model for binary time-series data: the effects of air pollution on school absences in the Southern California Children's Health Study. **Statistics in medicine**, v. 24, a. 7, p.1103-15, 2005.
- RUDER, S. An Overview of Gradient Descent Optimization Algorithms. **arXiv**, 2016.

SADRABADI, A. N.; ZNJIRCHI, S. M.; ABADI, H. Z. A.; HAJIMORADI, A. An optimized K-Nearest Neighbor algorithm based on Dynamic Distance approach. **2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)**, Mashhad, Iran, p. 1-7, 2020.

SAE. **JA1012 - A Guide to Reliability-Centered Maintenance (RCM) Standard**.

SAEZ, M.; MATORANA, F.; BARTON, K.; TILBURY, D. M. (2020, January 1). Context-Sensitive Modeling and Analysis of Cyber-Physical Manufacturing Systems for Anomaly Detection and Diagnosis. **Institute of Electrical and Electronics Engineers**, v. 17, a. 1, p. 29-40, 2020.

SAHAL, R.; BRESLIN, J. G.; ALI, M. I. Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. **Journal of Manufacturing Systems**, v. 54, p. 138-151, 2020.

SAKURADA, M.; YAIRI, T. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. 2Nd Workshop on Machine Learning for Sensory Data Analysis MLSDA'14, p. 4-1, New York, NY, USA, 2014.

SALEH, M.; ESA, Y.; ONUORAH, N.; MOHAMED, A. A. Optimal microgrids placement in electric distribution systems using complex network framework. 2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA), San Diego, CA, USA, p. 1036-1040, 2017.

SAMATAS, G. G.; MOUMGIAKMAS, S. S.; PAPAKOSTAS, G. A. Predictive Maintenance -- Bridging Artificial Intelligence and IoT. **Cornell University**. 2021.

SAMINE, S.; ZEMZAMI, M.; HMINA, N.; LAGACHE, M.; BELHOUIDEG, S. Using machine learning and deep learning in additive manufacturing. 2023 9th International Conference on Optimization and Applications (ICOA), Abu Dhabi, United Arab Emirates, p. 1-5, 2023.

SARKER, I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions. **SN Computer Science**, v. 2, n. 3, p. 160, 2021.

SATINET, C.; FOUSS, F. A Supervised Machine Learning Classification Framework for Clothing Products' Sustainability. **Sustainability**, v. 14, n. 3, p. 1334, 2022.

SCHMIDL, S.; WENIG, P.; PAPENBROCK, T. Anomaly Detection in Time Series: A Comprehensive Evaluation. **Proceedings of the VLDB Endowment**, v. 15, a. 9, p. 1779 - 1797, 2022;

SCHMIDL, S.; WENIG, P.; PAPENBROCK, T. Anomaly detection in time series: a comprehensive evaluation. **Proceedings of the VLDB Endowment**, v. 15, a. 9, p. 1779–1797, 2022.

SENGOTTAIYAN, N.; VIJAYAKUMAR, J.; KUMAR, H. S.; KALYANASUNDARAM, P. MARKKANDEYAN, S. Maximize the Production Process by Using a Novel Hybrid Model to Predict the Failure of Machine. 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, p. 1-10, 2022.

SEZER, E.; ROMERO, D.; GUEDEA, F.; MACCHI, M.; EMMANOUILIDIS, C. An Industry 4.0-Enabled Low Cost Predictive Maintenance Approach for SMEs, 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Stuttgart, Germany, p. 1-8, 2018.

SEZER, E.; ROMERO, D.; GUEDEA, F.; MACCHI, M.; EMMANOUILIDIS, C. An industry 4.0-enabled low cost predictive maintenance approach for SMEs: a use case applied to a cnc turning center. IEEE International conference on engineering, technology and innovation (ICE/ITMC), p. 1–8, 2018.

SHABANI, M.; IOSIFIDIS, A. Low-rank temporal attention-augmented bilinear network for financial time-series forecasting. 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, p. 2156-2161, 2020.

SHANECHI, M. M.; CHEMALI, J. J.; LIBERMAN, M.; SOLT, K.; BROWN, E. N. A brain-machine interface for control of burst suppression in medical coma. 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, p. 1575-1578, 2013.

SHEHZAD, Z. A.; SHAIKH, M. A.; ARIZ, M.; ZAKARIYA, M.; HUSSAIN, A. IoT & ML-Based Parameter Monitoring of 3-φ Induction Motors for Industrial Application. 2023 International Conference on Emerging Power Technologies (ICEPT), Topi, Pakistan, p. 1-5, 2023.

SHIN, J. H.; JUN, H. B.; On condition based maintenance policy. **Journal of Computational Design and Engineering**, v. 2, a. 2, p. 119–127, 2015.

SHU, W.; YAO, Y.; LYU, S.; LI, J.; CHEN, H. Short isometric shapelet transform for binary time series classification. **Knowledge and Information Systems**, v. 63, p. 2023–2051, 2021.

SILVA, A.; SOUZA, G. F. M. Prognosis Smart System AI-based Applied to Equipment Health Monitoring in 4.0 Industry Scenario. 2021 Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, p. 1-6, 2021.

SILVA, L.; PERES, S.; BOSCAROLI, C. **Introdução à Mineração de Dados com aplicações em R**. 1^a. ed. Rio de Janeiro: Elsevier, 2016.

SILVESTRIN, L. P.; HOOGENDOORN, M.; KOOLE, G. A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive Maintenance. **2019 IEEE Symposium Series on Computational Intelligence (SSCI)**, Xiamen, China, pp. 760-767, 2019

SINGH, M.; SAHOO K. S.; GANDOMI, A. H. An Intelligent-IoT-Based Data Analytics for Freshwater Recirculating Aquaculture System. **IEEE Internet of Things Journal**, v. 11, a. 3, p. 4206-4217, 2024.

SMOLOV, V.; ROZVADOVSKIY, A. Application of the Arduino platform for recording wind waves. **Morskoy gidrofizicheskiy zhurnal**, v. 36, 2020.

SOARES, S. G.; ARAÚJO, R. An on-line weighted ensemble of regressor models to handle concept drifts. **Engineering Applications of Artificial Intelligence**, v. 37, p. 392-406, 2015.

SONG, S.; LI, H.; Early warning signals for stock market crashes: empirical and analytical insights utilizing nonlinear methods. **EPJ Data Science**, v. 13, p. 16, 2024b.

SONG, S.; LI, H.; Time series synchronization in cross-recurrence networks: uncovering a homomorphic law across diverse complex systems. **New Journal of Physics**, v. 26, a. 1, p. 1367, 2024a.

SRIVASTAVA, A.; PARIDA, S. K. A Robust Fault Detection and Location Prediction Module Using Support Vector Machine and Gaussian Process Regression for AC

Microgrid, **IEEE Transactions on Industry Applications**, v. 58, a. 1, p. 930-939, 2022.

SRIVASTAVA, A.; PARIDA, S. K. Fault Isolation and Location Prediction using Support Vector Machine and Gaussian Process Regression for Meshed AC Microgrid. 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, p. 724-728, 2020.

SRIVASTAVA, N.; HINTON, G. E.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**, v. 15, a. 1, p. 1929-1958, 2014.

STEURTEWAGEN B.; VAN DEN POEL, D. Root Cause Analysis of Compressor Failure by Machine Learning. 2019 Petroleum and Chemical Industry Conference Europe (PCIC EUROPE), Paris, France, p. 1-5, 2019.

STEURTEWAGEN, B.; POEL, D. V. D. Adding interpretability to predictive maintenance by machine learning on sensor data, **Computers & Chemical Engineering**, v. 152, 2021.

SUE, J. A.; HASHOLZNER, R.; BRENDEL, J.; KLEINSTUBER, M.; TEICH, J. A Binary Time Series Model of LTE Scheduling for Machine Learning Prediction. 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W), Augsburg, Germany, p. 269-270, 2016.

SUN, K.; XIA, F.; LIU, J.; XU, B. V. Saikrishna and C. C. Aggarwal, Attributed Graph Force Learning. **IEEE Transactions on Neural Networks and Learning Systems**, v. 35, a. 4, p. 4502-4515, 2024.

SUN, W.; HUANG, M.; HE, Y.; LI, K. Design of tool-state monitoring system based on current method. **The Journal of Engineering**, v. 2019, a. 23, p. 9026-9030, 2019.

SUNEHAG, P.; SHAO, W.; HUTTER, M. Coding of non-stationary sources as a foundation for detecting change points and outliers in binary time-series. Proceedings of the Tenth Australasian Data Mining Conference, v. 134, p. 79-84, 2012.

SUNG, C. L.; HUNG, Y.; RITTASE, W.; ZHU, C.; JEFF WU, C. F. A Generalized Gaussian Process Model for Computer Experiments With Binary Time Series. **Journal of the American Statistical Association**, v. 115, a. 530, p. 945–956, 2020.

- SURANTHA, N.; GOZALI, I.D. Evaluation of the Improved Extreme Learning Machine for Machine Failure Multiclass Classification. **Electronics**, v. 12, p. 3501. 2023.
- SUSTO, G. A.; SCHIRRU, A.; PAMPURI, S.; MCLOONE, S.; BEGHI, A. Machine learning for predictive maintenance: A multiple classifier approach. **IEEE Transactions on Industrial Informatics**, v. 1, a. 3, p. 812–820, 2015.
- SUTSKEVER, I.; MARTENS, J.; DAHL, G.; HINTON, G. On the Importance of Initialization and Momentum in Deep Learning. In Proceedings of the 30th International Conference on Machine Learning (ICML-13), p. 1139-1147, 2013.
- SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems (NeurIPS 2014), 2014.
- SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. Cambridge: MIT press, 2018.
- TAGORE, V.; SUTRADHAR, B. C. Conditional inference in linear versus nonlinear models for binary time series. **Journal of Statistical Computation and Simulation**, v. 79, a. 7, p. 881–897, 2009.
- TAMILSELVAN, P.; WANG, P.; JAYARAMAN, R. Health diagnostics with unexampled faulty states using a two-fold classification method. 2012 IEEE Conference on Prognostics and Health Management, Denver, CO, USA, p. 1-11, 2021.
- Tang, J., & Zeng, J. (2021, June 18). Spatiotemporal gated graph attention network for urban traffic flow prediction based on license plate recognition data. **Computer-aided civil and infrastructure engineering**, v.31, a. 1, p. 3-23, 2021.
- TANUSKA, P.; SPENDLA, L.; KEBISEK, M.; ĎURIŠ, R.; STRÉMY, M. Smart anomaly detection and prediction for assembly process maintenance in compliance with industry 4.0. **Sensors**, v. 21, a. 7, p. 2376, 2021.
- TARIK, M.; MNIAI, A.; JEBARI, K. Hybrid feature selection and support vector machine framework for predicting maintenance failures. **Applied Computer Science**, v. 19, a. 2, p. 112–124, 2023.
- THEIS, L.; SHI, W.; CUNNINGHAM, A.; HUSZÁR, F. Lossy Image Compression with Compressive Autoencoders. 5th International Conference on Learning Representations (ICLR 2017), 2017.

TOBON-MEJIA, D. A.; MEDJAHER, K.; ZERHOUNI, N.; TRIPOT, G. Hidden Markov Models for failure diagnostic and prognostic. 2011 Prognostics and System Health Management Conference, Shenzhen, China, p. 1-8, 2011.

TRAN, D. A. T.; CHEN, Y.; JIANG, C. Comparative investigations on reference models for fault detection and diagnosis in centrifugal chiller systems, **Energy and Buildings**, v. 133, p. 246–256, 2016.

TRIVEDI, S.; BHOLA, S.; TALEGAONKAR, A.; GAUR, P.; Sharma, S. Predictive Maintenance of Air Conditioning Systems Using Supervised Machine Learning. 2019 20th International Conference on Intelligent System Application to Power Systems (ISAP), 2019.

Tsai, F. -K.; Chen, C. -C.; Chen, T. -F.; Lin, T. -J. Sensor Abnormal Detection and Recovery Using Machine Learning for IoT Sensing Systems. 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), Tokyo, Japan, p. 501-505, 2019.

TSCHANNEN, M.; BACHEM, O.; LUČIĆ, M. Recent Advances in Autoencoder-Based Representation Learning. **Cornell University**, 2018.

TV, V.; GUPTA, P.; MALHOTRA, P.; VIG, L.; SHROFF, G. Recurrent Neural Networks for Online Remaining Useful Life Estimation in Ion Mill Etching System. Proceedings of the Annual Conference of the PHM Society 2018, v. 10, a. 1, 2018.

UDOH, N.; UDOM, A.; OHAEGBUNEM, F. Optimal Maintenance Probabilities and Preventive Replacement Maintenance Policy for Photocopy Machines. **Journal of Reliability and Statistical Studies**, v. 14, a. 1, p. 263–284, 2021.

UHRICH, B.; HLUBEK, N.; HÄNTSCHEL, T.; RAHM, E. Using differential equations inspired machine learning for valve faults prediction. 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), Lemgo, Germany, p. 1-8, 2023.

UPASANE, S. J.; HAGRAS, H.; ANISI, M. H.; SAVILL, S.; TAYLOR, I.; MANOUSAKIS, K. A Type-2 Fuzzy-Based Explainable AI System for Predictive Maintenance Within the Water Pumping Industry. **IEEE Transactions on Artificial Intelligence**, v. 5, a. 2, p. 490-504, 2024.

UPPAL, M.; GUPTA, D.; MAHMOUD, A.; ELMAGZOUB, M.A.; SULAIMAN, A.; RESHAN, M.S.A.; SHAIKH, A.; JUNEJA, S. Fault Prediction Recommender Model for IoT Enabled Sensors Based Workplace. **Sustainability**, v. 15, p. 1060, 2023.

VACHTSEVANOS, G. J.; LEWIS, F.; HESS, A.; WU, B. **Intelligent Fault Diagnosis and Prognosis for Engineering Systems**, John Wiley & Sons Inc, United States of

VAIDYAN, V. M.; TYAGI, A. Towards Quantum Artificial Intelligence Electromagnetic Prediction Models for Ladder Logic Bombs and Faults in Programmable Logic Controllers. 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), Chennai, India, p. 1-6, 2022.

VAN ENGELEN, J. E.; HOOS, H. H. A survey on semi-supervised learning. **Machine Learning**, v. 109, n. 2, p. 373-440, 2020.

VELIBEYOGLU, I.; NOH, H. Y.; POZZI, M. A graphical approach to assess the detectability of multiple simultaneous faults in air handling units, **Energy and Buildings**, v. 184, p. 275-288, 2019.

VELIBEYOGLU, I.; NOH, H. Y.; POZZI, M. A graphical approach to assess the detectability of multiple simultaneous faults in air handling units. **Energy and Buildings**, v. 184, p. 275-288, 2019.

VERSCHEURE, O.; VLACHOS, M.; ANAGNOSTOPOULOS, A.; FROSSARD, P.; BOUILLET, E. YU, P. S. Finding Who Is Talking to Whom, VoIP Networks via Progressive Stream Clustering, Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, p. 667-677, 2006.

VINCENT, P.; LAROCHELLE, H.; BENGIO, Y.; MANZAGOL, P. A. Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on Machine learning. p. 1096-1103, 2008.

VIVEK B. V.; KUSUMA, S. M. Effective Quality Machine Learning to Predict Industry Equipment Failures. 2023 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, p. 1-6, 2023.

VIVEK, B. V.; KUSUMA, S. M. Effective Quality Machine Learning to Predict Industry Equipment Failures. 2023 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, p. 1-6, 2023.

VOGL, G. W.; WEISS, B. A.; HELU, M. A Review of Diagnostic and Prognostic Capabilities and Best Practices for Manufacturing, **Journal of Intelligent Manufacturing**, v. 30, p. 79–95. 2019.

WAHID, A.; BRESLIN, J. G.; INTIZAR, M. A. Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework. **Applied Sciences**, v. 12, p. 4221, 2022.

WAHID, A.; BRESLIN, J. G.; INTIZAR, M. A. Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework. **Applied Sciences**, v. 12, a. 9, p. 4221, 2022.

WAN, J.; TANG, S.; LI, D.; WANG, S.; LIU, C.; ABBAS, H.; VASILAKOS, A. V. A manufacturing big data solution for active preventive maintenance. **IEEE Transactions on Industrial Informatics**, v.13, p. 2039–2047, 2017.

WANG H.; ZHANG, X. Binary Time-Series Query Framework for Efficient Quantitative Trait Association Study. IEEE 13th International Conference on Data Mining, Dallas, TX, USA, p. 777-786, 2013.

WANG Z. W. A Method for Predicting the State of Sensors and Meters Based on Improved Generative Adversarial Networks. 2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Xi'an, China, p. 327-331, 2022.

WANG, B.; HUA, Q.; ZHANG, H.; TAN, X.; NAN, Y.; CHEN, R.; SHU, X.; Research on anomaly detection and real-time reliability evaluation with the log of cloud platform, **Alexandria Engineering Journal**, v. 61, a. 9, p. 7183–7193, 2022.

WANG, C.; WEN, C.; LU, Y. A fault diagnosis method by using extreme learning machine. 2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF), p. 318-322, 2015.

WANG, D.; XIAN, X.; SONG, C. Joint Learning of Failure Mode Recognition and Prognostics for Degradation Processes. **IEEE Transactions on Automation Science and Engineering**, p. 1-13, 2023.

WANG, J.; RAN, R.; FANG, B. GNPENet: A Novel Convolutional Neural Network With Local Structure for Fault Diagnosis. **IEEE Transactions on Instrumentation and Measurement**, v. 73, p. 1-16, 2024.

- WANG, K. Intelligent predictive maintenance (ipdm) system–industr 4.0 scenario, **WIT Transactions on Engineering Sciences**, v. 113, p.259–268, 2016.
- WANG, W.; PENG, Z.; WANG, S.; LI, H.; LIU, M.; XUE, L. ZHANG, N. IFP-ADAC: A Two-stage Interpretable Fault Prediction Model for Multivariate Time Series. 2021 22nd IEEE International Conference on Mobile Data Management (MDM), Toronto, ON, Canada, p. 29-38, 2021.
- WANG, X.; WANG, C. Time Series Data Cleaning with Regular and Irregular Time Intervals. Cornell University, 2020.
- WANG, Y. B. Multi-sensor information fusion and its application overview. **Journal of Sensor World**, v. 16, a.12, p. 6–9, 2010.
- WANG, Y.; WU, M.; LI, X.; XIE, L.; CHEN, Z. Multivariate Time Series Representation Learning via Hierarchical Correlation Pooling Boosted Graph Neural Network. **IEEE Transactions on Artificial Intelligence**, p. 1-13, 2023.
- WANG, Y.; XIANG, J.; MARKERT, R.; LIANG, M. Spectral kurtosis for fault detection, diagnosis and prognostics of rotating machines: A review with applications. **Mechanical Systems and Signal Processing**, v. 66-67, p. 679-698, 2016.
- WARRIACH, E. U.; OZCELEBI T.; LUKKIEN, J. J. A Comparison of Predictive Algorithms for Failure Prevention in Smart Environment Applications, **2015 International Conference on Intelligent Environments**, Prague, Czech Republic, p. 33-40, 2015.
- WARRIACH, E.; OZCELEBI, T.; LUKKIEN, J. A Comparison of Predictive Algorithms for Failure Prevention in Smart Environment Applications, 2015 International Conference on Intelligent Environments, p. 33-40, 2015.
- WASON, R. Deep learning: Evolution and expansion. **Cognitive Systems Research**, v. 52, p. 701-708, 2018.
- WILLIAMS, J. L.; FISHER J. W.; WILLSKY, A. S. Approximate Dynamic Programming for Communication-Constrained Sensor Network Management. **IEEE Transactions on Signal Processing**, v. 55, a. 8, p. 4300-4311, 2007.

- WU, G.; LIU, F.; XU, S. Compiling method of automobile fault diagnosis information. **EDP Sciences**, v. 355, p. 03073-03073, 2022.
- WU, H.; HUANG, A.; SUTHERLAND, J. W. Avoiding Environmental Consequences of Equipment Failure via an LSTM-Based Model for Predictive Maintenance. **Procedia Manufacturing**, v. 43, p. 666-673, 2020.
- WU, R.; CUI, Y. A parameter-driven logit regression model for binary time series. **Journal of Time Series Analysis**, v. 35, 2014.
- WU, S. J.; GEBRAEEL, N.; LAWLEY, M. A.; YIH, Y. A neural network integrated decision support system for condition-based optimal predictive maintenance policy. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, v. 37, a. 2, p. 226–236, 2007.
- WU, Z.; LUO, H.; YANG, Y.; LI, P.; ZHU, X.; JI, Y.; WU, B. K-PdM: KPI-Oriented Machinery Deterioration Estimation Framework for Predictive Maintenance Using Cluster-Based Hidden Markov Model. **IEEE Access**, v. 6, p. 41676-41687, 2018.
- XINYUAN, Z.; YAOMIN, Y.; YUE, R.; XIANGXING, G.; HONGLIANG, W.; WEI, G. A Deep Learning-Based Model for Secondary Prediction on Deep-Sea Collector Plumes. 2023 IEEE International Conference on Mechatronics and Automation (ICMA), Harbin, Heilongjiang, China, p. 713-718, 2023.
- XIU, Y.; REN, X.; ZHANG, T.; CHEN, Y.; JIANG, L.; LI, D.; WANG, X.; ZHAO, L.; CHAN, - W. K. Time Labeled Visibility Graph for Privacy-Preserved Physiological Time Series Classification. 2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, p. 280-284, 2022.
- XU, G.; LIU, M.; JIANG, Z.; SHEN, W.; HUANG, C. Online Fault Diagnosis Method Based on Transfer Convolutional Neural Networks, **IEEE Transactions on Instrumentation and Measurement**, v. 69, a. 2, p. 509-520, 2020.
- XU, H.; MA, R.; YAN, L.; MA, Z. Two-stage prediction of machinery fault trend based on deep learning for time series analysis. **Digital Signal Processing**, p. 117, 2021.

XU, Z.; GUO, Y.; SALEH, J. H. Remaining useful life prediction with uncertainty quantification: development of a highly accurate model for rotating machinery. **Cornell University**, 2021.

YAGUO, L.; NAIPENG, L.; LIANG, G.; NINGBO, L.; TAO, Y.; JING, L. Machinery health prognostics: A systematic review from data acquisition to RUL prediction, **Mechanical Systems and Signal Processing**, v.104, p.799-834, 2018.

YAN, K.; MA, L.; DAI, Y.; SHEN, W.; JI, Z.; XIE, D. Cost-sensitive and sequential feature selection for chiller fault detection and diagnosis, **International Journal of Refrigeration**, v. 86, p. 401– 409, 2018.

YANG, J.; XIE, G.; YANG Y.; LI, X. A Rotating Machinery Fault Diagnosis Method for High-Speed Trains Based on Improved Deep Learning Network. 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, p. 440-444, 2018.

YARKAN, S.; ARSLAN, H. Binary Time Series Approach to Spectrum Prediction for Cognitive Radio. 2007 IEEE 66th Vehicular Technology Conference, Baltimore, MD, USA, p. 1563-1567, 2007.

YE, M. An improved CNN structure model for image classification recognition. **JCP**, v. 13, a. 12, p. 1349-1356, 2018.

YIN, R. K. **Case study research: design and methods**. 4^a. ed. Thousand Oaks, California: Sage, 2009.

YOTOV,O. B.; ALEKSIEVA-PETROVA, A. P. Analysis of Sensor Signals with Machine and Deep Learning Algorithms. 2023 International Scientific Conference on Computer Science (COMSCI), Sozopol, Bulgaria, p. 1-5, 2023.

YU, J.; Mechatronics Fault Prediction and Diagnosis Based on Multi Sensor Information Fusion, **Journal of Physics: Conference Series**, v. 1982, 2021.

YUGAPRIYA, M.; JUDESON, A. K. J. JAYANTHY, S. Predictive Maintenance of Hydraulic System using Machine Learning Algorithms. 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, p. 1208-1214, 2022.

ZAHARIA, M.; XIN, R. S.; WENDELL, P.; DAS, T.; ARMBRUST, M.; DAVE, A.; MENG, X.; ROSEN, J.; VENKATARAMAN, S.; FRANKLIN, M. J.; GHODSI, A.; GONZALEZ, J.; SHENKER, S.; STOICA, I. Apache Spark: A Unified Engine for Big Data Processing.

Communications of the ACM, v. 59, n. 11, 2016.

ZANG, Y.; FIONDELLA, L. A Network Reliability Analysis Method for Complex Systems based on Complex Network Theory. 2022 Annual Reliability and Maintainability Symposium (RAMS), Tucson, AZ, USA, p. 1-6, 2022.

ZHANG X, ZHANG G, LI Y, RAN Y, WANG H, GONG X. A novel fault diagnosis approach of a mechanical system based on meta-action unit. **Advances in mechanical engineering**, v. 11, a. 2, 2019.

ZHANG, J. ; YUAN, C.; ZENG, W.; WANG, C. Fault Detection and Isolation of Uncertain Nonlinear Parabolic PDE Systems. **Cornell University**, 2022.

ZHANG, P.; HU, W.; CAO, W.; CHEN, L.; WU, M. Multi-Fault Diagnosis of Hydraulic Systems Based on Fully Convolutional Networks. 2022 13th Asian Control Conference (ASCC), Jeju, Republic of Korea, p. 631-636, 2022.

ZHANG, Y. Support Vector Machine Classification Algorithm and Its Application. In: LIU, C.; WANG, L.; YANG, A. **Information Computing and Applications**. ICICA 2012. Communications in Computer and Information Science, v. 308, Berlin: Springer 2012.

ZHANG, Y.; ZHANG, C.; WANG, S.; CHEN, R.; TOMOVIC, M. M. Performance degradation based on importance change and application in dissimilar redundancy actuation system. **Mathematics**, v. 10, p. 843, 2022.

ZHANHONG, J.; MICHAEL, J. R.; SANTLE, C. K. S.; CHENLU, Z.; SAMAN, C.; YOUNG, M. L. A timeseries supervised learning framework for fault prediction in chiller systems, **Energy and Buildings**, v. 285, p. 112876, 2023.

ZHAO, R.; WANG, D.; YAN, R.; MAO, K.; SHEN, F.; WANG, J. Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks. **IEEE Transactions on Industrial Electronics**, v. 65, a. 2, p. 1539-1548, 2018.

ZHAO, R.; YAN, R.; CHEN, Z.; MAO, K.; WANG, P.; GAO, R. X. Deep learning and its applications to machine health monitoring, **Mechanical Systems and Signal Processing**, v. 115, p. 213-237, 2019.

ZHAO, S.; SONG, J.; ERMON, S. Towards Deeper Understanding of Variational Autoencoding Models. **Cornell University**, 2017.

ZHAO, Y. Fault Prediction Method and System of Acquisition Terminal Based on Bayesian Network Optimization Algorithm. 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), Dharwad, India, p. 1-7, 2023.

ZHAO, Y.; LI, T.; ZHANG, X.; ZHANG, C. Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future. **Renewable and Sustainable Energy Reviews**, v. 109, p. 85–101, 2019.

ZHONG, G.; HUANG, K. **Semi-supervised learning: Background, applications and future directions**. Hauppauge: Nova, 2018

ZHONG, Z.; FAN, Q.; ZHANG, J.; MA, M.; ZHANG, S.; SUN, Y.; LIN, Q.; ZHANG, Y.; PEI, D. A Survey of Time Series Anomaly Detection Methods in the AIOps Domain. Cornell University, 2023.

ZHOU, H.; SUN, G.; SHA, F.; WANG, L.; HU J.; GAO, Y. Internet financial fraud detection based on a distributed big data approach with Node2vec. **IEEE Access**, v. 9, p. 43378-43386, 2021.

ZHU, M.; CHEN, Y.; YANG C.; ZHAO, D. ATPG of Digital Electronic Systems BIST Based on D-PL Chaotic Model. 2012 Second International Conference on Intelligent System Design and Engineering Application, Sanya, China, p. 1200-1203, 2012.

ZHU, R.; PENG, W.; HAN, Y.; HUANG, C. -G. Intelligent Health Monitoring of Machine Tools Using a Bayesian Multibranch Neural Network. **IEEE Sensors Journal**, v. 22, a. 12, p. 12183-12196, 2022.

ZONTA, T.; COSTA, C. A.; RIGHI, R. R.; LIMA, M. J.; TRINDADE, E. D.; LI, G. P. Predictive maintenance in the Industry 4.0: A systematic literature review. **Computers & Industrial Engineering**, v. 150, p.106889, 2020.

ZÜFLE, M.; AGNE, J.; GROHMANN, J.; DORTOLUK, I.; KOUNEV, S. A Predictive Maintenance Methodology: Predicting the Time-to-Failure of Machines in Industry 4.0. 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), p. 1-8, 2021.

APÊNDICE I – PROTOCOLO DE REVISÃO DA LITERATURA I

Para realização da revisão sistemática da literatura sobre aprendizado de máquina aplicado à análise e detecção de falhas de máquinas e equipamentos, foi utilizada a mais recente metodologia Transparent Reporting of Systematic Reviews and Meta-Analyses (PRISMA), que é descrita neste apêndice.

SELEÇÃO DOS ARTIGOS PARA IDENTIFICAR AS APLICAÇÕES DE APRENDIZADO DE MÁQUINA EM PREDIÇÃO DE FALHAS

Com o objetivo claramente estabelecido de identificar como o aprendizado de máquina é aplicado na previsão de falhas no contexto de máquinas e equipamentos industriais e sensores, a pesquisa foi conduzida nas bases de dados Scopus, Web of Science e IEEE Xplore. Os critérios de elegibilidade focaram em artigos publicados em periódicos ou apresentados em congressos, restringindo-se somente a trabalhos escritos em inglês e não se limitou a busca a um determinado intervalo de tempo (PAGE *et al.*, 2021).

Como estratégia de busca, foram consideradas palavras-chaves e termos de pesquisa mais comumente relacionados com aprendizado de máquina, como *machine learning*, *deep learning*, *artificial intelligence* e *AI*, junto com os termos *failure prediction*, *fault prediction*, *defect prediction* e *time to failure* para identificar a literatura relevante, conforme aprestado na tabela 16, (PAGE *et al.*, 2021).

Aplicou-se critérios de exclusão para garantir a relevância dos artigos selecionados. Primeiramente, excluiu-se os artigos que não continham nenhuma das palavras-chave estabelecidas no resumo, indicando uma desconexão potencial com o tema central da pesquisa. Além disso, descartou-se os artigos que, embora estivessem relacionados a campos de estudo amplos, não abordavam especificamente o aprendizado de máquina ou seus cognatos. Isso inclui trabalhos que, apesar de se enquadrarem em temas mais amplos da inteligência artificial ou da análise de dados, não se focavam em técnicas de aprendizado de máquinas.

Por último, priorizou-se a relevância prática ao excluir artigos que não tratavam de aplicações industriais, especialmente aquelas relacionadas a máquinas,

equipamentos e sensores. Essa escolha se deu devido à especificidade do tipo de dados abordados nesta pesquisa, que busca *insights* aplicáveis no contexto industrial, onde os tipos de dados e desafios são substancialmente diferentes de outros campos.

Tabela 16 – Protocolo de revisão da literatura I

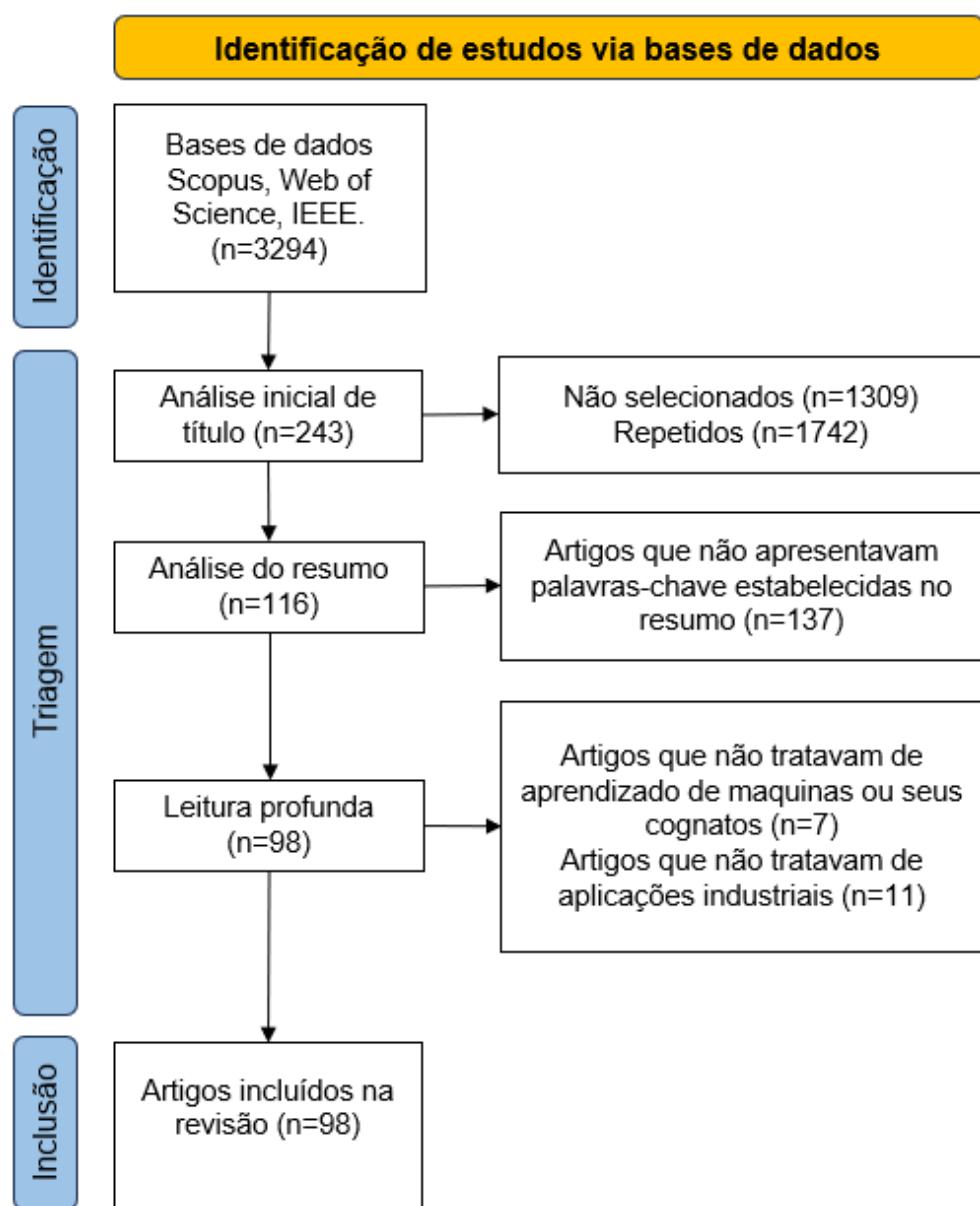
Objetivo	Identificar artigos que abordem aplicação de aprendizado de máquina em predição de falhas.
Termos de pesquisa	<p>“<i>machine learning</i>” AND “<i>failure prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>machine learning</i>” AND “<i>fault prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>machine learning</i>” AND “<i>defect prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>machine learning</i>” AND “<i>time to failure</i>” AND “<i>sensor*</i>”</p> <p>“<i>deep learning</i>” AND “<i>failure prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>deep learning</i>” AND “<i>fault prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>deep learning</i>” AND “<i>defect prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>deep learning</i>” AND “<i>time to failure</i>” AND “<i>sensor*</i>”</p> <p>“<i>artificial intelligence</i>” AND “<i>failure prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>artificial intelligence</i>” AND “<i>fault prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>artificial intelligence</i>” AND “<i>defect prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>artificial intelligence</i>” AND “<i>time to failure</i>” AND “<i>sensor*</i>”</p> <p>“<i>ai</i>” AND “<i>failure prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>ai</i>” AND “<i>fault prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>ai</i>” AND “<i>defect prediction</i>” AND “<i>sensor*</i>”</p> <p>“<i>ai</i>” AND “<i>time to failure</i>” AND “<i>sensor*</i>”</p>
Bases de dados	Scopus, Web of Science e IEEE Xplore
Tipos de publicação	Artigos publicados em periódicos ou congressos
Idioma	Inglês
Período	A pesquisa não se limitou a um período
Critérios de inclusão	Artigos sobre predição de falhas e aprendizado de máquina
Critérios de exclusão	<p>Artigos que não apresentavam nenhuma das palavras chaves no resumo; artigos que não tratavam de aprendizado de máquina ou seus cognatos;</p> <p>Artigos que não tratavam de aplicações industriais, tais como máquinas, equipamentos e sensores.</p>

Fonte: Elaborado pelo autor.

A busca resultou em 3294 artigos até o início de abril de 2024 que tratavam sobre o tema aprendizado de máquina e seus cognatos combinados individualmente com os termos *machine learning*, *deep learning*, *artificial intelligence* e *AI*. Após uma revisão dos títulos, foram identificados 1742 artigos repetidos entre as bases pesquisadas que foram descartados, bem como 1309 artigos por não serem aderentes ao escopo desta pesquisa, restando assim 243 artigos. Após a leitura do

resumo, foram selecionados 116 artigos que tratavam especificamente sobre algum aspecto de aprendizado de máquina e predição de falhas e após a uma leitura profunda destes artigos, foram identificados 98 artigos que tratam de algum aspecto sobre a aplicação de aprendizado de máquina em detecção de falhas no contexto de máquinas e equipamentos industriais e sensores, conforme descrito na Figura 34.

Figura 24 – Fluxograma de revisão da literatura I.



Fonte: Elaborado pelo autor.

ANÁLISE DA DIVERSIDADE DE ALGORITMOS APLICADO A PREDIÇÃO DE FALHAS EM MÁQUINAS E EQUIPAMENTOS.

Em um campo em constante evolução como aprendizado de máquina, a escolha dos algoritmos é importante para o sucesso de uma pesquisa. Essa escolha não apenas reflete as tendências atuais e capacidades dos métodos disponíveis, mas também revela abordagens únicas e inovadoras adotadas por pesquisadores em seus estudos. Para compreender melhor essa dinâmica, é apresentado uma tabela compilada que destaca os autores de publicações acadêmicas recentes, principais algoritmos utilizados e área de aplicação de seus trabalhos.

Os dados apresentados não apenas mostram os algoritmos amplamente adotados, como SVM, Random Forest e redes neurais, mas também destaca escolhas mais inovadoras e especializadas, refletindo a constante evolução e adaptação no campo. Cada entrada na tabela permite uma análise sobre como diferentes pesquisadores abordam problemas variados no vasto domínio do aprendizado de máquinas. A tabela 01 facilita a identificação de padrões e tendências comuns, além de promover o reconhecimento de abordagens únicas e potencialmente inovadoras neste recorte de pesquisa.

Ao realizar a revisão sistemática da literatura, a abordagem não se limitou a um período de tempo específico. Buscou-se compreender amplamente as tendências e desenvolvimentos nesta área em constante evolução. A análise dos dados revelou aspectos interessantes que destacam tanto a evolução recente quanto a distribuição temporal dos estudos.

Descrição dos anos dos estudos:

- Quantidade: 98 estudos.
- Média do Ano: 2020.63, indicando uma tendência recente nos estudos.
- Desvio padrão: 2.83, mostrando uma variação moderada nos anos.
- Ano mínimo: 2008.
- Ano máximo: 2024.

Quartis:

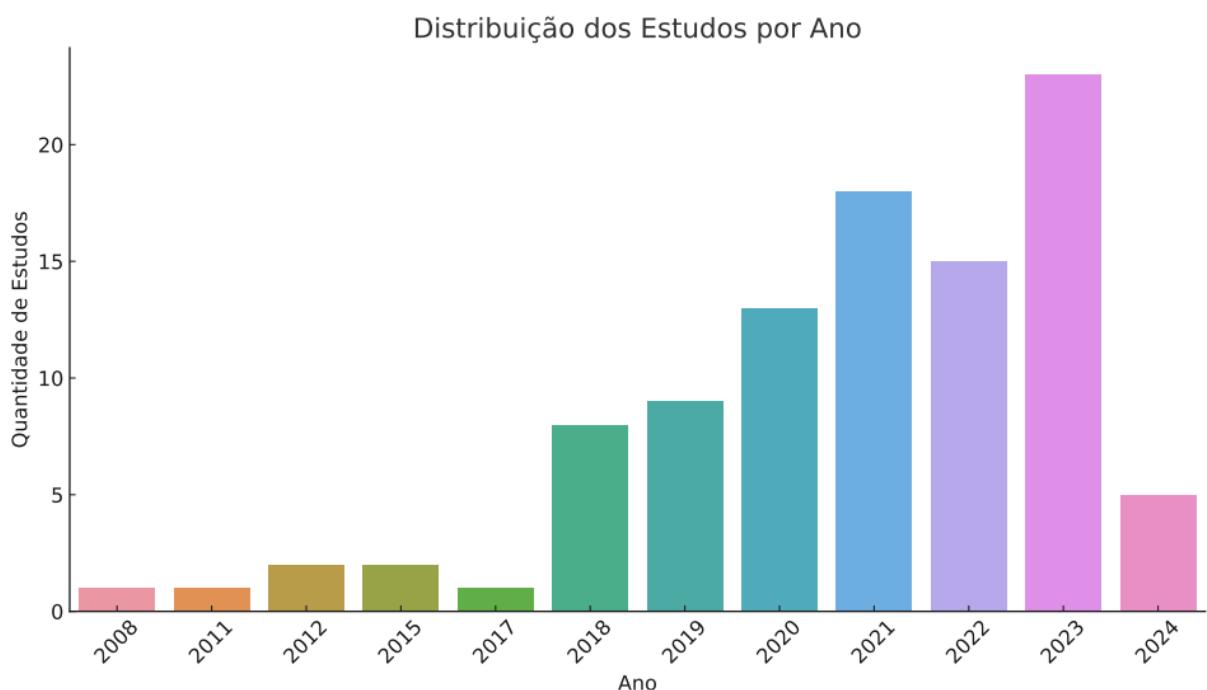
- 25% dos estudos são de 2020 ou antes.
- 50% (mediana) são de 2021 ou antes.

- 75% são de 2023 ou antes.

A análise temporal na Figura 35 mostra a distribuição dos estudos por ano.

Com base na análise temporal da distribuição dos estudos por ano pode-se observar uma tendência crescente em publicações recentes, indicando um aumento no interesse e na pesquisa nessa área de máquinas e equipamentos industriais. Um pico notável em anos recentes também sugere que o tópico se tornou particularmente relevante.

Figura 35 – Análise temporal da distribuição dos estudos.



Fonte: Elaborado pelo autor.

Também se evidencia a versatilidade do aprendizado de máquina pela ampla gama de aplicações práticas identificadas na revisão sistemática. Com 65 aplicações únicas mencionadas nos estudos, fica claro que o alcance do aprendizado de máquina se estende por diversos domínios e desafios industriais. Isso sugere uma forte tendência na utilização de machine learning para interpretar e agir sobre dados capturados por sensores, uma aplicação fundamental em muitas áreas tecnológicas.

Além disso, sistemas hidráulicos e equipamentos industriais também são frequentemente explorados, com cinco e quatro ocorrências, respectivamente. Essas aplicações destacam a relevância do aprendizado de máquina na otimização e

manutenção de sistemas complexos e equipamentos, enfatizando seu papel vital em melhorar a eficiência e a segurança em contextos industriais.

Percebe-se também um aumento no uso de técnicas de aprendizado profundo e um interesse crescente em técnicas avançadas como CNN (Redes Neurais Convolucionais) e LSTM (Long Short-Term Memory). Isso reflete a evolução contínua na capacidade e complexidade dos métodos de aprendizado de máquina. O surgimento e popularização crescente de GAN (Generative Adversarial Networks) nos anos mais recentes também sinalizam inovação e busca por métodos mais sofisticados.

A revisão sistemática conduzida foi direcionada para abranger estudos específicos sobre falhas, máquinas e equipamentos industriais, e o uso de sensores, estando alinhada com a questão central de pesquisa da tese. Esta abordagem focalizada permitiu uma análise mais aprofundada e relevante dos dados na esfera de interesse. Consequentemente, os dados refletiram uma predominância significativa no uso de séries temporais, com 74 dos estudos empregando essa técnica. Isso sublinha a importância das séries temporais na análise e previsão de falhas em ambientes industriais, bem como na monitorização eficaz de máquinas através de sensores.

No que diz respeito ao uso de dados de séries temporais binárias, observa-se que a grande maioria dos estudos, 95 no total, analisou dados de séries temporais, no entanto nenhum dos estudos capturados por esta revisão sistemática analisou dados de séries temporais binárias. Isso indica que os tipos de dados e os tipos de sensores que geram estes dados diferem dos dados e dos sensores abordados na proposta neste trabalho.

Esta revisão sistemática da literatura revela não apenas as tendências atuais no campo de aprendizado de máquinas no contexto de máquinas e equipamentos industriais, mas também aponta para áreas menos exploradas, como séries temporais binárias geradas por sensores indutivos (que são maioria em máquinas ferramentas), preparando o terreno para a próxima fase deste trabalho.

APÊNDICE II – PROTOCOLO DE REVISÃO DA LITERATURA II

Para realização da revisão sistemática da literatura sobre séries temporais binárias também foi utilizada metodologia PRISMA, descrita neste apêndice.

SELEÇÃO DOS ARTIGOS SOBRE SÉRIES TEMPORAIS BINÁRIAS

Como não se foi possível identificar similaridade nos tipos de dados analisados nos estudos sobre aprendizado de máquinas aplicado a análise e identificação de falhas em máquinas industriais, se fez necessária a condução de uma segunda revisão sistemática da literatura sobre o tipo de dado específico em que se prende analisar neste trabalho, a saber, séries temporais binárias.

A pesquisa foi conduzida nas bases de dados Scopus, Web of Science e IEEE Xplore. Os critérios de elegibilidade focaram em artigos publicados em periódicos ou apresentados em congressos e capítulos de livros, restringindo-se somente a trabalhos escritos em inglês e não se limitou a busca a um período (PAGE *et al.*, 2021).

Como estratégia de busca, foi considerado o termo *binary time series* para identificar a literatura relevante exclusivamente sobre séries temporais binárias (PAGE *et al.*, 2021). Aplicou-se critérios de exclusão para garantir a relevância dos artigos selecionados. Primeiramente, excluiu-se os artigos que não continham o termo de pesquisa no título, indicando uma desconexão potencial com o tema central da pesquisa. Além disso, descartou-se os artigos que os dados analisados não eram séries temporais binárias, conforme aprestado na tabela 16.

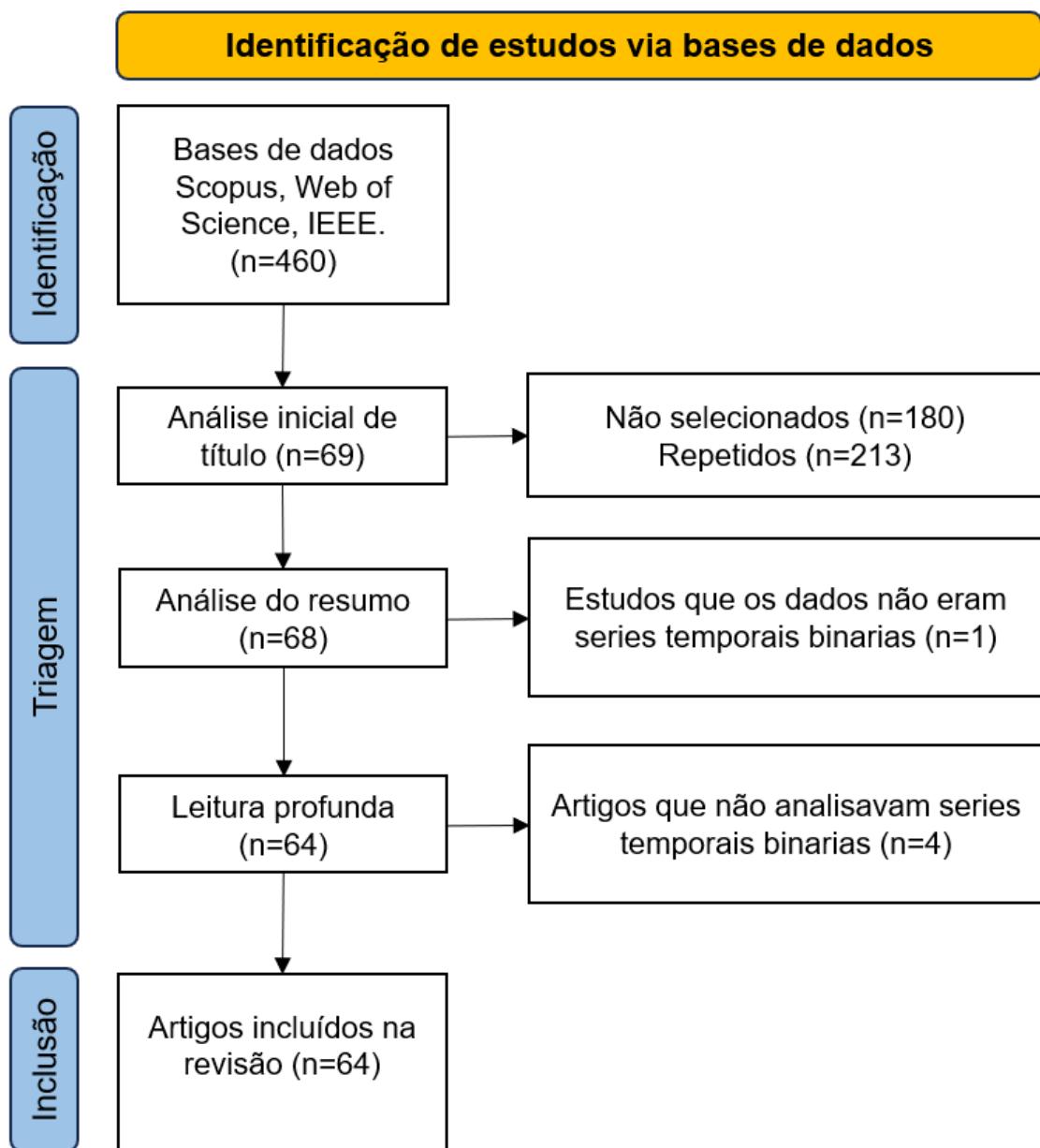
Tabela 16 – Protocolo de revisão da literatura II.

Objetivo	Identificar artigos que abordem métodos de análise de séries temporais binárias.
Termo de pesquisa	“ <i>binary time series</i> ”
Bases de dados	Scopus, Web of Science e IEEE Xplore
Tipos de publicação	Artigos publicados em periódicos ou congressos, capítulos de livro.
Idioma	Inglês
Período	A pesquisa não se limitou a um período
Critérios de inclusão	Estudos sobre séries temporais binárias
Critérios de exclusão	Estudos em que os dados não eram séries temporais binárias Estudos que não analisavam séries temporais binárias

Fonte: Elaborado pelo autor.

A busca resultou em 460 artigos de abril de 2024 que tratavam sobre o tema séries temporais binárias. Após uma revisão dos títulos, foram identificados 213 artigos repetidos entre as bases pesquisadas que foram descartados, bem como 180 artigos por não serem aderentes ao escopo desta pesquisa, restando assim 69 artigos. Após uma checagem do tipo de dado utilizado nos estudos, foram selecionados 68 artigos em que os dados eram séries temporais binárias, dos quais foram identificados 64 artigos que analisam séries temporais binárias, conforme descrito na Figura 36.

Figura 36 – Fluxograma de revisão da literatura II.



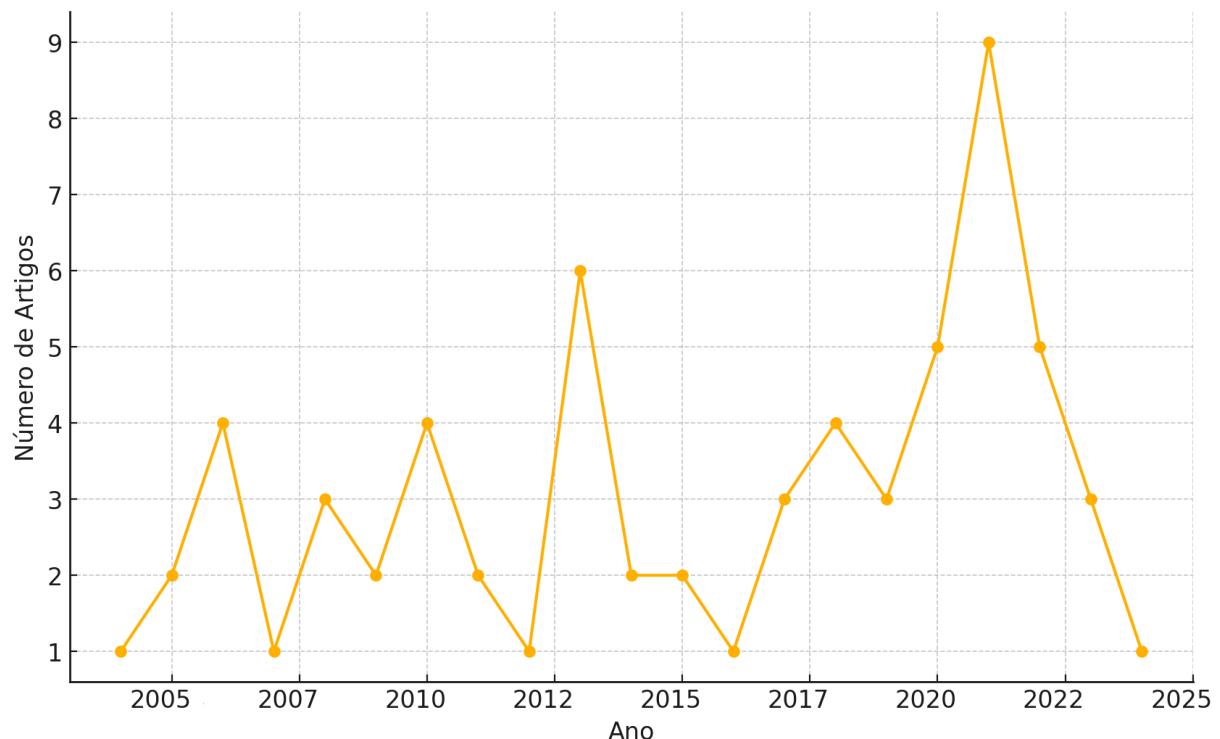
Fonte: Elaborado pelo autor.

ANÁLISE DE SÉRIES TEMPORAIS BINÁRIAS.

A análise de séries temporais binárias tem sua aplicação em uma variedade de contextos, desde a previsão de eventos discretos até a modelagem de comportamentos complexos. Esta revisão sistemática visa explorar as tendências e técnicas emergentes nesta área, com base em uma análise detalhada de artigos publicados que utilizam séries temporais binárias.

A partir da análise temporal da distribuição dos estudos por ano pode-se observar uma variação no número de publicações por ano, com alguns anos apresentando um número maior de artigos do que outros. Um pico notável em anos recentes também sugere que o tópico se tornou particularmente relevante, conforme descrito na Figura 37.

Figura 37 – Análise temporal da distribuição dos artigos.

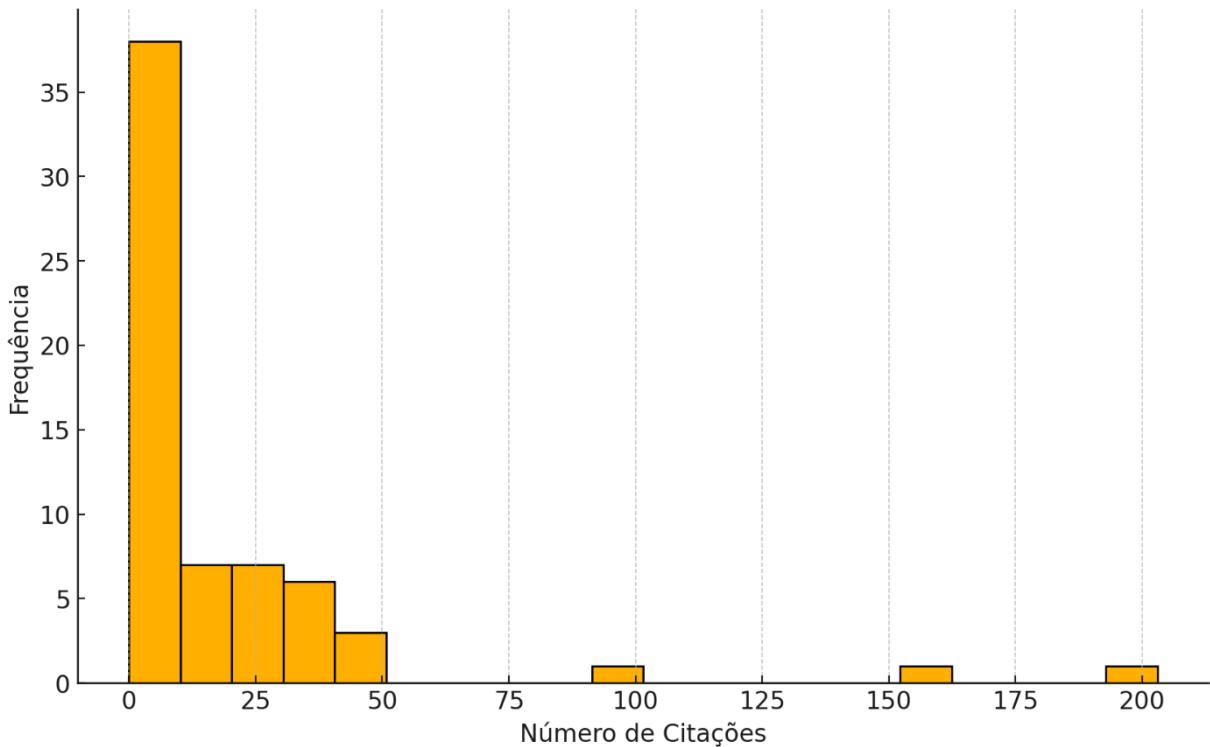


Fonte: Elaborado pelo autor.

A análise das citações revela uma significativa variabilidade na visibilidade e impacto dos artigos. A média de citações é de 18,5 por artigo, com um desvio padrão de 33,85. Enquanto alguns artigos alcançam até 203 citações, a maioria tem um número relativamente baixo, com uma mediana de 7 citações. Isso indica que, embora

existam trabalhos de grande influência, muitos artigos ainda não receberam ampla atenção na comunidade científica, conforme mostrado na Figura 38.

Figura 38 – Distribuição de citações dos artigos.



Fonte: Elaborado pelo autor.

O histograma mostra a frequência dos artigos em diferentes intervalos de número de citações. A maioria dos artigos tem menos de 50 citações, com uma concentração significativa entre 0 e 10 citações. Há alguns artigos com um número muito alto de citações, mas são casos excepcionais.

ANÁLISE DAS DESCRIÇÕES DOS DADOS

A análise das descrições dos dados nos artigos revela uma ampla gama de aplicações. Desde competições esportivas e atividade neuronal até redes complexas e mercados financeiros, os dados binários são aplicados em diversos cenários. As descrições dos dados nos artigos revelam os seguintes padrões e categorias comuns:

- Descrição Geral: A maioria dos artigos (16 no total) utiliza a descrição genérica séries temporais binárias ou variações dela, sem fornecer detalhes específicos sobre os dados analisados.

- Competição Esportiva: Algumas séries temporais binárias são relacionadas a eventos específicos, como uma competição de remo entre as universidades de Oxford e Cambridge (3 artigos).
- Atividade Neuronal: Outras séries temporais binárias representam a atividade neuronal (4 artigos).
- Redes Complexas: Algumas séries representam a dinâmica dos nós em uma rede complexa (1 artigo).
- Ocupação Espectral: Há séries temporais que consistem em medições síncronas de ocupação espectral (1 artigo).
- Poluição Ambiental: Algumas séries temporais analisam concentração de dióxido de enxofre (SO₂) (1 artigo).
- Mercados Financeiros: As séries temporais binárias também são aplicadas a mercados de ações de urso e touro nos Estados Unidos (1 artigo).
- Trajetórias de Movimento: Algumas séries consistem nas trajetórias de movimento do *Caenorhabditis elegans* (1 artigo).

TÉCNICAS DE ANÁLISE DE SÉRIES TEMPORAIS BINÁRIAS

Nos últimos anos, algumas técnicas de análise de séries temporais binárias destacam-se pela popularidade crescente, incluindo métodos estatísticos tradicionais e abordagens de aprendizado de máquina, como redes neurais convolucionais e AutoML. A combinação de técnicas tradicionais e modernas reflete a evolução contínua da pesquisa, adaptando-se às novas tecnologias e demandas analíticas. As técnicas que aparecem com mais frequência recentemente incluem:

- Modelos estatísticos: 2 artigos
- PPGLM (Generalized Linear Models for Panel Data): 1 artigo
- Transformada de Walsh-Hadamard, Transformada de Fourier: 1 artigo
- Análise de trajetória de movimento, Identificação de características distintivas, Uso da função findpeaks do MATLAB: 1 artigo
- Proposição de método de análise discriminante baseada em séries temporais binárias: 1 artigo
- Modelos probit dinâmicos, Distribuições unificadas de skew-normais (sun), Métodos de Monte Carlo: 1 artigo

- Modelo Caótico D-PL, Análise de Correlação: 1 artigo
- Algoritmo de expectativa-maximização (EM): 1 artigo
- Modelos de Regressão Logística, AR(p), modelos autorregressivos: 1 artigo
- Short Isometric Shapelets Transform (SIST): 1 artigo
- Entropia de Similaridade de Comprimento de Ramo, Detecção de Falhas, Differential Dynamic Time Warping (DTW), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT): 1 artigo
- CNN, U-Net, Binary Cross Entropy, Adam Optimizer: 1 artigo
- Modelos Lineares de Efeitos Mistos (LMM), Modelos Lineares Generalizados de Efeitos Mistos (GLMM), Modelos Autoregressivos de Efeitos Mistos (AR-GLMM): 1 artigo
- Custo de Erro de Tempo Proposto (STE), Gradient Descent Algorithm: 1 artigo
- Regressão Logística, Cadeias de Markov Monte Carlo (MCMC), Marginal Likelihood, Deviance Information Criteria (DIC): 1 artigo
- New Discrete Autoregressive Moving-Average (NDARMA), Generalized Binary Autoregressive Moving-Average (gbARMA): 1 artigo
- Isolation Forest, Local Outlier Factor, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), kMeans, AE, VAE, SVM, AutoML: 1 artigo
- Modelo de Processo Gaussiano Generalizado para respostas binárias, Função de Correlação Exponencial de Potência, Modelos de Séries Temporais Binárias, Autoregressão (AR): 1 artigo
- Regressão Logística, Modelo Autoregressivo (AR), Método de Verossimilhança Parcial: 1 artigo
- Abordagem Bayesiana, Metropolis-Hastings (MH): 1 artigo
- Rede Neural Convolucional Bayesiana (BCNN): 1 artigo

Entre essas técnicas, algumas se destacam pela sua popularidade recente:

- Modelos estatísticos: Continuam sendo amplamente utilizados.
- Redes Neurais Convolucionais (CNNs): Indicando um interesse crescente em técnicas de aprendizado profundo.
- Métodos Bayesianos e MCMC: Estão ganhando tração, mostrando um interesse em abordagens probabilísticas.

- Isolation Forest e AutoML: Indicando uma tendência crescente para técnicas de aprendizado de máquina e automação de modelos.

Essa diversidade sugere que os estudos sobre séries temporais binárias estão explorando uma ampla gama de métodos, tanto tradicionais quanto modernos, para abordar diferentes tipos de problemas e dados. A combinação de técnicas tradicionais e modernas reflete a evolução contínua da pesquisa, adaptando-se às novas tecnologias e demandas analíticas.

ANÁLISE DOS OBJETIVO DOS ARTIGOS

A análise dos objetivos dos artigos revela uma variedade de metas e focos de pesquisa, dentre esses, alguns exemplos representativos são:

- Modelos Estatísticos e Controle: Avaliar a controlabilidade de um modelo estatístico específico, o point-process generalized linear model (PPGLM). Este objetivo mostra um interesse em entender e controlar modelos estatísticos aplicados a séries temporais binárias.
- Análise de Relações e Covariáveis: Analisar a relação entre uma série temporal binária e covariáveis estocásticas e dependentes do tempo. Foco na compreensão das relações entre séries temporais binárias e outras variáveis influenciadoras.
- Modelagem com Lacunas: Modelagem estatística de séries temporais binárias com lacunas, focando em desenvolver modelos de regressão para lidar com a natureza binária dos dados e a presença de lacunas nas observações ao longo do tempo. Desenvolvimento de modelos para dados incompletos.
- Caracterização da Autossimilaridade: Caracterização da auto-similaridade de séries temporais binárias por meio de um novo método que envolve a estimativa do perfil de entropia de similaridade de comprimento de ramo. Desenvolvimento de novos métodos para caracterizar séries temporais binárias.
- Previsão Ambiental: Previsão de séries temporais binárias de concentração de dióxido de enxofre (SO₂) utilizando o Modelo Aditivo Generalizado (GAM) com função de ligação desconhecida. Aplicação de modelos preditivos em dados ambientais.

Por outra perspectiva, os artigos também podem ser agrupados por categorias comuns de objetivos, sendo elas:

- Desenvolvimento e Avaliação de Modelos: Muitos artigos focam no desenvolvimento, avaliação e controle de modelos estatísticos específicos aplicados a séries temporais binárias.
- Análise de Relações e Covariáveis: Análise das relações entre séries temporais binárias e outras variáveis, muitas vezes com foco em covariáveis estocásticas e dependentes do tempo.
- Previsão e Detecção de Padrões: Previsão de eventos futuros ou detecção de padrões específicos em séries temporais binárias, aplicados a diferentes contextos, incluindo ambientais e financeiros.
- Caracterização e Similitude: Desenvolvimento de métodos para caracterizar a autossimilaridade e outras propriedades das séries temporais binárias.
- Dados Incompletos: Criação de modelos e técnicas para lidar com lacunas e dados incompletos em séries temporais binárias.