

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE MINAS GERAIS - *CAMPUS* SABARÁ

ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Wilkson Araujo Costa Lima

**ESTUDO DE VIABILIDADE DE UM MÉTODO BASEADO EM INTELIGÊNCIA
ARTIFICIAL PARA CLASSIFICAÇÃO DE ANOMALIAS EM MÁQUINA
EMBALADORA USANDO DADOS COLETADOS EM PRODUÇÃO, PARA
AUXILIAR NA MANUTENÇÃO.**

Sabará
2026

WILKSON ARAUJO COSTA LIMA

**ESTUDO DE VIABILIDADE DE UM MÉTODO BASEADO EM INTELIGÊNCIA
ARTIFICIAL PARA CLASSIFICAÇÃO DE ANOMALIAS EM MÁQUINA
EMBALADORA USANDO DADOS COLETADOS EM PRODUÇÃO, PARA
AUXILIAR NA MANUTENÇÃO.**

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus* Sabará para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Luiz Guilherme Hilel Drumond
Silveira

Sabará
2026

Lima, Wilkson Araújo Costa

L732e

Estudo de viabilidade de um método baseado em inteligência artificial para classificação de anomalias em máquina embaladora usando dados coletados em produção, para auxiliar na manutenção. [manuscrito]. / Wilkson Araújo Costa Lima. - 2026.

56 f. : il.

Orientador: Prof. Me. Luiz Guilherme Hilel Drumond Silveira.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto Federal de Minas Gerais, *Campus* Sabará.

1. Banco de dados - Modelagem. – Monografia. 2. Máquinas - Manutenção e reparos. – Monografia. 3. Inteligência artificial. – Monografia. 4. Localização de falhas (Engenharia). – Monografia. 5. Fábricas - Manutenção. – Monografia. 6. Análise de séries temporais. – Monografia. 7. Redes neurais (Computação). – Monografia. I. Silveira, Luiz Guilherme Hilel Drumond. II. Instituto Federal de Minas Gerais, *Campus* Sabará. III. Bacharelado em Engenharia de Controle e Automação. IV. Título.

CDU 004.89

César dos Santos Moreira / CRB6-2229
Biblioteca do IFMG *Campus* Sabará



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS

Campus Sabará
Diretoria de Ensino, Pesquisa e Extensão Conselho de Área
- Informática e Comunicação
Rodovia MGC 262, Km 10 - Bairro Sobradinho - CEP 34590-390 - Sabará - MG
- www.ifmg.edu.br

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Aos nove dias do mês de fevereiro do ano de dois mil e vinte e seis, às dezenove horas, sob a presidência do professor Luiz Guilherme Hilel Drumond, reuniu-se a banca examinadora composta pelos professores abaixo relacionados, para a defesa do Trabalho de Conclusão de Curso (TCC) do discente Wilkson Araujo Costa Lima, matrícula nº 0065961, do curso de Engenharia de Controle e Automação, do IFMG *campus* Sabará.

O trabalho intitulado ESTUDO DE VIABILIDADE DE UM MÉTODO BASEADO EM INTELIGÊNCIA ARTIFICIAL PARA CLASSIFICAÇÃO DE ANOMALIAS EM MÁQUINA EMBALADORA USANDO DADOS COLETADOS EM PRODUÇÃO, PARA AUXILIAR NA

MANUTENÇÃO DE MÁQUINA foi apresentado e submetido à apreciação da banca. Após exposição, arguição e deliberação, a banca atribuiu a nota final de 70 pontos (de um total de 80). Somados aos 18 pontos atribuídos ao aluno pelo docente responsável pela disciplina de Projeto II, o aluno ficou com um total de 88 pontos, resultando em **aprovado**, condicionado ao cumprimento das orientações e prazos estabelecidos pelas normas acadêmicas institucionais.

Compuseram a Banca Examinadora:

- Membro 1 (Presidente): Luiz Guilherme Hilel Drumond
- Membro 2: Mariella Maia Quadros
- Membro 3: Carlos Alexandre Silva

O discente deverá apresentar a versão final do trabalho em formato PDF e depositá-la no repositório institucional até o dia 27 de fevereiro. O não cumprimento dessas exigências implicará na não contabilização das horas referentes aos componentes curriculares de TCC I e TCC II no sistema acadêmico.

A sessão foi encerrada às vinte horas e quarenta minutos. Para constar, eu, Luiz Guilherme Hilel Drumond, redigi a presente ata que após lida publicamente, foi aprovada e assinada pelos membros da banca examinadora.

Sabará, 09 de fevereiro de 2026.



Documento assinado eletronicamente por **Luiz Guilherme Hilel Drumond Silveira, Professor**, em 10/02/2026, às 18:38, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Carlos Alexandre Silva, Professor EBTT**, em 10/02/2026, às 19:41, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Mariella Maia Quadros, Professora EBTT**, em 11/02/2026, às 18:24, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2618015** e o código CRC **D651EBE8**.

23714.001466/2025-48

2618015v1

Dedico esta monografia aos meus pais Wilson e Adriana, em reconhecimento ao apoio e incentivo constantes que tornaram possível a realização deste trabalho.

AGRADECIMENTOS

Neste momento, agradeço principalmente aos meus pais por todo o apoio que me deram ao longo da minha trajetória acadêmica. Sempre estiveram presentes, incentivando-me a seguir em frente e a me dedicar cada vez mais aos meus objetivos.

RESUMO

Este trabalho avalia a viabilidade do uso de um Autoencoder baseado em redes neurais recorrentes do tipo *Long Short-Term Memory* (LSTM) para a detecção de anomalias em sinais operacionais de uma máquina embaladora *Vega Shrink-Wrapper*. O modelo é treinado exclusivamente com dados considerados normais, aprendendo o padrão temporal típico de operação da máquina, e a identificação de anomalias é realizada por meio do erro de reconstrução das janelas de dados, calculado pelo Erro de reconstrução (MSE). Utiliza-se um conjunto de dados reais públicos contendo séries temporais de sensores industriais coletados ao longo de seis meses de operação. A metodologia envolve a preparação e o janelamento dos dados, o treinamento do Autoencoder LSTM e a avaliação do desempenho do modelo em dois cenários de divisão entre dados de treino e teste (70/30 e 90/10), utilizando métricas como acurácia, precisão, recall, F1-score e curvas Precision–Recall. Como resultados, foram calculadas métricas de desempenho que possibilitaram a comparação direta entre os dois cenários de divisão dos dados.

Palavras-chave: Detecção de anomalias. Autoencoder LSTM. Séries temporais. Manutenção preditiva. Inteligência Artificial.

ABSTRACT

This work evaluates the prediction of using an Autoencoder based on Long Short-Term Memory (LSTM) recurrent neural networks for anomaly detection in operational signals of a machine packaged with Vega Shrink-Wrapper. The model is trained exclusively with data considered normal, learning the typical temporal pattern of machine operation, and anomaly identification is performed through the occurrence error of data windows, calculated by the Occurrence Error (MSE). A publicly available real dataset containing time series from industrial sensors available over six months of operation is used. The methodology involves data preparation and windowing, training the LSTM Autoencoder, and evaluating the model's performance in two data splitting scenarios (70/30 and 90/10), using metrics such as accuracy, precision, recall, F1-score, and Precision-Recall curves. As a result, performance analyses were calculated that allowed a direct comparison between the two data splitting scenarios.

Keywords: Anomaly detection. LSTM Autoencoder. Time series. Predictive maintenance. Artificial Intelligence.

LISTA DE ILUSTRAÇÕES

Figura 1 – Janela do tempo	21
Figura 2 – Estrutura do encoder e decoder.	22
Figura 3 – Máquina embaladora Vega Shrink-Wrapper em perspectiva	23
Figura 4 – Gráfico da distribuição de dados ao longo dos meses	28
Figura 5 – Gráfico do Precision-Recall 70/30	38
Figura 6 – Gráfico do Precision-Recall 90/10	39
Figura 7 – Matriz de confusão para teste de 30%	40
Figura 8 – Matriz de confusão para teste de 10%	40
Figura 9 – Comparação falso alarme e falha não detectada.	42
Figura 10 – Gráfico da distribuição dos erros 30%	45
Figura 11 – Gráfico da distribuição dos erros 10%	45
Figura 12 – Gráfico da série do erro de reconstrução 70/30 dia 14/05 .	47
Figura 13 – Gráfico da série do erro de reconstrução 90/10 dia 14/05.	47
Figura 14 – Gráfico da série do erro de reconstrução 70/30 dia 22/06.	48
Figura 15 – Gráfico da série do erro de reconstrução 90/10 dia 22/06.	48
Figura 16 – Gráfico da série do erro de reconstrução 70/30 dia 15/06.	49
Figura 17 – Gráfico da série do erro de reconstrução 90/10 dia 15/06.	49

LISTA DE TABELAS

Tabela 1 – Comparação entre os diferentes resultados	41
Tabela 2 – Comparação de erro do sistema.	43
Tabela 3 – Tabela de diagnóstico 70/30	50
Tabela 4 – Tabela de diagnóstico 90/10	50

SUMÁRIO

1. INTRODUÇÃO	14
1.1 Objetivos	16
1.1.1 Objetivo geral	16
1.1.2 Objetivos específicos	16
1.2 Justificativa	16
2. REVISÃO BIBLIOGRÁFICA	18
3. FUNDAMENTAÇÃO TEÓRICA	20
3.1 Inteligência Artificial como ferramenta para analisar dados	20
3.2 Arquitetura LSTM	21
3.3 Autoencoders	22
3.3.1 Encoder	22
3.3.2 Decoder	22
4. METODOLOGIA	23
4.1 Objeto de Estudo: A Máquina Embaladora Vega Shrink-Wrapper	23
4.2 Análise Exploratória dos Dados	24
4.3 Análise Exploratória dos Dados	25
4.3.1 Autoencoder LSTM	25
4.3.2 Justificativa do uso do LSTM	26
4.4 Período de referência	26
4.5 Arquitetura do Autoencoder LSTM	29
4.6 Erro de reconstrução	29
4.7 Avaliação do Desempenho do Modelo	30
4.7.1 Curva ROC	30
4.7.2 Sensibilidade (Recall)	31
4.7.3 Curva Precision–Recall	31
4.7.4 F1-score	31
4.7.5 Acurácia	31
4.8 Processo de treinamento	32
4.9 Preparação dos dados	33
4.10 Matriz de Confusão	35
4.11 Definição do Limiar (Threshold)	35
4.11.1 Duas configurações de treinamento da rede	36
5. RESULTADOS	36
5.1 Análise da Matriz de Confusão	39
5.2 F1-Score	43
5.3 Distribuição dos erros	44
5.4 Amostras dos erros	46
5.5 Resultado da rede	49
6. CONCLUSÃO	51
REFERÊNCIAS	52

LISTA DE ABREVIATURAS E SIGLAS

ABIEF – Associação Brasileira da Indústria de Embalagens Plásticas Flexíveis

AE – Autoencoder

AUC – Área Sob a Curva (area sobre curva)

CSV – *Comma-Separated Values*

FN – Falso Negativo

FNR – Taxa de Falsos Negativos (do inglês, False Negative Rate)

FP – Falso Positivo

FPR – Taxa de Falsos Positivos (do inglês, False Positive Rate)

GPU – Graphics Processing Unit

IA – Inteligência Artificial

IFMG – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais

LSTM – Memória de Longo Prazo (do inglês, *Long Short-Term Memory*)

MSE – Erro de reconstrução (do inglês, Mean Squared Error)

PR – Curva Precisão–Revocação (do inglês, *Precision–Recall*)

PR-AUC – Área Sob a Curva Precision–Recall

RAM – Random Access Memory

RNA – Rede Neural Artificial

RNR – Redes Neurais Recorrentes

ROC – Curva Característica de Operação do Receptor (Receiver Operating Characteristic)

ROC-AUC – Área Sob a Curva ROC

SCADA – Supervisão, Controle e Aquisição de Dados

SE – Sensibilidade

TCC – Trabalho de Conclusão de Curso

VN – Verdadeiro Negativo

VP – Verdadeiro Positivo

1. INTRODUÇÃO

A indústria de embalagens desempenha um papel fundamental na economia global, com o setor de bebidas apresentando crescimento contínuo na demanda por embalagens. No contexto brasileiro, esse mercado tem se expandido significativamente, exigindo sistemas de produção cada vez mais automatizados e confiáveis. Segundo dados da Associação Brasileira da Indústria de Embalagens Plásticas Flexíveis (ABIEF, 2024), no Brasil, houve um aumento de 2,5% da produção de embalagens flexíveis e alta de 7,6% do faturamento do setor entre 2023 e 2024. Esse desempenho é impulsionado principalmente pelo mercado de cervejas e refrigerantes, que responde por um grande volume total de embalagens produzidas no país.

Neste contexto, as máquinas embaladoras como a *Vega Shrink-Wrapper* são componentes críticos nesse processo, sendo responsáveis pela etapa final de acondicionamento dos produtos. A *Vega Shrink-Wrapper* é uma máquina industrial de empacotamento que realiza o agrupamento e o envolvimento de produtos com filme termoencolhível, formando fardos compactos para transporte e armazenamento. Seu funcionamento envolve sistemas mecânicos, elétricos e de controle automatizado que garantem padronização das embalagens. No entanto, uma pesquisa global da *True Cost of Downtime*, com 72 grandes multinacionais, revelou que plantas industriais perdem, em média, 323 horas de produção por ano devido a falhas de equipamentos (METSO, 2024). Esse dado evidencia a importância da manutenção eficaz, especialmente no controle de falhas relacionadas ao desgaste de componentes mecânicos.

Por sua vez, grande parte dos sistemas utilizados nos processos industriais estão suscetíveis a falhas, seja devido a erros no modelo, mau funcionamento de sensores e atuadores, ou perturbações externas. Essas falhas são comportamentos não esperados ou desvios que podem acontecer em um sistema (MONTESUMA *et al.*, 2022). Porém, nesse contexto utilizando a máquina embaladora, a principal possível falha seria em equipamentos como a lâmina de corte e motor de atuadores, que com o passar do tempo vão perdendo sua eficiência. Isso pode ocasionar em um aumento no tempo de trabalho ou até parada da máquina. Esses desvios quando não identificados e tratados da forma correta podem evoluir para um grande prejuízo ou até mesmo acidentes.

Métodos convencionais de monitoramento de desgaste apresentam

limitações significativas, com uma taxa de detecção de falhas incipientes inferior a 65% (WANG *et al.*, 2021). Essa deficiência resulta em paradas não planejadas que representam até 35% dos custos de manutenção em plantas de envase (LI *et al.*, 2020). A IA, através de modelos como aprendizado de máquina e redes neurais, dá a possibilidade com outras técnicas a análise e interpretação de grandes volumes de dados operacionais das máquinas, identificando padrões que precedem falhas. Isso auxilia processos de manutenção, reduzindo custos e minimizando tempo de inatividade. (SANTOS *et al.*, 2024). Com isto, observa-se um espaço para a aplicação da inteligência artificial, que possui grande potencial para solucionar problemas analisando o sistema e apontando anomalias.

Nesse contexto, este trabalho propõe o uso de Inteligência Artificial para classificar anomalias na máquina Vega Shrink-Wrapper, com foco na identificação de desgastes durante sua operação. Inicialmente, será realizada a análise exploratória e a preparação dos dados provenientes dos sensores da máquina, incluindo etapas de tratamento, normalização e organização das variáveis.

Em seguida, será desenvolvido e treinado um modelo de rede neural recorrente do tipo LSTM, estruturado como Autoencoder, utilizando exclusivamente dados considerados normais para que o modelo aprenda o padrão típico de funcionamento do equipamento. Após o treinamento, o erro de reconstrução será utilizado como critério para identificar possíveis desvios operacionais.

O desempenho do modelo será avaliado por meio de métricas como acurácia, precisão, recall, F1-score e AUC, permitindo uma análise quantitativa da capacidade de detecção de anomalias. Além disso, serão comparadas diferentes proporções de divisão entre dados de treino e teste (70/30 e 90/10), com o objetivo de analisar seus impactos na estabilidade, sensibilidade e generalização do modelo.

Também será investigada a distribuição dos erros de reconstrução, buscando identificar tendências, níveis de sobreposição e separação entre dados normais e anômalos. Por fim, será analisada a aplicação prática do método como ferramenta de apoio à manutenção preditiva, discutindo suas limitações e possíveis melhorias.

Dessa forma, o estudo busca contribuir para a redução de paradas não planejadas e para o aumento da confiabilidade do sistema produtivo.

1.1 Objetivos

Nesta seção serão apresentados os objetivos deste trabalho, divididos em objetivo geral e objetivos específicos, com a finalidade de orientar o desenvolvimento da pesquisa e delimitar o escopo das análises realizadas.

1.1.1 Objetivo geral

Avaliar a viabilidade de um modelo de inteligência artificial para detecção de anomalias em sinais operacionais de uma máquina embaladora *Vega Shrink-Wrapper*, utilizando dados reais de produção. O objetivo é determinar se o modelo é capaz de distinguir padrões normais de padrões potencialmente associados a falhas ou degradação mecânica.

1.1.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Realizar a análise exploratória e preparação dos dados provenientes dos sensores da máquina embaladora.
- Desenvolver e treinar um modelo Autoencoder LSTM utilizando apenas dados considerados normais.
- Avaliar o desempenho do modelo na detecção de anomalias por meio de métricas como acurácia, precisão, recall, F1-score e AUC.
- Comparar diferentes proporções de divisão treino/teste (70/30 e 90/10) e analisar seus impactos na estabilidade e sensibilidade do modelo.
- Investigar a distribuição dos erros de reconstrução para identificar tendências, sobreposição e separação entre classes normais e anômalas.
- Analisar a aplicação prática do método como ferramenta de apoio à manutenção preditiva da máquina, discutindo limitações e potenciais melhorias.

1.2 Justificativa

O presente trabalho justifica-se pela importância econômica e tecnológica do tema e pelo seu potencial impacto na eficiência produtiva. A indústria de embalagens, setor estratégico para a economia brasileira, enfrenta desafios significativos relacionados à manutenção de equipamentos críticos como a *Vega Shrink-Wrapper*.

Uma pesquisa da ABB e Sapio (2023) indica que indústrias de alimentos e bebidas perdem entre US\$ 4 mil e US\$ 30 mil por hora com paradas não programadas. Embora amplamente utilizados, os sistemas tradicionais de monitoramento de condição enfrentam dificuldades na identificação de anomalias em seus estágios primários. Pesquisas indicam que a acurácia desses métodos para falhas incipientes é limitada, frequentemente não atingindo níveis satisfatórios de confiabilidade diagnóstica (Silva & Oliveira, 2022).

Nesta visão é possível ver um cenário muito favorável para implementação de um sistema de inteligência artificial, a fim de otimizar o processo de monitoramento de máquinas.

As técnicas inteligentes são ferramentas inspiradas em processos naturais e/ou biológicos e pertencem ao campo da inteligência computacional. Apresentam bons resultados em problemas de reconhecimento de padrões e diagnóstico, e assim, podem ser adaptadas para resolver problemas de monitoramento e identificação de falhas estruturais na engenharia. (DEODATO et al., 2022).

Neste sentido, métodos de análise de anomalias com inteligência artificial, podem ser ferramentas importantes para auxiliar técnicos e engenheiros a identificar e caracterizar grandes variações no processo. Tendo potencial vantagem em comparação com métodos convencionais, como uma simples análise de um técnico da situação da máquina.

2. REVISÃO BIBLIOGRÁFICA

A competitividade e as constantes mudanças na economia impõem às empresas a necessidade de reduzir seus custos e atuar de forma responsável em relação à segurança e ao meio ambiente, o que representa um desafio contínuo para o setor de manutenção (Bandeira et al., 2016). Corroborando esta visão, como pontua Orives (2019), “os desafios da área de manutenção tornaram-se mais complexos, mas continuam sendo os mesmos: reduzir o tempo de equipamento parado, reduzir os custos de peças de reposição em estoque e aumentar a utilização do ativo”. Para atender a essas demandas, torna-se essencial a adoção de tecnologias e abordagens mais inteligentes que auxiliem e otimizem todo o sistema de manutenção.

Nesse contexto, a Inteligência Artificial (IA) surge como uma das mais promissoras tecnologias capazes de enfrentar tais desafios. O campo da IA vem sendo construído a partir de contribuições de diferentes áreas científicas, como Filosofia, Matemática, Neurociência, Psicologia, Linguística, Informática e Engenharia da Computação, sendo considerado, assim, interdisciplinar (DEAN, 1995; GOMES, 2010; RUSSELL; NORVIG, 2013). Essa base ampla confere à IA a robustez necessária para modelar sistemas complexos, como o comportamento de máquinas industriais.

Uma aplicação muito eficaz da IA na manutenção se dá por meio do Aprendizado de Máquina, especificamente com as Redes Neurais Artificiais (RNA). Um exemplo prático dessa aplicação é encontrado em Kang (2020), onde o autor fez testes com Redes Neurais para detecção de falhas em uma planta química composta por diversos componentes. No trabalho, foi avaliado um total de 21 falhas, e os métodos foram treinados com dois bancos de dados diferentes, um com 5 amostras e outro com 40. Como resultado, foi obtido um índice de acerto de 75,5% se treinada com somente 5 amostras e 87,3% se treinada com as 40 amostras. Este resultado evidencia não apenas a viabilidade do método, mas também a importância crucial da quantidade de dados para aumentar a confiabilidade dos sistemas de manutenção preditiva.

No contexto, o trabalho de Mine (2024) apresenta um estudo de caso prático e relevante. O objetivo geral da pesquisa foi desenvolver modelos baseados em Redes Neurais do tipo LSTM, que são redes temporais com memória, capazes de classificar falhas em uma bomba de cavacos, um equipamento crítico no processo de

produção de celulose. A metodologia empregada envolveu a coleta de dados de sensores de um equipamento real ao longo de um ano. O autor testou 24 configurações de modelo distintas, variando sistematicamente a "janela de leitura" (período de dados passados, de 1 a 6 horas) e a "janela de previsão" (antecedência da previsão, de 24 a 96 horas). Essa abordagem permitiu avaliar como a estruturação dos dados de treinamento afeta o desempenho do modelo preditivo.

Como principal resultado, o estudo demonstrou uma variação significativa no desempenho dos modelos conforme a configuração das janelas. O autor ressalta que, para o cenário industrial, a especificidade da capacidade do modelo de identificar corretamente uma falha real é uma métrica de maior impacto do que a acurácia geral, pois falhas não detectadas podem comprometer a operação. A melhor configuração, denominada "Modelo V", alcançou uma especificidade de 84,49% para uma previsão com 96 horas de antecedência, utilizando uma janela de leitura de 4 horas. Este resultado contrasta fortemente com outras configurações, que apresentaram especificidade inferior a 10%.

Dessa forma, Mine (2024) evidencia empiricamente que a forma como os dados temporais são tratados são cruciais para a confiabilidade de um sistema de manutenção preditiva, reforçando a ideia de que a performance não depende apenas da escolha do algoritmo, mas também de uma criteriosa preparação e estruturação dos dados.

Diferentemente dos estudos que buscam a previsão ou a classificação direta de falhas específicas, o presente trabalho adota uma abordagem baseada na detecção de desvios operacionais a partir de sinais de sensores. O foco está na identificação de comportamentos anômalos em relação ao padrão normal de funcionamento da máquina, sem a necessidade de rótulos prévios de falha ou de modelos supervisionados. Dessa forma, a proposta se concentra em fornecer indícios de degradação operacional que possam subsidiar ações de manutenção, atuando como uma ferramenta de apoio à tomada de decisão em ambientes industriais.

3. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os conceitos necessários para compreender o funcionamento da rede RNA usando o modelo Autoencoder LSTM aplicado à classificação de anomalias em séries temporais industriais. Abrange os fundamentos de manutenção preditiva, séries temporais, redes neurais, arquiteturas recorrentes e o princípio matemático por trás dos autoencoders utilizados neste trabalho.

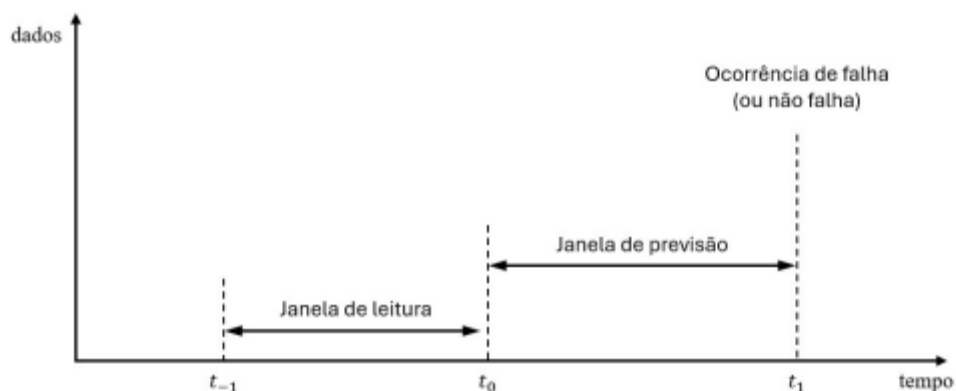
3.1 Inteligência Artificial como ferramenta para analisar dados

A Inteligência Artificial é um campo interdisciplinar da ciência da computação que busca desenvolver sistemas capazes de realizar tarefas que normalmente exigiriam inteligência humana, como aprendizado, raciocínio e reconhecimento de padrões. Seu potencial para analisar grandes volumes de dados e modelar sistemas complexos a torna uma ferramenta poderosa para enfrentar os desafios da manutenção preditiva.

Conforme Silva (2024), a detecção de falhas em sistemas industriais consiste em reconhecer desvios em relação ao comportamento esperado, gerados por desgaste, erros operacionais ou perturbações externas. Esses desvios, se não tratados, podem causar falhas com consequências como instabilidade, riscos à segurança e perdas de produção. Métodos tradicionais de monitoramento apresentam limitações, especialmente na detecção de falhas incipientes, o que evidencia a necessidade de abordagens mais eficientes, como aquelas baseadas em Inteligência Artificial.

Segundo Leukel, González e Riekert (2023), a abordagem para prever a ocorrência de falhas é feita com base na definição de dois intervalos de tempo, sendo a janela de previsão e janela de leitura, conforme Figura 1.

Figura 1 – Janela do tempo



Fonte: Leukel, González e Riekert (2023).

O intervalo de previsão corresponde ao período anterior em que se antecipa a possível ocorrência da falha. Já a janela de observação determina o período de tempo cujos dados serão analisados para realizar essa previsão.

3.2 Arquitetura LSTM

O LSTM é uma variante das Redes Neurais Recorrentes desenvolvida para mitigar o problema do desaparecimento do gradiente, comum em arquiteturas recorrentes tradicionais. Essa característica é alcançada por meio da introdução de estruturas internas denominadas portões de entrada, esquecimento e saída, responsáveis por controlar o fluxo de informações ao longo do tempo. Além disso, a LSTM possui um estado interno de memória que permite o armazenamento seletivo de informações relevantes, possibilitando a captura de dependências temporais de curto e longo prazo nos dados (HOCHREITER; SCHMIDHUBER, 1997).

Devido a essas características, a LSTM é adequada para aplicações envolvendo séries temporais, como previsão de sinais, identificação de padrões temporais anômalos e reconstrução de sequências. No contexto deste trabalho, a arquitetura LSTM é empregada tanto na etapa de codificação (encoder) quanto na etapa de decodificação (decoder) do Autoencoder, permitindo modelar a dinâmica temporal dos sinais e reconstruir sequências representativas do comportamento normal do sistema.

3.3 Autoencoders

Autoencoders são redes neurais projetadas para aprender uma representação compacta dos dados. Eles possuem duas partes:

3.3.1 Encoder

O encoder é a etapa do Autoencoder responsável por mapear os dados de entrada em uma representação de menor dimensão. Nessa etapa, são extraídas as características mais relevantes dos sinais, reduzindo redundâncias e preservando a dinâmica temporal do sistema. No presente trabalho, o encoder é implementado por meio de camadas LSTM, permitindo a captura das dependências temporais das séries de dados e a representação do comportamento normal de operação da máquina.

3.3.2 Decoder

Reconstrói a entrada original a partir do vetor.

1. Se o dado segue o padrão normal a reconstrução é boa.
2. Se o dado está fora do padrão, a reconstrução é ruim.

Essa diferença é quantificada pelo Erro de reconstrução (MSE), usado como indicador de anomalia.

Figura 2 – Estrutura do encoder e decoder.



Fonte: Elaboração própria (2026).

4. METODOLOGIA

Este capítulo descreve os procedimentos metodológicos adotados para o desenvolvimento do sistema de monitoramento de desgaste na máquina embaladora *Vega Shrink-Wrapper*, conforme os objetivos definidos anteriormente. A pesquisa possui uma abordagem quantitativa e aplicada, utilizando técnicas de modelagem computacional para criar um modelo de RNA e encoder LSTM capaz de apreender dados e responder se há comportamentos fora do esperado.

As etapas seguiram uma sequência lógica, que abrange desde a preparação do ambiente de desenvolvimento e a compreensão dos dados até a construção, treinamento e avaliação do modelo.

4.1 Objeto de Estudo: A Máquina Embaladora *Vega Shrink-Wrapper*

A máquina *Vega Shrink-Wrapper* como mostra na Figura 3 é um equipamento essencial para o empacotamento de garrafas com filme termo retrátil, utilizado em fábricas alimentícias. Sendo responsável pela etapa final de acondicionamento dos produtos. Seu processo operacional está no transporte das garrafas por uma esteira, onde são envolvidas por um filme plástico que é cortado e aquecido para formar a embalagem final.

Figura 3 – Máquina embaladora *Vega Shrink-Wrapper* em perspectiva



Fonte: OCME (2025).

Em máquinas embaladoras do tipo shrink-wrapper, as falhas mais recorrentes estão associadas ao desgaste progressivo de componentes mecânicos e conexões submetidas a operação contínua. Elementos como correntes, engrenagens, mancais, rolamentos, acoplamentos, guias lineares e conexões pneumáticas sofrem degradação natural devido ao atrito, vibração e esforços cíclicos. Esse desgaste pode provocar desalinhamentos, aumento de folgas mecânicas, vibrações excessivas e perda de precisão no posicionamento, comprometendo a qualidade da selagem e o sincronismo entre etapas do processo.

Além disso, conexões elétricas e terminais podem sofrer afrouxamento ou oxidação ao longo do tempo, especialmente em ambientes industriais com variação térmica e vibração constante. Esse fenômeno pode gerar falhas intermitentes, ruídos em sinais de sensores e leituras incorretas, impactando diretamente sistemas de controle e monitoramento. Segundo Mobley (2002), a maioria das falhas em equipamentos industriais é precedida por um período de degradação detectável, geralmente associado a aumento de vibração, temperatura e folgas mecânicas.

4.2 Análise Exploratória dos Dados

A base do modelo preditivo são os dados. Nesta etapa, o foco é entender e preparar o conjunto de dados para o treinamento da RNA.

O estudo utiliza um banco de dados público disponibilizado na plataforma Kaggle, referente a um ano de produção de uma máquina embaladora (KAGGLE, 2018).

A primeira etapa prática consiste em uma análise exploratória para reconhecer as variáveis mais relevantes que indicam o desgaste da máquina. Temos ao total 8 entradas de sensores no banco de dados.

Dentro do banco de dados as variáveis estão atribuídas cada uma com um nome e será mantido o mesmo neste trabalho. Como apresentado a seguir:

Timestamp: Tempo de rodagem de cada leitura de dados. **Cut::Motor_Torque:** Medida do torque exercido pelo motor do módulo de corte do filme termo retrátil. Esta é uma das variáveis mais críticas, pois o aumento do torque pode indicar uma maior resistência ao corte, sendo um sintoma direto da perda de eficiência e do desgaste da lâmina. **Cut::CTRL_Position_controller::* (Actual_position, Actual_speed, Lag_error):** Conjunto de dados do controlador de posição do atuador de corte. O *Lag Error* (erro de atraso) representa a diferença entre a posição comandada e a posição real do atuador. Um erro de atraso crescente é um forte indicativo de que o sistema

mecânico está com dificuldades para responder aos comandos, o que pode ser causado pelo desgaste da lâmina ou outros problemas mecânicos associados. A análise desta variável está diretamente alinhada ao objetivo de usar "atrasos na execução de comandos" para o dimensionamento do desgaste.

pSvolFilm::CTRL_Position_controller::* (Actual_position, Actual_speed, Lag_error): Dados referentes ao controle de posição do motor de desbobinamento do filme. Embora não esteja diretamente ligada ao corte, a sincronia e a tensão do filme são essenciais para o bom funcionamento da máquina. Anomalias nesta variável podem indicar problemas indiretos que afetam o processo de corte ou ser usados para contextualizar as leituras do motor de corte.

pSpintor::VAX_speed: Velocidade do atuador responsável por empurrar ou agrupar as garrafas. A velocidade deste componente influencia o ritmo de toda a operação. Variações podem ser correlacionadas com o desempenho geral da máquina e, conseqüentemente, com o estresse sobre o mecanismo de corte.

4.3 Análise Exploratória dos Dados

Este trabalho utiliza um conjunto de ferramentas e passos organizados para treinar o modelo e classificar a falha. Conforme apresentado nas seções seguintes.

4.3.1 Autoencoder LSTM

O modelo usado é um autoencoder com LSTM. Ele tem duas partes:

Encoder (codificador): recebe uma janela do sinal e condensa essa sequência em um vetor pequeno. Como exemplo, considere uma janela de dados de um sensor de vibração que, em condições normais, apresenta oscilações regulares e de baixa variabilidade. O encoder não memoriza cada ponto do sinal individualmente, mas aprende a representar esse comportamento por meio de poucos valores que descrevem sua forma geral. Caso a vibração passe a apresentar impactos ou oscilações irregulares, essa nova janela não será adequadamente representada no espaço latente aprendido, contribuindo para o aumento do erro de reconstrução no processo de decodificação.

Decodificador: O decodificador recebe o vetor compacto gerado pelo encoder e tem como objetivo reconstruir a janela original do sinal ao longo do tempo. Utilizando camadas LSTM, ele expande essa representação resumida, tentando reproduzir a sequência temporal com o mesmo padrão, forma e comportamento do sinal de

entrada.(GOODFELLOW; BENGIO; COURVILLE, 2016).

O treinamento do modelo é realizado exclusivamente com dados representativos do regime normal de operação do sistema. Dessa forma, o autoencoder aprende o padrão típico de comportamento dos sensores, isto é, a forma como as variáveis medidas oscilam quando a máquina está operando corretamente.

Durante a etapa de inferência, correspondente à aplicação do modelo sobre os dados de teste e novos dados não utilizados no treinamento, o autoencoder recebe um novo trecho de sinais e tenta reconstruí-lo com base exclusivamente nos padrões aprendidos a partir do regime normal de operação. Caso o comportamento observado esteja de acordo com o regime normal, a reconstrução realizada pelo autoencoder tende a ser precisa, resultando em um erro de reconstrução baixo. Por outro lado, quando ocorre um comportamento fora do padrão — como variações abruptas, ruídos atípicos ou falhas no processo — o modelo apresenta maior dificuldade em reconstruir o sinal, o que se reflete em um aumento do erro de reconstrução, medido neste trabalho pelo erro de reconstrução.

Esse erro de reconstrução é, portanto, utilizado como um indicador de anomalia, sendo comparado a um limiar de decisão previamente definido. Valores de MSE inferiores ao limiar são classificados como comportamento normal, enquanto valores superiores indicam a presença de uma anomalia no sistema monitorado.

4.3.2 Justificativa do uso do LSTM

A LSTM é uma RNA com memória. Em vez de olhar cada ponto isolado, ela lê a sequência na ordem e decide o que lembrar e o que esquecer por meio de portas internas (entrada, esquecimento e saída). Na prática, contribui a:

1. Captar tendências lentas, ciclos e atrasos entre variáveis.
2. Diferenciar ruído rápido de mudanças de regime.
3. Lidar de forma mais eficiente com séries temporais industriais, nas quais o contexto recente influencia diretamente a interpretação do estado atual do sistema (GOODFELLOW; BENGIO; COURVILLE, 2016).

4.4 Período de referência

Para realizar o processo de treinamento, é necessário identificar no banco

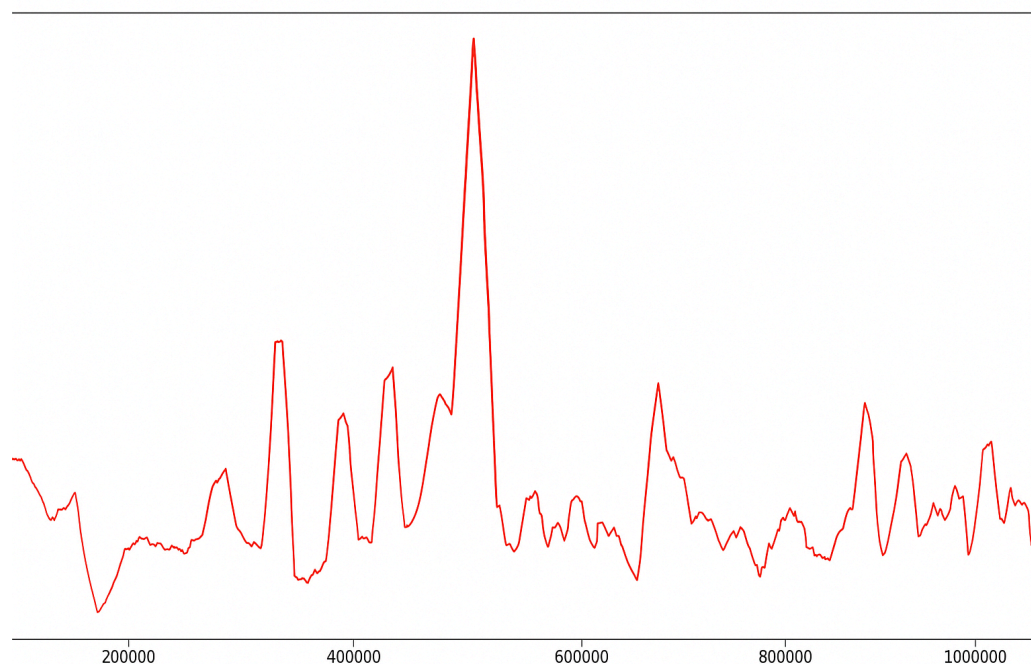
de dados o que pode ser considerado como comportamento normal da máquina, de modo que esse padrão sirva como referência para a posterior identificação de anomalias.

Como forma de obter uma visão global do comportamento da máquina ao longo do tempo, foi construído um índice escalar a partir de todas as variáveis de entrada selecionadas, provenientes dos 8 sensores do sistema, como posições, velocidades e erros de seguimento dos diferentes atuadores.

De forma conceitual, esse índice escalar pode ser entendido como um valor único que resume, em cada instante de tempo, o estado médio da máquina considerando simultaneamente todas as variáveis medidas. Por exemplo, em um determinado instante, cada sensor fornece um valor específico; ao calcular a média desses valores, obtém-se um único número que representa o nível global do sinal naquele momento. Esse procedimento permite reduzir a complexidade dos dados multivariados sem perder a noção geral do comportamento do sistema.

Inicialmente, todas as variáveis dos sensores foram normalizadas, de forma a padronizar suas escalas e evitar que sensores com maior magnitude numérica exercessem influência desproporcional no cálculo do índice. Em seguida, foi calculada a média instantânea entre as variáveis normalizadas, gerando uma série temporal escalar representativa do estado global do sistema. Posteriormente, sobre essa série foi aplicada uma média móvel utilizando a função *rolling* da biblioteca Pandas, com o objetivo de suavizar oscilações pontuais e evidenciar a tendência geral de funcionamento da máquina ao longo do tempo. O resultado desse processo é o gráfico apresentado na Figura 4, que representa o comportamento médio do sistema ao longo do período analisado.

Figura 4 – Gráfico da distribuição de dados ao longo dos meses



A cada 100.000 valores representa um mês de tempo. Sendo que toda a coleta são 12 meses. É possível notar que, até o mês 4 existiram mudanças nas médias dos valores, porém, no mês 5 apresentou uma enorme variação. Por esse motivo será considerado do primeiro mês até o quarto como o período normal da máquina e o quinto e sexto mês como o período a ser avaliado pela rede.

Embora o conjunto de dados disponibilizado contemple um período mais extenso de operação, neste trabalho optou-se por restringir a análise aos seis primeiros meses, sendo os quatro iniciais utilizados como referência de comportamento normal e os dois subsequentes destinados à etapa de diagnóstico. Essa escolha fundamenta-se no fato de que os restantes dos meses não diferem tanto como no começo do ano. Assim não agregando tanto para análise.

Ao longo do período analisado, foram considerados apenas os dias em que a máquina esteve efetivamente em operação, uma vez que períodos de parada não representam o comportamento dinâmico do processo produtivo. Dessa forma, até o quarto mês foram contabilizados 36 dias válidos de operação contínua, utilizados como referência do regime normal de funcionamento.

Para os meses cinco e seis, foram registrados 13 dias de operação, os quais foram destinados à etapa de avaliação e diagnóstico do modelo.

Para cada dia de operação considerado no estudo, foram coletadas, em

média, aproximadamente 2.000 amostras de dados de cada sensor da máquina.

4.5 Arquitetura do Autoencoder LSTM

No presente trabalho, foi implementado um Autoencoder baseado em redes LSTM com o objetivo de classificar o comportamento dos sinais da máquina embaladora e identificar desvios operacionais por meio do erro de reconstrução. O modelo foi treinado exclusivamente com dados representativos do regime normal de operação, de modo a aprender padrões temporais característicos desse estado.

A arquitetura adotada é composta por duas partes: encoder e decoder. O encoder é formado por duas camadas LSTM, com 128 e 64 neurônios, respectivamente, seguidas por uma camada densa com 32 unidades, responsável por gerar a representação do sistema. Essa etapa realiza a compressão das sequências de entrada, preservando as informações temporais mais relevantes dos sinais.

O decoder possui estrutura simétrica ao encoder, sendo composto por duas camadas LSTM com 64 e 128 neurônios, respectivamente, seguidas por uma camada de saída que reconstrói as sequências originais com o mesmo número de variáveis da entrada. Essa configuração permite que o modelo tente reproduzir, a partir da representação latente, o comportamento observado nos dados de entrada.

A disposição das camadas na forma 128–64–32–64–128, foi adotada pois segundo Lima (2021), os autoencoders tradicionalmente apresentam uma arquitetura simétrica, na qual o codificador (encoder) e o decodificador (decoder) possuem estruturas equivalentes, sendo separados por uma camada central de menor dimensionalidade, denominada *bottleneck*. Essa camada impõe uma restrição ao modelo, forçando-o a aprender e representar apenas as características mais relevantes dos dados de entrada.

O desempenho do Autoencoder é avaliado por meio do erro de reconstrução, calculado a partir da diferença entre os sinais de entrada e os sinais reconstruídos, sendo valores elevados interpretados como indícios de comportamento anômalo.

4.6 Erro de reconstrução

O funcionamento do Autoencoder LSTM baseia-se na capacidade de aprender o comportamento normal dos sinais ao longo do tempo e reproduzi-lo na saída. Quando o Autoencoder recebe uma janela (um bloco de 120 amostras), ele tenta reconstruir esse mesmo bloco. Ou seja, avalia se a tendência original está

acontecendo. Se o comportamento da janela for normal, o modelo consegue reconstruir com facilidade. Se for anômalo, ele reconstrói mal, gerando um alto erro.

O modelo utiliza a seguinte equação para calcular:

$$MSE = \frac{1}{n} \sum (X_i - \hat{X}_i)^2$$

1. X_i = valor original da amostra i
2. \hat{X}_i = valor reconstruído pelo Autoencoder
3. n = número total de valores dentro da janelas
4. $(X_i - \hat{X}_i)^2$ = erro de reconstrução de cada ponto

4.7 Avaliação do Desempenho do Modelo

Antes da análise dos resultados, faz-se necessária a definição dos critérios utilizados para avaliar o desempenho do RNA. Considerando a natureza do problema e o desbalanceamento entre classes, são adotadas métricas pontuais e curvas de desempenho que permitem avaliar tanto a capacidade de identificação de eventos anômalos quanto o equilíbrio entre detecções corretas e alarmes falsos. A seguir, são apresentados os conceitos utilizados para essa avaliação.

4.7.1 Curva ROC

A Curva Característica de Operação do Receptor (*Receiver Operating Characteristic* – ROC) é utilizada para avaliar o desempenho de classificadores binários por meio da relação entre a taxa de verdadeiros positivos (sensibilidade) e a taxa de falsos positivos, considerando diferentes valores de limiar de decisão. O desempenho global do classificador é sintetizado pela Área Sob a Curva (*Area Under the Curve* – AUC), na qual valores mais próximos de 1 indicam maior capacidade do modelo em discriminar entre as classes. A métrica AUC avalia o comportamento do classificador ao longo de todos os limiares possíveis, fornecendo uma medida de desempenho adequada para comparação entre modelos, por ser menos sensível à distribuição das classes e à escolha do limiar de decisão (LI, 2024).

4.7.2 Sensibilidade (*Recall*)

A sensibilidade, também denominada *recall*, mede a capacidade do modelo em identificar corretamente os eventos anômalos. Essa métrica corresponde à razão entre o número de verdadeiros positivos e o total de eventos que realmente pertencem à classe anômala. Valores mais elevados de sensibilidade indicam menor ocorrência de falhas não detectadas, sendo uma métrica fundamental em problemas de detecção de anomalias e manutenção preditiva (POWERS, 2011; RUFF *et al.*, 2021).

4.7.3 Curva *Precision–Recall*

A Curva Precisão–Revocação (*Precision–Recall*) relaciona a precisão do modelo com a sensibilidade (*recall*) para diferentes valores de limiar de decisão. Essa curva é especialmente adequada para cenários com desbalanceamento entre classes, como em problemas de detecção de anomalias, nos quais a ocorrência de eventos anômalos é significativamente inferior à de comportamentos normais. A área sob a curva *Precision–Recall* fornece uma medida complementar do desempenho do modelo, com foco na identificação correta de eventos raros, sendo mais informativa do que a curva ROC nesses contextos (SAITO; REHMSMEIER, 2015).

4.7.4 F1-score

O F1-score é uma métrica que combina precisão e sensibilidade em um único valor, sendo calculado a partir da média harmônica entre essas duas medidas. Essa métrica é amplamente utilizada para avaliar o equilíbrio entre a identificação correta de eventos anômalos e a redução de alarmes falsos, sendo particularmente adequada em cenários com desbalanceamento entre classes, nos quais métricas globais podem fornecer interpretações equivocadas do desempenho do modelo (CHICCO; JURMAN, 2020).

4.7.5 Acurácia

A acurácia representa a proporção total de classificações corretas realizadas pelo modelo, considerando tanto janelas normais quanto anômalas. É calculada como a razão entre o número de previsões corretas (verdadeiros positivos e verdadeiros negativos) e o total de amostras avaliadas (CHICCO; JURMAN, 2020)..

Além de outras métricas como a **precisão** que indica a confiabilidade das detecções de anomalia realizadas pelo modelo. Essa métrica expressa a proporção de

janelas corretamente classificadas como anômalas em relação ao total de janelas sinalizadas como anômalas. E a **especificidade** que avalia a capacidade do modelo em reconhecer corretamente janelas normais, sendo definida como a proporção de comportamentos normais corretamente classificados em relação ao total de amostras realmente normais.

4.8 Processo de treinamento

Os treinos e testes da rede foram realizados em um notebook Acer Nitro 5, com processador Ryzen 5 3600, placa de vídeo NVIDIA GTX 1650 e 16 GB de memória RAM. O código foi escrito e executado no *Visual Studio Code*.

No desenvolvimento do projeto, foram utilizadas bibliotecas no ecossistema Python, cada uma com uma função específica dentro do fluxo de processamento, modelagem e avaliação dos dados:

1. **os**

Utilizada para manipulação de diretórios e caminhos de arquivos, permitindo acessar pastas, verificar a existência de arquivos e organizar a leitura dos dados de forma estruturada.

2. **glob**

Empregada para localizar e listar automaticamente conjuntos de arquivos CSV dentro das pastas, facilitando a leitura de grandes volumes de dados distribuídos em múltiplos arquivos.

3. **numpy**

Utilizada para operações numéricas eficientes, manipulação de vetores e matrizes, além de servir como base para diversas operações matemáticas envolvidas no processamento dos dados.

4. **pandas**

Responsável pela leitura, organização e manipulação dos dados tabulares, permitindo o tratamento das séries temporais, seleção de variáveis, limpeza dos dados e integração com outras bibliotecas do projeto.

5. **sklearn.preprocessing (StandardScaler)**

Utilizada para a normalização dos dados, ajustada exclusivamente com base

nos dados considerados normais. Essa etapa garante que as variáveis apresentem escala adequada para o treinamento do modelo, evitando que diferenças de magnitude influenciem o aprendizado da rede.

6. **sklearn.metrics**

Empregada para o cálculo das métricas de desempenho do modelo, como acurácia, precisão, *recall*, *F1-score*, curvas ROC e *Precision–Recall*, permitindo uma avaliação quantitativa dos resultados obtidos.

7. **joblib**

Utilizada para salvar e carregar objetos do projeto, como modelos treinados e escaladores, possibilitando a reutilização do modelo sem a necessidade de novo treinamento.

8. **tensorflow**

Utilizado para a implementação e o treinamento da rede neural, oferecendo suporte eficiente ao aprendizado e à manipulação de modelos baseados em redes neurais.

9. **tensorflow.keras**

API de alto nível utilizada para a construção da arquitetura do autoencoder LSTM, definição das camadas, funções de ativação, processo de treinamento e inferência do modelo.

10. **matplotlib.pyplot**

Utilizada para a geração de gráficos e visualizações, incluindo séries temporais, histogramas do erro de reconstrução, matrizes de confusão e curvas de desempenho, auxiliando na análise e interpretação dos resultados.

4.9 Preparação dos dados

A preparação dos dados foi realizada conforme as etapas descritas a seguir.

1. **Leitura dos arquivos CSV**, organizados em pastas distintas correspondentes a dados de operação normal e a dados contendo eventos anômalos. Essa separação permite que o modelo seja treinado apenas com comportamentos

considerados normais e posteriormente avaliado em situações fora do padrão.

2. **Seleção das variáveis numéricas**, considerando apenas as colunas comuns entre os diferentes conjuntos de dados. Essa etapa garante que todas as janelas de entrada possuam o mesmo conjunto de variáveis, evitando inconsistências na estrutura dos dados fornecidos ao modelo.
3. **Limpeza e imputação dos dados**, aplicando interpolação e preenchimento *forward/backward* para tratar valores ausentes. Esse procedimento é utilizado para remover lacunas nas séries temporais, preservando a continuidade do sinal e evitando a exclusão de trechos relevantes dos dados.
4. **Normalização dos dados**, ajustada exclusivamente com base nos dados normais e posteriormente aplicada aos demais conjuntos. Dessa forma, evita-se o vazamento de informação do conjunto de teste para o treinamento, assegurando que o modelo aprenda apenas o padrão de operação normal da máquina.
5. **Janelamento dos dados**, no qual as séries temporais são segmentadas em janelas contendo 120 amostras, com avanço (*stride*) de 10 amostras entre janelas consecutivas. Esse procedimento permite capturar a dinâmica temporal do sinal, fazendo com que cada janela represente um pequeno trecho contínuo da operação da máquina.

Isso significa que, para cada dia de coleta, foram formados blocos (janelas) onde entram:

- **120 leituras do sensor 1**,
 - **120 leituras do sensor 2**,
 - ... e assim sucessivamente, para todas as **8 variáveis de sensor** mantidas na análise.
6. **Divisão dos dados em conjuntos de treinamento e teste**, considerando as proporções **70/30** e **90/10**, utilizadas para a comparação entre os modelos avaliados.

4.10 Matriz de Confusão

Matriz de confusão é uma tabela que compara o que o modelo previu com o que era de fato. Em cada célula aparece a quantidade de janelas que caíram naquela combinação. Assim, é possível ver:

1. **Verdadeiros Positivos (TP)**: era anomalia e o modelo previu anomalia.
2. **Verdadeiros Negativos (TN)**: era normal e o modelo previu normal.
3. **Falsos Positivos (FP)**: era normal, mas o modelo alarmou (prevê anomalia).
4. **Falsos Negativos (FN)**: era uma anomalia, mas o modelo não apontou.

A matriz de confusão é uma ferramenta fundamental para avaliar o desempenho do modelo, pois permite analisar de forma clara onde o sistema acerta e onde erra. Diferentemente de métricas globais, como a acurácia, ela mostra como esses erros acontecem, destacando principalmente os falsos positivos e os falsos negativos, que possuem impactos diferentes no contexto da manutenção preditiva. Por esse motivo, a matriz de confusão é adotada como a principal base de análise deste trabalho, servindo de apoio para a interpretação das métricas calculadas e para a comparação entre as diferentes configurações de treinamento avaliadas.

4.11 Definição do Limiar (Threshold)

Sobre a decisão do valor de escolha do limiar de decisão, ou seja o que vai ser considerado anomalia ou não para este trabalho, seguiu-se o que é apresentado na literatura, utilizando abordagens estatísticas para a definição desse parâmetro. Conforme descrito no estudo de Moallemi *et al.* (2022), que emprega autoencoders para detecção de anomalias, o limiar pode ser definido a partir de medidas estatísticas do erro de reconstrução obtidas durante o regime normal de operação. Essa abordagem considera a média do erro acrescida de múltiplos do desvio padrão, de forma a estabelecer um limite estatisticamente fundamentado para a separação entre comportamentos normais e anômalos.

Nesse contexto, o limiar de decisão adotado neste trabalho foi calculado conforme a equação:

$$Threshold = \mu + K \cdot \sigma$$

Em que μ representa a média do erro de reconstrução calculada a partir dos dados classificados como normais, σ corresponde ao desvio padrão desse erro, e K é um coeficiente que define o nível de tolerância estatística adotado. Neste estudo, foi utilizado $K=3$, valor indicado pelo estudo de Moallemi. Por estar associado a um intervalo de confiança de aproximadamente 99,7% dos dados, assumindo uma distribuição aproximadamente normal dos erros.

Dessa forma, erros de reconstrução que excedem esse limiar são considerados estatisticamente improváveis de ocorrerem sob condições normais de operação, sendo, portanto, classificados como anomalias. Essa metodologia permite que o limiar seja ajustado automaticamente às características estatísticas do conjunto de dados analisado, tornando o processo de decisão menos dependente de valores arbitrários e mais coerente com o comportamento real do sistema monitorado.

4.11.1 Duas configurações de treinamento da rede

Com o objetivo de analisar a influência da proporção entre dados de treinamento e teste no desempenho do modelo, foram adotadas duas configurações distintas de treinamento da rede. Na primeira configuração, 70% do conjunto de dados é utilizado para treinamento e 30% para teste. Na segunda, 90% dos dados são destinados ao treinamento e 10% à fase de teste.

Como exemplo, no caso da configuração 70/30, os dados correspondentes ao período inicial de operação da máquina até o quarto mês, são utilizados exclusivamente para o treinamento da rede. Os 30% restantes dos dados são então empregados na etapa de teste, sendo para os meses cinco e seis, com o objetivo de avaliar a habilidade do modelo em reconstruir sinais não vistos durante o treinamento e, conseqüentemente, identificar possíveis desvios de comportamento.

A comparação entre as duas configurações possibilita avaliar como o aumento da quantidade de dados de treinamento impacta o erro de reconstrução,

5. RESULTADOS

Nesta etapa serão apresentados os resultados obtidos a partir do treinamento e da avaliação das duas redes neurais propostas, permitindo uma comparação direta entre os modelos. O objetivo desta análise é compreender como diferentes proporções entre dados de treinamento e teste influenciam o desempenho do sistema, especialmente no que se refere à classificação de anomalias.

O processo de definição do limiar de decisão (*threshold*) foi realizado de forma automática pela rede, utilizando as estatísticas do erro de reconstrução calculado a partir dos dados normais durante o treinamento. Como resultado, foram obtidos valores distintos para cada configuração analisada:

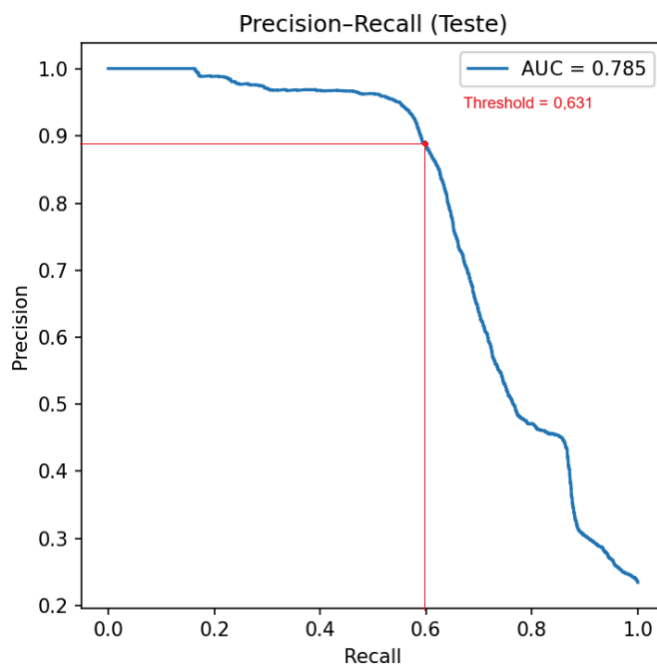
1. **Configuração 70/30:** *threshold* = 0,631
2. **Configuração 90/10:** *threshold* = 0,812

Observa-se uma diferença significativa entre os valores de limiar obtidos nas duas configurações. Essa variação está diretamente relacionada às diferentes médias do erro de reconstrução calculadas para cada modelo durante o treinamento. Esses limiares definem o ponto a partir do qual o erro de reconstrução de uma janela de dados passa a ser interpretado como indicativo de comportamento anômalo, influenciando diretamente a sensibilidade do sistema de detecção.

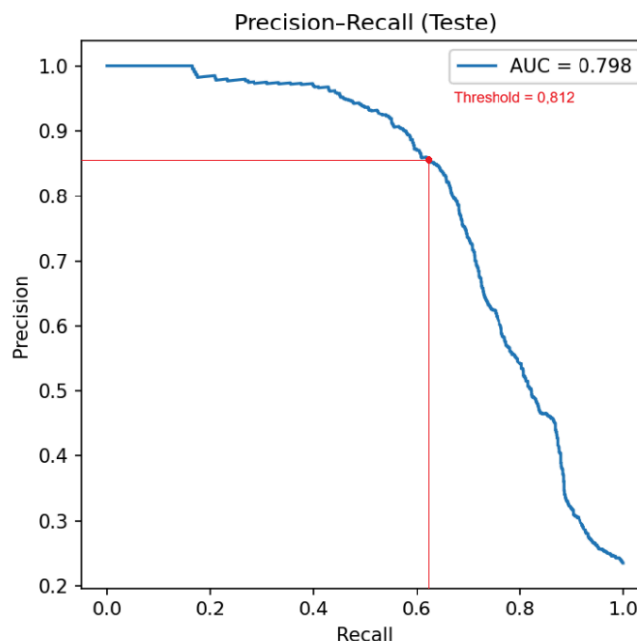
Todo o processo de treinamento da rede e testes levou cerca de nove horas e trinta minutos para a rede de 70/30 e nove horas para a rede 90/10 para apresentar os resultados de cada módulo de treino.

Para verificar se a rede escolheu bem valores de limiar, podemos analisar a curva entre *Precision* e *Recall*. Nas figuras 5 e 6, é apresentado o resultado dessa curva, cuja métrica associada é a área sob a curva (PR-AUC). Diferentemente do *F1-score*, que avalia o desempenho do modelo para um único valor de limiar de decisão, a PR-AUC analisa o comportamento do classificador ao longo de todos os possíveis valores de *threshold*, permitindo uma avaliação global do equilíbrio entre precisão e *recall*.

Essa abordagem possibilita compreender como o desempenho do modelo varia conforme o critério de decisão é ajustado. E analisar os valores escolhidos para limiar.

Figura 5 – Gráfico do *Precision-Recall* 70/30

Para a configuração 70/30 da Figura 5, o threshold de 0,631 posiciona o sistema em um ponto caracterizado por alta precisão ($\approx 89\%$) e *recall* moderado ($\approx 60\%$). Esse comportamento indica que a maioria dos alertas gerados corresponde efetivamente a eventos anômalos, reduzindo falsos alarmes. Entretanto, parte das anomalias reais não é detectada, evidenciando um compromisso entre confiabilidade dos alertas e sensibilidade do sistema.

Figura 6 – Gráfico do *Precision-Recall* 90/10

A configuração 90/10 da Figura 6 apresentou AUC superior (0,798) em relação ao cenário 70/30 (0,785), indicando melhor desempenho global do classificador. Entretanto, ao analisar o ponto operacional definido pelo threshold estatístico (0,812), observa-se que ambos os modelos operam em níveis semelhantes de recall ($\approx 60\%$). Nesse ponto, o modelo 70/30 apresenta maior precisão, enquanto o 90/10 apresenta maior dispersão de falsos alarmes. Esse resultado indica que, apesar do melhor desempenho global do modelo 90/10, o critério estatístico adotado para definição do limiar limitou o aproveitamento do ganho de sensibilidade proporcionado pelo maior volume de dados de treinamento.

Pode-se chegar na conclusão que os resultados dos valores de limiar adotados pela rede são valores satisfatórios, pois geram um bom ponto de equilíbrio para a rede, tendo bom controle entre *precision* e *recall*.

5.1 Análise da Matriz de Confusão

Nesta seção serão apresentados os resultados da avaliação do modelo por meio da matriz de confusão. A partir dessa análise, será possível compreender o comportamento do modelo em diferentes configurações de divisão dos dados de teste.

Figura 7 – Matriz de confusão para teste de 30%

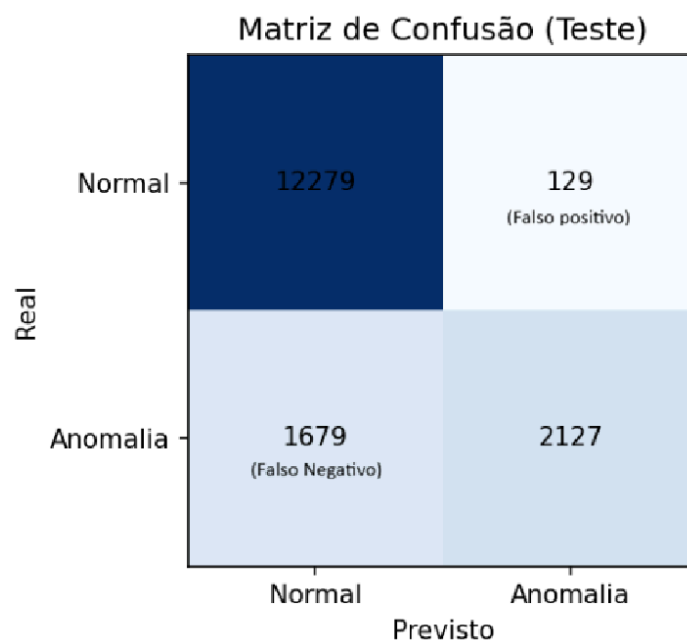


Figura 8 – Matriz de confusão para teste de 10%

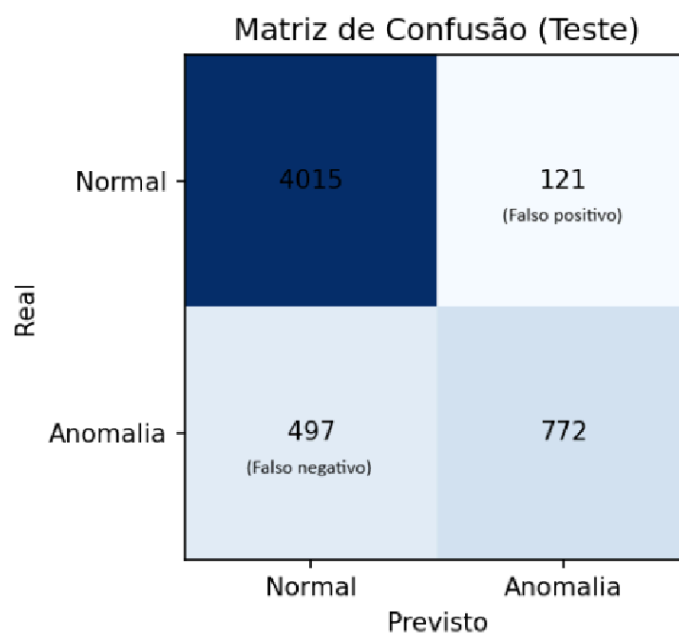


Tabela 1 – Comparação entre os diferentes resultados

Métrica	Teste 30%	Teste 10%
Acurácia	88,33%	88,98%
Precisão	86,45%	94,28%
Recall	55,97%	60,84%
Especificidade	97,08%	98,86%

Após o primeiro treinamento da figura 7 utilizando 30% dos dados para teste, foi obtido uma taxa de **acurácia** de 88,33%. Essa taxa representa o percentual de previsões corretas feitas pelo modelo em relação ao total de amostras avaliadas. Para exemplificar, suponha que o modelo tenha avaliado 100 amostras no conjunto de teste. Se ele classificou corretamente 88 dessas amostras e errou 12, a acurácia será:

$$Acurácia = \frac{88}{100} = 0,8833 = 88,33\%$$

Na outra configuração, empregando na figura 8 de 10% dos dados para teste, o desempenho não apresentou melhora significativa. A acurácia atingiu 88,98%, acompanhada de uma pequena redução nas taxas de erro. Como a variação é mínima podemos chegar a conclusão que nesta métrica ambos chegaram no mesmo resultado.

Ao analisar a métrica de **precisão**, observa-se um aumento na configuração 90/10, que passou de 86,45% para 94,28%. De forma prática, isso significa que, a cada 100 vezes que o modelo indica a presença de uma anomalia, aproximadamente 94 correspondem de fato a situações anômalas, enquanto apenas cerca de 6 seriam alarmes incorretos.

Em comparação, na configuração 70/30, esse número de alertas incorretos é maior. Assim, o modelo tende a alertar com mais frequência situações que, na realidade, ainda se comportam como normais. Na prática operacional, isso pode ser entendido como um sistema que chama a atenção do operador mais vezes sem necessidade.

Portanto, a maior precisão observada no cenário 90/10 representa um comportamento mais confiável no momento em que o sistema decide emitir um alerta, reduzindo a ocorrência de falsos alarmes e tornando a interpretação dos avisos mais objetiva.

Em relação ao **recall**, também é observada uma melhora no cenário 90/10, com o valor aumentando de 55,97% para 60,84%. De forma intuitiva, essa métrica

indica quantas anomalias reais o modelo consegue identificar corretamente.

Essa diferença evidencia que, no cenário 90/10, o modelo apresenta maior capacidade de identificação de padrões anômalos no processo. Na prática, isso reduz a chance de uma anomalia ocorrer sem que o sistema emita qualquer tipo de alerta.

Analisando a **especificidade**, observa-se um aumento de 97,08% para 98,86% no cenário 90/10. Essa métrica está relacionada à capacidade do sistema de reconhecer corretamente situações normais, evitando classificá-las como anomalia.

Em um contexto operacional, essa diferença pode ser entendida como um sistema que tende a interromper a operação apenas quando necessário. Ou seja, há menos alertas gerados em momentos em que o equipamento está funcionando normalmente, reduzindo intervenções desnecessárias e aumentando a confiança do operador nas sinalizações emitidas.

Para complementar a análise das métricas de classificação, a Figura 9 apresenta a comparação entre os principais tipos de erro do sistema, considerando as diferentes proporções de dados utilizados nos testes.

Figura 9 – Comparação falso alarme e falha não detectada.

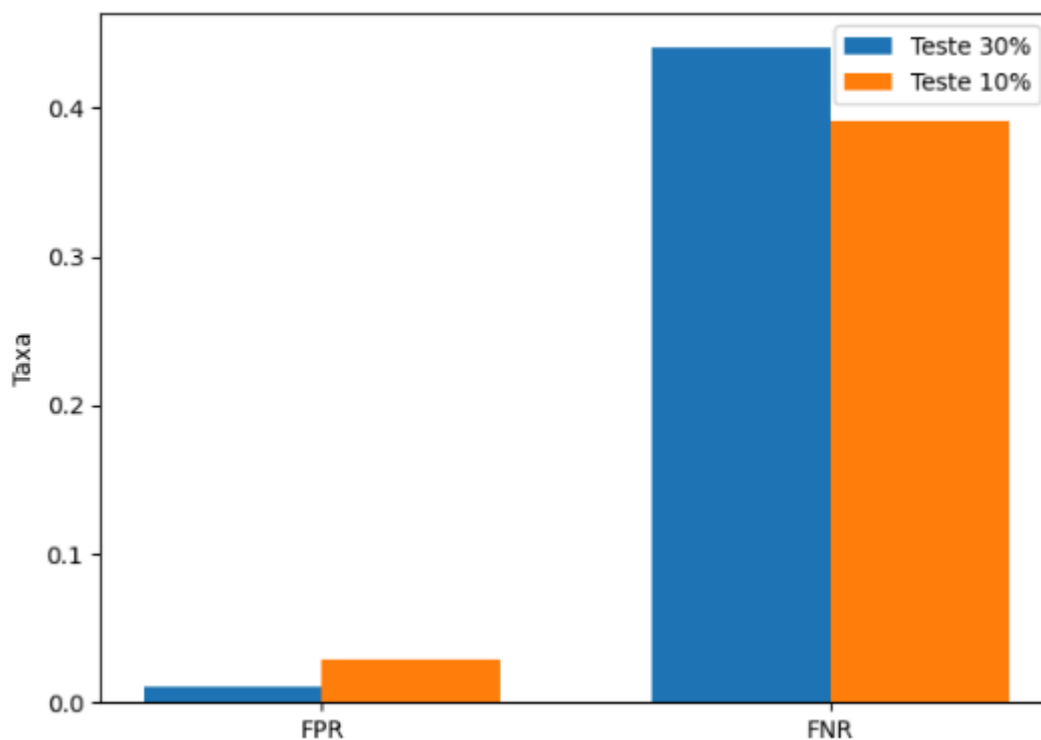


Tabela 2 – Comparação de erro do sistema.

FPR – Taxa de falso alarme	FNR – Taxa de falha não detectada
Teste 30% ≈ 1,04%	Teste 30% ≈ 44,11%
Teste 10% ≈ 2,93%	Teste 10% ≈ 39,16%

A Tabela 2 apresenta a comparação entre as taxas de falso positivo e falsos negativos do sistema. Com isto observa-se que o modelo treinado com a configuração 90/10 apresentou aumento da taxa de falso alarme (FPR), passando de aproximadamente 1,04% para 2,93%. Entretanto, esse comportamento é acompanhado por uma redução da taxa de falha não detectada (FNR), que caiu de 44,11% para 39,16%.

Esse comportamento está diretamente relacionado ao maior volume de dados normais disponibilizado durante o treinamento na configuração 90/10. A maior quantidade de amostras permite que o modelo aprenda de forma mais consistente os padrões típicos de operação, aumentando sua capacidade de identificar desvios sutis no comportamento do processo.

5.2 F1-Score

Nesta seção é apresentado o cálculo do F1-score para as duas configurações de teste adotadas no estudo (30% e 10%). O objetivo é comparar o equilíbrio entre precisão e *recall* em cada cenário, utilizando diretamente os valores obtidos a partir das respectivas matrizes de confusão. Sendo que a fórmula utilizada é a seguinte:

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Caso 1 — Teste de 30%, Valores da matriz:

1. TP = 2127
2. FP = 129
3. FN = 1679

$$\text{Cálculo: } F1 = \frac{2 \cdot 2127}{2 \cdot 2127 + 129 + 1679} = \frac{4257}{6062} = 0,7017 = 70,17\%$$

Caso 2 — Teste de 10%, valores da matriz:

1. TP = 772
2. FP = 121
3. FN = 497

$$\text{Cálculo: } F1 = \frac{2 \cdot 772}{2 \cdot 772 + 121 + 497} = \frac{1544}{2162} = 0,7142 = 71,42\%$$

A análise do F1-score evidencia diferenças importantes no comportamento dos modelos avaliados. No cenário com 30% dos dados destinados ao teste, o modelo apresentou um comportamento mais restritivo na classificação de anomalias, caracterizado por uma menor taxa de falsos alarmes, porém com maior quantidade de falhas não detectadas. Esse fator impactou negativamente o valor do F1-score, uma vez que o *recall* foi reduzido em função do elevado número de falsos negativos.

Por outro lado, o modelo avaliado com 10% dos dados para teste apresentou maior sensibilidade à detecção de anomalias, reduzindo a taxa de falhas não detectadas, ainda que ao custo de um pequeno aumento na taxa de falsos alarmes. Esse equilíbrio entre precisão e *recall* resultou em um F1-score ligeiramente superior, indicando desempenho global mais adequado para o problema de detecção de anomalias.

Dessa forma, a análise do F1-score reforça que a escolha do modelo mais apropriado deve considerar o perfil de risco da aplicação, especialmente em ambientes industriais onde a não detecção de falhas pode representar impactos significativos à operação.

5.3 Distribuição dos erros

Os gráficos da figura 10 e 11 mostram a distribuição do erro de reconstrução calculado para cada janela de dados individualmente, após o treinamento da rede. É importante destacar que esse erro não representa a evolução do treinamento do modelo ao longo das épocas, nem está relacionado ao erro de otimização durante o

aprendizado da rede. Este gráfico é para complementar a matriz de confusão, para ser possível analisar por outra perspectiva como está o comportamento do erro de reconstrução e o que está sendo considerado como anomalias ou não.

Figura 10 – Gráfico da distribuição dos erros 30%

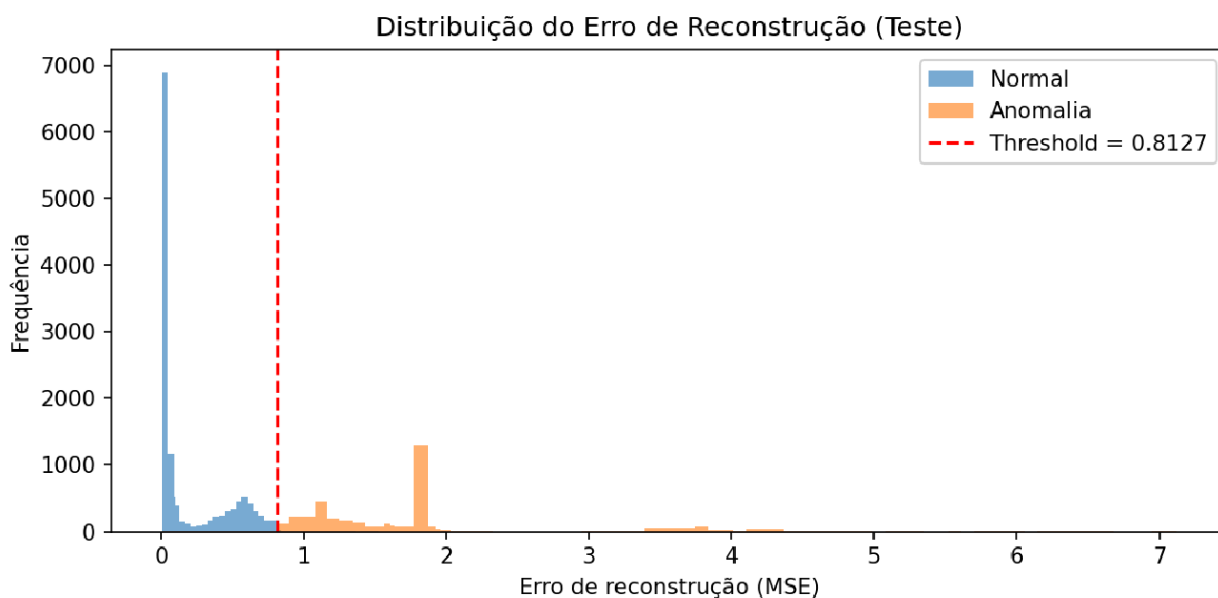
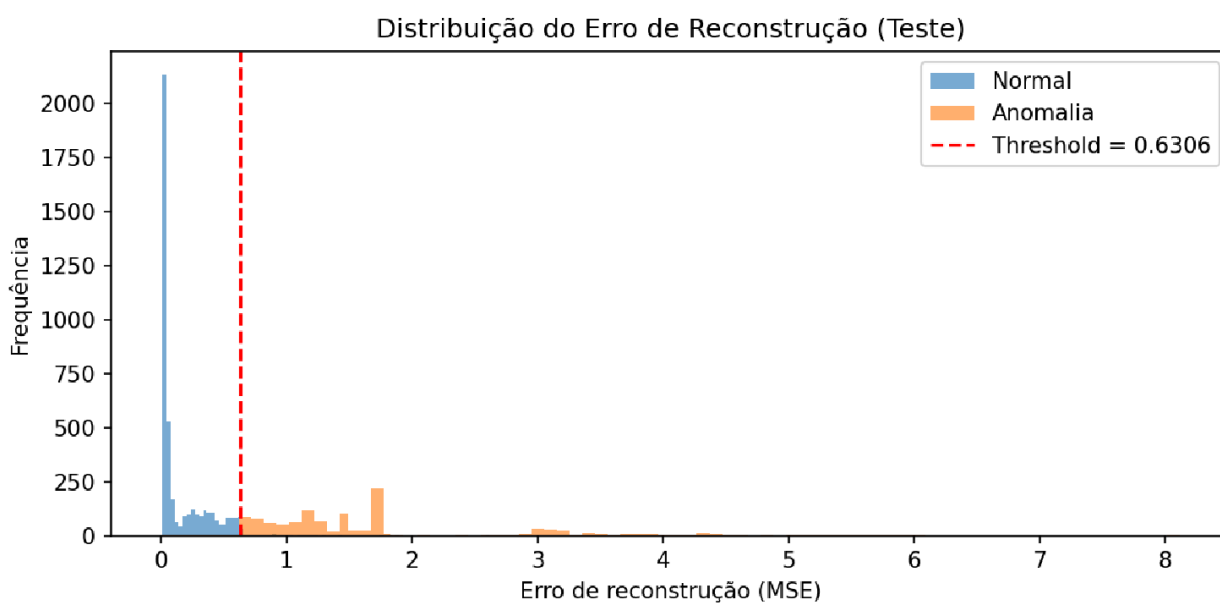


Figura 11 – Gráfico da distribuição dos erros 10%



Nas figuras 10 e 11, observa-se a distribuição do erro de reconstrução para as janelas do conjunto de testes. Os valores localizados à esquerda da linha

tracejada, representados em azul, são as janelas classificadas como normais, enquanto os valores à direita da linha de *threshold*, em laranja, são classificados como anomalias. Dessa forma, os gráficos ilustram de forma visual como a rede neural toma sua decisão de classificação com base no erro de reconstrução de cada janela.

Ao comparar os dois cenários avaliados, é possível observar diferenças principalmente na forma como as anomalias são distribuídas em relação ao limiar de decisão. Isto reforça de como alterações no treinamento da rede pode gerar diferentes resultados.

Observa-se uma diferença significativa no formato da distribuição do erro de reconstrução da classe normal entre os dois cenários avaliados. No modelo treinado com menor volume de dados normais Figura 10, a distribuição apresenta um pico mais concentrado e formato mais estreito, indicando menor dispersão do erro e maior consistência nas reconstruções realizadas pelo autoencoder.

Em contrapartida, Figura 11, a distribuição do erro normal apresenta maior espalhamento e menor concentração em torno de valores baixos, refletindo maior variabilidade no desempenho de reconstrução. Esse comportamento está associado à menor quantidade de dados disponíveis para treinamento, o que limita a capacidade do modelo em capturar todas as variações do regime normal de operação.

Na Figura 10, o valor de 0.8127 do *threshold* definido desloca o ponto de decisão para a direita, fazendo com que janelas com erros de reconstrução mais elevados sejam classificadas como anomalias. Esse ajuste resulta em um comportamento mais restritivo do modelo, influenciando a forma como desvios do padrão normal são identificados.

Na figura 11, o limiar de decisão assume um valor menor de 0.6306, alterando o ponto de separação entre janelas normais e anômalas. Nesse caso, variações menores no erro de reconstrução passam a ser consideradas no processo de decisão, modificando a distribuição das janelas classificadas como anômalas.

5.4 Amostras dos erros

Nesta seção são apresentados os resultados obtidos para dias específicos de coleta de dados, dentro dos dias efetivos de operação. São analisados três cenários distintos com o objetivo de ilustrar o comportamento da rede neural em diferentes situações operacionais, ou seja, esses dias não foram escolhidos ao acaso,

foram justamente para observar diferentes comportamentos do erro de reconstrução .

Figura 12 – Gráfico da série do erro de reconstrução 70/30 dia 14/05 .

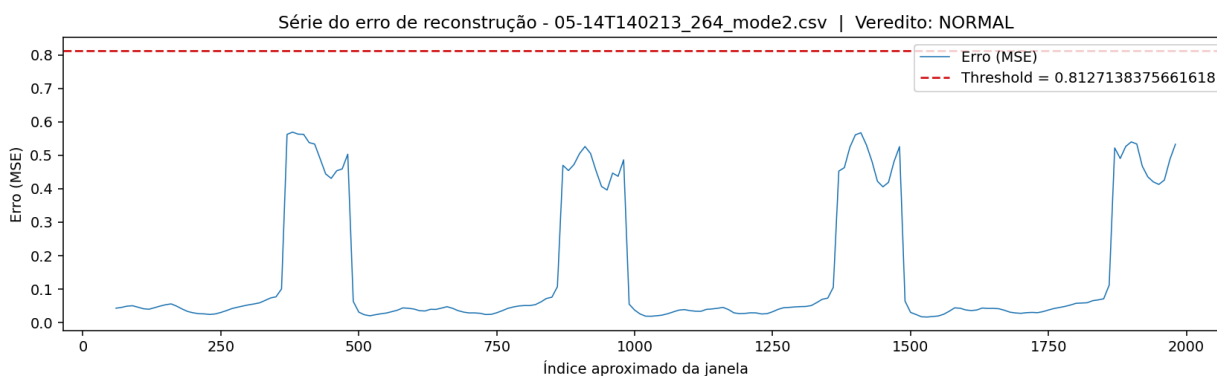
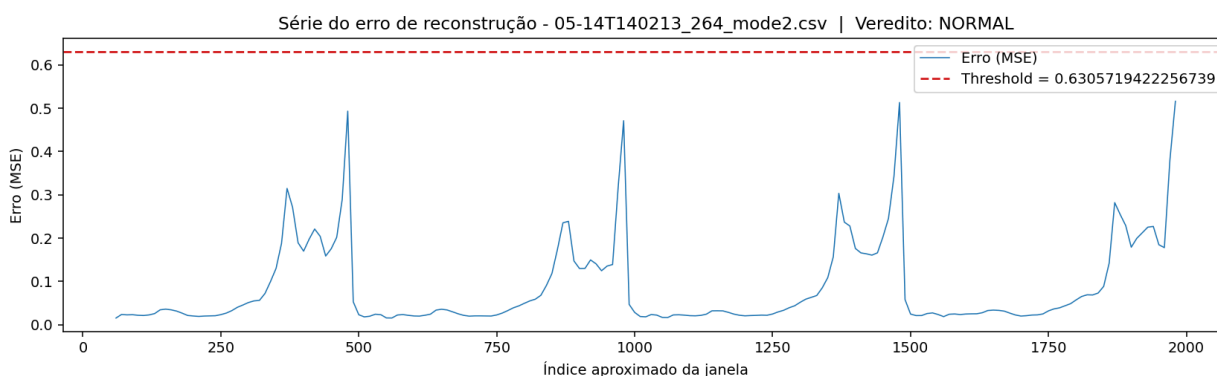


Figura 13 – Gráfico da série do erro de reconstrução 90/10 dia 14/05.



As Figuras 12 e 13 apresentam, respectivamente, os gráficos da série temporal do erro de reconstrução para o dia 14/05, considerando as configurações 70/30 e 90/10. Em ambos os casos, observa-se que o erro de reconstrução permanece abaixo do limiar de decisão ao longo de toda a série, resultando no resultado de operação normal para esse dia.

Embora o comportamento geral seja semelhante nos dois cenários, é possível notar pequenas diferenças no comportamento do erro de reconstrução. No caso da configuração 70/30 (Figura 12), o erro apresenta variações mais suaves e um comportamento mais estável ao longo do tempo. Já na configuração 90/10 (Figura 13), observa-se maior variabilidade nos valores do erro, ainda que sem ultrapassar o limiar definido.

Essas diferenças indicam que, mesmo quando resultado da rede é o mesmo (normal), alterações na configuração do modelo influenciam a forma como o erro de reconstrução se distribui ao longo da série temporal, ou seja, o resultado foi

como normal porém os gráficos apresentaram diferenças entre si.

Figura 14 – Gráfico da série do erro de reconstrução 70/30 dia 22/06.

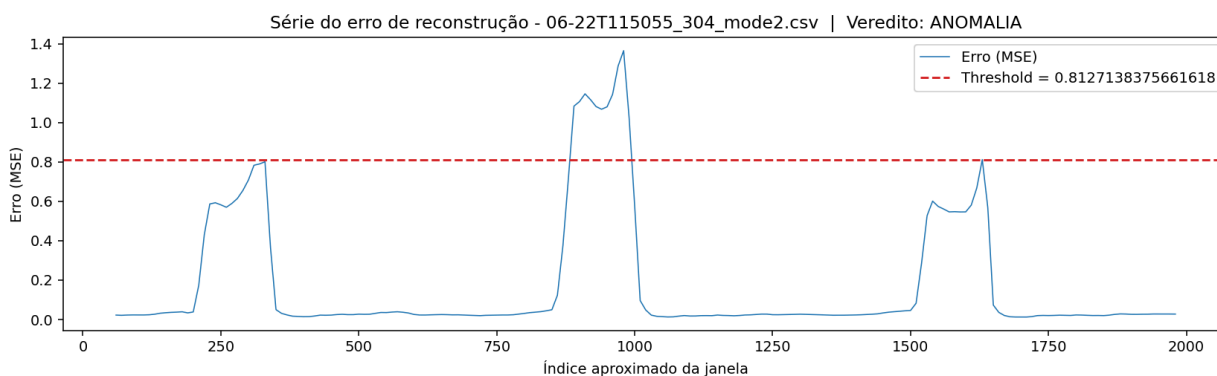
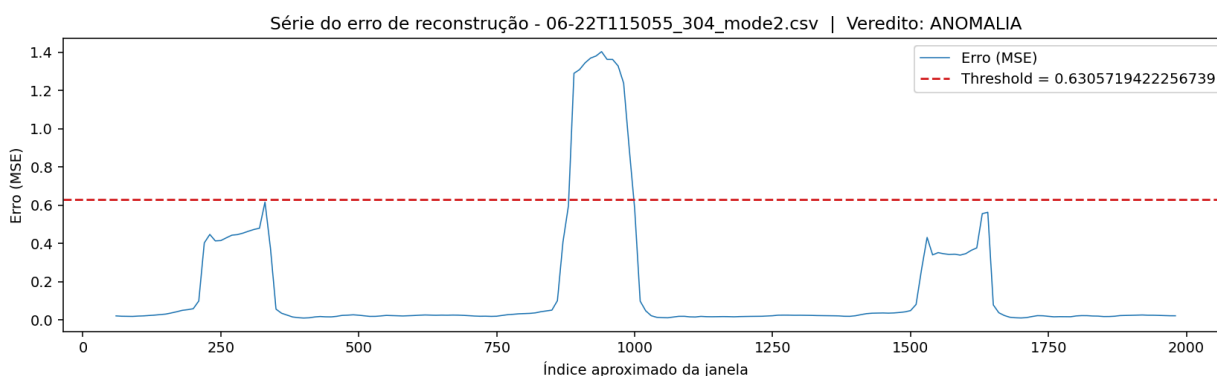


Figura 15 – Gráfico da série do erro de reconstrução 90/10 dia 22/06.



Já nas figuras 14 e 15, o erro de reconstrução ultrapassa o limiar de decisão em parte da série, resultando na conclusão de operação anômala. Observa-se que o cruzamento do *threshold* ocorre principalmente na região central dos gráficos. Esse padrão é consistente nas duas configurações, reforçando a classificação do dia como anômalo.

Apesar da mesma conclusão, nota-se diferença no comportamento do erro de reconstrução em comparação com o exemplo anterior. Na configuração 90/10, o erro apresenta variações mais suaves e um perfil mais estável quando comparado à configuração 70/30, que exhibe picos mais acentuados ao longo da série.

Figura 16 – Gráfico da série do erro de reconstrução 70/30 dia 15/06.

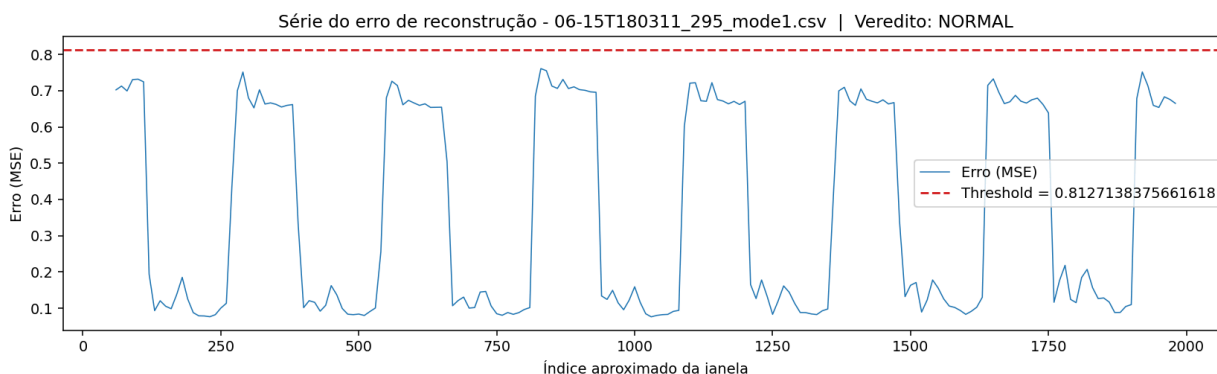
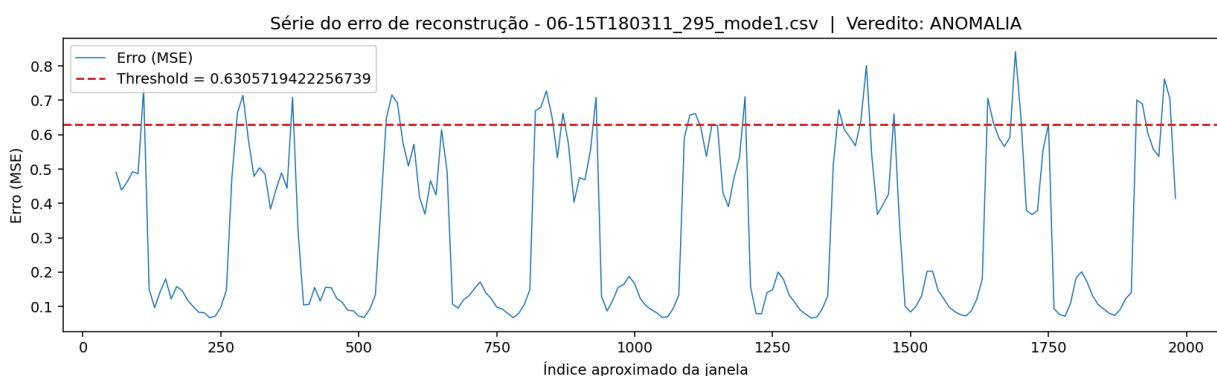


Figura 17 – Gráfico da série do erro de reconstrução 90/10 dia 15/06.



No caso das figuras 16 e 17, observa-se que os modelos apresentam resultados distintos para o mesmo dia: enquanto a configuração 70/30 classifica o comportamento como normal, a configuração 90/10 indica a ocorrência de anomalia. Na Figura 16 correspondente à configuração 70/30, o erro de reconstrução permanece abaixo do limiar de decisão ao longo de toda a série, apesar das variações observadas. Já na Figura 17, configuração 90/10, parte das janelas ultrapassa o *threshold*, resultando na classificação do dia como anômalo. Essa diferença evidencia o impacto de se treinar o modelo de formas distintas e diferentes valores de limiar. Além de diferentes tamanhos de disponibilidade de dados para treinamento. Pequenas alterações na forma como o modelo é treinado refletem em diferentes resultados.

5.5 Resultado da rede

Nesta seção serão apresentados os resultados do que foi classificado como anomalia e como normal pela rede, nos meses 5 e 6. A tabela apresenta, para cada dia analisado, as seguintes informações: a quantidade de arquivos do banco de

dados disponíveis naquele dia, o total de janelas de dados coletadas (ou seja, o número total de amostras analisadas) e quantas dessas janelas foram classificadas como anômalas, de acordo com o tipo de treinamento utilizado.

Tabela 3 – Tabela de diagnóstico 70/30

Dia	Quantidade de arquivos	Janelas totais	% anomalias	Status
14-05	3	579	7,74%	ANOMALIA
15-05	3	579	29,35%	ANOMALIA
16-05	4	772	21,80%	ANOMALIA
22-05	4	772	6,86%	ANOMALIA
26-05	6	1.158	4,26%	NORMAL
13-06	4	772	2,33%	NORMAL
14-06	3	579	2,85%	NORMAL
15-06	4	772	4,39%	NORMAL
19-06	2	386	1,51%	NORMAL
20-06	4	772	8,81%	ANOMALIA
22-06	6	1.158	6,87%	ANOMALIA
28-06	1	193	5,61%	ANOMALIA
29-06	5	965	8,09%	ANOMALIA

Tabela 4 – Tabela de diagnóstico 90/10

Dia	Quantidade de arquivos	Janelas totais	% anomalias no dia	Status no arquivo final
14-05	3	579	5,55%	ANOMALIA
15-05	3	579	32,57%	ANOMALIA
16-05	4	772	28,41%	ANOMALIA
22-05	4	772	8,83%	ANOMALIA
26-05	6	1158	6,28%	ANOMALIA
13-06	4	772	3,42%	NORMAL
14-06	3	579	3,38%	NORMAL
15-06	4	772	12,87%	ANOMALIA
19-06	2	386	4,07%	NORMAL
20-06	4	772	6,09%	ANOMALIA
22-06	6	1158	7,32%	ANOMALIA
28-06	1	193	7,80%	ANOMALIA
29-06	5	1158	11,64%	ANOMALIA

Em dias como, 14/05, 15/05, 19/05, 22/06 e 29/06, ambas as configurações classificam o dia como anomalia, porém o modelo 90/10 apresenta percentuais mais elevados de janelas anômalas, indicando maior intensidade na detecção dos desvios.

Por outro lado, em dias classificados como normais, como 13/06, 14/06 e 19/06, ambos os modelos apresentam percentuais baixos de anomalias e concordam no resultado final, tendo o mesmo comportamento do sistema nesses períodos.

Algumas diferenças pontuais podem ser observadas na classificação de determinados dias. Um exemplo é o dia 15/06, classificado como normal na configuração 70/30 e como anomalia na configuração 90/10. Esse resultado reflete diretamente a maior sensibilidade do modelo 90/10, que passa a considerar como anômalas variações que, no modelo 70/30, ainda permanecem dentro do limite aceitável.

Situação semelhante ocorre em dias como 26/06, onde o percentual de anomalias no modelo 90/10 é superior ao observado no modelo 70/30, impactando o status final do dia.

6. CONCLUSÃO

A necessidade de monitorar o comportamento de máquinas industriais de

forma contínua tem se tornado cada vez mais relevante, principalmente para reduzir paradas não planejadas e apoiar estratégias de manutenção preditiva. Nesse contexto, este trabalho teve como objetivo avaliar a viabilidade de um método baseado em Autoencoder LSTM para a detecção de anomalias em sinais operacionais de uma máquina embaladora, utilizando dados reais de produção.

O método utilizou um limiar de decisão definido de forma automática, a partir das estatísticas do erro de reconstrução, permitindo classificar as janelas de dados como normais ou anômalas de maneira adaptativa. Para garantir uma análise consistente dos sinais ao longo do tempo, os dados foram segmentados em janelas de 120 amostras, com avanço de 10 amostras entre janelas consecutivas.

Com base nos resultados obtidos, é possível afirmar que o objetivo proposto de projetar um modelo para apoio à manutenção alcançado de forma satisfatória. O sistema desenvolvido demonstrou capacidade de identificar desvios em relação ao padrão normal de operação, conforme evidenciado pelas métricas de desempenho e pelas análises das matrizes de confusão e curvas *Precision-Recall*.

Em relação aos objetivos específicos, observou-se que a etapa de preparação dos dados permitiu estruturar adequadamente as séries temporais dos sensores, viabilizando o treinamento do *autoencoder* apenas com dados normais. O modelo foi capaz de aprender o comportamento típico do processo e gerar erros de reconstrução para a separação entre padrões normais e anômalos. Além disso, a comparação entre diferentes proporções de divisão treino/teste (70/30 e 90/10) evidenciou variações no equilíbrio entre sensibilidade e taxa de falsos alarmes, permitindo avaliar o impacto dessa escolha na estabilidade do sistema.

Apesar dos resultados positivos, os valores de FNR e a sobreposição parcial entre distribuições de erro indicam que o modelo ainda apresenta pequenas limitações, especialmente na detecção de valores de anomalias sutis. Dessa forma, o estudo demonstra a viabilidade da abordagem proposta como ferramenta de apoio à manutenção preditiva, porém também evidencia a necessidade de ajustes adicionais e validações futuras para aplicações em ambiente industrial real. Ou seja, um próximo passo seria buscar diferentes estratégias para refinar a rede. Mesmo que seja necessário usar outro modelo de rede a fim de obter melhores resultados.

REFERÊNCIAS

ABB; Sapio Research. *O Valor da Confiabilidade: Relatório da Pesquisa ABB 2023 – Perspectiva da Indústria sobre Manutenção e Confiabilidade*. 2023.

ABIEF. **Produção e faturamento do setor de flexíveis aumentaram em 2024**. São Paulo, 2025. Disponível em: <https://www.abief.org.br/noticias/producao-e-faturamento-do-setor-de-flexiveis-aumentaram-em-2024>. Acesso em: 25 jun. 2025.

BANDEIRA, H. C.; COSTA, D. O.; LOPES, R. S. **Estudo da manutenção centrada em confiabilidade aplicada a um sistema de refrigeração**. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 36., 2016, João Pessoa. *Anais...* João Pessoa: ABEPRO, 2016.

CHICCO, D.; JURMAN, G. **The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation**. *BMC Genomics*, v. 21, n. 1, p. 1–13, 2020.

DEAN, T. L. **Artificial intelligence: theory and practice**. Menlo Park: Addison-Wesley Publishing Company, 1995.

DEODATO, K.; SOUZA, S. S. F. **Diagnóstico de falhas estruturais em um edifício utilizando inteligência artificial**. *Brazilian Journal of Development*, Curitiba, v. 8, n. 5, p. 41029–41048, 2022.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Cambridge: MIT Press, 2016.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2010.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. **Long Short-Term Memory**. *Neural Computation*, Cambridge, v. 9, n. 8, p. 1735–1780, 1997.

KAGGLE. **One-year industrial component degradation dataset** 2018. Disponível em: <https://www.kaggle.com/datasets/inIT-OWL/one-year-industrial-component-degradation>. Acesso em: 25 jun. 2025.

KANG, J.-L. **Visualization analysis for fault diagnosis in chemical processes using recurrent neural networks**. *Journal of the Taiwan Institute of Chemical Engineers*,

Amsterdam: Elsevier, v. 112, p. 137–151, 2020.

KOUTSOUPAKIS, E.; STOILAREVAS, N. **A hybrid deep learning approach for bearing fault diagnosis using continuous wavelet transform and attention-enhanced spatiotemporal feature extraction.** *Sensors*, Basel, v. 24, n. 9, p. 2712, 2024. DOI: 10.3390/s24092712. Disponível em: <https://www.mdpi.com/1424-8220/24/9/2712>. Acesso em: 31 jul. 2025.

LEI, Y. et al. **Machinery health prognostics: a systematic review from data acquisition to RUL prediction.** *Mechanical Systems and Signal Processing*, v. 104, p. 799–834, 2018.

LEUKEL, J.; GONZÁLEZ, J.; RIEKERT, M. **Machine learning-based failure prediction in industrial maintenance: improving performance by sliding window selection.** *International Journal of Quality & Reliability Management*, v. 40, n. 6, p. 1449–1462, 2023. Disponível em: <https://doi.org/10.1108/IJQRM-12-2021-0439>. Acesso em: 24 jun. 2025.

LIMA, Anderson Ferreira de. **Estudo sobre detecção de anomalias com autoencoders.** 2021. Dissertação (Mestrado em Engenharia) – Universidade Nove de Julho, São Paulo, 2021.

LI, Jing. *Area under the ROC Curve has the most consistent evaluation for binary classification.* PLOS ONE, v. 19, n. 12, e0316019, 23 dez. 2024.

METSO. **Estudo revela o verdadeiro custo das paradas não planejadas na indústria.** Metso, 2024. Disponível em: <https://www.metso.com/pt/news/2024/true-cost-of-downtime/>. Acesso em: 25 jun. 2025.

MOALLEMI, A. et al. Exploring Scalable, Distributed Real-Time Anomaly Detection for Bridge Health Monitoring. *IEEE Internet of Things Journal*, [S.l.], p. 1–1, 2022.

MINE, F. H. **Detecção de falhas em ativos críticos utilizando redes neurais artificiais.** 2024. 51 f. Trabalho de Conclusão de Curso (Especialização em Ciência de Dados) – Instituto de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, 2024.

MONTESUMA, E. F.; MULAS, M.; CORONA, F.; MBOULA, F.-M. N. **Cross-domain fault diagnosis through optimal transport for a CSTR process.** *Chemical Engineering Research and Design*, p. 1–6, 2022.

OCME. **Vega HT – shrink-wrapping machine for medium-high speed.** 2025. Disponível em: <https://www.ocme.com/en/vega-ht-p271>. Acesso em: 28 jul. 2025.

ORIVES, T. L. **Realidade aumentada na manutenção: um processo de colaboração.** 2019. Artigo (Especialização em Manufatura Avançada) – Universidade do Vale do Rio dos Sinos, São Leopoldo, 2019. Disponível em: http://www.repositorio.jesuita.org.br/bitstream/handle/UNISINOS/9649/Tiago%20da%20Luz%20Orives_.pdf. Acesso em: 5 jul. 2025.

POWERS, D. M. W. **Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation.** *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2011.

RUFF, L. et al. **A unifying review of deep and shallow anomaly detection.**

Proceedings of the IEEE, v. 109, n. 5, p. 756–795, 2021.

SAITO, T.; REHMSMEIER, M. **The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets.** *PLoS ONE*, v. 10, n. 3, e0118432, 2015.

SANTOS, N. P.; SILVA, L. C.; CARDOSO, E. G. **Aplicação da inteligência artificial no diagnóstico de falhas de máquinas.** *Brazilian Journal of Development*, Curitiba, v. 10, n. 11, p. e74593, 2024.

SILVA, Patricia Lopes. *Métodos de inteligência artificial para detecção de falhas industriais aplicados em um sistema de manufatura: uma análise comparativa de desempenho.* Dissertação (Mestrado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2024.

SILVA, J. R.; OLIVEIRA, M. A. Diagnóstico avançado de falhas em sistemas industriais: uma abordagem comparativa. *Journal of Mechanical Engineering*, v. 12, n. 4, p. 45-60, 2022.

