

UNIVERSIDADE FEDERAL DE J
FACULDADE DE ENGENH
PROGRAMA DE PÓS-GRADUAÇÃO EM

Matheus Júnior Silveira D

Matheus Júnior Silveira D

**Aplicação de Autocodificadores Convolucionais p
em Estruturas**

Dissertação de
grama de Pós
Civil da Facu
versidade Fede
sito parcial à
em Engenharia
Estruturas e
Mecânica das

Matheus Júnior Silveira Dalcin

Título: Aplicação de Autocodificadores Convolucionais para Identifica

Dissertação ap
da Universidad
obtenção do
concentração: I

Aprovada em 17 de outubro de 2024.

BANCA EXAMINADORA

Prof. Dr. Flávio de Souza Barbosa - Orientad

Universidade Federal de Juiz de Fora

Prof. Dr. Alexandre Abrahão Cury

Universidade Federal de Juiz de Fora

Dra. Rafaelle Piazzaroli Finotti Amaral

Universidade Federal de Juiz de Fora

Profa. Dra. Graciela Nora Doz de Carvalho

Universidade de Brasília

Prof. Dr. Francisco de Assis das Neves

Universidade Federal de Ouro Preto

AGRADECIMENTOS

Primeiramente, agradeço a Deus pelas oportunidades ao longo desta jornada. Tua graça e misericórdia foram fundamentais durante este processo.

À minha esposa, Talita Dalcin, pelo companheirismo e apoio essenciais para que eu pudesse focar na pesquisa e superação de minha profissão Engenheiro Marcelo Jr. Dalcin e à minha mãe Viviane Dalcin pela inspiração e incentivos ao longo do caminho. À irmã Marcelly Dalcin pela amizade e força nos momentos difíceis.

Ao meu orientador, Prof. Dr. Flávio de Souza Barbosa, pelas ideias e *insights* valiosos que foram fundamentais para o desenvolvimento da minha dissertação. Sua dedicação e conhecimento foram inspiradores e fundamentais para a conclusão da dissertação. Ao meu co-orientador, Prof. Dr. Alexandre Piazzaroli Finotti Amaral, pela contribuição, orientação e compartilhamento de conhecimentos e experiências que enriqueceram este trabalho. A todos que de alguma forma contribuíram para este trabalho, meus agradecimentos.

Aos membros desta banca, Prof. Dr. Francisco de Assis e Profa. Dra. Graciela Nora Doz de Carvalho, pela disponibilidade em avaliar este trabalho.

Aos meus amigos e colegas de profissão Eng.º Leonardo, Eng.ª Júlia Kunsch, Eng.º Clayton Silva, Cássio Motta e todos os demais que passaram momentos ao longo desses anos de mestrado.

Gostaria também de agradecer ao Programa de Pós-graduação em Engenharia Civil (UEFIE) à FAPEMIC (PAPC 23071 036173/2023) pelo apoio financeiro.

RESUMO

O monitoramento contínuo da integridade estrutural é uma das principais preocupações da engenharia moderna, devido a recentes incidentes que evidenciam a importância de se considerar o comportamento dinâmico das estruturas. De forma geral, isso é feito por meio de sensores acoplados às estruturas analisadas, que fornecem dados específicos capazes de identificar o estado da integridade. No presente trabalho tem como foco a aplicação de técnicas de aprendizado de máquina e inteligência artificial, com ênfase em autocodificadores, para avaliar a integridade das estruturas por meio da análise de dados. Quatro estratégias de análise baseadas nos erros de reconstrução e na distância de Mahalanobis dos espaços latentes, obtidos por meio de autocodificadores convolucionais, são utilizadas. A metodologia é avaliada por meio de dois conjuntos de dados. O primeiro conjunto, proveniente do Laboratório de Sinais e Imagens da Universidade de São Paulo, representa uma estrutura com cenários de danos controlados. Por outro lado, o conjunto Z-24, representa uma estrutura real sujeita a variações de carga. Os resultados obtidos evidenciam a eficácia das estratégias em ambientes reais, uma vez que, em ambos os casos, foi possível se detectar danos de forma satisfatória, indicando que a abordagem via CAE pode ser uma alternativa viável.

Palavras-chave: Monitoramento da Integridade Estrutural; Autocodificador. Convolução.

ABSTRACT

Continuous monitoring of structural integrity has been a challenge for many years, with several incidents that highlight the importance of ensuring safe behavior of structures. In general, this monitoring is performed by sensors attached to the analyzed structures, which provide data that can be used for identifying their integrity status. In this context, this paper explores the use of technologies such as machine learning and artificial intelligence, specifically convolutional autoencoders, to assess the integrity of structures based on the collected data. To this end, four analysis strategies based on dynamic signals and the Mahalanobis distance of latent space representations based on convolutional autoencoders, are used. The proposed methodology is evaluated using two data sets. The first set, a double-clamped frame structure from the Images Laboratory of the University of Juiz de Fora, represents controlled damage scenarios. On the other hand, the second application is a real structure subject to variations of external factors. Through the analysis, the effectiveness of the strategies in controlled environments and in real cases, both cases, it was possible to detect structural changes in the data. This demonstrates that the CAE approach can be effective in real structure monitoring.

Keywords: Structural Health Monitoring. Neural Networks. Convolutional Autoencoders. Latent Space. Mahalanobis Distance. Artificial Intelligence.

LISTA DE ILUSTRAÇÕES

- Figura 1 – Acidentes Nacionais e Internacionais
- Figura 2 – Arquitetura de uma rede neural artificial M
- Figura 3 – Funções de Ativação
- Figura 4 – Funcionamento de uma rede neural convolu
- Figura 5 – Arquitetura de uma rede neural autocodific
- Figura 6 – Fluxograma da metodologia proposta
- Figura 7 – Pórtico Bi-Engastado
- Figura 8 – Corte esquemático da estrutura
- Figura 9 – Gráfico de resposta típica de um acelerômet
- Figura 10 – Modelos de treinamento - Domínio do Temp
- Figura 11 – Arquitetura autocodificador convolucional
- Figura 12 – Predição do Sinal
- Figura 13 – Erro Quadrático Médio - Modelo A
- Figura 14 – Erro Quadrático Médio - Modelo B
- Figura 15 – Erro Quadrático Médio - Modelo C
- Figura 16 – Erro Quadrático Médio - Modelo D
- Figura 17 – Matriz de Confusão - Modelo A
- Figura 18 – Matriz de Confusão - Modelo B
- Figura 19 – Matriz de Confusão - Modelo C
- Figura 20 – Matriz de Confusão - Modelo D
- Figura 21 – Distância de Mahalanobis - Estágio 01
- Figura 22 – Distância de Mahalanobis - Estágio 02
- Figura 23 – Distância de Mahalanobis - Estágio 03

Figura 37	–	Predição do Sinal
Figura 38	–	Erro Quadrático Médio - Modelo A
Figura 39	–	Erro Quadrático Médio - Modelo B
Figura 40	–	Erro Quadrático Médio - Modelo C
Figura 41	–	Erro Quadrático Médio - Modelo D
Figura 42	–	Matriz de Confusão - Modelo A
Figura 43	–	Matriz de Confusão - Modelo B
Figura 44	–	Matriz de Confusão - Modelo C
Figura 45	–	Matriz de Confusão - Modelo D
Figura 46	–	Distância de Mahalanobis - Estágio 01
Figura 47	–	Distância de Mahalanobis - Estágio 02
Figura 48	–	Distância de Mahalanobis - Estágio 03
Figura 49	–	Distância de Mahalanobis - Estágio 04
Figura 50	–	Distância de Mahalanobis - Estágio 05
Figura 51	–	Matriz de Confusão - Estágio 01
Figura 52	–	Matriz de Confusão - Estágio 02
Figura 53	–	Matriz de Confusão - Estágio 03
Figura 54	–	Matriz de Confusão - Estágio 04
Figura 55	–	Matriz de Confusão - Estágio 05
Figura 56	–	Distância de Mahalanobis - Acelerômetros C
Figura 57	–	Matriz de Confusão - Acelerômetros Compi
Figura 58	–	Distância de Mahalanobis - Acrescentando
Figura 59	–	Matriz de Confusão - Acrescendo Danos
Figura 60	–	Ponte Z24
Figura 61	–	Predição do Dano

Figura 76	–	Matriz de Confusão - Estágio 01
Figura 77	–	Matriz de Confusão - Estágio 02
Figura 78	–	Matriz de Confusão - Estágio 03
Figura 79	–	Matriz de Confusão - Estágio 04
Figura 80	–	Distância de Mahalanobis - Acelerômetros C
Figura 81	–	Matriz de Confusão - Acelerômetros Compi
Figura 82	–	Distância de Mahalanobis - Acrescentando
Figura 83	–	Matriz de Confusão - Acrescendo Danos
Figura 84	–	Gráfico no domínio da frequência de uma res metro
Figura 85	–	Modelos de treinamento - Domínio da Freq
Figura 86	–	Arquitetura autocodificador convolucional
Figura 87	–	Arquitetura autocodificador convolucional
Figura 88	–	Predição do Sinal - Primeira Otimização
Figura 89	–	Predição do Sinal - Segunda Otimização
Figura 90	–	Erro Quadrático Médio - Modelo A
Figura 91	–	Erro Quadrático Médio - Modelo B
Figura 92	–	Erro Quadrático Médio - Modelo C
Figura 93	–	Matriz de Confusão - Modelo A
Figura 94	–	Matriz de Confusão - Modelo B
Figura 95	–	Matriz de Confusão - Modelo C
Figura 96	–	Distância de Mahalanobis - Estágio 01
Figura 97	–	Distância de Mahalanobis - Estágio 02
Figura 98	–	Distância de Mahalanobis - Estágio 03
Figura 99	–	Distância de Mahalanobis - Estágio 04

Figura 114	– Distância de Mahalanobis - Estágio 01	. . .
Figura 115	– Distância de Mahalanobis - Estágio 02	. . .
Figura 116	– Distância de Mahalanobis - Estágio 03	. . .
Figura 117	– Distância de Mahalanobis - Estágio 04	. . .
Figura 118	– Matriz de Confusão - Estágio 01
Figura 119	– Matriz de Confusão - Estágio 02
Figura 120	– Matriz de Confusão - Estágio 03
Figura 121	– Matriz de Confusão - Estágio 04
Figura 122	– Distância de Mahalanobis - Acelerômetros	C
Figura 123	– Matriz de Confusão - Acelerômetros	Compi
Figura 124	– Distância de Mahalanobis - Acrescendo Danos	
Figura 125	– Matriz de Confusão - Acrescendo Danos	.

LISTA DE ABREVIATURAS

AE	<i>Autocodificador. Do inglês, Autoencoder</i>
ANN	<i>Rede Neural Artificial. Do inglês, Artificial Neural Network</i>
CAE	<i>Autocodificador Convolutacional. Do inglês, Convolutional Autoencoder</i>
CNN	<i>Rede Neural Convolutacional. Do inglês, Convolutional Neural Network</i>
CVAE	<i>Autocodificador Convolutacional Variacional. Do inglês, Variational Convolutional Autoencoder</i>
DAE	<i>Autocodificador Profundo. Do inglês, Deep Autoencoder</i>
DCDAE	<i>Deep Convolutional Denoising Autoencoder</i>
DL	<i>Aprendizado profundo. Do inglês, Deep Learning</i>
ID	<i>Índice de Dano</i>
ML	<i>Aprendizado de Máquina. Do inglês, Machine Learning</i>
MLP	<i>Multi-Layer Perceptron</i>
MSE	<i>Erro Quadrático Médio. Do inglês, Mean Squared Error</i>
SAE	<i>Autocodificador Esparso. Do inglês, Sparse Autoencoder</i>
SHM	<i>Structural Health Monitoring</i>
USP	<i>Universidade de São Paulo</i>
UFJF	<i>Universidade Federal de Juiz de Fora</i>
LIS	<i>Laboratório de Imagens e Sinais</i>

SUMÁRIO

1	INTRODUÇÃO
1.1	CONTEXTO E MOTIVAÇÃO
1.2	OBJETIVOS
1.3	ORGANIZAÇÃO DO TRABALHO
2	REVISÃO DA LITERATURA
3	REFERENCIAL TEÓRICO
3.1	Modelos de Aprendizado Supervisionado
3.1.1	Rede Neural Artificial - ANN
3.1.2	Rede Neural Convolutacional - CNN
3.2	Modelos de Aprendizado Não Supervisionado
3.2.1	Autocodificador - AE
3.2.2	Autocodificador Convolutacional - CAE
3.3	Estratégias para Detecção de Danos
3.3.1	Erro Quadrático Médio - MSE
3.3.2	Distância de Mahalanobis
4	METODOLOGIA
5	APLICAÇÃO 1: PÓRTICO
5.1	Domínio do Tempo
5.1.1	Divisão dos dados
5.1.2	Otimização dos parâmetros
5.1.3	Arquitetura da Rede
5.1.4	Treinamento da Rede
5.1.5	Estratégias para detectar danos

6.2.1	Divisão dos dados
6.2.2	Otimização dos parâmetros
6.2.3	Arquitetura da Rede
6.2.4	Treinamento da rede
6.2.5	Estratégias para detectar danos
7	CONCLUSÕES
7.0.1	TRABALHOS FUTUROS
	REFERÊNCIAS
8	APÊNDICE A
9	APÊNDICE B

1 INTRODUÇÃO

1.1 CONTEXTO E MOTIVAÇÃO

As estruturas civis existentes, como arranha-céus e pontes, necessitam de contínuas inspeções e monitoramento para garantir a segurança e garantir o comportamento dinâmico previsto. A identificação prévia de danos estruturais contribui para proteger a vida útil calculada nesses projetos [1].

Nos últimos anos, múltiplos incidentes em estruturas civis ocorreram. Um exemplo recente foi a colisão de um navio com a ponte Rio-Niterói, ocorrida em 2019. Além disso, houve tremores de terra nas regiões de Miracatu Iguaçu e Itaipava, no Brasil que levantaram preocupações sobre a estabilidade das estruturas. Um evento recente, registrado no Brasil, e com potencial de causar danos significativos ocorreu em 2023, quando um ciclone extratropical, em São Paulo, afetou a região de Bom Jardim da Serra, ultrapassou os 120 km/h.

Embora as normativas brasileiras e mundiais vigentes sejam baseadas em cálculos estruturais [5] [6] [7], incidentes naturais ou causados por eventos extremos podem levar estruturas ao colapso, resultando em perdas materiais e humanas. Exemplos de colapso da ponte em Gênova, Itália, em 2018, e o acidente da ponte do México em 2021, mostrados na Figura 1.

De acordo com Cury e Cremona [8], além dos incidentes naturais, as estruturas também estão sujeitas a mudanças prolongadas e graduais ao longo do seu uso, como corrosão, deformações excessivas, desgastes e fadiga.

Figura 1 – Acidentes Nacionais e Internacionais



(a) Colapso de Viaduto em Gênova, 2018 [9]



(b) Colapso de Ponte em 2021 [9]



(c) Colapso de Ponte no Estado do Amazonas, 2022 [10]



(d) Estrutura de ponte em 2024

Ao implementar um SHM, torna-se possível planejar a manutenção preventiva, preservando a integridade das estruturas de metal. Com esse abandono do SHM, maximiza-se a vida útil das

pode ser influenciada por vários fatores, como mudanças (radiação solar) e operacionais (condições de carga de veículo). Essas metodologias de SHM [16] [17].

Alternativamente uma promissora área de pesquisa em técnicas de monitoramento baseadas em dados dinâmicos do tempo, quer seja no domínio da frequência. Nesses casos os dados pelo SHM podem ser processadas por meio de técnicas de aprendizado de máquina (do inglês *Machine Learning*) para o reconhecimento de danos ou danos estruturais. Uma das técnicas aplicadas em dados é a rede neural artificial (ANN, *Artificial Neural Network*) [18].

Embora essas estratégias tenham demonstrado resultados, muitas pesquisas ainda se concentram em algoritmos de aprendizado supervisionado. No entanto, rotular diversos estágios danificados ou não é caro e custoso, especialmente em aplicações práticas. Isso tem motivado abordagens de aprendizado não supervisionado, que incluem o autoencoder (do inglês *Autoencoder*) [20] [21]. Um AE é uma arquitetura não supervisionada que consiste em duas partes fundamentais: o codificador (encoder) e o decodificador (decoder). O codificador é responsável por transformar a entrada em uma representação de menor dimensionalidade, onde informações importantes da entrada podem ser obtidas, resultando em uma forma manufaturada. Por sua vez, o decodificador tem a função de reconstruir a saída a partir da obtida pelo encoder, gerando uma saída o mais próxima possível da entrada original.

No presente trabalho será analisado um tipo específico de autoencoder: os autoencoders convolucionais (CAE, do inglês *Convolutional Autoencoder*).

1.2 OBJETIVOS

O objetivo geral desse trabalho é avaliar estratégias para a detecção de danos em estruturas. Nesse sentido são apresentados os seguintes objetivos:

- O uso da distância de Mahalanobis como métrica para a detecção de danos;
- A influência no domínio (tempo e frequência) da detecção de alterações estruturais;
- Os parâmetros e arquiteturas dos CAE que indicam a melhor performance.

1.3 ORGANIZAÇÃO DO TRABALHO

O estudo possui a seguinte estruturação dos capítulos:

- Capítulo 1: Destaca a importância do tema abordado e os resultados alcançados nesta dissertação de mestrado;
- Capítulo 2: Apresenta uma revisão da bibliografia sobre integridade estrutural, com destaque para o uso de redes neurais e aprendizado profundo na detecção de danos e análise do comportamento das redes neurais artificiais e auto-redes neurais em relação à sua aplicabilidade no algoritmo proposto;
- Capítulo 3: Aplica os conceitos analisados a um modelo de uma estrutura de concreto armado submetida a um carregamento dinâmico.

2 REVISÃO DA LITERATURA

Com o crescente número de trabalhos envolvendo S, este capítulo tem como objetivo de abordar estratégias na literatura que contribuíram para o desenvolvimento apresentados são voltados para a detecção de danos em pontes, túneis, viadutos e pórticos estruturais.

Em sua tese de doutorado, Finotti [20], avaliou a esparsos (SAE, do inglês *Sparse Autoencoder*) como ferramenta para analisar dados dinâmicos e detectar comportamentos anômalos e uso de aprendizado supervisionado, empregando o algoritmo (SVM, do inglês *Support Vector Machine*) como classificação numérico de viga e à uma ponte monitorada em João autora propôs uma metodologia de detecção de alterações usando o SAE para caracterizar respostas dinâmicas e de controle T^2 de Hotelling, calculadas a partir dos p Um critério automatizado baseado no erro de reconstrução e generalização dos pontos T^2 foi apresentado, mostrando nem sempre garante boa detecção de alterações estruturais não-supervisionado foi aplicada a três casos: um pórtico ponte Z24, e uma torre na Itália. Cabe ressaltar que, nas temperatura foi considerada.

FINOTTI *et al.* [22] investigaram o uso do SAE na por meio da análise de sinais de vibração. A metodolog

Com base nos dados provenientes da ponte Z-24 na Itália, FINOTTI *et al.* [21] apresentaram uma abordagem estruturais, combinando o uso do SAE com o gráfico de caracterizou as respostas dinâmicas das estruturas utilizadas identificou o surgimento de comportamentos anômalos *Shewhart T*, calculado a partir das características extraídas analisou a utilização da Análise de Componentes Principais (*Component Analysis*), como uma técnica de redução de modelos baseados em PCA/T² tenham apresentado resultados com SAE mostrou-se superior na detecção de mudanças e que a metodologia proposta não apenas detectou anomalias em situações analisadas, mas também demonstrou ser menos associados ao processamento de dados, quando comparado disso, a abordagem mostrou-se capaz de lidar com a influência variações de temperatura, sem a necessidade de algoritmos feitos.

O estudo conduzido por FINOTTI *et al.* [23] vê com SAE na detecção de anomalias estruturais. A estratégia utilizada para extrair características significativas dos sinais monitoradas classificadas pela técnica SVM. Esse processo é contextualizado SHM. A eficácia do método proposto foi avaliada utilizando de um modelo numérico de viga e de um viaduto rodoviário obtidos comprovam a capacidade do SAE em extrair parâmetros dinâmicos, ressaltando seu potencial para aplicações no

treinamento rotulados, ou seja, adotando um aprendizado supervisionado. Este trabalho apresenta uma técnica realizando o comparativo da eficiência de modelos de redes neurais de múltiplas camadas (MLP, do inglês *Multi-Layer Perceptron*) e Autoencoders (CAE). O CAE foi especialmente empregado para lidar com dados de vibração em pontes, enquanto o MLP foi utilizado para dados de entrada em 1D. Apesar de ambos serem modelos de redes neurais, o CAE, destacam-se suas componentes, incluindo camadas de convolução, camadas de transpostas, *pooling* e *upsampling*, cuja função é reduzir a dimensionalidade dos dados. O estudo ressalta que o autocodificador convolucional é uma ferramenta valiosa para lidar com dados de vibração em pontes, permitindo a identificação de danos estruturais complexos nos dados. Os resultados obtidos pela aplicação do CAE mostram alta precisão na detecção de danos, consolidando a tecnologia como uma ferramenta valiosa para o SHM em pontes rodoviárias e ferroviárias.

Neto *et al.* [26] avaliaram a aplicação de diferentes modelos de redes neurais para a identificação de danos em estruturas de pórticos bidimensionais em uma ponte Z-24. A análise focou em quatro autocodificadores e na aplicação da estatística T^2 de Hotelling, com o objetivo de detectar danos em três estruturas. Os valores da camada latente dos autocodificadores foram tratados como parâmetros no T^2 , e os resultados foram comparados entre os diferentes modelos estruturais. Os resultados mostraram-se promissores na identificação de danos, e os demais autocodificadores também se destacaram na identificação de danos.

Em contrapartida, Resende *et al.* [27] realizaram a identificação de danos estruturais por meio da aplicação de modelos de redes neurais e extratores de recursos de dados dinâmicos brutos coletados em uma ponte de frequência. A abordagem envolveu a classificação das

Rastin, Amiri e Darvishan [18] expuseram sobre a aplicação de uma rede de detecção de danos estruturais não supervisionada baseada em uma rede convolucional profundo. Os autores discutem a metodologia utilizada no modelo, bem como os resultados obtidos em um modelo de saúde de pontes. O artigo também discute as limitações e possíveis melhorias para trabalhos futuros. Além disso, ele fornece uma visão existente sobre técnicas de SHM e compara o método proposto.

O artigo desenvolvido pelos autores Shang *et al.* [27] foca em extrair características de danos de medições de campo de ruído, levando em consideração as incertezas decorrentes de ruído de fundo. Para tal, empregam um modelo de autocodificador com denoising de ruído (DCDAE, do inglês *Deep Convolutional Denoising Autoencoder*). Eles testaram a estratégia em dados de vibração de uma ponte e concluíram que se pode detectar com sucesso danos estruturais em condições de ruído variáveis. Em síntese, o artigo discute as implicações da abordagem e sugere possíveis direções para pesquisas futuras.

Em outro artigo, Zhang *et al.* [28] apresentam um método para detectar danos em túneis, utilizando uma abordagem não supervisionada baseada em análise variacional convolucional (CVAE, do inglês *Convolutional Variational Autoencoder*) e análise de pacote de *wavelet*. A relevância desse estudo reside na análise significativa na engenharia civil, que é a identificação de danos e o monitoramento quase em tempo real do estado de saúde de estruturas. Além disso, de anormalidades mesmo durante o funcionamento normal. O artigo também aborda sobre o pré-processamento dos dados.

danos moderados. No entanto, destacam que a abordagem precisa e robusto para a detecção de danos nas cordoal dados de aceleração específicos para um veículo em ques

Apresentando avanços significativos, os estudos an a constante evolução das abordagens de SHM em conju Nesse sentido, o presente trabalho compara a eficácia e CAE em dados provenientes de um pórtico estrutural [3] foi identificar automaticamente os diferentes cenários de d o uso do CAE em estratégias SHM.

A Tabela 1 exhibe uma síntese das principais refe deste estudo.

Tabela 1 – Estado da Art

Autores	Obras
Rafaelle Finotti.	Inteligência artificial aplicada ao monitoramento de estruturas: detecção de alterações mecânico-estruturais baseada no uso de redes neurais autocodificadoras esparsas para a caracterização de respostas dinâmicas, 2022.
Rafaelle Finotti; Flávio Barbosa; Alexandre Cury; Roberto Pimen-	Numerical and experimental evaluation of structural changes using sparse auto-encoders and svm applied to dynamic responses. 2021.

Tabela 1 – (continuação)

Autores	Obras
Rafaelle Finotti; Flávio Barbosa; Alexandre Cury; Roberto Pimentel.	Novelty detection using sparse auto-encoders to characterize structural vibration responses, 2022.
Osama Abdeljaber; Onur Avci; Serkan Kiranyaz; Moncef Gabbouj; Daniel J.Inman.	Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. Journal of Sound and Vibration. Elsevier Ltd, 2017.
Valentina Giggioni; Ilaria Venazi; Valentina Poggioni; Alina Baia; Alfredo Milani; Filippo Ubertini.	Deep Neural Networks for unsupervised damage detection on the Z24 Bridge. ECCOMAS Congress, 2022.
Marcos Spínola Neto; Rafaelle Finotti; Flávio Barbosa; Alexan-	Structural Damage Identification Using Autoencoders: A Comparative Study. Buildings, 2024.

Tabela 1 – (continuação)

Autores	Obras
Zhiqiang Shang; Limin Sun; Ye Xia; Wei Zhang.	Vibration-based damage detection for bridges by deep convolutional denoising autoencoder. Structural Health Monitoring, 2020.
Yonglai Zhang; Xiongyao Xie; Hongqiao Li; Biao Zhou.	An Unsupervised Tunnel Damage Identification Method Based on Convolutional Variational Auto-Encoder and Wavelet Packet Analysis. Sensors, MDPI, 2022.
Lucas Resende; Rafaelle Finotti; Flávio Barbosa; Alexandre Cury.	Structural damage detection with autoencoding neural networks. XLIII CILAMCE 2022.
Kanghyeok Lee; Seunghoo Jeong; Sung-Han Sim;	A Novelty Detection Approach for Tendons of Prestressed Concrete Bridges Based on a Convolutio-

3 REFERENCIAL TEÓRICO

As Redes Neurais Artificiais são estruturas computacionais interconectadas que se baseiam, em sua concepção, no funcionamento do cérebro humano. Elas aprendem com exemplos e experiências, utilizando algoritmos para ajustar os pesos das conexões entre suas unidades. As ANNs são uma ferramenta influente no estudo de SHM, e sua aplicação tem mostrado resultados em termos de eficiência e precisão no monitoramento e detecção de danos. Nesse sentido, essa abordagem foi essencial para o desenvolvimento de sistemas de SHM.

No contexto do SHM, as ANNs são capazes de processar grandes volumes de dados sensoriais e realizar análises avançadas de processamento de sinais. Isso é especialmente útil, pois as estruturas de dados podem ser complexas, exigindo a captura de relações não triviais entre as variáveis. As redes neurais se mostram particularmente adequadas para o SHM devido à sua capacidade de detectar sutis e entender comportamentos não lineares, o que possibilita a identificação de danos e a previsão de falhas estruturais.

Ao serem utilizadas no SHM, as ANNs permitem otimizar os processos de monitoramento existentes, reduzir os custos associados e melhorar a confiabilidade dos resultados. Isso se deve ao fato de que os dados coletados de sensores distribuídos nas estruturas, tornando possível identificar padrões de comportamento saudáveis e detectar desvios ou danos. A seguir apresenta-se um resumo sobre a inserção de ANNs em sistemas de SHM, tanto supervisionado e não supervisionado.

3.1.1 Rede Neural Artificial - ANN

No ano de 1962, Rosenblatt desenvolveu um estudo de redes neurais individuais (conhecido como modelo McCulloch-Pitts - uma estruturação de três camadas denominada *Perceptron*), um dos primeiros modelos de rede neural artificial, inspirado nos neurônios biológicos. Contudo, o entusiasmo inicial não persistiu porque uma camada de saída limitava sua capacidade de resolver problemas inadequado para tarefas mais desafiadoras e não lineares.

Essa limitação foi superada com a elaboração do algoritmo de retropropagação (em inglês, *backpropagation*), proposta por Paul Werbos em 1970 e por Hinton e Williams em 1986. A retropropagação é um avanço crucial no treinamento de redes neurais de múltiplas camadas, permitindo a aprendizagem de Perceptrons (MLP). Esse avanço permitiu o aprendizado de problemas linearmente separáveis [32].

No aprendizado supervisionado, o processo de *retropropagação* ajusta os pesos das conexões entre os neurônios na rede, comparando as saídas previstas e as saídas desejadas durante o treinamento. Existem duas fases principais. Primeiro, na fase de propagação para frente, os dados de entrada são alimentados na rede neural e propagados para a camada de saída. Cada neurônio na rede aplica uma função de ativação aos sinais recebidos das conexões de entrada, gerando uma saída. Na segunda fase (*Backward Phase*), o algoritmo compara as saídas previstas (rótulos) dos dados de treinamento, calculando o erro entre as previsões e os rótulos. O erro é, então, retropropagado

unidirecional, da camada de entrada para a camada de saída, assim uma rede *feedforward*. Cada camada oculta da rede é formada por neurônios abstratos e complexos das camadas anteriores, permitindo a extração hierárquica dos dados de entrada.

Figura 2 – Arquitetura de uma rede neural

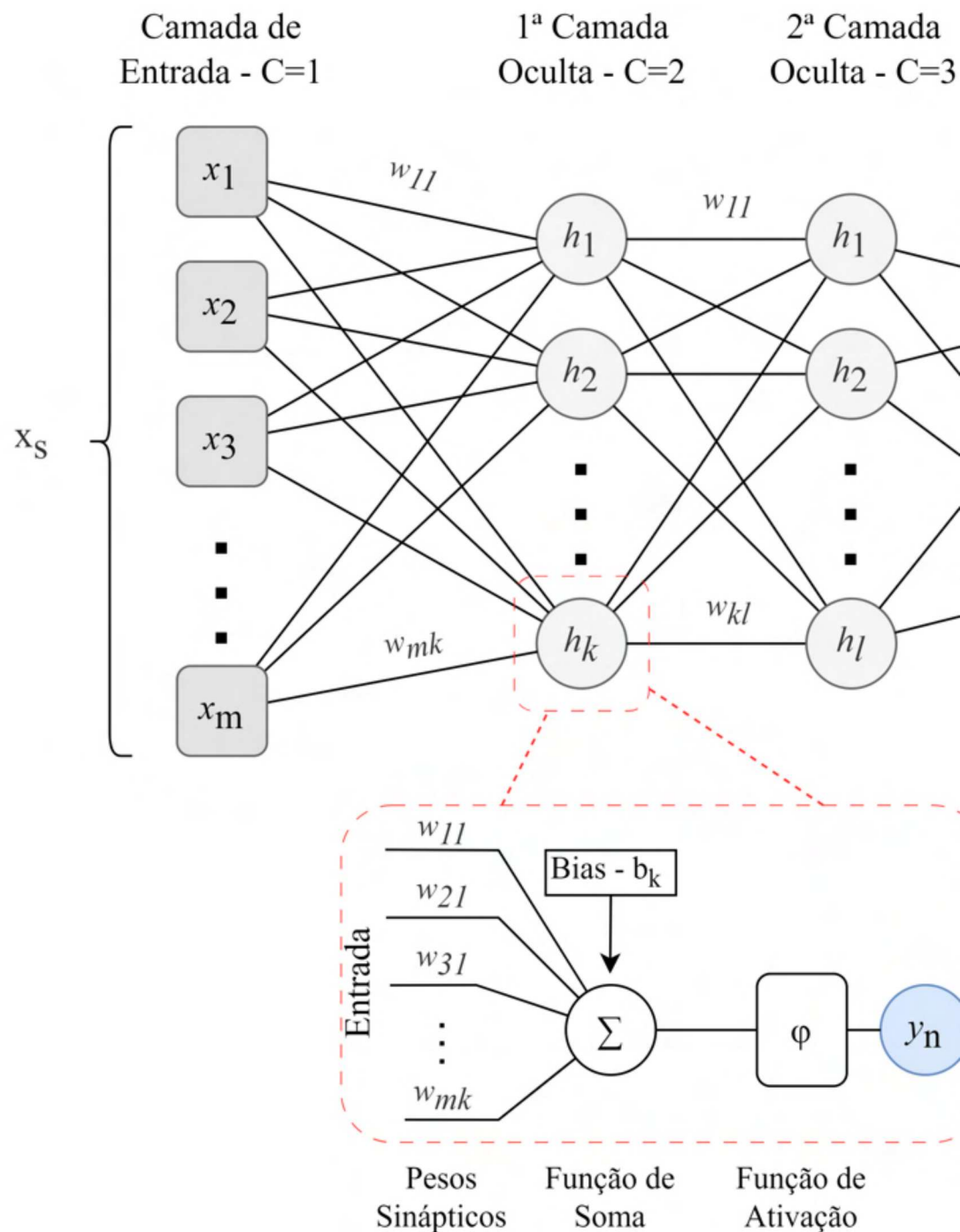
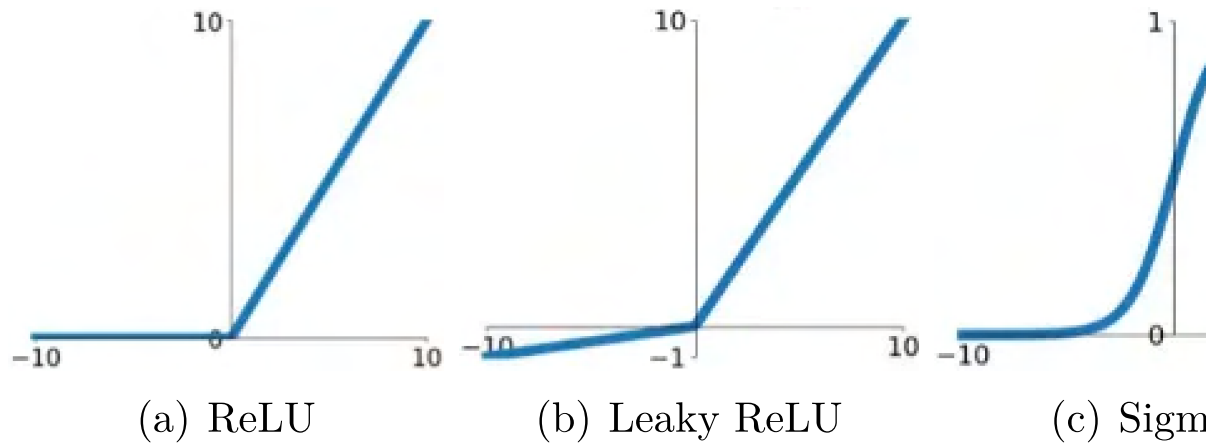


Figura 3 – Funções de Ativação



Fonte: Adaptado, Jadon [35],

Os dados de entrada da rede neural são organizados e S representa o número de parâmetros a serem analisados e n é o tamanho da amostra. Como se trata de aprendizado supervisionado, cada entrada é associada a uma matriz alvo $\mathbf{T}_{n \times S}$, que indica as n classes esperadas ou seja, as saídas desejáveis utilizadas no cálculo do erro.

Embora os dados sejam fornecidos à rede em forma de vetor, cada elemento um a um. Isso é ilustrado na *MLP*, que possui m parâmetros na camada oculta e n saídas possíveis, conforme mostrado na Figura 4. Para cada entrada x_s (onde $s = 1, 2, \dots, S$), existe um vetor de saída y_s (onde y_s é o vetor de saída da rede, correspondente a essa entrada).

Sob a perspectiva do SHM, o valor de entrada x_m representa a m -ésima característica extraída do sinal dinâmico s . O elemento y_n corresponde ao n -ésimo elemento do vetor de saída y , que representa a classe do sinal s classificado pela rede. Por

Por fim, essa versatilidade do MLP e da retropropagação em problemas complexos impulsionou a pesquisa e o desenvolvimento de redes neurais, como a Rede Neural Convolutiva (CNN, *Convolutional Neural Network*), para o processamento de imagens, ampliando significativamente o campo do ML e da IA.

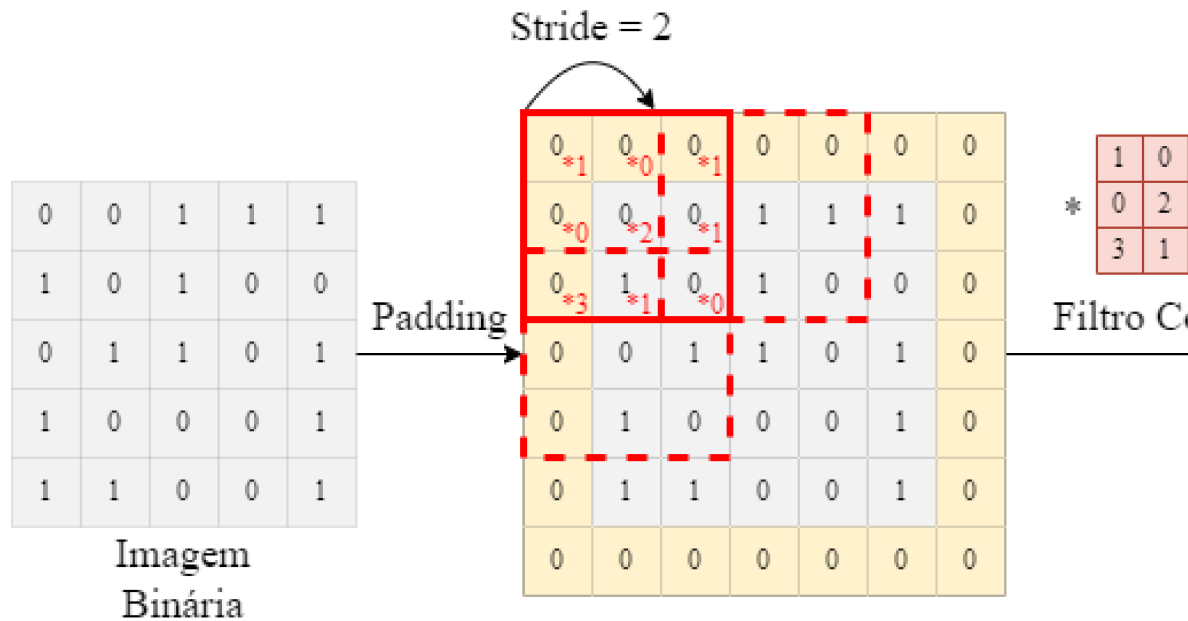
3.1.2 Rede Neural Convolutiva - CNN

Redes neurais convolucionais são um campo especializado programadas especialmente para o processamento de dados de imagens e sinais de vibração. As CNNs são aplicadas em tarefas como detecção de objetos, visão computacional, reconhecimento de imagens, destacando-se por alcançar resultados promissores em tarefas de aprendizado e extração automatizada de parâmetros. É uma técnica influente na análise e compreensão de informações visuais.

A operação de convolução desempenha um papel fundamental e consiste em aplicar filtros (ou *kernels*) convolucionais sobre a entrada. O produto entre os valores do filtro e os valores correspondentes da entrada em cada posição. Em seguida, os produtos são somados para obter o valor resultante em cada posição, formando o que é conhecido como "mapa de características". Este processo é repetido para diversas posições da entrada, resultando em mapas de características que capturam diferentes padrões e informações relevantes da entrada.

A figura 4, demonstra de modo prático o funcionamento de uma CNN, apresentando parâmetros fundamentais como passo (ou *Stride*), filtro (ou *Kernel*) e agrupamento (ou *Pool*).

Figura 4 – Funcionamento de uma rede ne



Fonte: Do Autor, 2023

- Kernel:** Conhecido como filtro, desempenha um papel fundamental na convolução em redes neurais convolucionais. Sua principal função é extrair características relevantes na entrada. Cada valor no filtro representa um peso que é multiplicado pelos dados de entrada para obter resultados para obter o valor de saída em cada posição. Ao aplicar o filtro em todas as posições da entrada, resultando em uma saída específica daquele filtro às características da entrada.

Os filtros são aprendidos durante o treinamento da rede. Os valores são estabelecidos aleatoriamente, e a rede aprende durante o treinamento, utilizando o método de retropropagação para ajustar os pesos que sejam capazes de capturar parâmetros relevantes, como bordas, texturas, padrões, entre outros.

Contudo, resolve-se esse problema acrescentando padding 1D, nas bordas ao redor dos dados antes de aplicar o filtro. Os padding são preenchidos com valores específicos, como zero, para garantir que o filtro seja capaz de se posicionar corretamente nas bordas. Existem dois tipos distintos de *padding*s utilizados: o *Valid Padding* (sem preenchimento), em que nenhuma informação é adicionada na imagem, ocasionando a redução do tamanho da saída; e o *Same Padding* (preenchimento igual), que insere um certo número de padding na imagem, visando obter uma saída da convolução com o mesmo tamanho da entrada. Entretanto, para alcançar isso, o número de *pixels* de padding em cada lado depende do tamanho do filtro.

Sendo assim, o *padding* é importante, pois possibilita a preservação da informação nas bordas da entrada, sendo útil em operações de bordas espaciais precisas, como detecção de bordas ou correlação. Além disso, é crucial ao usar camadas convolucionais em redes neurais, pois, à medida que a informação percorre a rede, a ausência de padding resulta em uma progressiva redução da dimensionalidade de saída, levando a recursos cada vez menores. Isso tem o potencial de causar a perda de informações espaciais e, eventualmente, diminuir a capacidade da rede de aprender características importantes.

- ***Stride***: "Passo", em português, é um atributo fundamental que define o deslocamento do filtro pela entrada durante a operação de convolução. Quanto maior o *stride*, quantos *pixels* são pulados a cada passo da convolução.

Durante a convolução, ao usar um *stride* superior a 1, a saída tem uma dimensão reduzida em relação à entrada. Essa característica é útil para

complexidade computacional, reduzir o tamanho dos mapas de recursos relevantes em diferentes arquiteturas de CNN.

- ***Pooling***: O "agrupamento", é uma etapa primordial para reduzir a dimensão espacial dos mapas de recursos gerados, tornando a representação mais compacta e eficiente. No exemplo 4, foi aplicado o *Max Pooling*. Nesse processo, um filtro é aplicado na camada convolucional com um determinado tamanho. O valor máximo no interior do filtro é mantido, enquanto os demais são descartados. O resultado é um mapa de recursos onde as características mais importantes são retidas. Essa técnica é robusta a pequenas variações na posição das características, contribuindo para a extração de parâmetros dominantes.

Além do *Max Pooling*, tem-se o *Average Pooling*. Nesse caso, na mesma forma que no caso anterior. Contudo, em vez de manter o valor máximo, o cálculo do valor médio dentro do filtro é retido. Isso também reduz a dimensão do mapa de recursos e mantém as características dominantes, colabora para reduzir o impacto do ruído, tornando a rede mais robusta.

A principal vantagem do *pooling* é a redução da quantidade de parâmetros a serem aprendidos. Isso ajuda a evitar o *overfitting*, que ocorre quando o modelo se ajusta aos dados de treinamento e não generaliza bem para novos dados. No entanto, é importante notar que o *pooling* também reduz a informação espacial, pois está reduzindo o tamanho dos mapas de recursos.

3.2 Modelos de Aprendizado Não Supervisionado

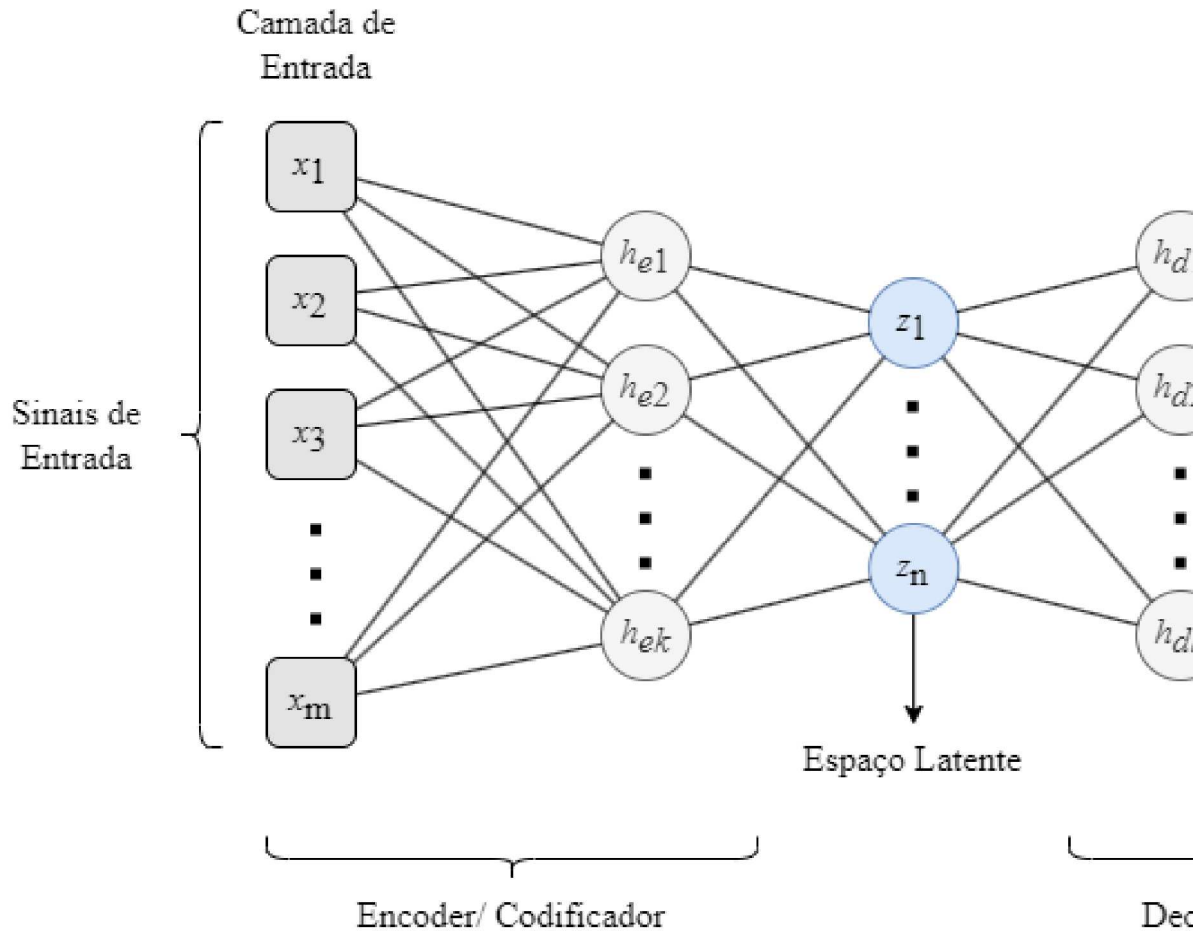
Outro modelo de aprendizado é o não supervisionado, que não necessita de rótulos para ajustar os pesos. O objetivo é descobrir padrões, estruturas ou vínculos entre os dados sem conhecimento prévio esperada, sendo geralmente usado em clusterização. O ajuste é feito de várias maneiras, como o algoritmo *K-means* ou AE, através da minimização da perda de entrada e saída ou da capacidade de reconstrução dos dados.

3.2.1 Autocodificador - AE

Autocodificador, *do inglês Autoencoder*, é um modelo de rede neural *feedforward*, desenvolvido com uma abordagem de aprendizado não supervisionado. Sua principal finalidade é aprender uma codificação, isto é, transformar sinais de entrada de modo que seja possível reconstruí-los a partir de uma representação comprimida (extração de características). Esse modelo é utilizado para tarefas de redução de dimensionalidade. Uma das vantagens com dimensões reduzidas é a rapidez de processamento. Assim como outras redes neurais, o treinamento dos autocodificadores é realizado por otimizadores baseados em retropropagação e gradiente descendente.

Um autocodificador característico consiste em uma rede neural que atua como um codificador (*Encoder*) e um decodificador (*Decoder*). A figura 5 evidencia a arquitetura, onde a primeira seção, o codificador, é programado para transformar os dados de entrada em uma representação comprimida. Geralmente, o codificador é composto por várias camadas densas *fully connected*, que gradativamente reduzem a dimensão dos dados.

Figura 5 – Arquitetura de uma rede neural



Fonte: Do Autor, 2023

A ideia central presente no autoencoder é que o espaço latente contenha as informações mais significativas. Decorrente da técnica de codificação, a camada oculta é formada por características (z_n) das entradas, enquanto o decodificador reconstrói tais entradas na camada de saída (y_m), conforme exemplo

$$x_m = \varphi(w_{nm}x_m + b),$$

$$y_m = \varphi'(w'_{mn}h_n + b'),$$

3.2.2 Autocodificador Convolutacional - CAE

O Autocodificador Convolutacional, do inglês *Convolutional Autoencoder*, substitui as camadas convolutacionais no lugar das camadas totalmente conectadas. Esta abordagem com o aprendizado não supervisionado caracteriza-se por ser não linear. Seu principal objetivo é reduzir a dimensionalidade dos dados de entrada para gerar uma representação comprimida no espaço latente.

A arquitetura do CAE segue uma estrutura análoga à dos modelos totalmente conectados. Aplica-se camadas convolutacionais para condensar as informações e reduzir a dimensão espacial. Para aprimorar a eficiência, é possível utilizar camadas convolutacionais responsáveis por reduzir a dimensionalidade e garantir a preservação da informação. Camadas convolutacionais transpostas são empregadas para gerar a representação comprimida e, opcionalmente, camadas convolutacionais para restaurar a dimensão original da entrada.

As camadas de aplainar e remodelar, do inglês *flatten* e *reshape*, são necessárias para o correto funcionamento do CAE. A camada *flatten* da camada latente, se refere ao processo de converter um vetor multidimensional em um vetor unidimensional, facilitando o processamento no decodificador. Posteriormente, esses dados lineares são **conectados** até o espaço latente. Porém, devido à saída linear, é fundamental inserir a camada *reshape*, que transforma os dados em vetores multidimensionais. Mais adiante, ao analisar as arquiteturas, os termos serão analisados novamente.

Durante o treinamento, o CAE se dedica a aprender a gerar a representação comprimida no espaço latente a partir da representação comprimida no espaço latente.

Embora os CAE sejam tradicionalmente aplicados (por exemplo), o seu uso em sinais unidimensionais com estrutura é simples e pode ser entendido como uma aplicação em imagens, considerando uma das dimensões unitária.

3.3 Estratégias para Detecção de Danos

As redes treinadas, com suas particularidades, podem indicar alterações comportamentais nas estruturas. Neste capítulo, são apresentadas algumas estratégias para análise das reconstruções das redes que são usadas em específico: o Erro Quadrático Médio de reconstrução no espaço latente.

3.3.1 Erro Quadrático Médio - MSE

O erro quadrático médio é uma medida estatística que compara os sinais observados e os sinais previstos por um modelo. Nas estruturas, o MSE pode ser usado para comparar a resposta de uma base em um modelo não danificado) com a resposta real. Quando estão presentes, eles geralmente causam mudanças nas propriedades de rigidez, amortecimento ou massa. Tais alterações se refletem na resposta da estrutura danificada e o MSE pode ser utilizado para identificar danos.

3.4)

$$MSE = L(w, w', b, b') = \operatorname{argmin}_{w, w', b, b'} \frac{1}{m} \sum_{i=1}^m (x_i - v_i)^2$$

onde x_i e v_i denotam o i -ésimo termo do sinal de entrada e saída, respectivamente.

Ambas as abordagens, MSE e a distância de Ma de danos ou alterações estruturais, pois comparam o estado da estrutura com seu estado atual, identificando discrepâncias de danos. No entanto, é importante notar que essas técnicas em conjunto com outras metodologias de análise estrutural são precisas e confiáveis.

4 METODOLOGIA

Tendo em vista alcançar o escopo proposto pelo trabalho, foi utilizado no *software Matlab 2023a*® de um autocodificador convolucional para sinais provenientes de um pórtico ensaiado no laboratório da Ponte Z24. A metodologia do estudo compreendeu as seguintes etapas:

- **Conjunto de Dados:** Utilizou-se dois conjuntos de dados de acordo com a metodologia proposta. A primeira aplicação é proveniente de dados coletados pelos autores FINOTTI *et al.* [30] e a segunda aplicação é proveniente de dados coletados no laboratório da Ponte Z24.
- **Pré-Processamento:** Nesta etapa, os dados de cada aplicação foram utilizados para formar uma matriz contendo os sinais separados por canal e por tipo de dano. Posteriormente, esses sinais são normalizados por canal de sinal pelo módulo da sua respectiva amplitude máxima, resultando em matrizes multidimensionais que variam entre -1 e 1.
- **Tratamento dos dados:** Após os dados serem convertidos para o domínio dos dados (Tempo ou Frequência), sendo analisados tanto no domínio do tempo quanto no domínio da frequência, buscando avaliar se há possibilidade de redução do número de pontos no eixo da frequência, com a finalidade de otimizar o tamanho dos dados.
- **Divisão dos Dados:** Após o tratamento dos dados, os dados são divididos manualmente entre treinamento, validação e teste, utilizando conjuntos fixos, conforme a sistemática de treinamento e teste.
- **Otimização dos parâmetros:** Antes de treinar a rede neural, é necessário otimizar os parâmetros da rede neural, utilizando o método de otimização de gradiente descendente.

- * **Verificação do erro:** Mediante a reconstrução de uma métrica para verificar os erros de treinamento, validação e teste. Valores maiores são atribuídos a dados de origem distintos do treinamento.

– **Estratégia 02:**

- * **Espaço Latente:** Após o aprendizado, a rede treinada vai decompor os novos sinais (Validação). Vale ressaltar que os espaços latentes são utilizados no estágio de dano.

- * **Distância de Mahalanobis:** Por meio da matriz de covariância calcula-se a distância de Mahalanobis para cada novo dado como a distribuição de referência para cada um dos dados dos espaços latentes de treino referencial.

– **Estratégia 03:**

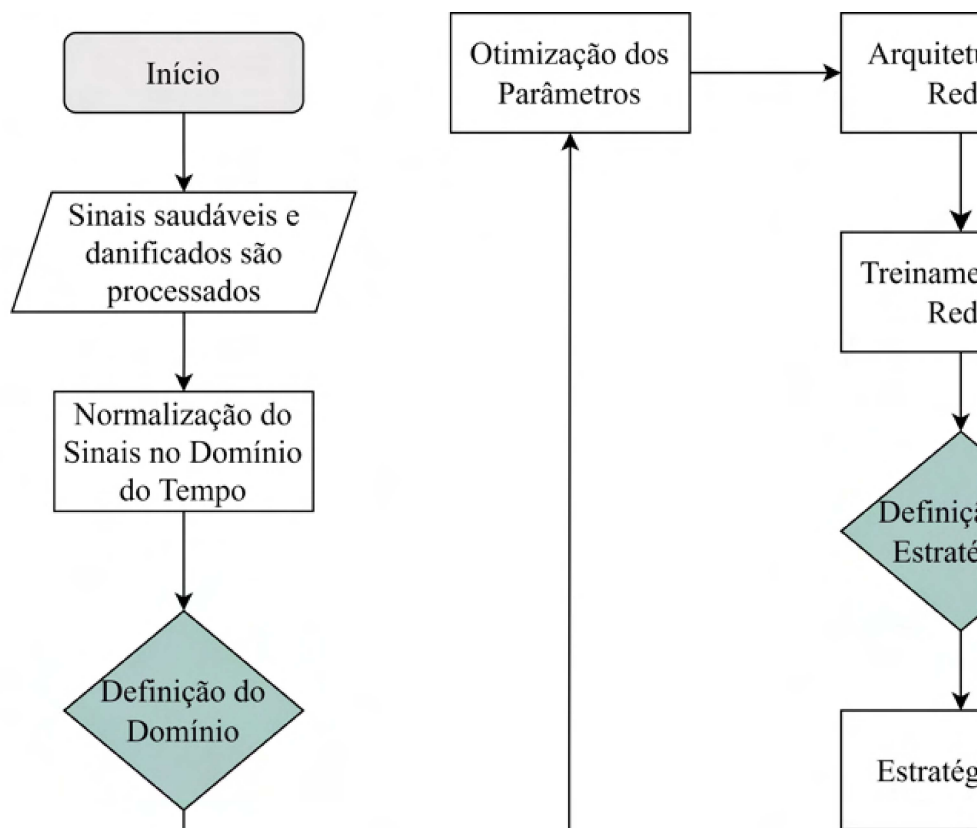
- * **Espaço Latente:** Após o aprendizado, a rede treinada vai decompor os novos sinais (Validação). Nesta estratégia os espaços latentes dos dados de treino são apenas uma matriz por estágio de dano.

- * **Distância de Mahalanobis:** A sistemática é semelhante à estratégia anterior calculando a distância de Mahalanobis nos espaços latentes, tendo como a distribuição de referência a média e covariância, os dados dos espaços latentes de treino referencial.

- **Classificação:** Dados com níveis de MSE/Distância acima de um limite, denominado de limiar entre os tipos de danos, são classificados como um novo estado de danificação. Esse limiar é determinado com base na distribuição de MSE/Distância de Mahalanobis.
- **Matriz de Confusão:** Após a classificação dos dados, os resultados são organizadas em uma matriz que apresenta os resultados positivos e negativos, bem com as métricas de acurácia, precisão e recall.

A figura 6 apresenta de modo sucinto o fluxograma da metodologia proposta neste estudo. É possível verificar as quatro estratégias de treinamento da rede serão evidenciadas no domínio dos sinais.

Figura 6 – Fluxograma da metodologia proposta



Sendo assim, buscando validar a metodologia por meio de uma análise sobre a detecção de danos estruturais em um pórtico biengastado e a ponte Z-24.

O pórtico biengastado representa uma estrutura com dois pontos de apoio, enquanto a ponte Z-24 representa uma estrutura real sujeita a cargas. Essa abordagem visa facilitar o estudo para avaliação e identificação de possíveis danos. A capacidade de detectar danos estruturais é essencial para assegurar a contínua segurança das infraestruturas, em que permite a otimização de recursos na manutenção.

5 APLICAÇÃO 1: PÓRTICO

Neste contexto, se utilizou um conjunto de dados bi-engastado de alumínio, submetido a diversas colisões. O pórtico foi ensaiado no seio do Laboratório de Imagens e a estrutura apresenta 6 barras chatas cujas dimensões são de 15,875mm de largura por 1,587 mm de espessura, com 4 porcas. Quatro acelerômetros piezoelétricos foram acoplados à estrutura conforme mostrado na figura 7.

Figura 7 – Pórtico Bi-Engastado

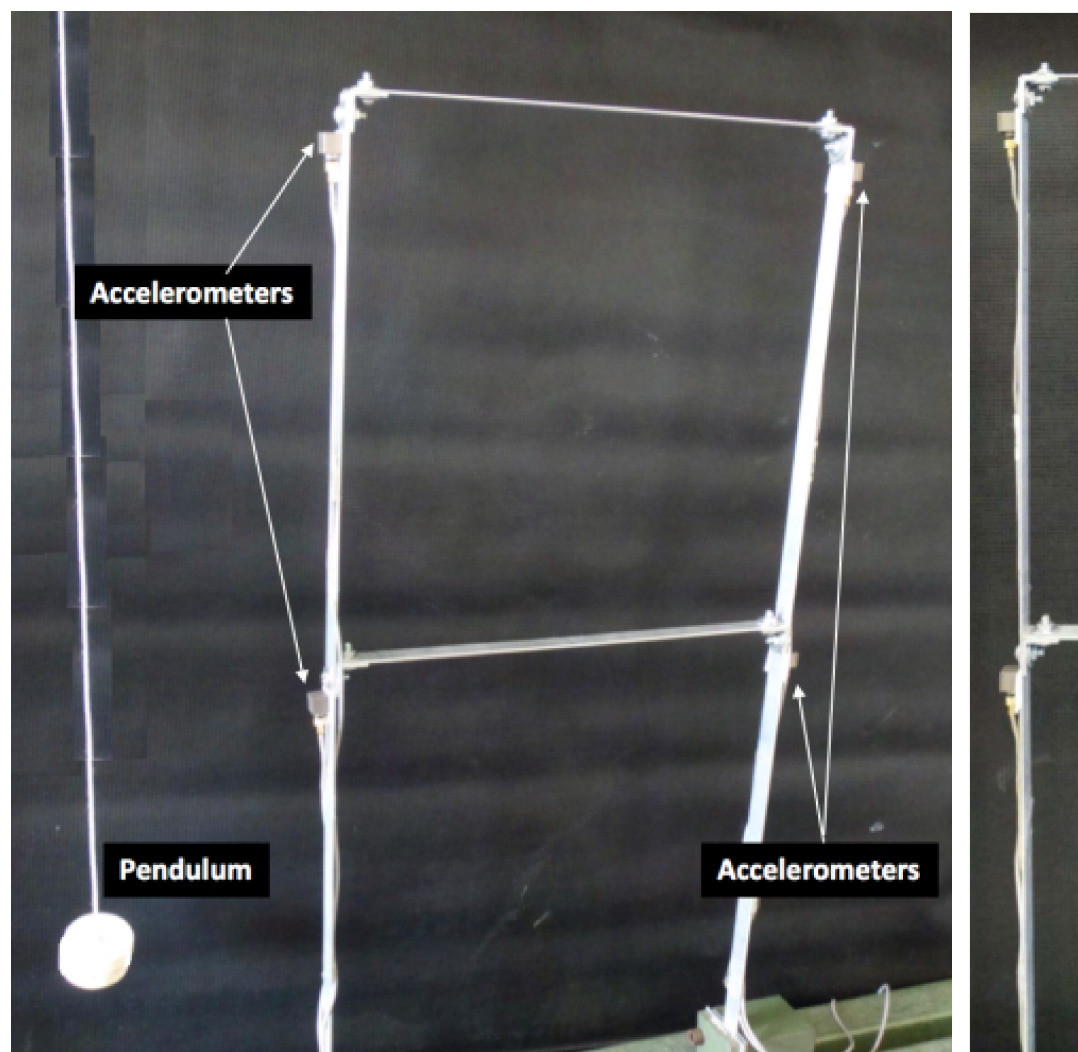
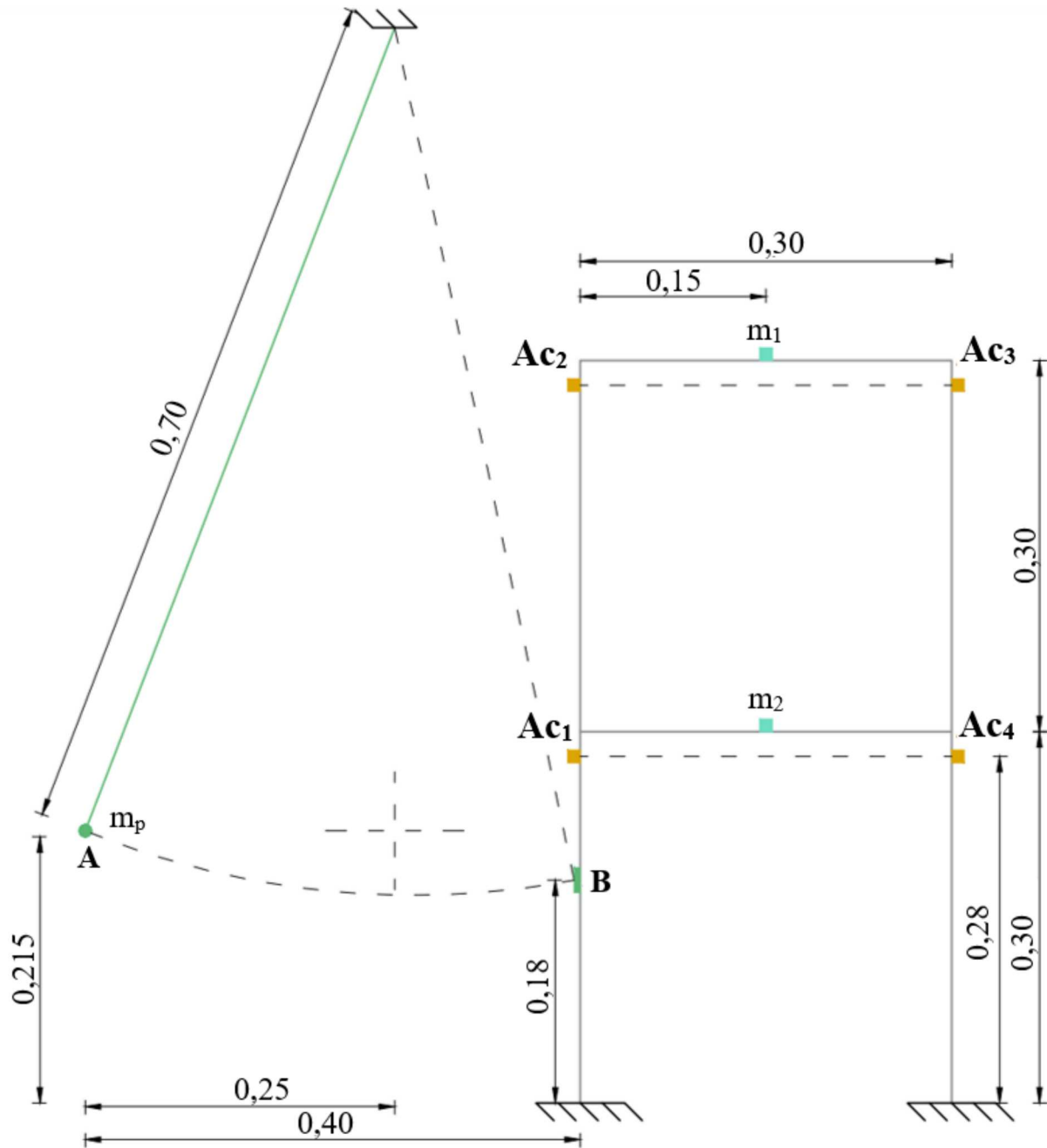


Figura 8 – Corte esquemático da



Fonte: Adapado Motta *et al.* [37]

Foram simulados um total de 5 estágios de danos

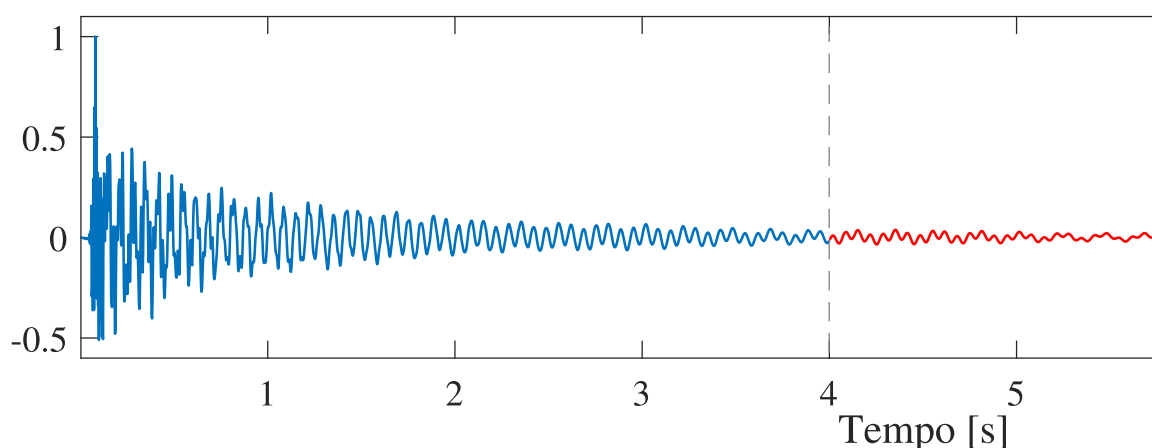
- **Estágio 01:** Não há massas acopladas à estrutura, sendo esta uma estrutura \tilde{c} livre de danos.

ampliar as análises para detecção foram empregados o domínio do tempo e da frequência. Nesta etapa, os dados são concatenados para formar uma matriz contendo os sinais e por estágio de dano.

5.1 Domínio do Tempo

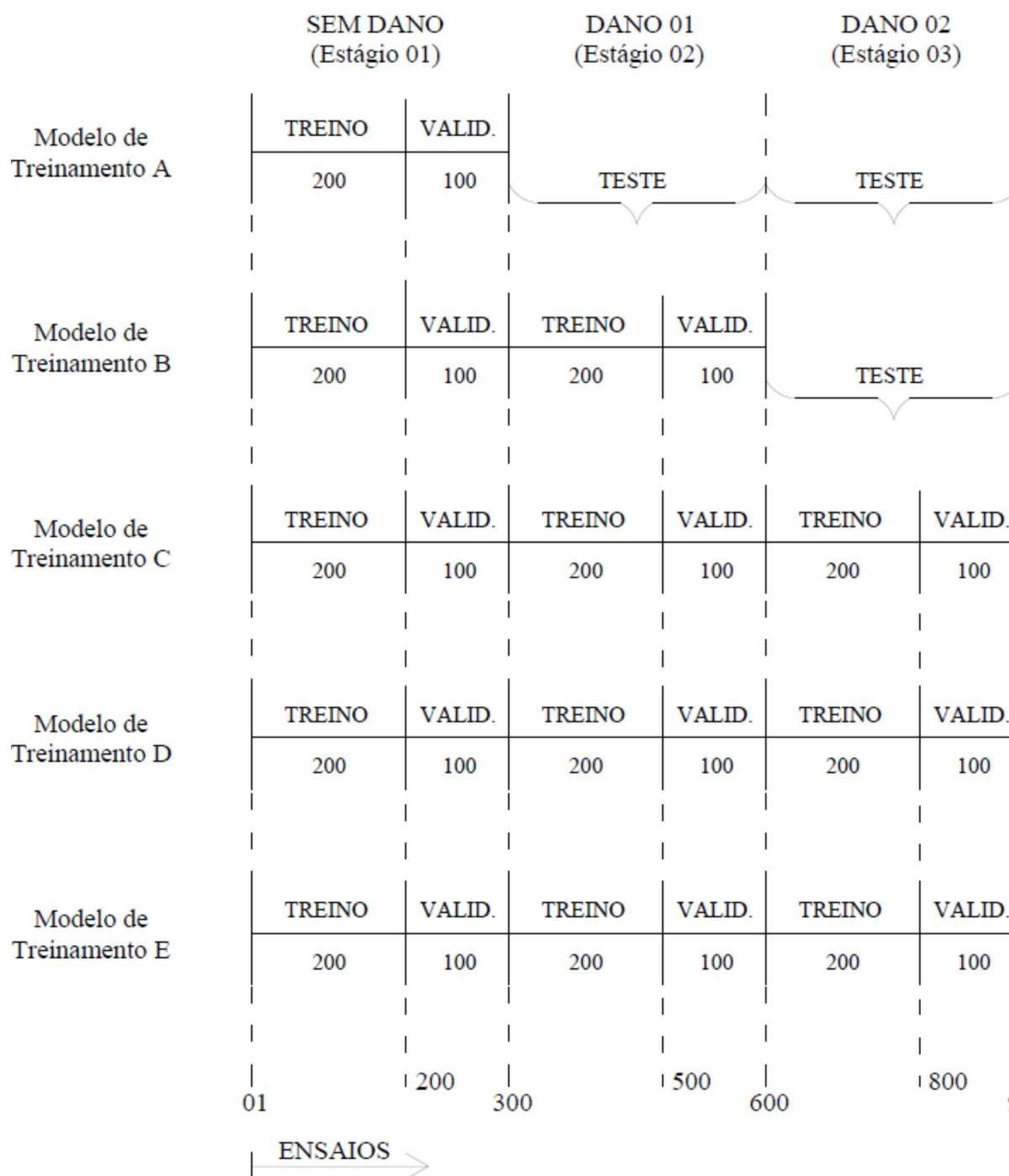
A figura 9 apresenta um sinal típico obtido pelo domínio do tempo. Após 2000 pontos (4 segundos), a estrutura apresenta amplitudes relativamente baixas e, por esse motivo, as respostas diretas são analisadas a partir de 4 segundos. Posteriormente, esses sinais são normalizados por cada sinal pelo módulo da sua respectiva amplitude máxima, resultando em sinais unidimensionais que variam entre -1 e 1. Após a normalização, os dados são divididos entre treinamento, validação e testes, definidos conforme o protocolo de rede neural.

Figura 9 – Gráfico de resposta típica de um acelerômetro



- Teste: O teste do Modelo A usou os 300 ensa
- **Modelo B:**
 - Treinamento: Este modelo foi treinado com Estágio 1 e 200 ensaios do Estágio 2.
 - Validação: A avaliação foi feita a partir dos res 1 e 100 ensaios do Estágio 2.
 - Teste: O teste do Modelo B usou os 300 ensa
- **Modelo C:**
 - Treinamento: Este modelo foi treinado com Estágio 1, 200 ensaios do Estágio 2 e 200 ensa
 - Validação: A avaliação foi feita a partir dos res 1, 100 ensaios do Estágio 2 e 100 ensaios do E
 - Teste: O teste do Modelo C usou os 300 ensa
- **Modelo D:**
 - Treinamento: Este modelo foi treinado com Estágio 1, 200 ensaios do Estágio 02, 200 ensa Estágio 4.
 - Validação: A avaliação foi feita a partir dos res 1, 100 ensaios do Estágio 2, 100 ensaios do Est
 - Teste: O teste do Modelo D usou os 300 ensa

Figura 10 – Modelos de treinamento - D



Fonte: Do autor, 2024

O Modelo A visa avaliar se a estratégia via CAE... dados do Estágio 1 dos demais estágios. Já o Modelo B...

dados distintos daqueles usados no treinamento sejam a subsequente, onde um certo tipo de dano já fora identificado. Para dados provenientes do Estágio 1 (estrutura sã) e Estágio 2 (estrutura danificada), o algoritmo identificará comportamentos dinâmicos no treinamento, associando-os ao Estágio 3. Essa estratégia evita danos, sempre retreinando o algoritmo a medida que surge a estrutura.

O modelo E consistiu um treinamento, compreendendo de todos os estágios de dano. Este modelo visa compreender os danos precedentes através da identificação simultânea de todos os danos no modelo.

5.1.2 Otimização dos parâmetros

A arquitetura da rede neural utilizada neste trabalho foi otimizada no processo de otimização de parâmetros. Os parâmetros fixos foram as características que, independentemente da arquitetura, de uma rede neural. As funções de ativações utilizadas entre as camadas e a última camada devido às amplitudes do sinal de saída. Foram utilizadas duas métricas de detecção de danos (MSE e distância de Hamming). Observar que a primeira métrica devido a análise ser por iterativa, treina com acréscimos de danos a cada iteração totalizando 1000 danos distintos, a segunda métrica treina de maneira única com 1000 danos que corresponde apenas quatro redes treinadas.

Nessa otimização, conforme demonstrado na T

Tabela 2 – Análise Pórtico - Parâmetros Fixos

ANÁLISE PÓRTICO	
PARÂMETROS FIXOS	
Domínio de Treinamento	Ten
Dimensão Sinal de Entrada	[2000
Solucionador	Ad
Função de Ativação	Re
Análise da rede	Erro de Re
Modelos de Treinamento	A, B,
Nº de Estratégias para Detecção	1
Tipo do Autocodificador	CA
Nº de Camadas Convolucionais	2
Nº de Estágios	5
Estágios de Treinamento	4
Acelerômetros	4
Nº de Datas de Treinamento [Análise Estática]	4

3 – Análise Pórtico - Parâmetros Variáveis

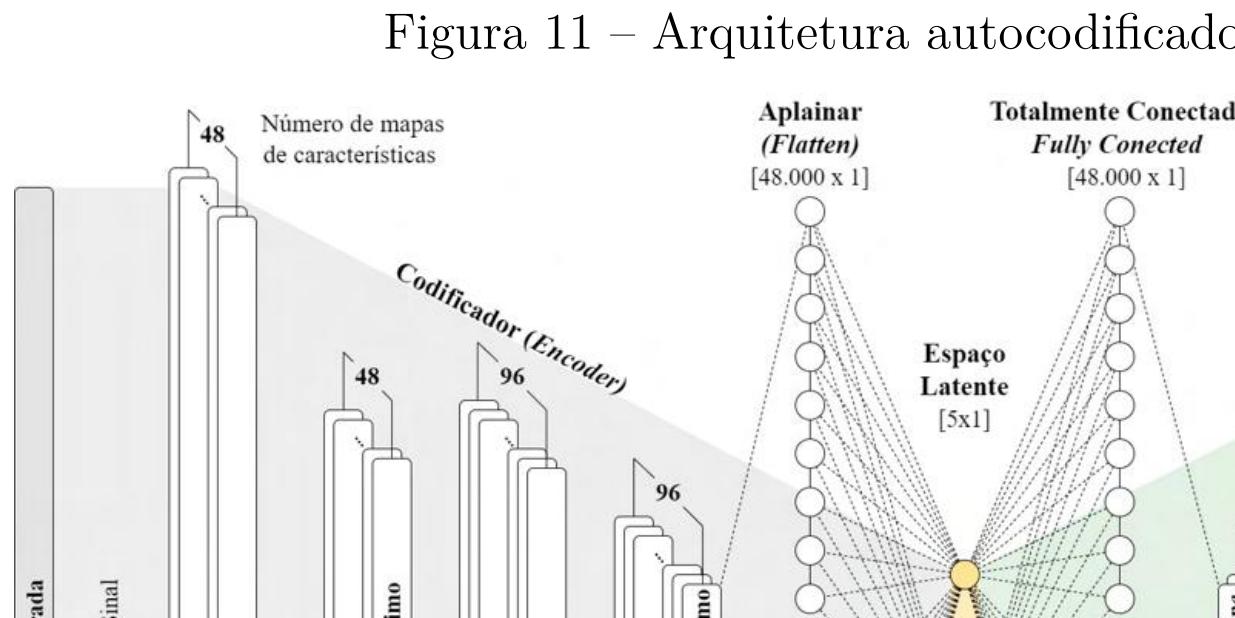
ANÁLISE PÓRTICO													
PARÂMETROS VARIÁVEIS													
	25			50			75			100			
1	Ac_1			Ac_1			Ac_1			Ac_1			
	8	2	4	8	2	4	8	2	4	8	2	4	8
0	300	75	150	300	75	150	300	75	150	300	75	150	300
0	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500
REDES													
	6	7	8	9	10	11	12	13	14	15	16	17	18
	24			24			24			24			
253	0,52726	0,3443	0,4145	0,5259	0,3502	0,4166	0,50545	0,33647	0,4145	0,50674	0,35005	0,41441	0,50385
318	0,13923	0,0594	0,0860	0,1384	0,0614	0,0868	0,12795	0,05688	0,08599	0,12854	0,061341	0,085945	0,12708
0	197	72	120	189	71	124	187	71	123	181	70	116	223
3,5	4931,6	7233,7	6047,4	4727,3	7110,4	6233,2	4697,8	7110,5	6194,3	4549	7050,2	5841,1	5580,5
38	06:48	05:55	07:01	06:53	06:09	06:00	07:14	05:56	05:55	07:19	29:00	05:59	07:17
3	24	25	26	27	28	29	30	31	32	33	34	35	36
	36			36			36			36			
018	0,48861	0,3303	0,38998	0,47595	0,33151	0,38411	0,45888	0,32161	0,39224	0,47257	0,32156	0,38486	0,54502
324	0,11977	0,05477	0,076119	0,11351	0,055344	0,073835	0,10555	0,051871	0,077011	0,11176	0,051886	0,074135	0,14852
9	225	72	117	201	70	118	224	71	118	209	72	126	243
9,5	5646,5	7256,2	5852,6	5026,6	7081,5	5930,3	5614,8	7176,8	5936,8	5228,6	7239	6326,6	6078
35	07:23	06:31	06:50	07:42	06:03	07:04	07:37	06:17	07:09	07:43	06:09	07:09	08:05
	42	43	44	45	46	47	48	49	50	51	52	53	54
	48			48			48			48			
83	0,52356	0,31714	0,36828	0,47528	0,3364	0,37991	0,48406	0,31623	0,36614	0,51127	0,32251	0,37498	0,48079
599	0,1437	0,050466	0,067862	0,11327	0,056712	0,07221	0,1178266	0,050143	0,067069	0,1346	0,052258	0,070358	0,11848
4	194	71	98	183	70	125	234	71	118	214	72	115	259

5.1.3 Arquitetura da Rede

Na análise da otimização, os parâmetros que propo para o objetivo do estudo foram os seguintes: a dimensã como 5, o número de filtros foi configurado para 48, e a d foi estabelecida igual a 2.

A utilização da camada de normalização em lote tem a função de normalizar as ativações de cada camada calculando a média e o desvio padrão das ativações do l as ativações de cada neurônio com base nesses valores. ativações próximas à média zero e ao desvio padrão um, a e otimizado.

O número de filtros, um dos parâmetros da cama longo de toda as análises. Para o preenchimento, visan se utilizou o *Same Padding*. O *Stride* e o tamanho do f convolucional, sendo mostrado na figura 11.



hiperparâmetros como taxa de aprendizado (do inglês *Initial Learning Rate*), tamanho dos dados de treinamento (do inglês *Batch Size*), entre outros, foram definidos pelo software. Sendo assim, de acordo com os parâmetros utilizados, o modelo foi treinado visando detectar alterações ou danos estruturais, obtendo-se uma arquitetura de camadas convolucionais e o mesmo quantitativo de camada convolucional de saída.

Tabela 4 – Arquitetura da Rede Neuronal

Camadas	Dimensão de Entrada	Dimensão de Saída
Entrada (<i>Input</i>)	2000 x 1 x 1 x 1	-
Convolução - 01 (<i>Convolution</i>)	2000 x 1 x 1 x 1	2000 x 1 x 1 x 1
Agrupamento Máximo (<i>Max Pooling</i>)	2000 x 1 x 48 x 1	1000 x 1 x 1 x 1
Convolução - 02 (<i>Convolution</i>)	1000 x 1 x 48 x 1	1000 x 1 x 1 x 1
Agrupamento Máximo (<i>Max Pooling</i>)	1000 x 1 x 96 x 1	500 x 1 x 1 x 1
Aplainar (<i>Flatten</i>)	500 x 1 x 96 x 1	48000
Totalmente Conectada (<i>Fully Connected</i>)	48000 x 1	5 x 1
Totalmente Conectada (<i>Fully Connected</i>)	5 x 1	48000
Redimensionamento (<i>Reshape</i>)	48000 x 1	500 x 1 x 1 x 1
Desagrupamento Máximo (<i>Max Unpooling</i>)	500 x 1 x 96 x 1	1000 x 1 x 1 x 1
Deconvolução - 01 (<i>Transposed Convolution</i>)	1000 x 1 x 96 x 1	1000 x 1 x 1 x 1
Desagrupamento Máximo (<i>Max Unpooling</i>)	1000 x 1 x 48 x 1	2000 x 1 x 1 x 1

5.1.4 Treinamento da Rede

Mediante a definição da divisão dos dados para a rede, é possível treinar a rede de acordo com cada estratégia. Na estratégia 01, aplicou os modelos A, B, C e D visando a detecção de danos. Nas estratégias 02, 03 e 04, aplicaram o modelo E.

5.1.5 Estratégias para detectar danos

Neste estudo, foram aplicadas duas métricas para avaliar a performance das redes utilizando quatro estratégias distintas. A primeira estratégia utilizou o erro médio, que mede a diferença entre os sinais de entrada e os sinais de saída. As outras três estratégias empregaram a distância de Mahalanobis no espaço latente.

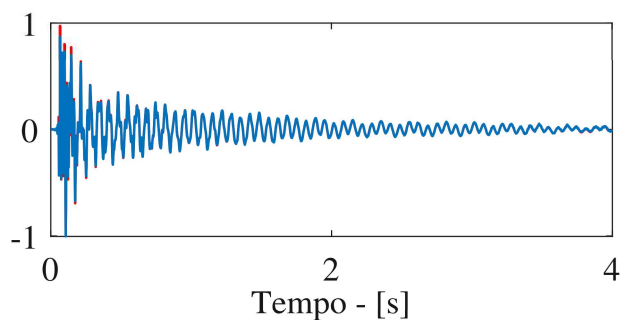
Resultados para a Estratégia por MSE - Estratégia 01

Cada rede treinada (por modelo e por acelerômetro específico, cuja função é predizer (reconstruir) os dados de entrada o mais possível. Sendo assim, é possível avaliar o erro quadrático médio entre os sinais de entrada e preditos. Espera-se que os sinais de reconstrução mais errática em comparação com os dados de treinamento tenham um comportamento semelhante aos dados de treinamento.

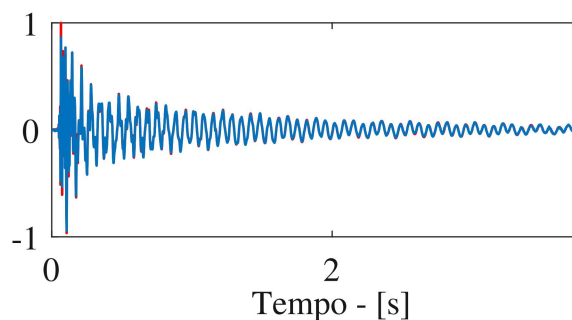
- Reconstrução dos sinais:

Nesta seção são apresentados os resultados das redes treinadas para o caso estudado. A figura 12 demonstra

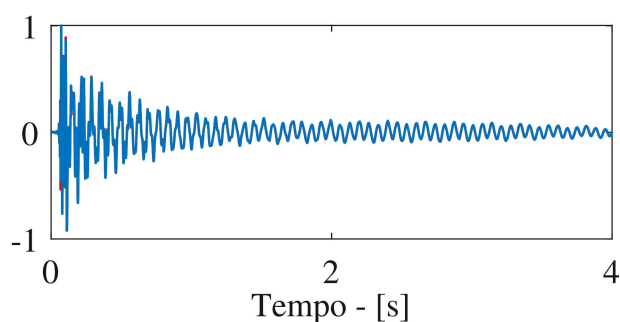
Figura 12 – Predição do Si



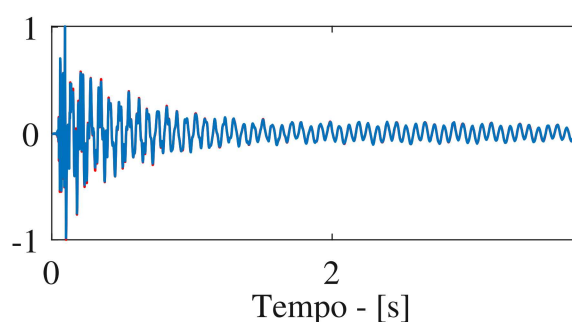
(a) Treino - Estágio 01



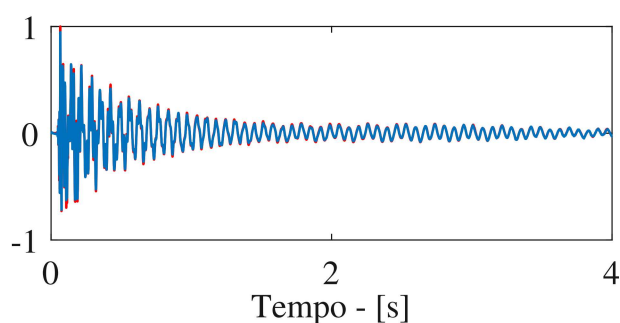
(b) Validação - Estágio 01



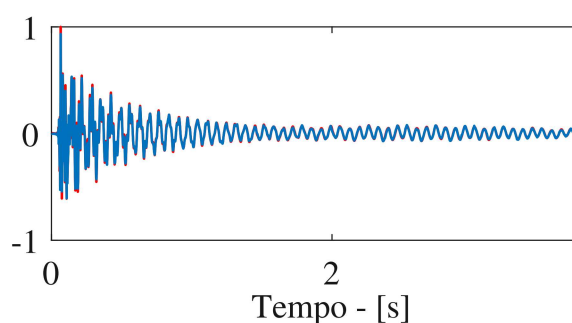
(d) Treino - Estágio 01+02



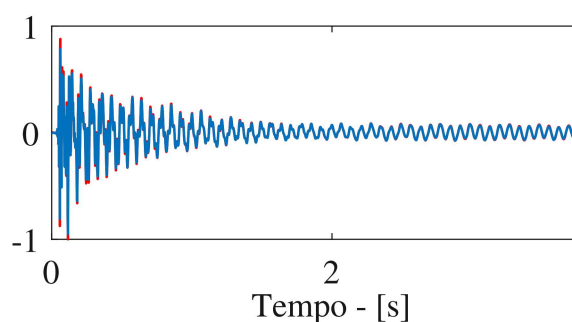
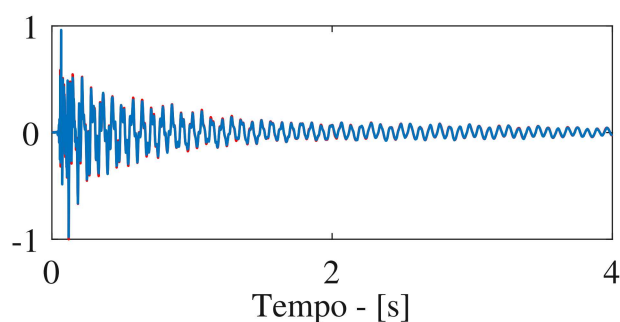
(e) Validação - Estágio 01+02



(g) Treino - Estágio 01+02+03



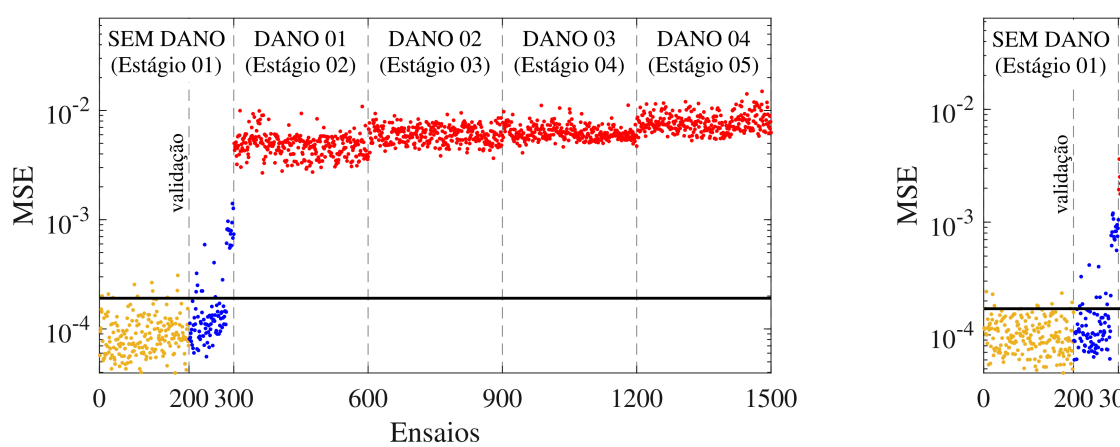
(h) Validação - Estágio 01+02+03



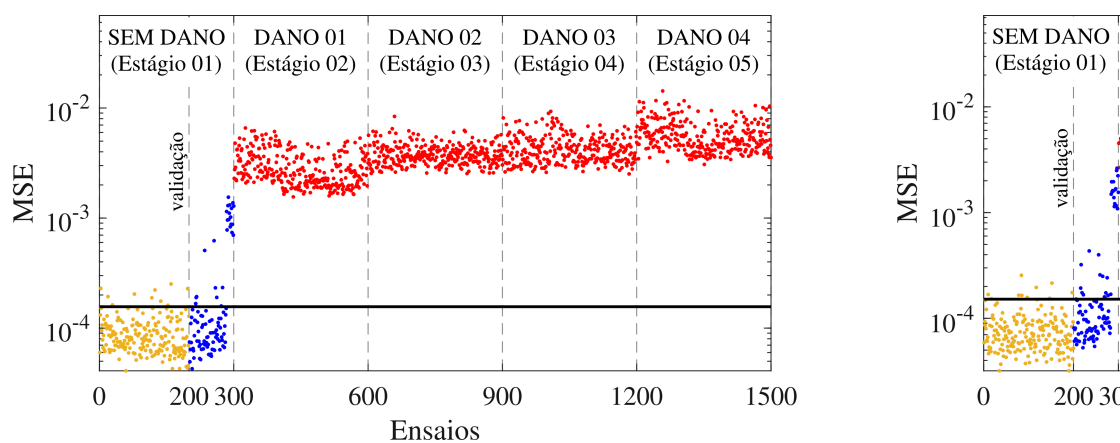
— Dado de entrada — Predito/Reconstru

no treinamento é possível identificar diferenças dos níveis de danos, permitindo se classificar diretamente os dados dos Estágios de Danos.

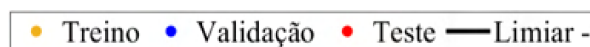
Figura 13 – Erro Quadrático Médio



(a) Acelerômetro 1

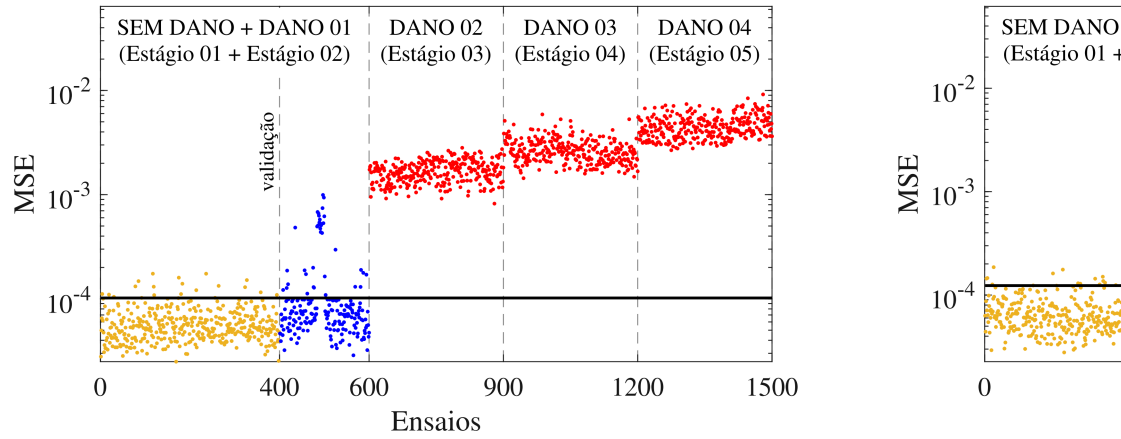


(c) Acelerômetro 3

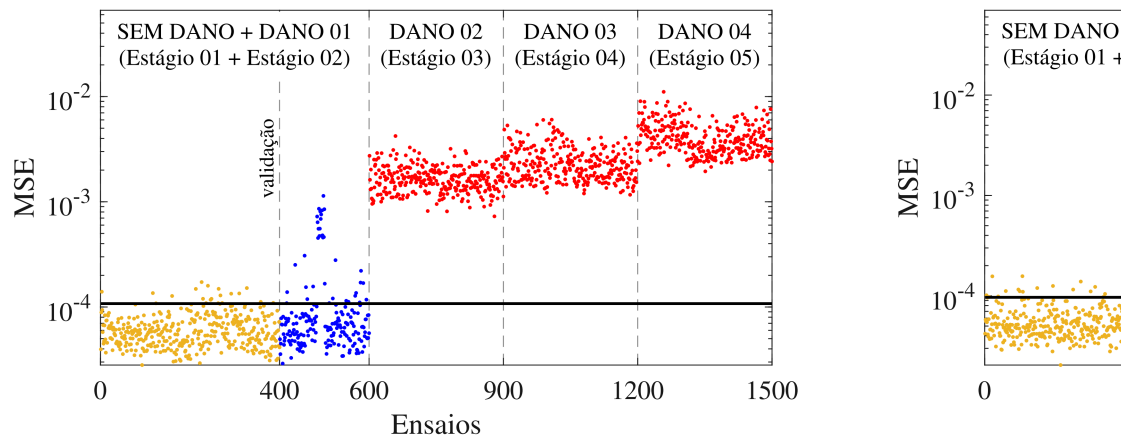


Fonte: Do Autor, 2024

Figura 14 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

● Treino ● Validação ● Teste — Limiar

Fonte: Do Autor, 2024

Figura 15 – Erro Quadrático Médio

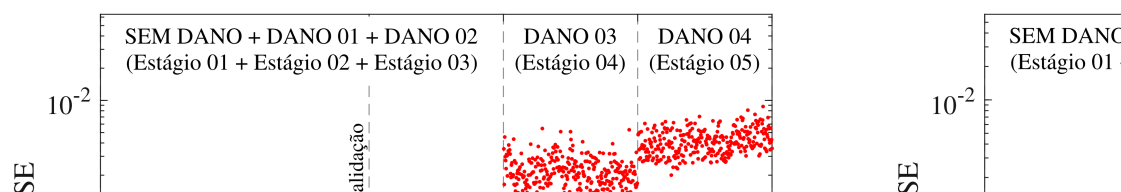
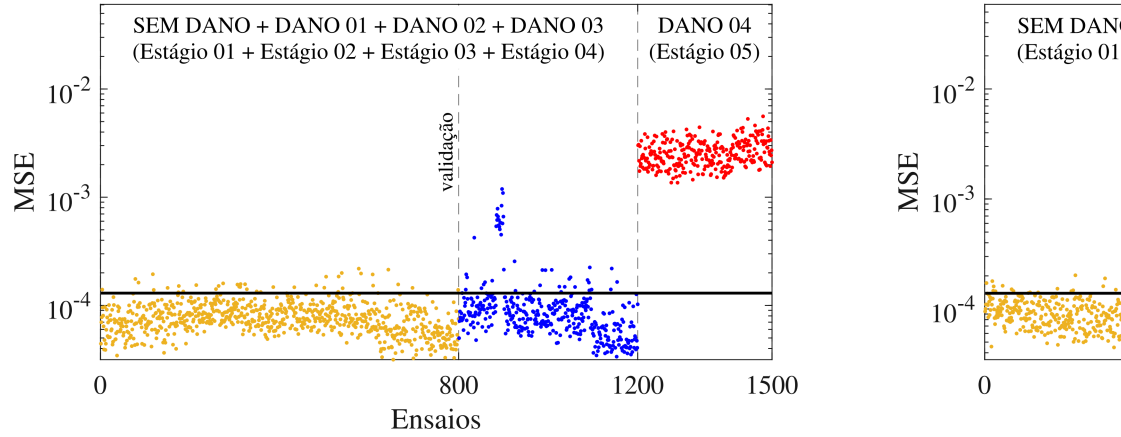
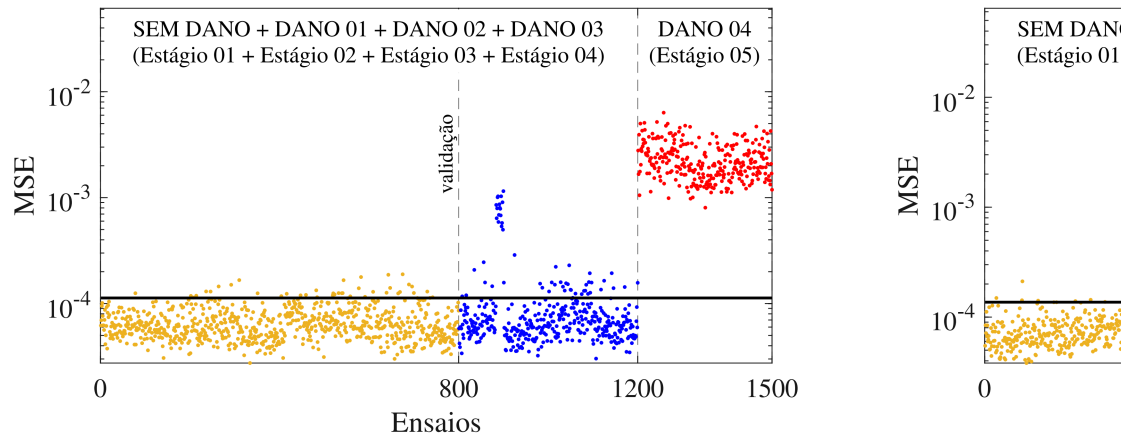


Figura 16 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

● Treino ● Validação ● Teste — Limiar

Fonte: Do Autor, 2024

As matrizes de confusão para os Modelos A, B, C 17, 18, 19 e 20, respectivamente. Observa-se um desempenho para ambos os Modelos, com uma taxa de acertos de classificação entre 90,61% e 98,01%. Ressalta-se que as matrizes de con

Figura 17 – Matriz de Confusão -

Dano Real	0	730 5,62%	270 2,08%	73,00% 27,00%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	97,80% 2,20%	97,92% 2,08%
		0	1+2+3+4	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	741 5,70%	0 0,00%	100,00% 0,00%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	97,80% 2,20%	97,92% 2,08%
		0	1+2+3+4	
		Dano Predito		

(b) Acel

Dano Real	0	726 5,58%	274 2,11%	72,60% 27,40%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	97,77% 2,23%	97,89% 2,11%
		0	1+2+3+4	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	0	695 5,35%	0 0,00%	100,00% 0,00%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	97,80% 2,20%	97,92% 2,08%
		0	1+2+3+4	
		Dano Predito		

(d) Acel

Fonte: Do Autor, 2024

Figura 18 – Matriz de Confusão -

Dano Real	0+1	1598 14,53%	402 3,65%	79,90% 20,10%
	3+4	0 0,00%	9000 81,35%	100,00% 0,00%
		100,00% 0,00%	97,80% 2,20%	97,92% 2,08%
		0+1	3+4	
		Dano Predito		

Dano Real	0+1	1634 14,85%	0 0,00%	100,00% 0,00%
	3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	97,80% 2,20%	97,92% 2,08%
		0+1	3+4	
		Dano Predito		

Figura 19 – Matriz de Confusão -

Dano Real	0+1+2	2533 28,14%	467 5,19%	84,43% 15,57%
	3+4	0 0,00%	6000 66,67%	100,00% 0,00%
		100,00% 0,00%	92,78% 7,22%	94,81% 5,19%
		0+1+2	3+4	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0+1+2	2547 28,30%		
	3+4	0 0,00%		
		100,00% 0,00%		
		0+1+2		
		Dano Predito		

(b) Acel

Dano Real	0+1+2	2451 27,23%	549 6,10%	81,70% 18,30%
	3+4	0 0,00%	6000 66,67%	100,00% 0,00%
		100,00% 0,00%	91,62% 8,38%	93,90% 6,10%
		0+1+2	3+4	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	0+1+2	2398 26,64%		
	3+4	0 0,00%		
		100,00% 0,00%		
		0+1+2		
		Dano Predito		

(d) Acel

Fonte: Do Autor, 2024

Figura 20 – Matriz de Confusão -

Dano Real	0+1+2+3	3490 49,86%	510 7,29%	87,25% 12,75%
	4	0 0,00%	3000 100,00%	

Dano Real	0+1+2+3	3555 50,79%		
	4	0 0,00%		

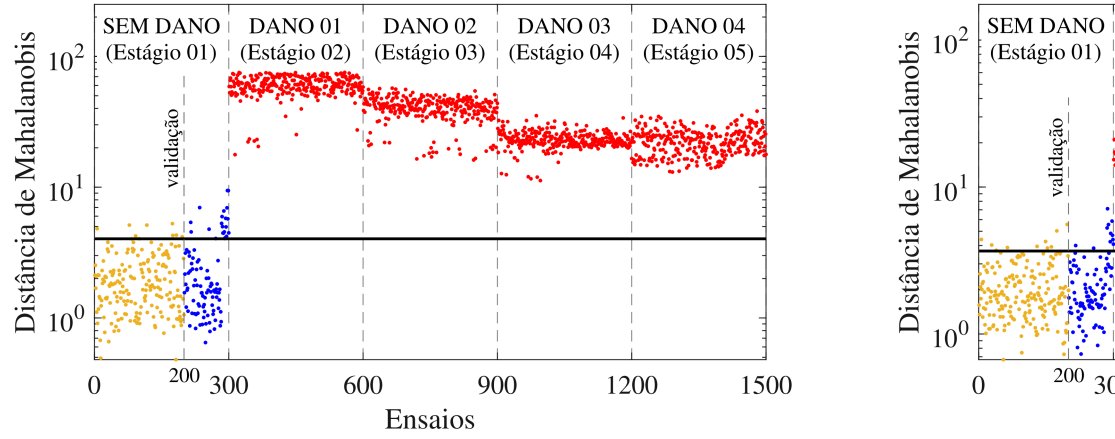
Resultados para Estratégia por Mahalanobis - Es

O espaço latente, isto é, espaço de menor dimensão, é definido pelos valores que podem indicar, de acordo com a metodologia utilizada, nos estágios de treinamento. Utilizando como base o espaço latente, foram desenvolvidas as estratégias de classificação. A primeira estratégia visa encontrar os pontos de referência através de uma distribuição de referência, separado por acelerômetro. Foi compilado os dados do espaço latente dos quatro acelerômetros para cada conjunto de dados, como o espaço latente apresentava uma dimensão de 20. A segunda estratégia de dados, como o espaço latente apresentava uma dimensão de 20, a primeira compilação foi uma matriz de $[20 \times 1]$ por ensaio. Em seguida, foram desenvolvidas as referências destes dados compilados. Por fim, a terceira estratégia de análise, uma distribuição de referência correspondente a cada conjunto de dados.

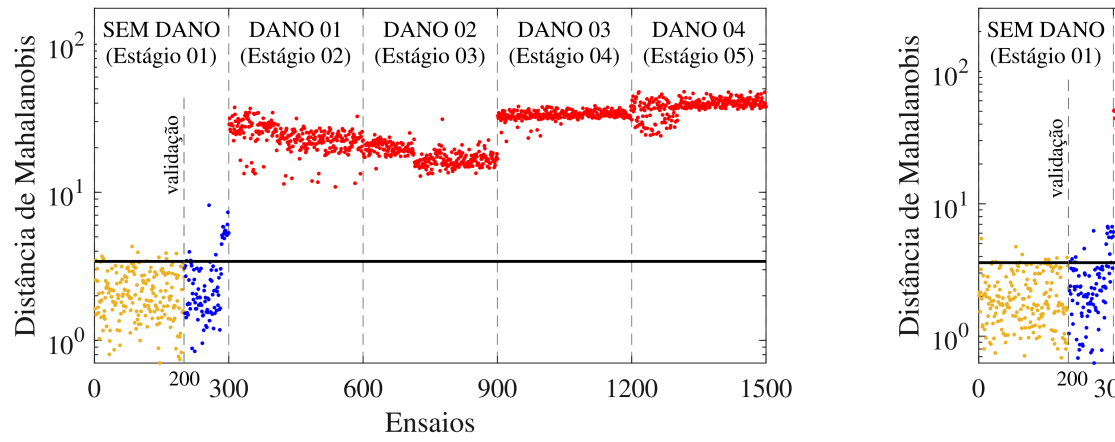
Sendo assim, a Distância de Mahalanobis foi aplicada para identificar os pontos calculando a distância de cada ponto em relação à média e ao desvio padrão da distribuição de referência. Essa medida permitiu identificar os pontos que indicariam possíveis danos na estrutura. É importante ressaltar que a Distância de Mahalanobis é particularmente útil quando há correlação entre as variáveis e uma detecção mais sensível de desvios significativos.

Cabe ressaltar que em todas as estratégias houve a preocupação de minimizar a aleatoriedade do sistema (os gráficos foram desenvolvidos por apresentarem resultados semelhantes). Nas figuras apresentadas, os valores de Mahalanobis os valores em vermelho, azul e amarelo representam os pontos de dados testados, validados, e a distribuição de referência (os pontos em preto) foram calculados através do 95-ésimo percentil da distribuição de Mahalanobis da distribuição de referência.

Figura 21 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

● Distribuição de Referência ● Validação ● Teste — Limite

Fonte: Do Autor, 2024

Figura 22 – Distância de Mahalanobis

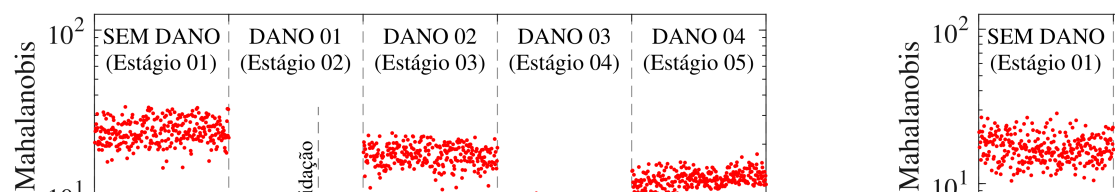
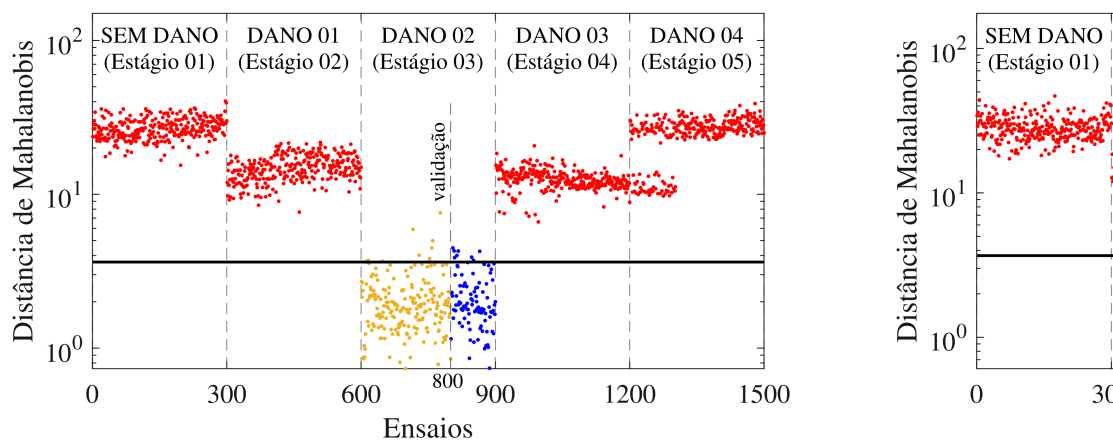
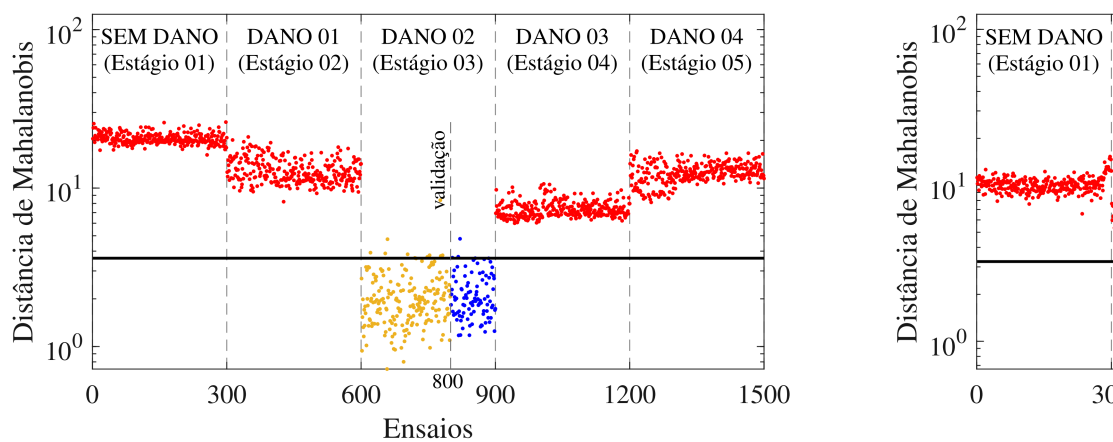


Figura 23 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

● Distribuição de Referência ● Validação ● Teste — Limite

Fonte: Do Autor, 2024

Figura 24 – Distância de Mahalanobis

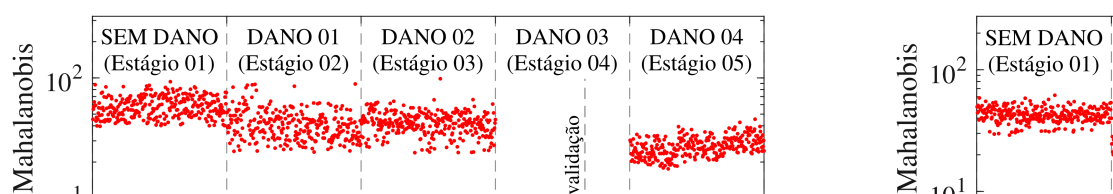
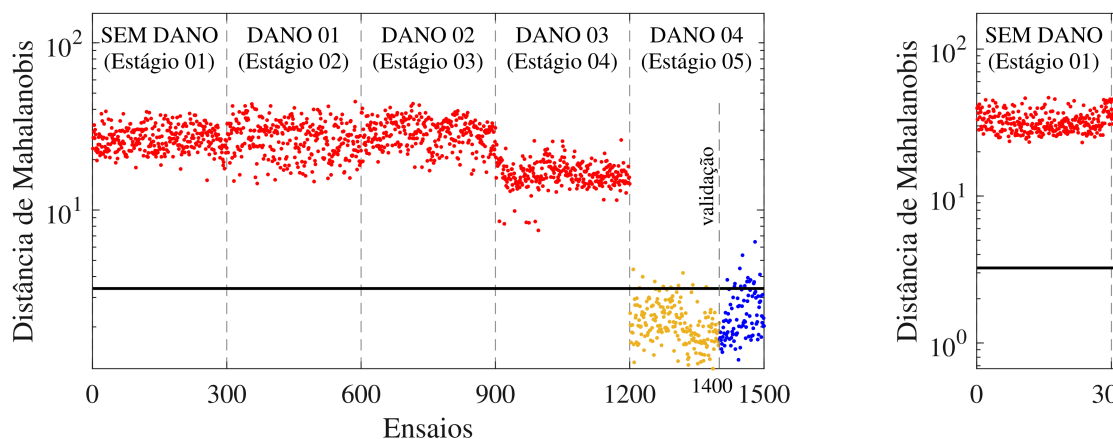
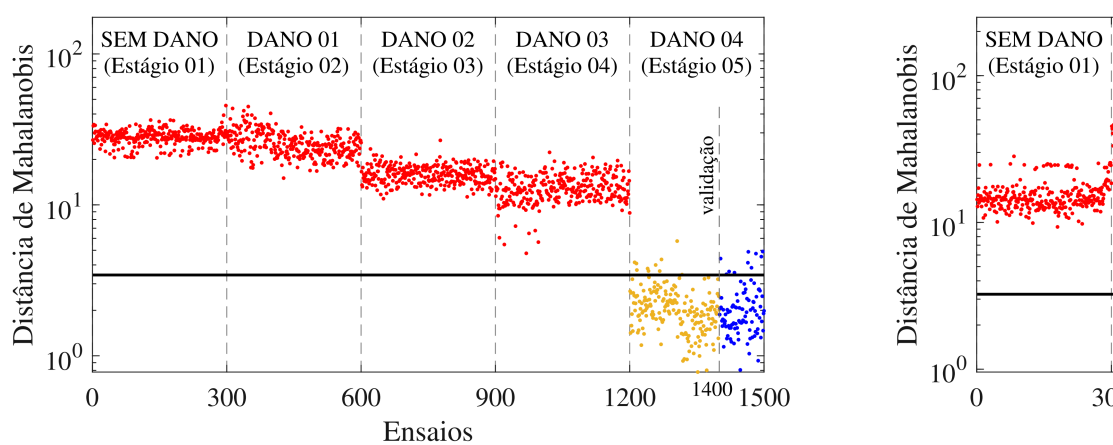


Figura 25 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

● Distribuição de Referência ● Validação ● Teste — Limite

Fonte: Do Autor, 2024

As matrizes de confusão para esta estratégia utilizadas nas Figuras 26, 27, 28, 29 e 30. Observa-se um desempenho para ambos os Modelos, com uma taxa de acertos de classificação entre 98,25% e 99,99%. Nota-se também que o número de

Figura 26 – Matriz de Confusão -

Dano Real	0	854 6,57%	146 1,12%	85,40% 14,60%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	98,80% 1,20%	98,88% 1,12%
		0	1+2+3+4	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	825 6,35%	
	1+2+3+4	5 0,04%	
		99,40% 0,60%	
		0	
		Dano Predito	

(b) Acel

Dano Real	0	783 6,02%	217 1,67%	78,30% 21,70%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	98,22% 1,78%	98,33% 1,67%
		0	1+2+3+4	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	0	791 6,08%	
	1+2+3+4	1 0,01%	
		99,87% 0,13%	
		0	
		Dano Predito	

(d) Acel

Fonte: Do Autor, 2024

Figura 27 – Matriz de Confusão -

Dano Real	1	979 7,53%	21 0,16%	97,90% 2,10%
	3+4	0	12000	100,00%

Dano Real	1	984 7,57%
	3+4	50

Figura 28 – Matriz de Confusão -

Dano Real	2	909 6,99%	91 0,70%	90,90% 9,10%
	0+1+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,25% 0,75%	99,30% 0,70%
		2	0+1+3+4	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	2	931 7,16%	0 0,00%	100,00% 0,00%
	0+1+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,25% 0,75%	99,30% 0,70%
		2	0+1+3+4	
		Dano Predito		

(b) Acel

Dano Real	2	944 7,26%	56 0,43%	94,40% 5,60%
	0+1+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,54% 0,46%	99,57% 0,43%
		2	0+1+3+4	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	2	918 7,06%	148 1,14%	86,12% 13,88%
	0+1+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,54% 0,46%	99,57% 0,43%
		2	0+1+3+4	
		Dano Predito		

(d) Acel

Fonte: Do Autor, 2024

Figura 29 – Matriz de Confusão -

Dano Real	3	973 7,48%	27 0,21%	97,30% 2,70%
	2+4	0 0,00%	12000 92,31%	100,00% 0,00%

Dano Real	3	990 7,62%	0 0,00%	100,00% 0,00%
	2+4	0 0,00%	12000 92,31%	100,00% 0,00%

Figura 30 – Matriz de Confusão -

Dano Real	4	905 6,96%	95 0,73%	90,50% 9,50%
	0+1+2+3	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,21% 0,79%	99,27% 0,73%
		4	0+1+2+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	4	986 7,58%	0 0,00%	100,00% 0,00%
	0+1+2+3	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,21% 0,79%	99,27% 0,73%
		4	0+1+2+3	
		Dano Predito		

(b) Acel

Dano Real	4	908 6,98%	92 0,71%	90,80% 9,20%
	0+1+2+3	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,24% 0,76%	99,29% 0,71%
		4	0+1+2+3	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	4	906 6,97%	0 0,00%	100,00% 0,00%
	0+1+2+3	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,21% 0,79%	99,27% 0,73%
		4	0+1+2+3	
		Dano Predito		

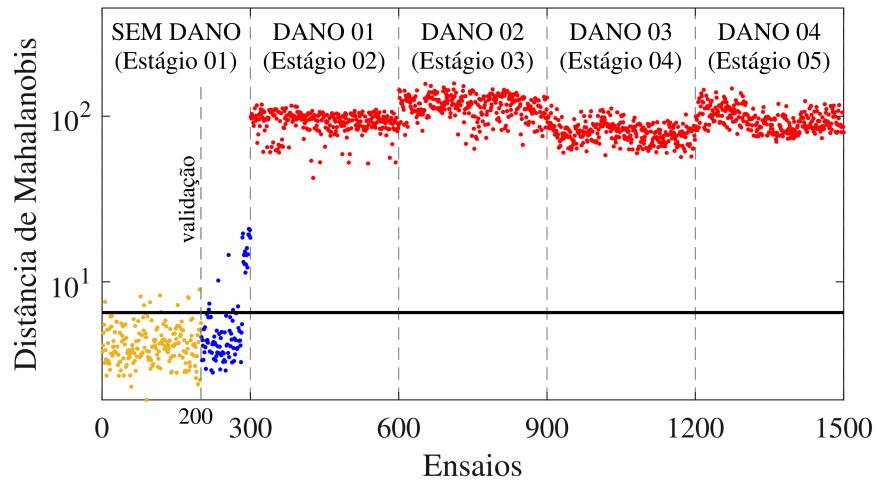
(d) Acel

Fonte: Do Autor, 2024

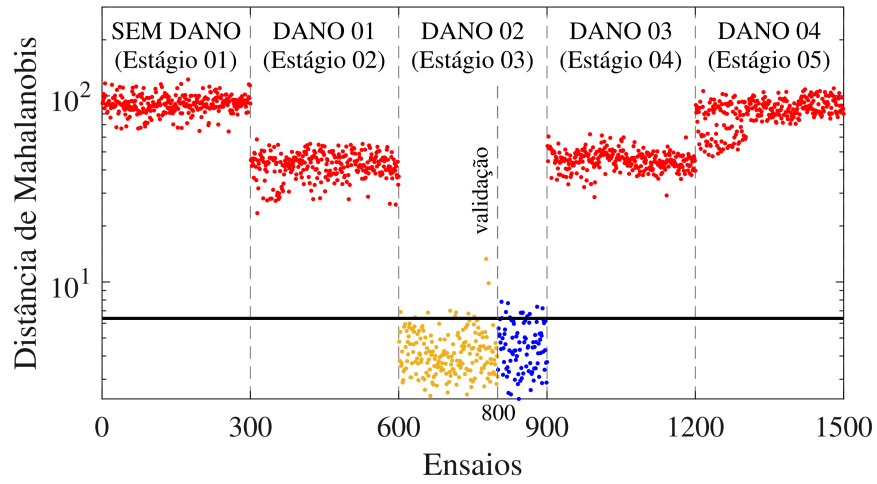
- Distância de Mahalanobis - Acelerômetros Compil

A segunda estratégia, correspondente a Figura 31, utiliza a distância de Mahalanobis com base na distribuição de referência dos cinco

Figura 31 – Distância de Mahalanobis - Acel



(a) Estágio 1



(c) Estágio 3

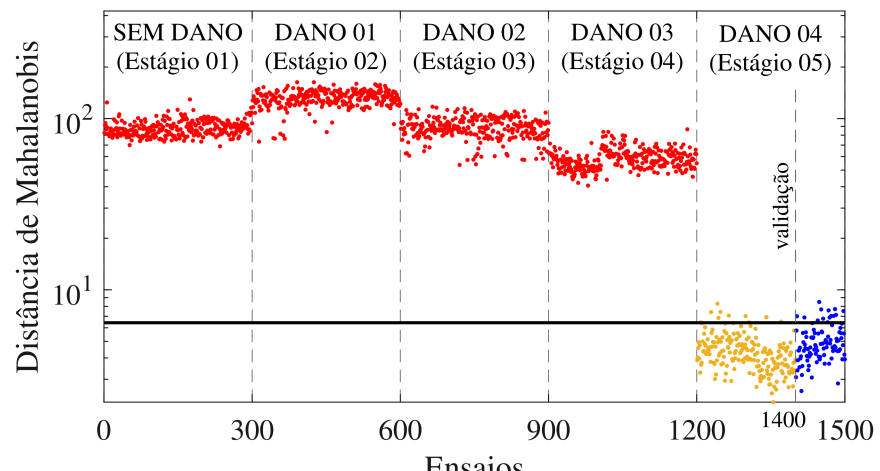


Figura 32 – Matriz de Confusão - Acelerô

Dano Real	0	1+2+3+4	
	752 5,78%	248 1,91%	75,20% 24,80%
	0 0,00%	12000 92,31%	100,00% 0,00%
1+2+3+4	100,00% 0,00%	97,98% 2,02%	98,09% 1,91%
	0	1+2+3+4	Dano Predito

(a) Estágio 1

Dano Real	1	0+2+3+4	
	952 7,32%	48 0,37%	95,20% 4,80%
	0 0,00%	12000 92,31%	100,00% 0,00%
0+2+3+4	100,00% 0,00%	99,60% 0,40%	99,63% 0,37%
	1	0+2+3+4	Dano Predito

(b) Estágio 2

Dano Real	3	0+1+2+4	
	965 7,42%	35 0,27%	96,50% 3,50%
	0 0,00%	12000 92,31%	100,00% 0,00%
0+1+2+4	100,00% 0,00%	99,71% 0,29%	99,73% 0,27%
	3	0+1+2+4	Dano Predito

(d) Estágio 4

Dano Real	4	0+1+2+3	
	871 6,70%	0 0,00%	100,00% 0,00%
	0 0,00%	12000 92,31%	100,00% 0,00%
0+1+2+3	100,00% 0,00%	99,71% 0,29%	99,73% 0,27%
	4	0+1+2+3	Dano Predito

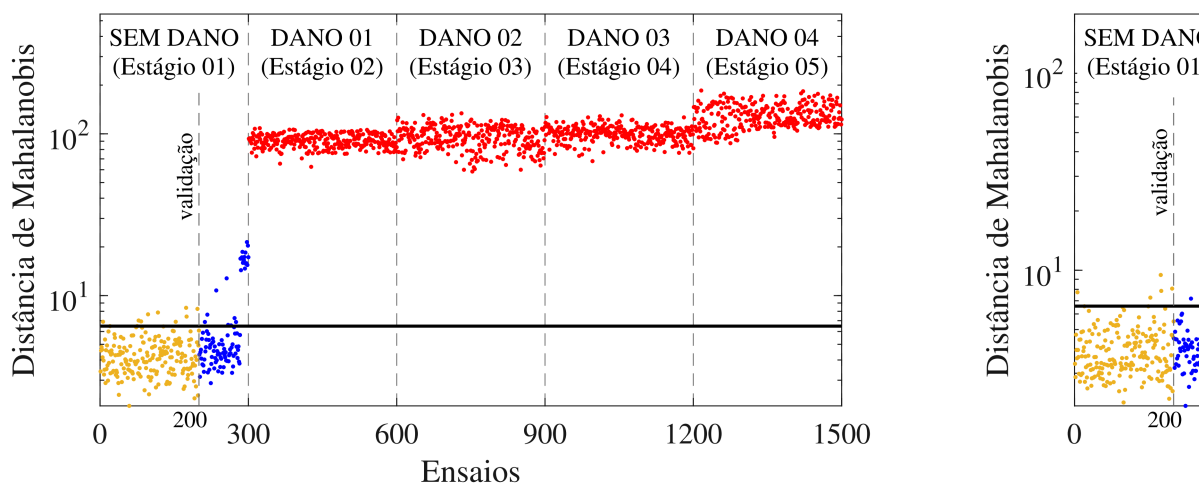
(e) Estágio 5

Fonte: Do Autor, 2024

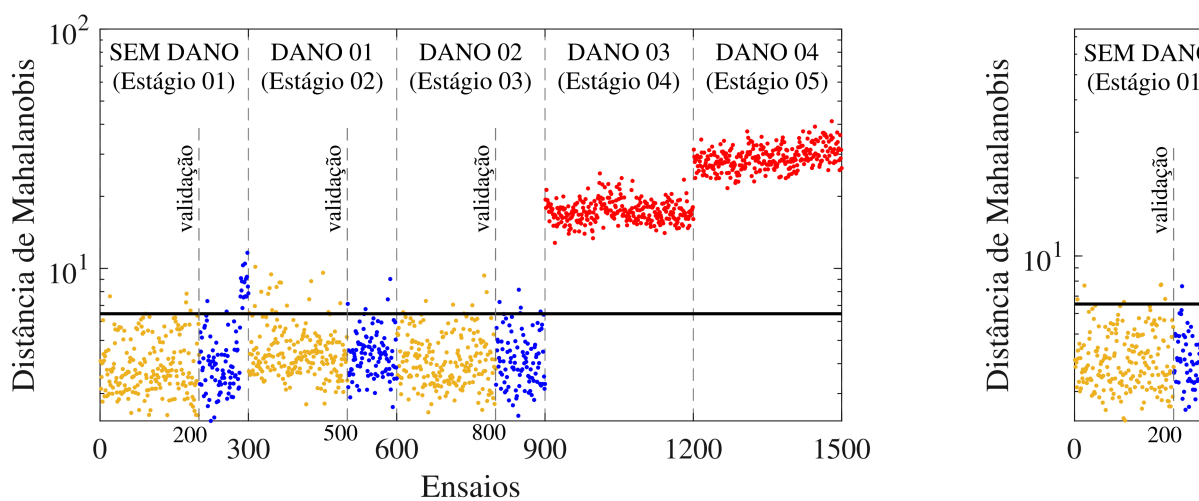
- Distância de Mahalanobis acrescentando Danos (E

A terceira estratégia, utilizou da compilação dos dados em uma matriz de dados. Contudo, ao invés de se analisar os dados de forma individual, foi utilizado o método de análise de componentes principais (PCA) para reduzir a dimensionalidade dos dados e identificar os principais fatores de variação. Este método é amplamente utilizado em estatística multivariada para simplificar a interpretação de dados complexos. A análise de componentes principais (PCA) é uma técnica estatística que permite reduzir a dimensionalidade de um conjunto de dados multivariados, mantendo a maior parte da informação contida nos dados. Isso é feito através da transformação dos dados originais em um novo conjunto de eixos ortogonais, chamados de componentes principais, que representam as direções de maior variabilidade nos dados. A primeira componente principal captura a maior parte da variância dos dados, enquanto as subsequentes capturam quantidades decrescentes de variância. Isso resulta em uma representação simplificada dos dados, onde os pontos de dados são projetados em um espaço de menor dimensão, facilitando a visualização e a interpretação dos padrões subjacentes. Além disso, o PCA pode ser utilizado para identificar outliers, agrupar dados semelhantes e melhorar a eficiência de algoritmos de aprendizado de máquina que lidam com dados de alta dimensão. No contexto da análise de dados de danos, o PCA pode ajudar a identificar os principais fatores que contribuem para a ocorrência de danos, permitindo uma análise mais focada e eficiente dos dados coletados.

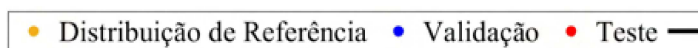
Figura 33 – Distância de Mahalanobis - Ac



(a) Estágio 1



(c) Estágio 3



Fonte: Do Autor, 2024

Os gráficos que ilustram as acurácias da terceira
disponíveis na Figura 34. Verifica-se um bom desempenho
ambas as variantes do Modelo, com uma média geral de

Figura 34 – Matriz de Confusão - Acr

Dano Real	0	752 5,78%	248 1,91%	75,20% 24,80%	Dano Real	0+1	1761 16,01%	
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%		2+3+4	0 0,00%	
		100,00% 0,00%	97,98% 2,02%	98,09% 1,91%			100,00% 0,00%	
		0	1+2+3+4			0+1		Dano P

(a) Estágio 1

(b) E

Dano Real	0+1+2	2728 30,31%	272 3,02%	90,93% 9,07%	Dano Real	0+1+2+3	3753 53,61%	
	3+4	0 0,00%	6000 66,67%	100,00% 0,00%		4	0 0,00%	
		100,00% 0,00%	95,66% 4,34%	96,98% 3,02%			100,00% 0,00%	
		0+1+2	3+4			0+1+2+3		Dano P

(c) Estágio 3

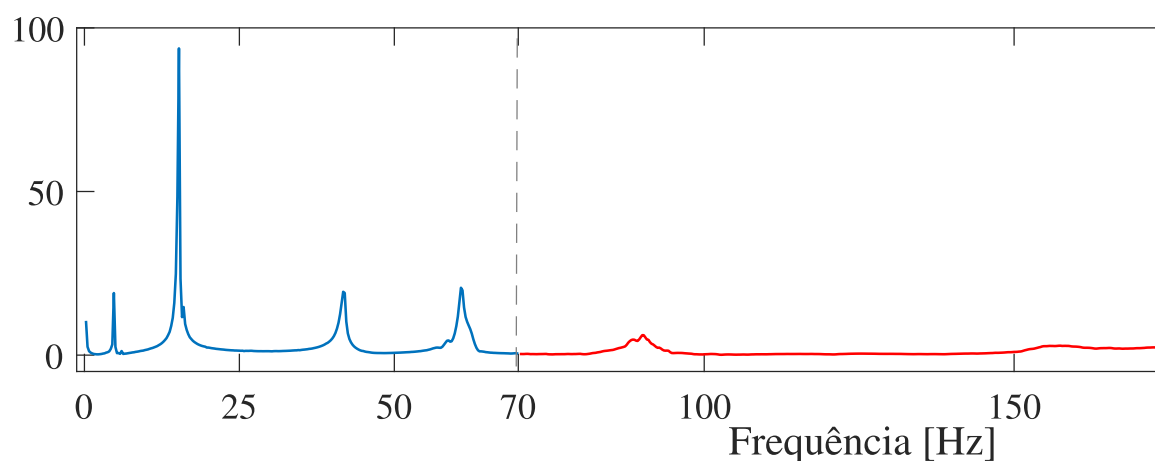
(d) E

Fonte: Do Autor, 2024

5.2 Domínio da Frequência

Contrastando com o domínio do tempo, o domínio da frequência é uma ferramenta central nesse domínio, permitindo a decomposição de sinais em suas componentes de frequência.

Figura 35 – Gráfico no domínio da frequência de uma res



Fonte: Do Autor, 2024

5.2.1 Divisão dos dados

A divisão dos dados de treinamento, validação e teste é discutida no item 5.1.1. Vale ressaltar que os modelos A, B, C e D utilizam a mesma estratégia de divisão, enquanto o modelo E, utiliza-se nas demais estratégias.

5.2.2 Otimização dos parâmetros

A arquitetura da rede neural aplicada neste estudo envolve um processo de otimização de parâmetros. Os parâmetros de otimização, apresentados no item 5, apresentam características que, independentemente da estratégia utilizada, proporcionar resultados mais eficazes. As funções de ativação utilizadas nas camadas convolucionais foram ReLU, com exceção da última camada de saída.

A tabela está segmentada conforme as duas estr

Tabela 5 – Análise Pórtico - Parâmetros Fixos

ANÁLISE PÓRTICO	
PARÂMETROS FIXOS	
Domínio de Treinamento	Frequência
Dimensão Sinal de Entrada	[280, 280]
Solucionador	Adaptativo
Função de Ativação	ReLU
Análise da rede	Erro de Rede
Modelos de Treinamento	A, B, C
Nº de Estratégias para Detecção	1
Tipo do Autocodificador	CA
Nº de Camadas Convolucionais	2
Nº de Estágios	5
Estágios de Treinamento	4
Acelerômetros	4
Nº de Dados de Treinamento [Amplitude Estática]	4

6 – Análise Pórtico - Parâmetros Variáveis

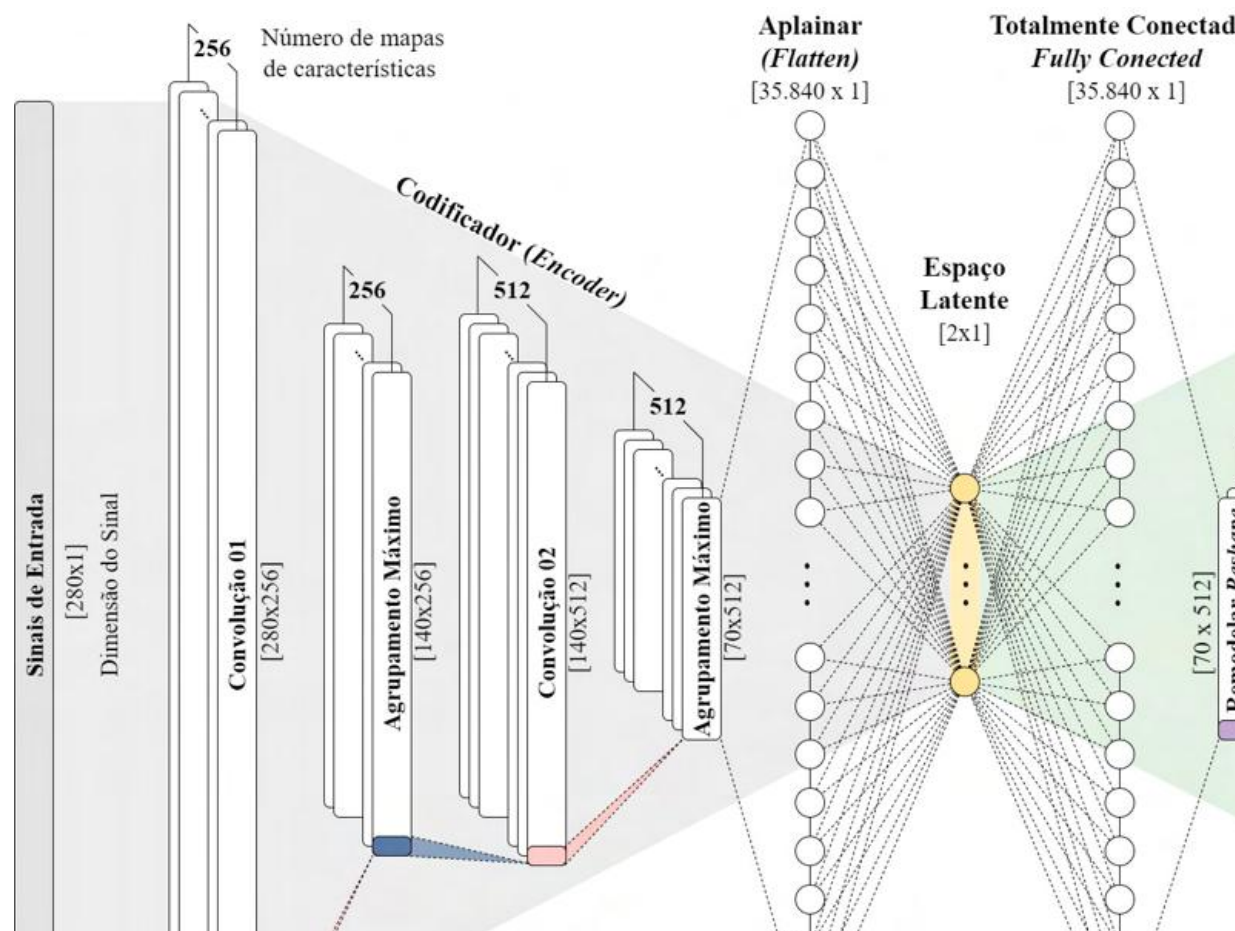
ANÁLISE PÓRTICO														
PARÂMETROS VARIÁVEIS														
5			10			15			25			50		
Ac_1			Ac_1			Ac_1			Ac_1			Ac_1		
4	8	2	4	8	2	4	8	2	4	8	2	4	8	
150	300	75	150	300	75	150	300	75	150	300	75	150	300	
7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500
REDES														
5	6	7	8	9	10	11	12	13	14	15	16	17	18	
128		128			128			128			128			
55	2,598	3,058	2,754	2,721	3,035	2,699	2,579	3,089	2,764	2,649	2,935	2,801	2,728	3,053
30	3,381	4,683	3,809	3,713	4,613	3,654	3,330	4,783	3,865	3,527	4,319	3,934	3,744	4,669
	112	165	65	105	149	65	115	208	64	113	207	61	115	177
0	5629	4150	6519	5263	3735	6570	5774	5203	6433	5677	5183	6116	5799	4444
9	06:01	06:15	05:29	06:06	06:28	05:27	06:08	06:06	05:42	06:09	06:13	05:38	06:17	06:19
	23	24	25	26	27	28	29	30	31	32	33	34	35	36
192		192			192			192			192			
4	2,613	2,997	2,506	2,489	3,014	2,480	2,623	2,858	2,397	2,615	2,993	2,497	2,600	2,964
37	3,417	4,495	3,167	3,116	4,550	3,099	3,455	4,093	2,902	3,423	4,483	3,133	3,383	4,397
	104	207	70	113	167	69	113	187	69	112	147	68	107	174

5.2.3 Arquitetura da Rede

Na análise de otimização, os parâmetros que propo para o objetivo do estudo foram os seguintes: a dimensã como 2, o tamanho do filtro foi configurado como 256, *Batch* foi estabelecida em 2.

O número de filtros, um dos parâmetros fundam foi mantido constante em cinco durante todas as análises preenchimento, optou-se pelo uso do *Same Padding* par Quanto ao *Stride* e ao tamanho do filtro, variaram co conforme ilustrado na Figura 36.

Figura 36 – Arquitetura autocodificada



Autoencoder (CAE) adotada consiste em duas camadas pelo mesmo número de camadas convolucionais transpos

Tabela 7 – Arquitetura da E

Camadas	Dimensão de Entrada	Dimensão de Saída
Entrada (<i>Input</i>)	280 x 1 x 1 x 1	-
Convolução - 01 (<i>Convolution</i>)	280 x 1 x 1 x 1	280 x 1 x 2
Agrupamento Máximo (<i>Max Pooling</i>)	280 x 1 x 256 x 1	140 x 1 x 2
Convolução - 02 (<i>Convolution</i>)	140 x 1 x 256 x 1	140 x 1 x 5
Agrupamento Máximo (<i>Max Pooling</i>)	140 x 1 x 512 x 1	70 x 1 x 5
Aplainar (<i>Flatten</i>)	70 x 1 x 512 x 1	35840
Totalmente Conectada (<i>Fully Conected</i>)	35840 x 1	2 x 1
Totalmente Conectada (<i>Fully Conected</i>)	2 x 1	35840
Redimensionamento (<i>Reshape</i>)	35840 x 1	70 x 1 x 5
Desagrupamento Máximo (<i>Max Unpooling</i>)	70 x 1 x 512 x 1	140 x 1 x 5
Deconvolução - 01 (<i>Transposed Convolution</i>)	140 x 1 x 512 x 1	140 x 1 x 2
Desagrupamento Máximo (<i>Max Unpooling</i>)	140 x 1 x 256 x 1	280 x 1 x 2
Deconvolução - 02 (<i>Transposed Convolution</i>)	280 x 1 x 256 x 1	280 x 1 x 1
Saída (<i>Output</i>)	280 x 1 x 1 x 1	-

próprias especificidades e focos, garantindo uma abordagem adequada para o estudo dos danos.

5.2.5 Estratégias para detectar danos

Neste estudo, foram aplicadas duas métricas para avaliar os danos, utilizando quatro estratégias distintas. A primeira estratégia é o erro médio, que mede a diferença entre os sinais de entrada e os sinais reconstruídos. As outras três estratégias empregaram a distância de Mahalanobis no espaço latente.

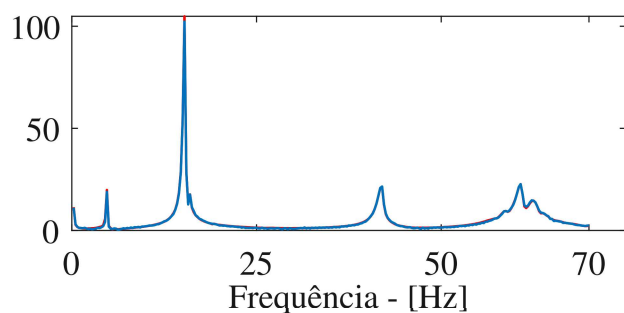
Resultados para a Estratégia por MSE - Estratégia por Erro Médio

Cada rede treinada, tanto por modelo quanto por espaço latente específico, cujo propósito é prever os danos com o menor erro possível durante a reconstrução. Dessa maneira, é possível calcular o erro médio resultante da discrepância entre os sinais de entrada e os sinais reconstruídos. Espera-se que os dados de teste apresentem uma reconstrução mais precisa em comparação com os dados de validação, que exibem um comportamento semelhante ao do treinamento.

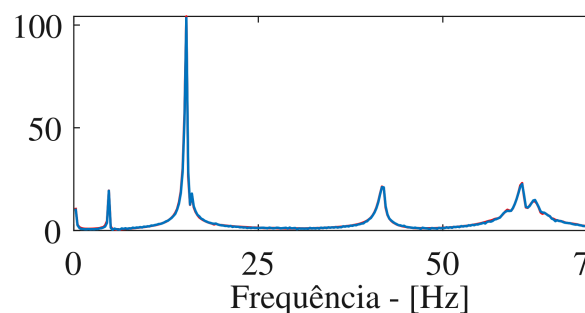
- Reconstrução dos Sinais:

Nesta seção, são apresentados os resultados provenientes do caso estudado, abordando agora o domínio da frequência. São mostrados exemplos típicos reconstruídos, comparando-os aos respectivos sinais originais. É perceptível uma reconstrução mais precisa nos dados de teste em comparação com os dados de validação.

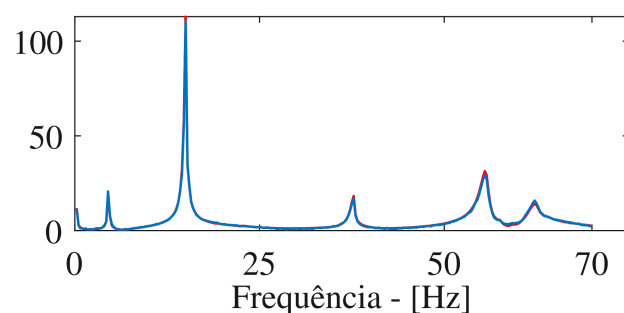
Figura 37 – Predição do Si



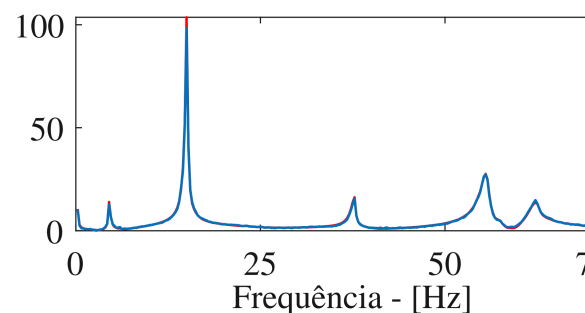
(a) Treino - Estágio 01



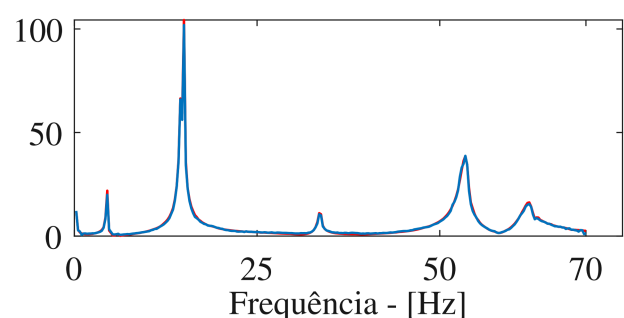
(b) Validação - Estágio 01



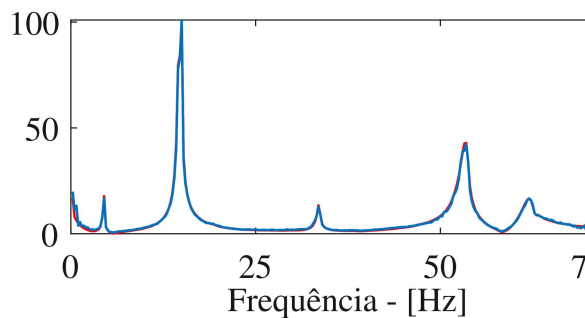
(d) Treino - Estágio 01+02



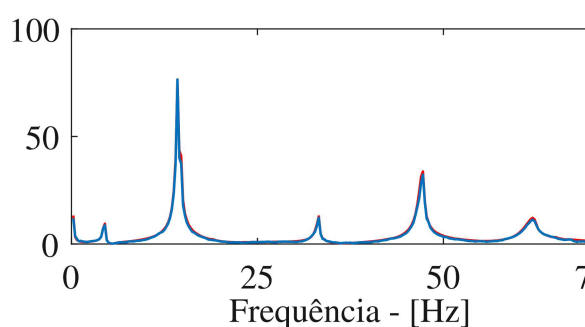
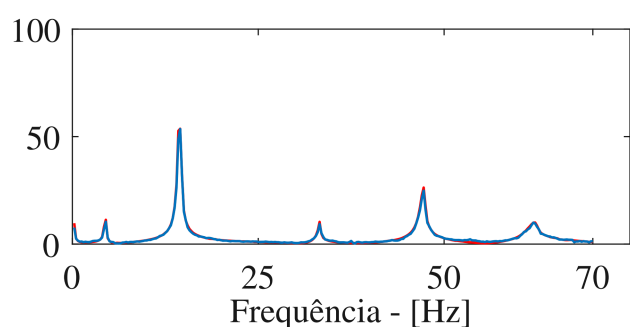
(e) Validação - Estágio 01+02



(g) Treino - Estágio 01+02+03



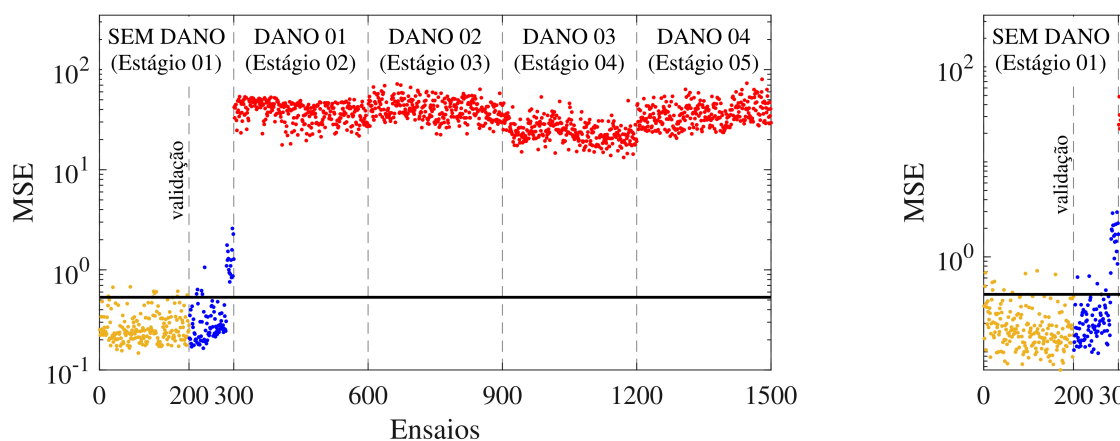
(h) Validação - Estágio 01+02+03



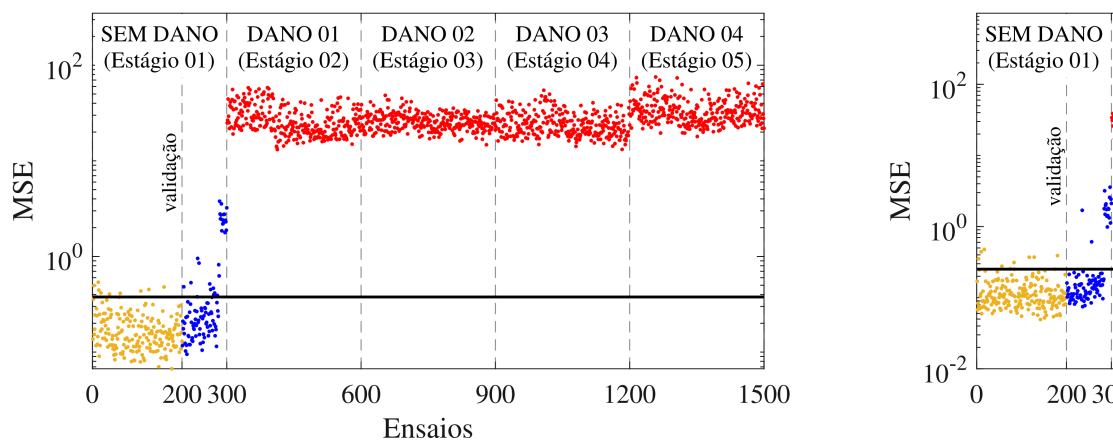
— Dado de entrada — Predito/Reconstruído

esses modelos, independentemente dos acelerômetros em acréscimo de danos durante o treinamento, torna-se viável de MSE dos dados de teste. Essa observação possibilita nos Estágios 1, 2, 3, 4 e 5.

Figura 38 – Erro Quadrático Médio



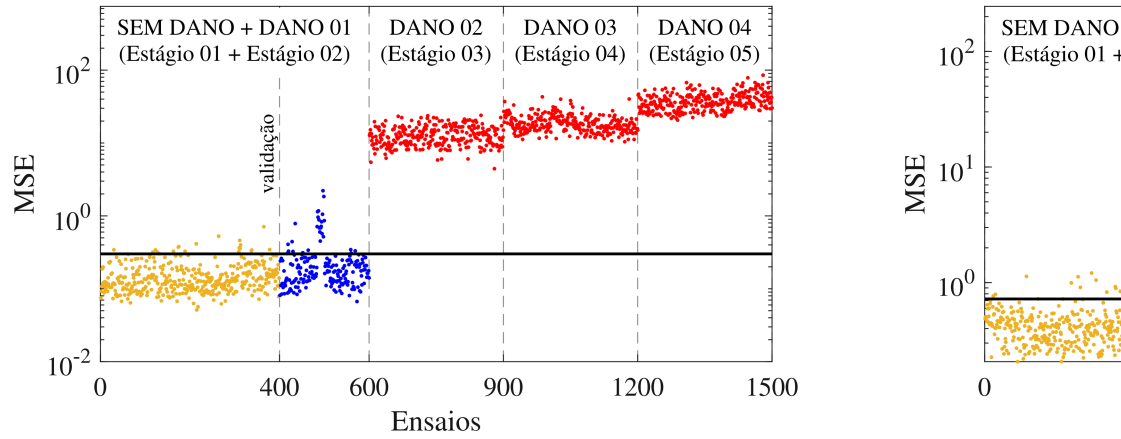
(a) Acelerômetro 1



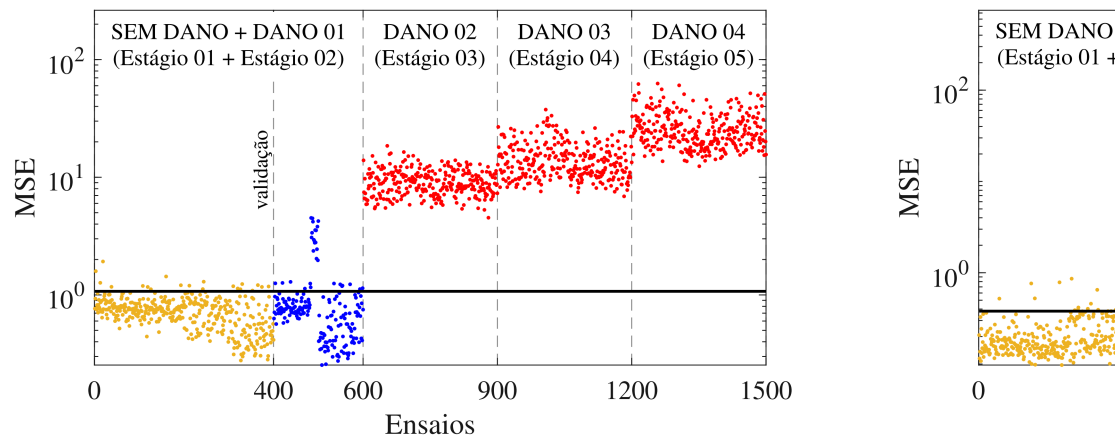
(c) Acelerômetro 3

● Treino ● Validação ● Teste — Limiar

Figura 39 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

● Treino ● Validação ● Teste — Limiar

Fonte: Do Autor, 2024

Figura 40 – Erro Quadrático Médio

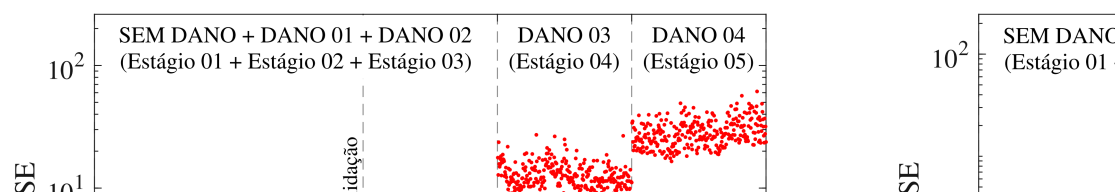
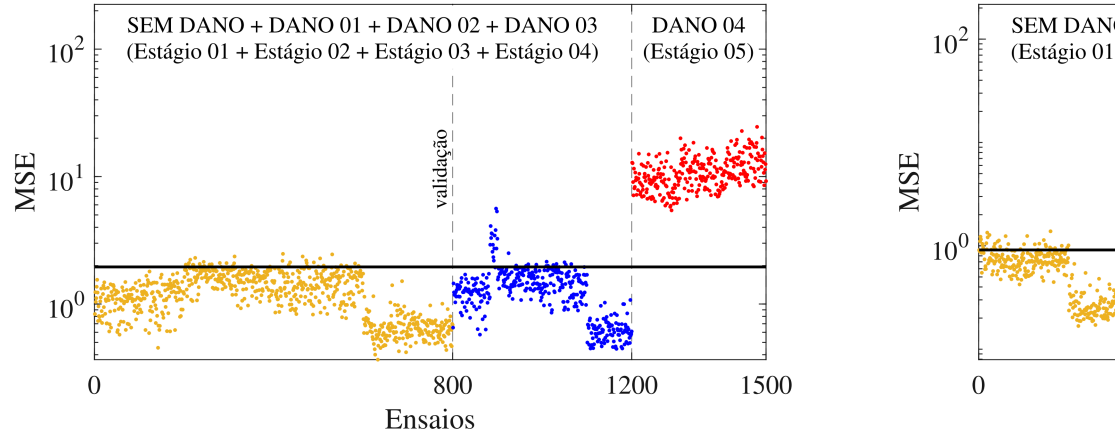
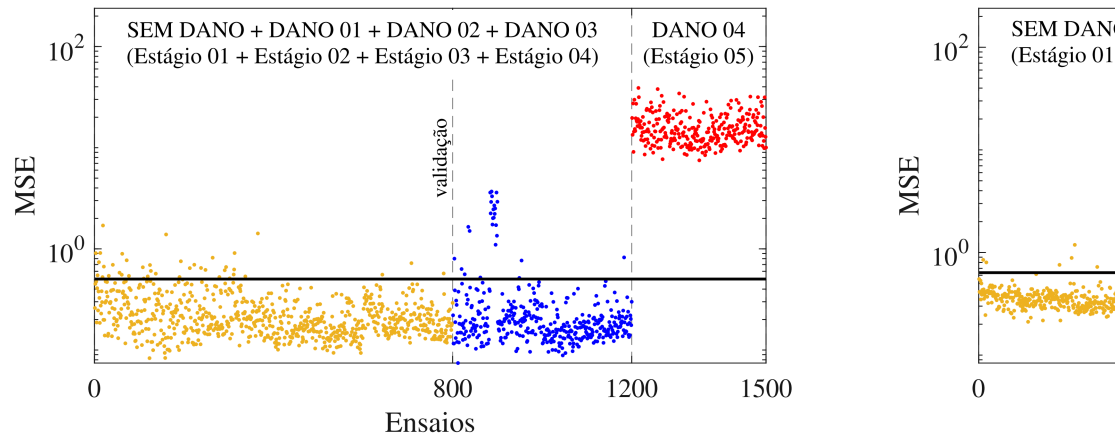


Figura 41 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

● Treino ● Validação ● Teste — Limiar

Fonte: Do Autor, 2024

As Figuras 42, 43, 44 e 45 exibem as matrizes de confusão para os Modelos A, B, C e D, respectivamente. É possível observar nas classificações para ambos os Modelos, apresentando uma classificação geral variando entre 94,44% e 98,19%. É importante

Figura 42 – Matriz de Confusão -

Dano Real	0	764 5,88%	236 1,82%	76,40% 23,60%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	98,07% 1,93%	98,18% 1,82%
		0	1+2+3+4	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	765 5,88%		
	1+2+3+4	0 0,00%		
		100,00% 0,00%		
		0		
		Dano Predito		

(b) Acel

Dano Real	0	758 5,83%	242 1,86%	75,80% 24,20%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	98,02% 1,98%	98,14% 1,86%
		0	1+2+3+4	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	0	745 5,73%		
	1+2+3+4	0 0,00%		
		100,00% 0,00%		
		0		
		Dano Predito		

(d) Acel

Fonte: Do Autor, 2024

Figura 43 – Matriz de Confusão -

Dano Real	0+1	1716 15,60%	284 2,58%	85,80% 14,20%
	3+4	0	9000	100,00%

Dano Real	0+1	1744 15,85%	
	3+4	0	

Figura 44 – Matriz de Confusão -

Dano Real	0+1+2	2714 30,16%	286 3,18%	90,47% 9,53%
	3+4	0 0,00%	6000 66,67%	100,00% 0,00%
		100,00% 0,00%	95,45% 4,55%	96,82% 3,18%
		0+1+2	3+4	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0+1+2	2733 30,37%		
	3+4	0 0,00%		
		100,00% 0,00%		
		0+1+2		
		Dano Predito		

(b) Acel

Dano Real	0+1+2	2704 30,04%	296 3,29%	90,13% 9,87%
	3+4	0 0,00%	6000 66,67%	100,00% 0,00%
		100,00% 0,00%	95,30% 4,70%	96,71% 3,29%
		0+1+2	3+4	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	0+1+2	2690 29,89%		
	3+4	0 0,00%		
		100,00% 0,00%		
		0+1+2		
		Dano Predito		

(d) Acel

Fonte: Do Autor, 2024

Figura 45 – Matriz de Confusão -

Dano Real	0+1+2+3	3611 51,59%	389 5,56%	90,28% 9,73%
		0 0,00%	3000 100,00%	
		0+1+2+3		
		Dano Predito		

Dano Real	0+1+2+3	3689 52,70%		
		0 0,00%		
		0+1+2+3		
		Dano Predito		

Resultados para Estratégia por Mahalanobis - Es

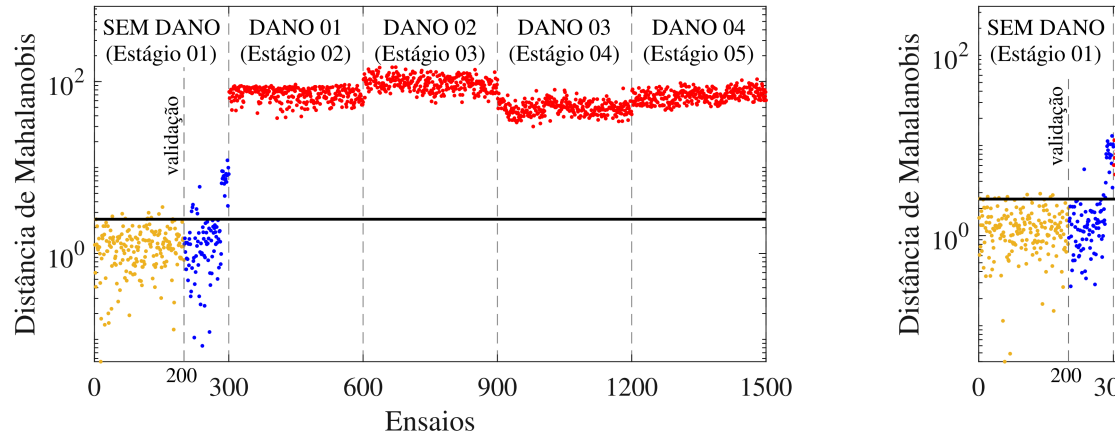
Com base no espaço latente, foram avaliadas três primeiras estratégias visou calcular a distância de Mahalanobis em relação a uma referência específica para cada acelerômetro. A segunda estratégia visou calcular o espaço latente dos quatro acelerômetros em uma única análise. O espaço latente apresentava uma dimensão de $[2 \times 1]$, essa com base em uma matriz de $[8 \times 1]$ por ensaio, a partir da qual se definiu a distribuição de referência. A estratégia buscou adicionar, a cada análise, uma distribuição de referência ao dano seguinte.

Dessa forma, a distância de Mahalanobis foi aplicada calculando a distância de cada ponto em relação à média da distribuição de referência. Essa medida permitiu identificar possíveis danos na estrutura. A distância de Mahalanobis é mais adequada quando há correlação entre as variáveis, proporcionando resultados mais precisos e desvios significativos.

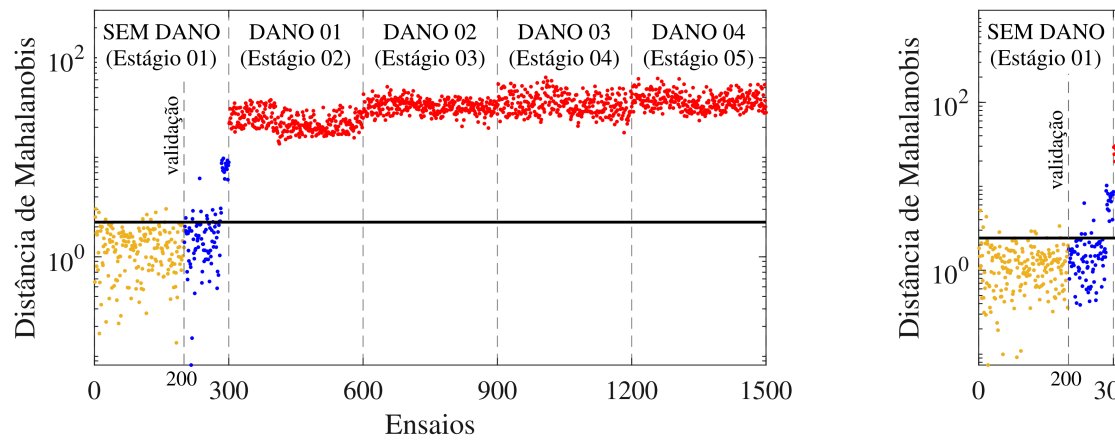
Na análise que compreende o espaço latente, nas figuras 1 e 2, o verde e amarelo correspondem, respectivamente, aos dados de referência e aos dados de dano. Os limiares (indicados pela linha preta) representam o 95º percentil dos valores da distância de Mahalanobis. É importante destacar que o treinamento foi conduzido para mitigar a aleatoriedade do sistema (os demais gráficos foram semelhantes).

- Distância de Mahalanobis - Por Acelerômetro (Est

Figura 46 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

● Distribuição de Referência ● Validação ● Teste — Limite

Fonte: Do Autor, 2024

Figura 47 – Distância de Mahalanobis

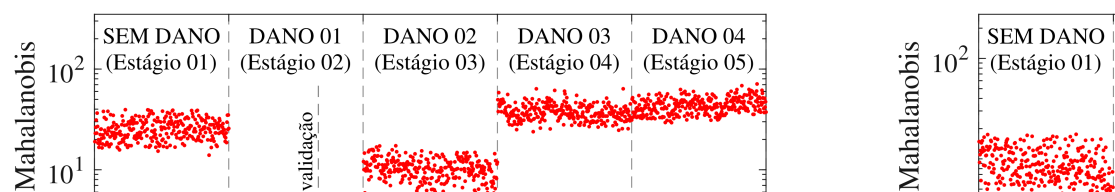
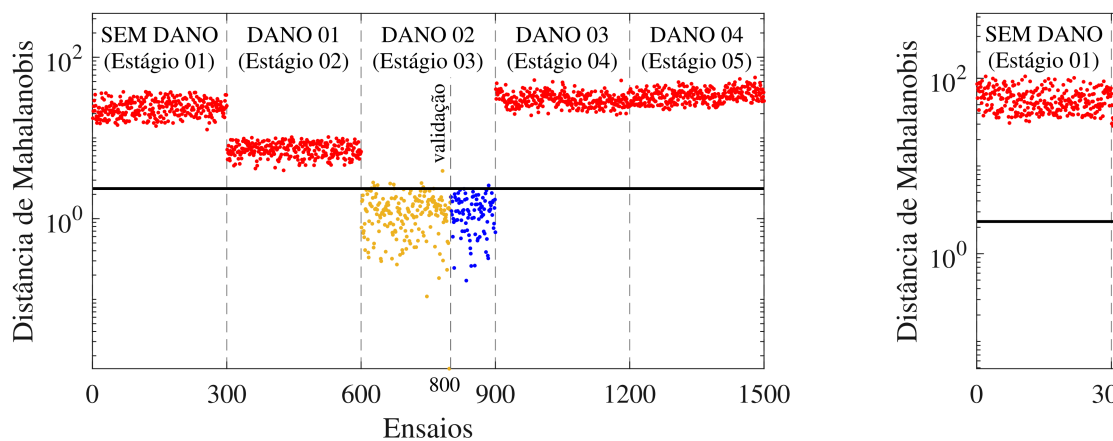
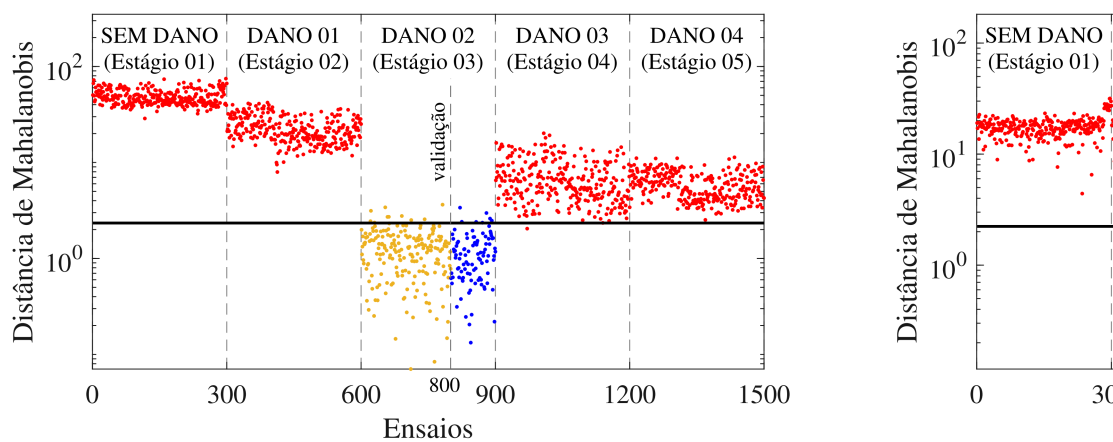


Figura 48 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

● Distribuição de Referência ● Validação ● Teste — Limite

Fonte: Do Autor, 2024

Figura 49 – Distância de Mahalanobis

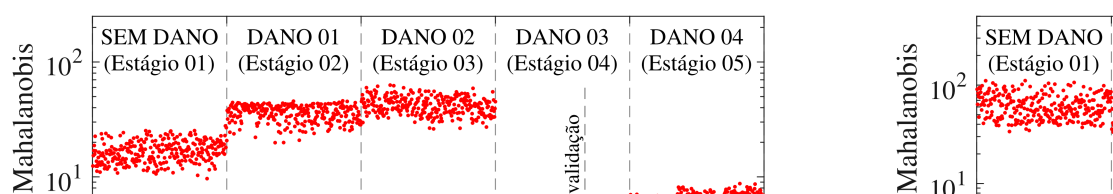
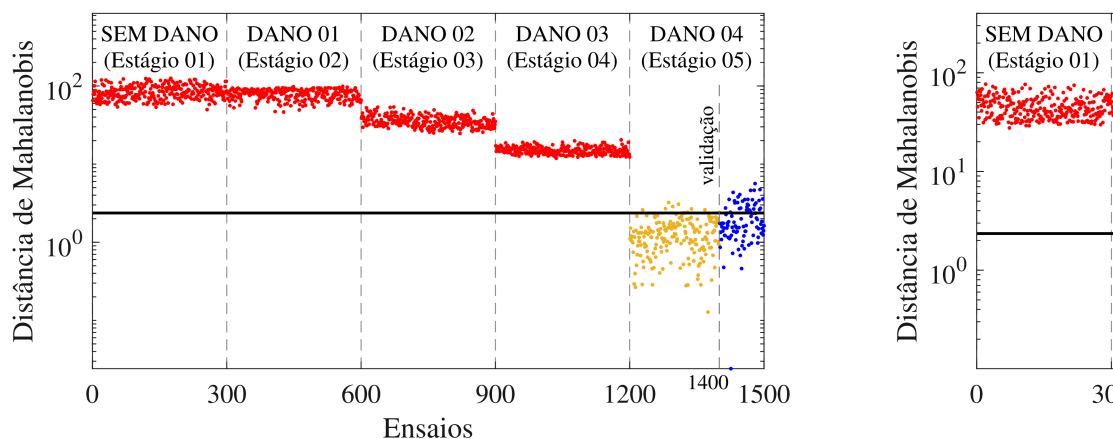
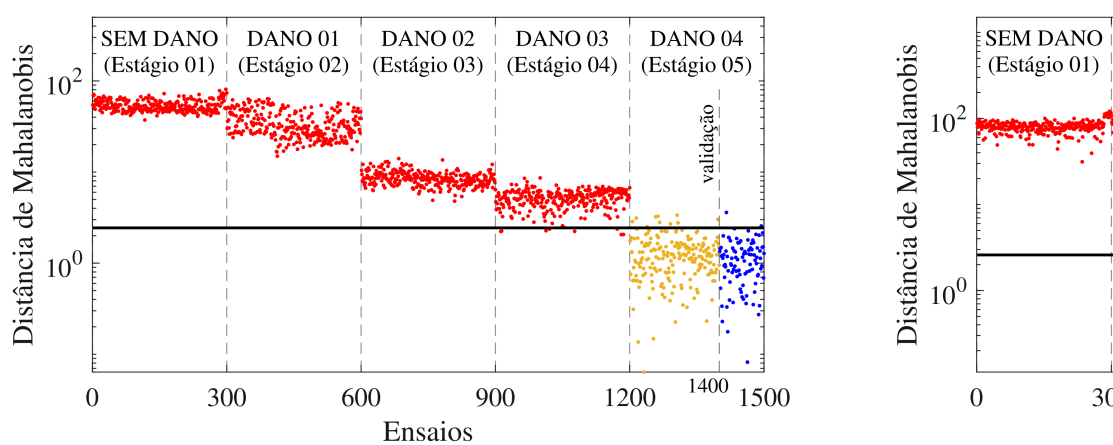


Figura 50 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

Fonte: Do Autor, 2024

As matrizes de confusão para esta estratégia do Mo são apresentadas nas Figuras 51, 52, 53, 54 e 55. Evidências nas classificações para ambos os Modelos, com uma taxa média que variou entre 82,01% e 99,68%. Contudo, c

Figura 51 – Matriz de Confusão -

Dano Real	0	786 6,05%	214 1,65%	78,60% 21,40%
	1+2+3+4	789 6,07%	11211 86,24%	93,43% 6,58%
		49,90% 50,10%	98,13% 1,87%	92,28% 7,72%
		0	1+2+3+4	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	793 6,10%	
	1+2+3+4	0 0,00%	
		100,00% 0,00%	
		0	
		Dano Predito	

(b) Acelo

Dano Real	0	732 5,63%	268 2,06%	73,20% 26,80%
	1+2+3+4	1 0,01%	11999 92,30%	99,99% 0,01%
		99,86% 0,14%	97,82% 2,18%	97,93% 2,07%
		0	1+2+3+4	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	0	757 5,82%	
	1+2+3+4	731 5,62%	
		50,87% 49,13%	
		0	
		Dano Predito	

(d) Acelo

Fonte: Do Autor, 2024

Figura 52 – Matriz de Confusão -

Dano Real	1	987 7,59%	13 0,10%	98,70% 1,30%
	1+3+4	654 5,03%	11346 87,87%	94,55% 5,45%

Dano Real	1	982 7,55%	
	1+3+4	32 0,25%	

Figura 53 – Matriz de Confusão -

Dano Real	2	966 7,43%	34 0,26%	96,60% 3,40%
	0+1+3+4	283 2,18%	11717 90,13%	97,64% 2,36%
		77,34% 22,66%	99,71% 0,29%	97,56% 2,44%
		2	0+1+3+4	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	2	974 7,49%		
	0+1+3+4	610 4,69%		
		61,49% 38,51%		
		2		
		Dano P		

(b) Acel

Dano Real	2	970 7,46%	30 0,23%	97,00% 3,00%
	0+1+3+4	1744 13,42%	10256 78,89%	85,47% 14,53%
		35,74% 64,26%	99,71% 0,29%	86,35% 13,65%
		2	0+1+3+4	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	2	960 7,38%		
	0+1+3+4	1 0,01%		
		99,90% 0,10%		
		2		
		Dano P		

(d) Acel

Fonte: Do Autor, 2024

Figura 54 – Matriz de Confusão -

Dano Real	3	961 7,39%	39 0,30%	96,10% 3,90%
	2+4	676	11324 94,37%	

Dano Real	3	991 7,62%	
	2+4	906	

Figura 55 – Matriz de Confusão -

Dano Real	4	771 5,93%	229 1,76%	77,10% 22,90%
	0+1+2+3	782 6,02%	11218 86,29%	93,48% 6,52%
		49,65% 50,35%	98,00% 2,00%	92,22% 7,78%
		4	0+1+2+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	4	964 7,42%		
	0+1+2+3	530 4,08%		
		64,52% 35,48%		
		4		
		Dano Predito		

(b) Acel

Dano Real	4	928 7,14%	72 0,55%	92,80% 7,20%
	0+1+2+3	2005 15,42%	9995 76,88%	83,29% 16,71%
		31,64% 68,36%	99,28% 0,72%	84,02% 15,98%
		4	0+1+2+3	
		Dano Predito		

(c) Acelerômetro 3

Dano Real	4	850 6,54%		
	0+1+2+3	0 0,00%		
		100,00% 0,00%		
		4		
		Dano Predito		

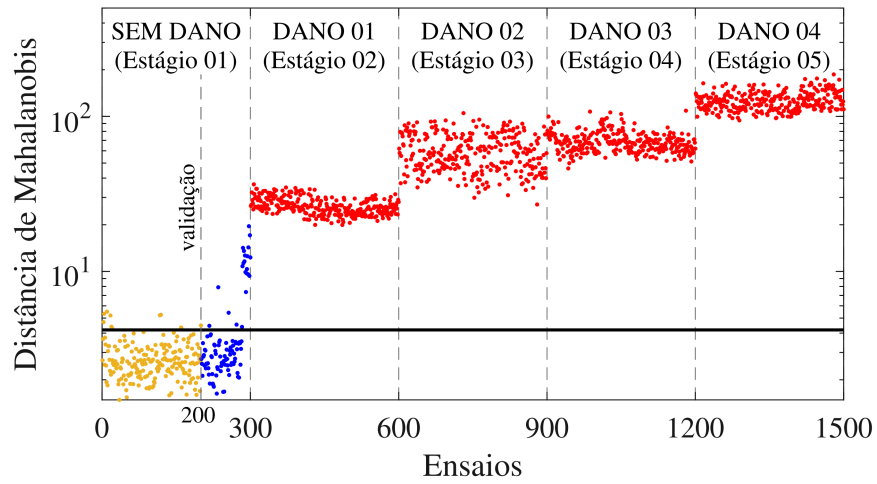
(d) Acel

Fonte: Do Autor, 2024

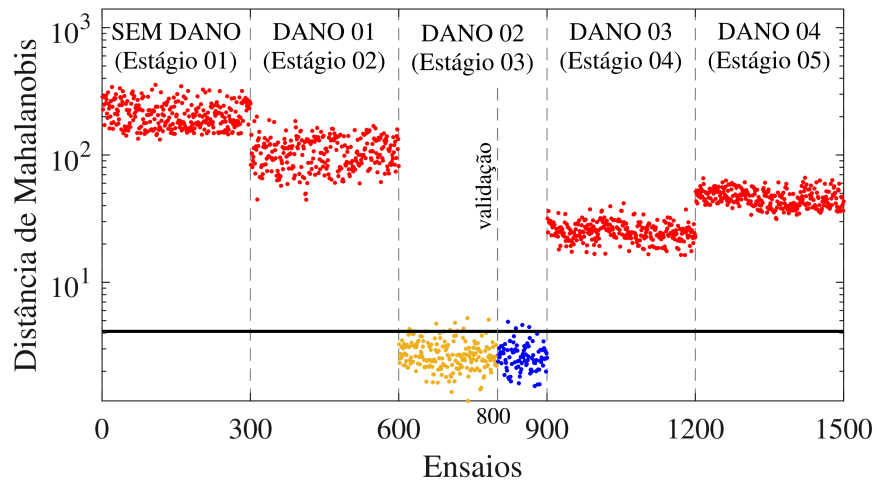
- Distância de Mahalanobis - Acelerômetros Compil

A segunda estratégia, representado pela Figura 31, utiliza a distância de Mahalanobis com base na distribuição de referência dos cinco

Figura 56 – Distância de Mahalanobis - Aceler



(a) Estágio 1



(c) Estágio 3

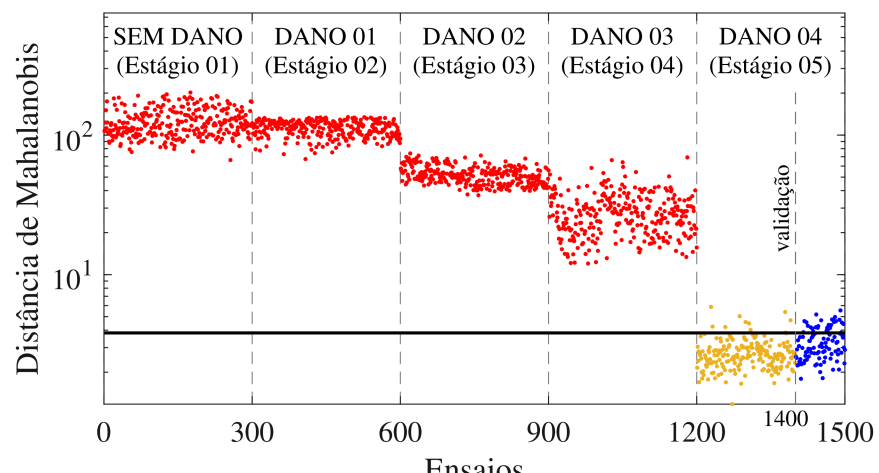


Figura 57 – Matriz de Confusão - Acelerô

Dano Real	0	772 5,94%	228 1,75%	77,20% 22,80%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	98,14% 1,86%	98,25% 1,75%
		0	1+2+3+4	
		Dano Predito		

(a) Estágio 1

Dano Real	1	983 7,56%	17 0,13%	98,30% 1,70%
	0+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,86% 0,14%	99,87% 0,13%
		1	0+2+3+4	
		Dano Predito		

(b) Estágio 2

Dano Real	3	971 7,47%	29 0,22%	97,10% 2,90%
	0+1+2+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,76% 0,24%	99,78% 0,22%
		3	0+1+2+4	
		Dano Predito		

(d) Estágio 4

Dano Real	4	747 5,75%	0 0,00%	100,00% 0,00%
	0+1+2+3	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	99,76% 0,24%	99,78% 0,22%
		4	0+1+2+3	
		Dano Predito		

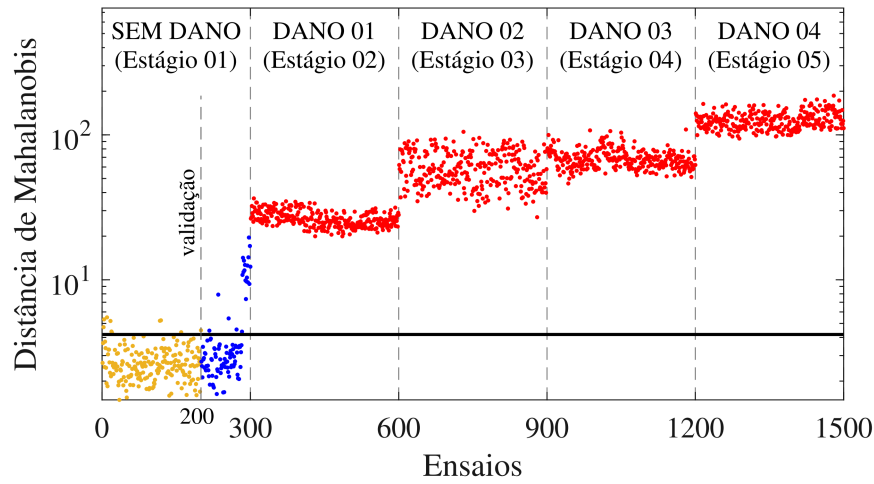
(e) Estágio 5

Fonte: Do Autor, 2024

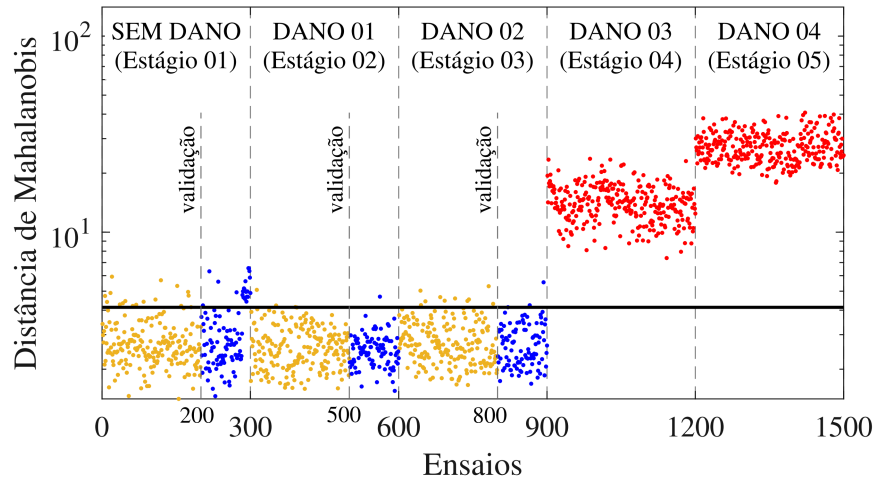
- Distância de Mahalanobis acrescentando Danos (E

A terceira estratégia adotou a abordagem de com
metros em uma única matriz. No entanto, em vez de anal

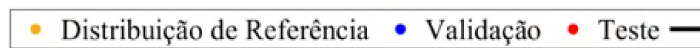
Figura 58 – Distância de Mahalanobis - Ac



(a) Estágio 1



(c) Estágio 3



Fonte: Do Autor, 2024

A análise das matrizes de confusão relacionadas a é apresentada na Figura 59. Verifica-se um desempenho três primeiros estágio de dano, revelando uma taxa média

Figura 59 – Matriz de Confusão - Acr

Dano Real	0	772 5,94%	228 1,75%	77,20% 22,80%
	1+2+3+4	0 0,00%	12000 92,31%	100,00% 0,00%
		100,00% 0,00%	98,14% 1,86%	98,25% 1,75%
		0	1+2+3+4	
		Dano Predito		

(a) Estágio 1

Dano Real	0+1	1780 16,18%		
	2+3+4	0 0,00%		
		100,00% 0,00%		
		0+1		
		Dano Predito		

(b) Estágio 2

Dano Real	0+1+2	2762 30,69%	238 2,64%	92,07% 7,93%
	3+4	10 0,11%	5990 66,56%	99,83% 0,17%
		99,64% 0,36%	96,18% 3,82%	97,24% 2,76%
		0+1+2	3+4	
		Dano Predito		

(c) Estágio 3

Dano Real	0+1+2+3	3785 54,07%		
	4	1629 23,27%		
		69,91% 30,09%		
		0+1+2+3		
		Dano Predito		

(d) Estágio 4

Fonte: Do Autor, 2024

6 APLICAÇÃO 2: PONTE Z-24

Na região de Berna, na Suíça, destaca-se a ponte como um viaduto rodoviário construído no ano de 19 Koppigen e Utzenstorf. Essa ponte clássica em viga caixão um vão principal de 30 metros e dois vãos laterais de 14

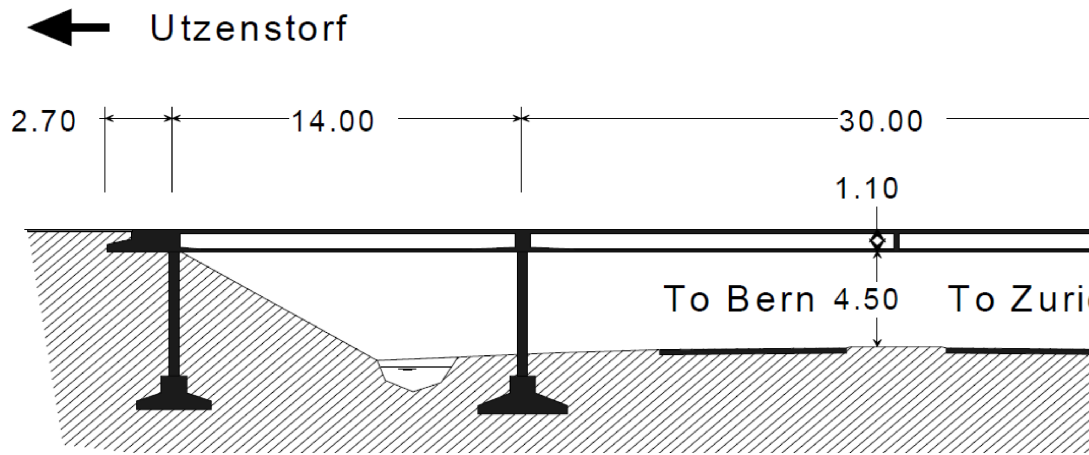
A necessidade de demolir a ponte surgiu devido acomodar uma nova linha ferroviária por baixo. No destruição, a estrutura foi instrumentada e submetida analisar a influência de diferentes estágios de danos reais da estrutura.

Figura 60 – Ponte Z24

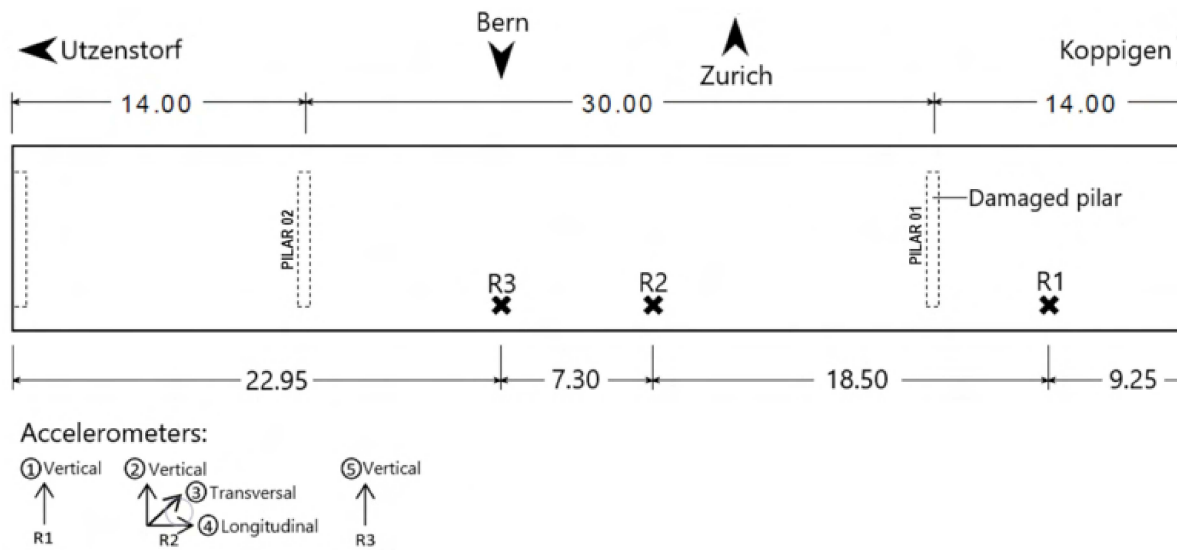


(a) Vista da Ponte Z24

Figura 61 – Predição do D



(a) Vista do corte longitudinal da



(b) Vista Superior

Fonte: Roeck, Peeters e Maeck

Os quatro estágios estruturais estudados, organizados ao longo do tempo, são definidos como:

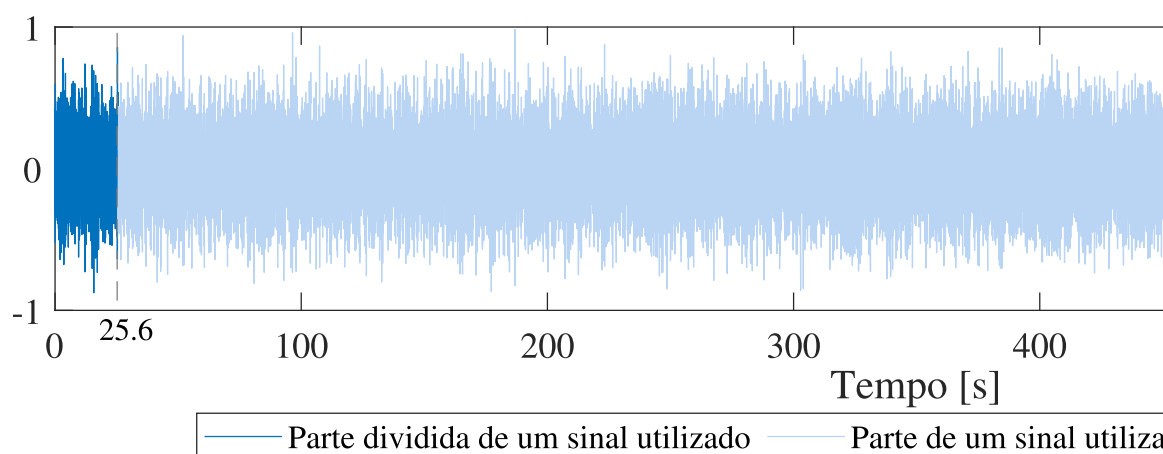
- **Estágio 01:** Não há presença de danos nem equi

ensaios são consolidados para formar uma matriz, na qual cada linha representa um ponto de tempo do acelerômetro e estágio de dano.

6.1 Domínio do Tempo

A figura 62 evidencia um sinal típico obtido pelo domínio do tempo em apenas um ensaio. É possível observar a extensão do sinal (65.530 pontos), e buscando otimizar o aprendizado da rede neural com sinais iguais, resultando em um sinal menor de 2560 pontos ou seja, os sinais passam por um processo de normalização, no qual o sinal é dividido em partes iguais de sua amplitude máxima. Esse procedimento resulta em sinais que variam entre -1 e 1. Após a etapa de normalização, segue o processo de treinamento, validação e testes, seguindo as diretrizes estabelecidas para o treinamento da rede neural.

Figura 62 – Gráfico de resposta típica de um acelerômetro



- **Modelo B:**

- Treinamento: Este modelo foi treinado com Estágio 1 e 168 ensaios do Estágio 2.
- Validação: A avaliação foi feita a partir dos resultados do Estágio 1 e 57 ensaios do Estágio 2.
- Teste: O teste do Modelo B usou os 225 ensaios do Estágio 2.

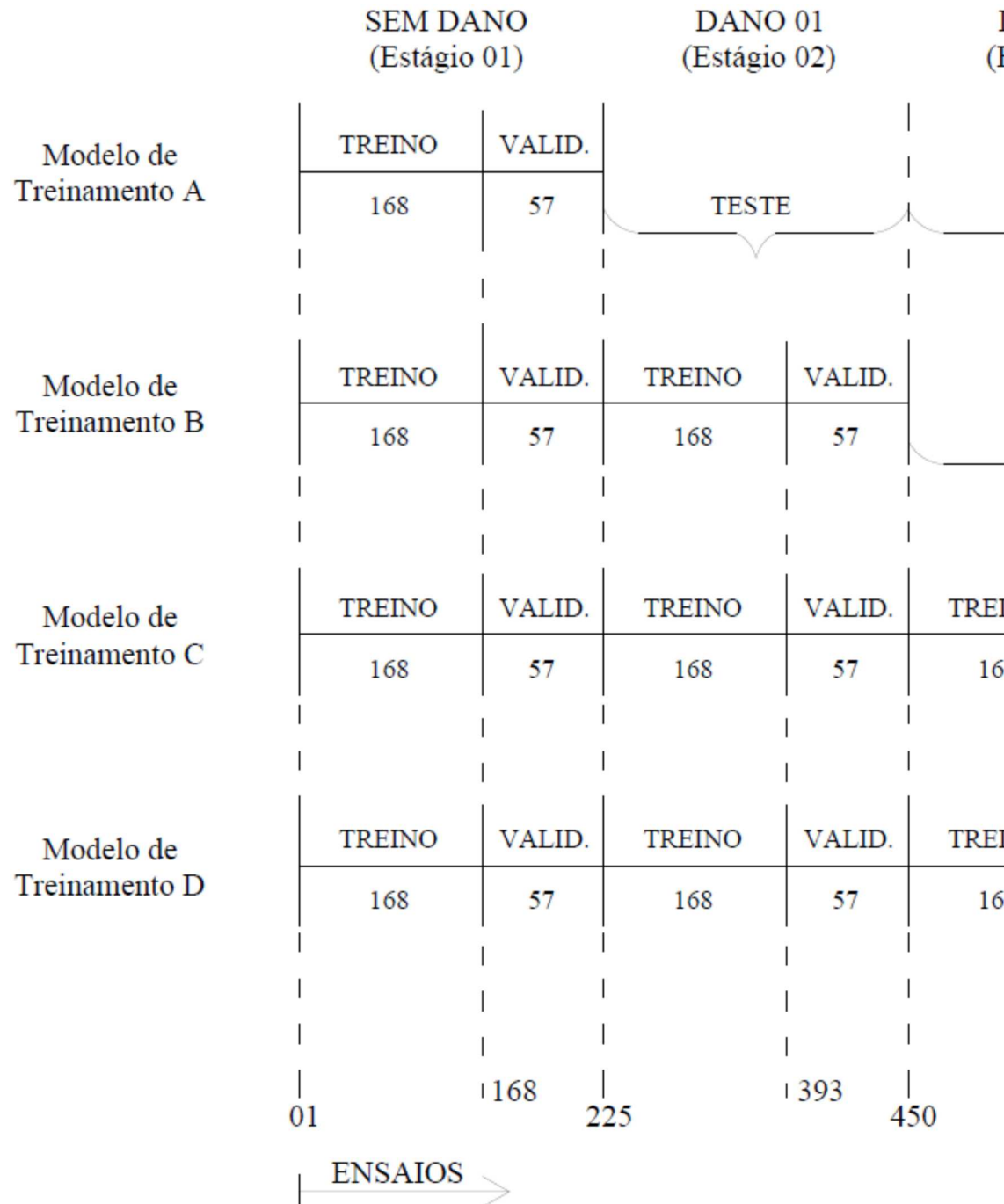
- **Modelo C:**

- Treinamento: Este modelo foi treinado com Estágio 1, 168 ensaios do Estágio 2 e 168 ensaios do Estágio 3.
- Validação: A avaliação foi feita a partir dos resultados do Estágio 1, 57 ensaios do Estágio 2 e 57 ensaios do Estágio 3.
- Teste: O teste do Modelo C usou os 225 ensaios do Estágio 3.

- **Modelo D:**

- Treinamento: Este modelo foi treinado com Estágio 1, 168 ensaios do Estágio 02, 168 ensaios do Estágio 4 e 168 ensaios do Estágio 5.
- Validação: A avaliação foi feita a partir dos resultados do Estágio 1, 57 ensaios do Estágio 2, 57 ensaios do Estágio 4 e 57 ensaios do Estágio 5.
- Teste: O teste do Modelo D é conforme cada

Figura 63 – Modelos de treinamento - D



Fonte: Do autor, 2024

6.1.2 Otimização dos parâmetros

A arquitetura da rede neural aplicada neste estudo

Tabela 8 – Análise Pórtico - Parâmetros

ANÁLISE PÓRTICO	
PARÂMETROS FIXOS	
Domínio de Treinamento	Ten
Dimensão Sinal de Entrada	[2560
Solucionador	Ad
Análise da rede	Erro de Re
Modelos de Treinamento	A, B
Nº de Estratégias para Detecção	1
Tipo do Autocodificador	CA
Nº de Camadas Convolucionais	3
Nº Estágios	4
Estágios de Treinamento	3
Acelerômetros	5
Nº de Redes Treinadas [Acel. x Estágios]	5 x 3
Nº de Redes Treinadas para média	1
Nº de Treinamento	15

Fonte: Do Autor, 2024

6.1.3 Arquitetura da rede

À vista disso, a análise que apresentou melhores resultados foi a que utilizou a divisão do sinal em 2560, dimensão do espaço latente de 10, dimensão do *Batch* de 2, a função de ativação *Relu* e o tamanho de *Batch* de 64 exemplifica a arquitetura da rede aplicada ao treinamento. A Figura 64 mostra três camadas convolucionais e as camadas transpostas u

Figura 64 – Arquitetura autocodificada

A tabela 9 evidencia as dimensões de entrada e saída bem como o quantitativo de camadas convolucionais e camadas totalmente conectadas. O sinal inicial apresentava uma dimensão de 2560, sendo reduzido para 320 no espaço latente com dimensão de 25. E posteriormente inicia-se a etapa da convolução transpostas.

Tabela 9 – Arquitetura da Rede

Camadas	Dimensão de Entrada	Dimensão de Saída
Entrada (<i>Input</i>)	2560 x 1 x 1 x 1	-
Convolução - 01 (<i>Convolution</i>)	2560 x 1 x 1 x 1	2560 x 1 x 1
Agrupamento Máximo (<i>Max Pooling</i>)	2560 x 1 x 128 x 1	1280 x 1 x 1
Convolução - 02 (<i>Convolution</i>)	1280 x 1 x 128 x 1	1280 x 1 x 1
Agrupamento Máximo (<i>Max Pooling</i>)	1280 x 1 x 256 x 1	640 x 1 x 1
Convolução - 03 (<i>Convolution</i>)	640 x 1 x 256 x 1	640 x 1 x 1
Agrupamento Máximo (<i>Max Pooling</i>)	640 x 1 x 512 x 1	320 x 1 x 1
Aplainar (<i>Flatten</i>)	320 x 1 x 512 x 1	163.840
Totalmente Conectada (<i>Fully Connected</i>)	163.840 x 1	25 x 1
Totalmente Conectada (<i>Fully Connected</i>)	25 x 1	163.840
Redimensionamento (<i>Reshape</i>)	163.840 x 1	320 x 1 x 1
Desagrupamento Máximo (<i>Max Unpooling</i>)	320 x 1 x 512 x 1	640 x 1 x 1

estratégia 01 aplicou os modelos A, B e C. Já as estratégias D, focando na identificação de danos estruturais.

6.1.5 Estratégias para detectar danos

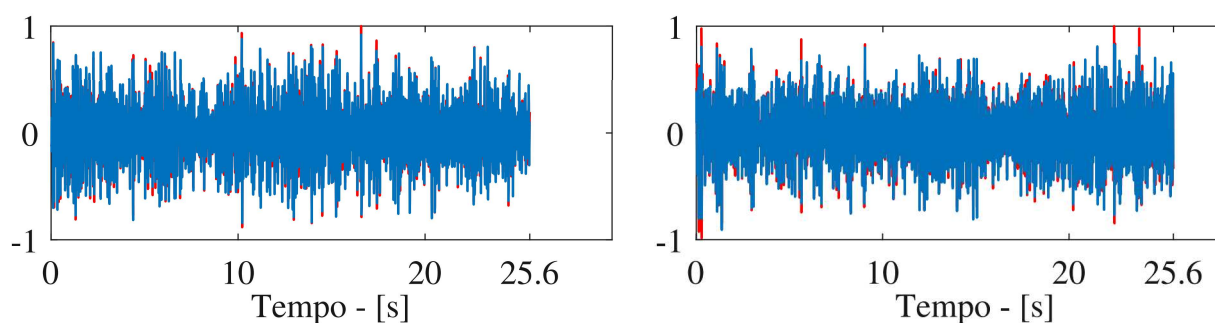
Visando detectar alterações estruturais no conjunto de modo semelhante as quatro estratégias utilizadas no

Resultados para a Estratégia por MSE - Estratégia

- Reconstrução dos Sinais:

A figura 65 demonstra os sinais típicos reconstruídos no exemplo da ponte, comparando-os aos respectivos sinais originais. Como se pode observar, os dados são reconstruídos de modo satisfatório em todos os pontos. Sendo assim, é importante estabelecer uma métrica que seja mais sensível aos estágios de danos. Espera-se que pela análise de variância ou pela distância de Mahalanobis, os estágios de danos sejam

Figura 65 – Predição do Sinal



(a) Treino - Estágio 01

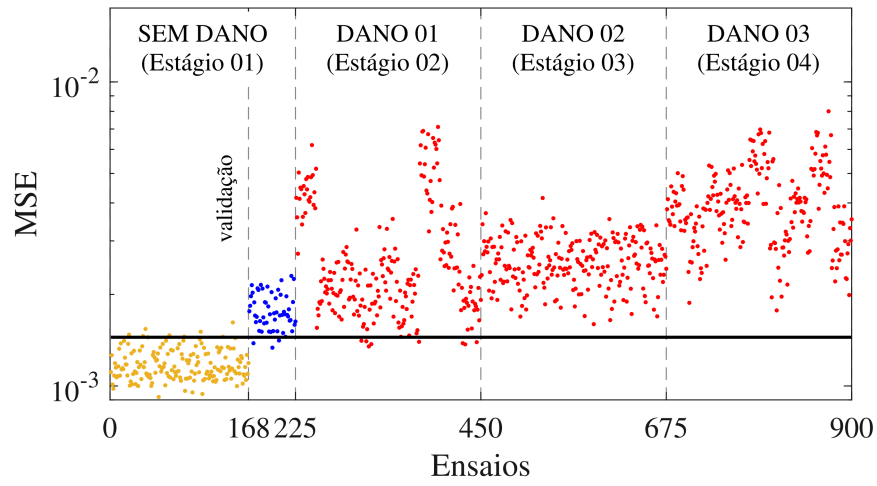
(b) Validação - Estágio 01

- MSE:

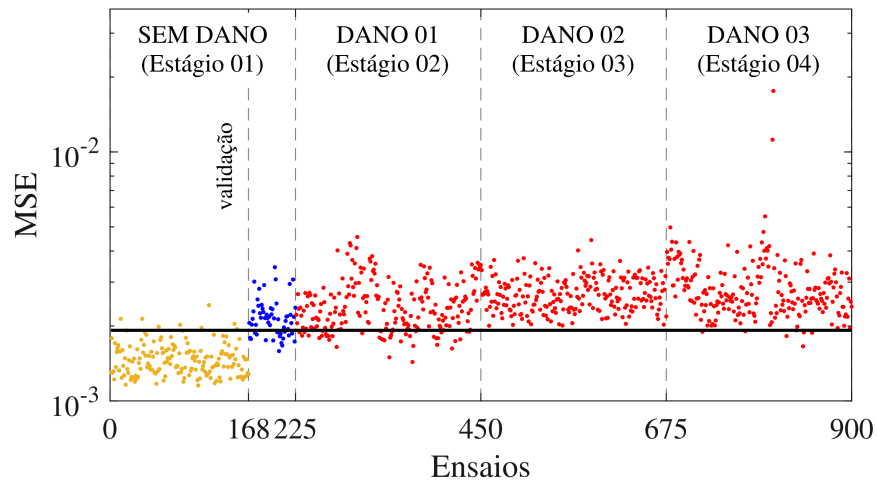
As Figuras 66, 67 e 68 apresentam os valores de MSE obtidos com um dos dez testes realizados (os demais gráficos apresentaram resultados semelhantes), para os Modelos A, B e C. É importante destacar que os MSE de teste para os modelos A e B, obtidos para os dados de validação os MSE foram acima do limite estabelecido. Isso indica altos erros de reconstrução se comparados com os dados de treinamento. O ponto fundamental da análise é o limiar, em alguns casos, os erros menores associados aos dados de teste, contudo os

Nas ilustrações apresentadas, os valores amarelos, correspondentes aos dados de treinamento, validação e teste, respectivamente. Nas gráficas, os limiares (marcados pela linha preta) foram de 0,05 percentil dos valores de Erro Quadrático Médio (MSE) de

Figura 66 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

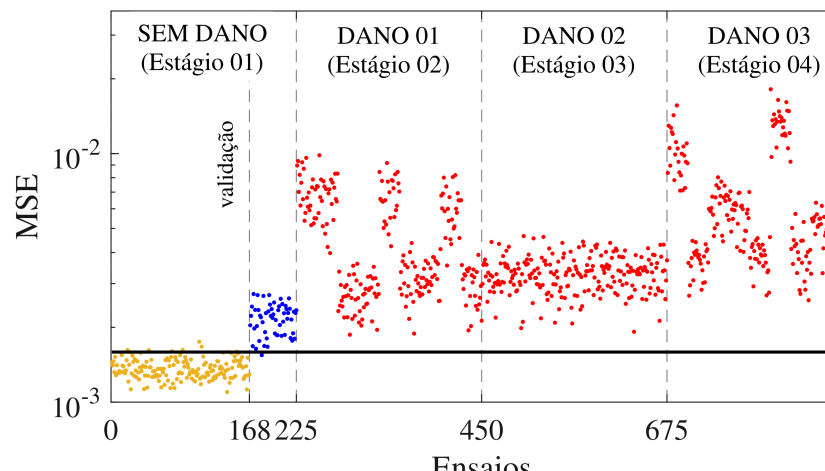
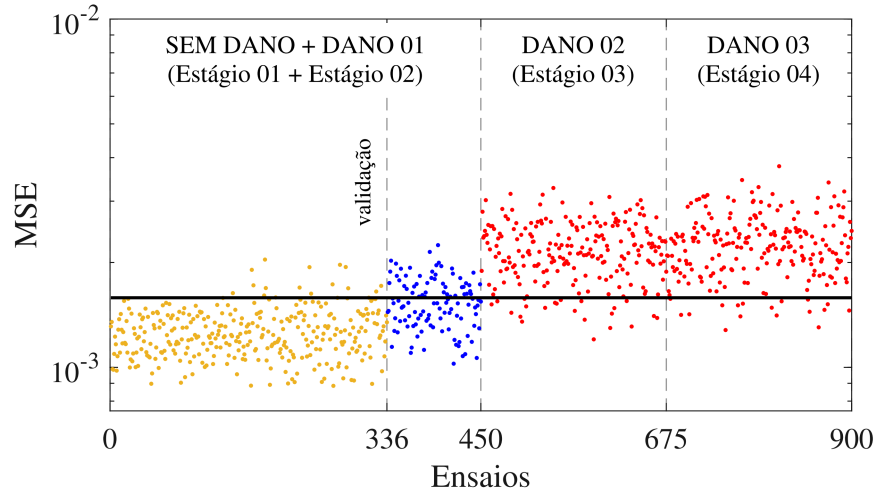
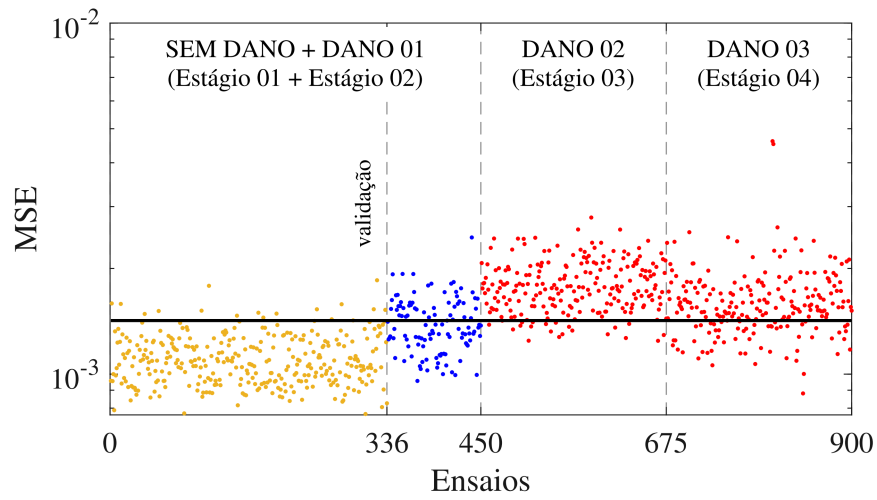


Figura 67 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

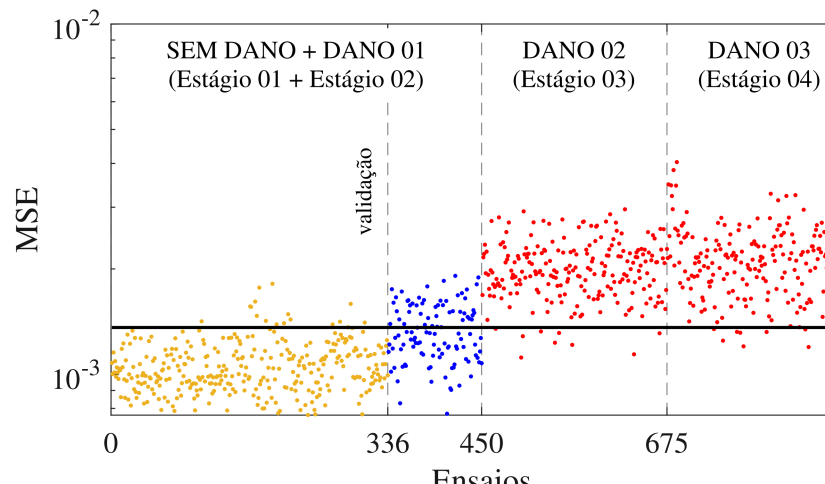
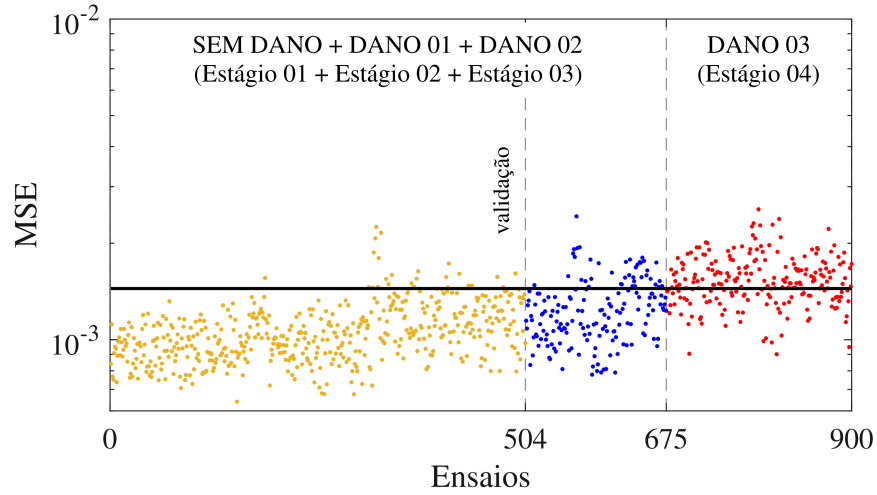
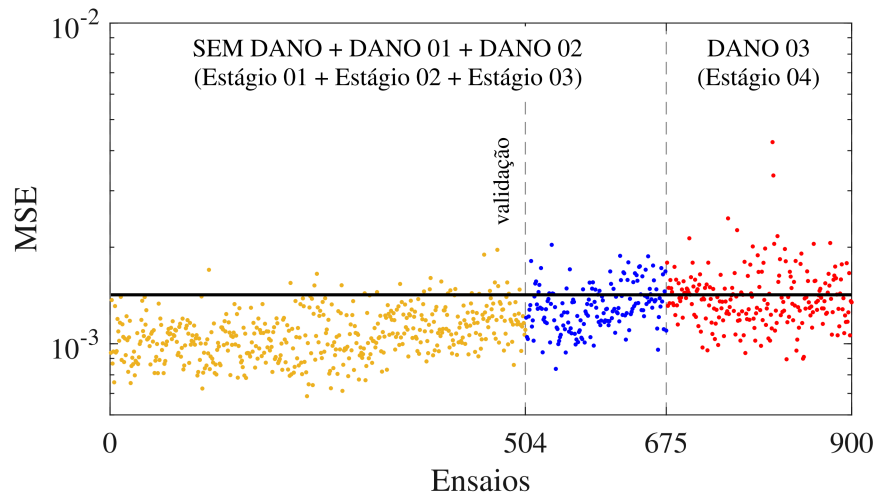


Figura 68 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

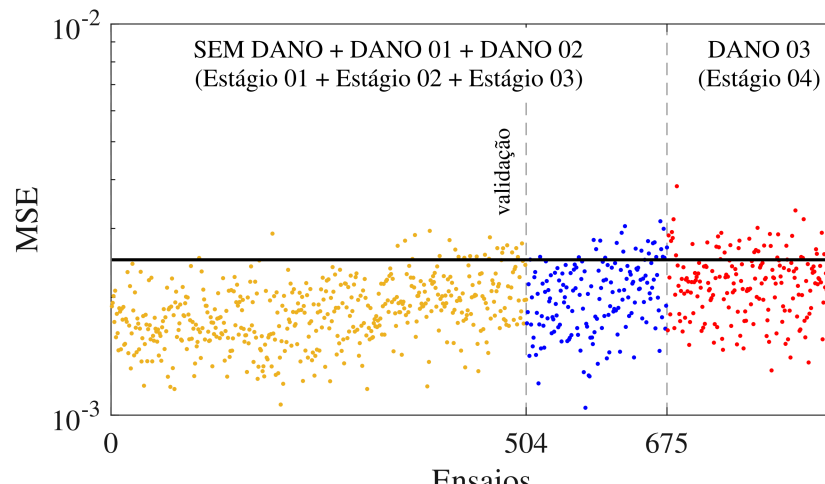


Figura 69 – Matriz de Confusão -

Dano Real	0	120 1,64%	450 6,15%	21,05% 78,95%
	1+2+3	171 2,34%	6579 89,88%	97,47% 2,53%
		41,24% 58,76%	93,60% 6,40%	91,52% 8,48%
		0	1+2+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	23 0,31%	547 7,47%	4,04% 95,96%
	1+2+3	1 0,01%	6749 92,20%	99,99% 0,01%
		95,83% 4,17%	92,50% 7,50%	92,51% 7,49%
		0	1+2+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0	298 4,07%	272 3,72%	52,28% 47,72%
	1+2+3	10 0,14%	6740 92,08%	99,85% 0,15%
		96,75% 3,25%	96,12% 3,88%	96,15% 3,85%
		0	1+2+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0	47 0,64%	
	1+2+3	15 0,20%	
		75,81% 24,19%	
		0	
		Dano P	

(e) Acele

Fonte: Do Autor, 2024

Figura 70 – Matriz de Confusão -

Dano Real	0+1	696 12,34%	444 7,87%	61,05% 38,95%
	3	544 2,34%	3956 89,88%	87,91% 2,53%
		41,24% 58,76%	93,60% 6,40%	91,52% 8,48%
		0	1+2+3	
		Dano Predito		

Dano Real	0+1	659 11,68%	481 8,53%	57,81% 42,19%
	3	213 0,31%	4287 92,20%	95,27% 0,01%
		95,83% 4,17%	92,50% 7,50%	92,51% 7,49%
		0	1+2+3	
		Dano Predito		

Figura 71 – Matriz de Confusão -

Dano Real	0+1+2	1384 34,95%	326 8,23%	80,94% 19,06%
	3	1109 28,01%	1141 28,81%	50,71% 49,29%
		55,52% 44,48%	77,78% 22,22%	63,76% 36,24%
		0+1+2	3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0+1+2	1373 34,67%	337 8,51%	80,29% 19,71%
	3	435 10,98%	1815 45,83%	80,67% 19,33%
		75,94% 24,06%	84,34% 15,66%	80,51% 19,49%
		0+1+2	3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0+1+2	1366 34,49%	344 8,69%	79,88% 20,12%
	3	1795 45,33%	455 11,49%	20,22% 79,78%
		43,21% 56,79%	56,95% 43,05%	45,98% 54,02%
		0+1+2	3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0+1+2	1385 34,97%		
	3	1228 31,01%		
		53,00% 47,00%		
		0+1+2		
		Dano P		

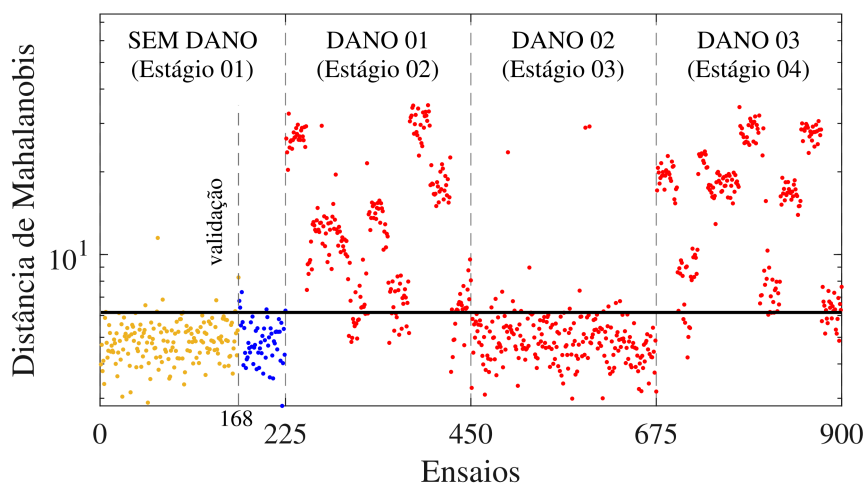
(e) Acele

Fonte: Do Autor, 2024

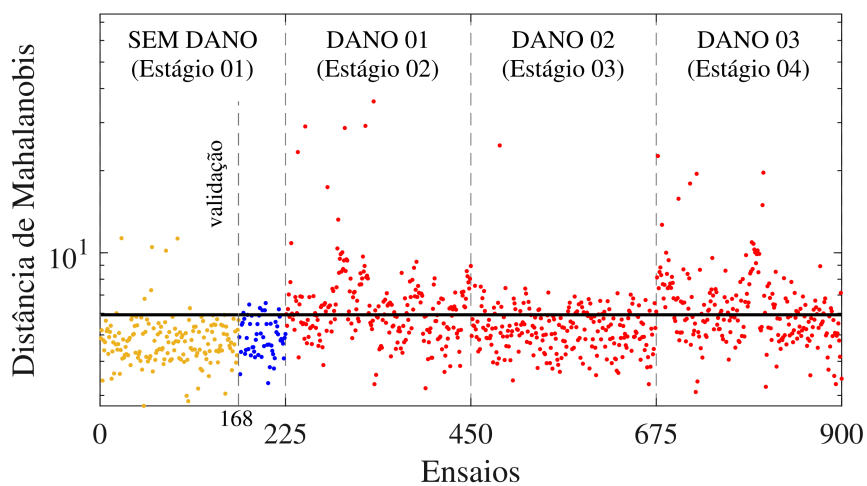
Resultados para Estratégia por Mahalanobis - Es

As estratégias 01, 02 e 03 do modelo D de treinamento Mahalanobis, visam distinguir os estágios de dano presentes, dispostas as estratégias com seus respectivos resultados.

Figura 72 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

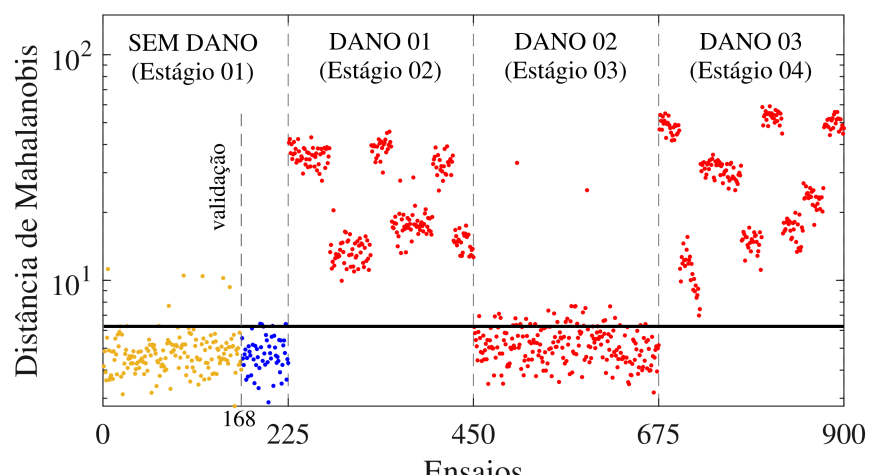
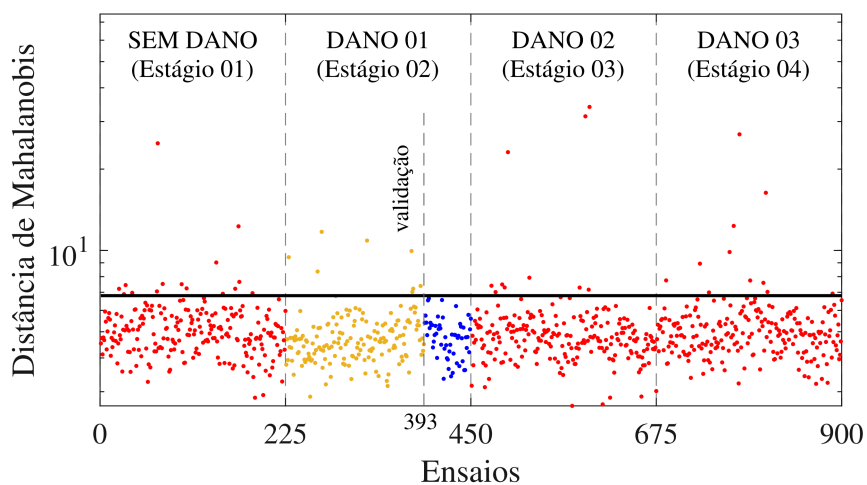
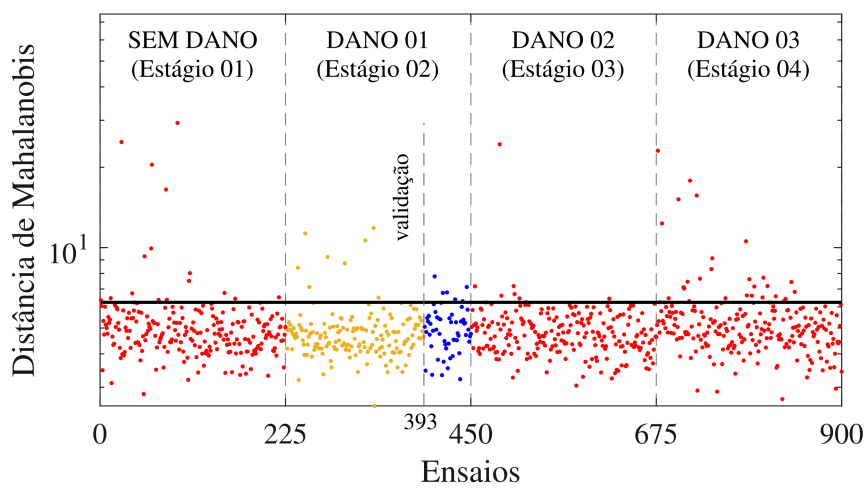


Figura 73 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

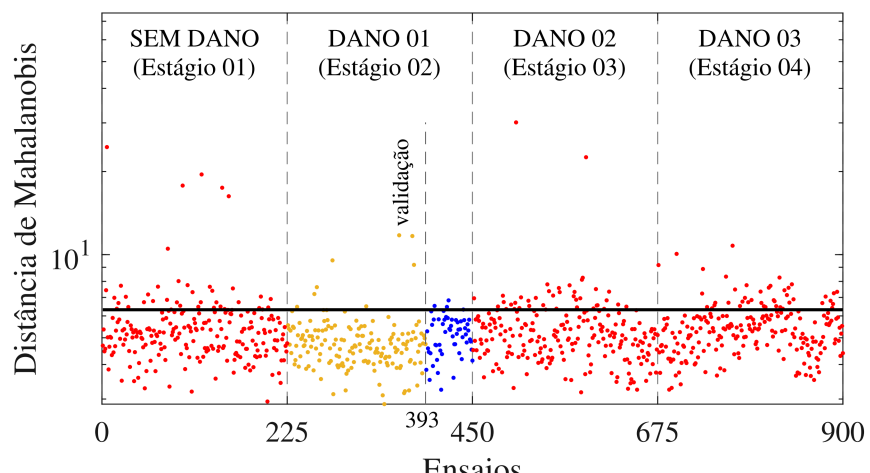
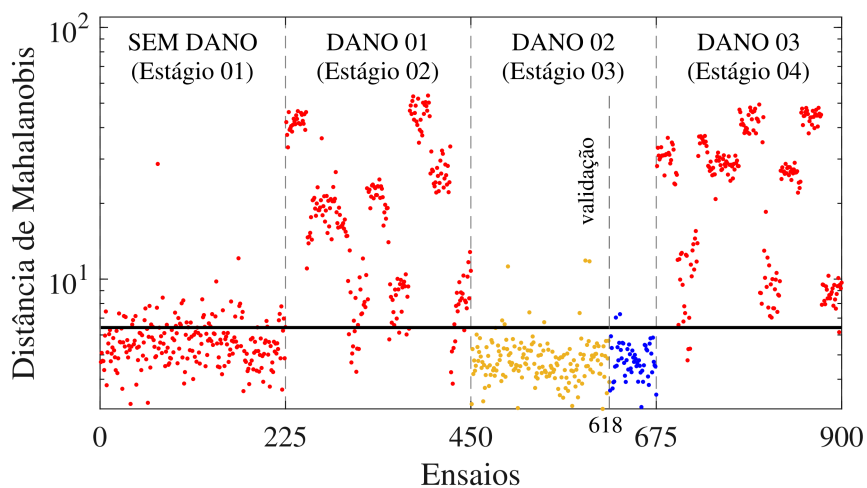
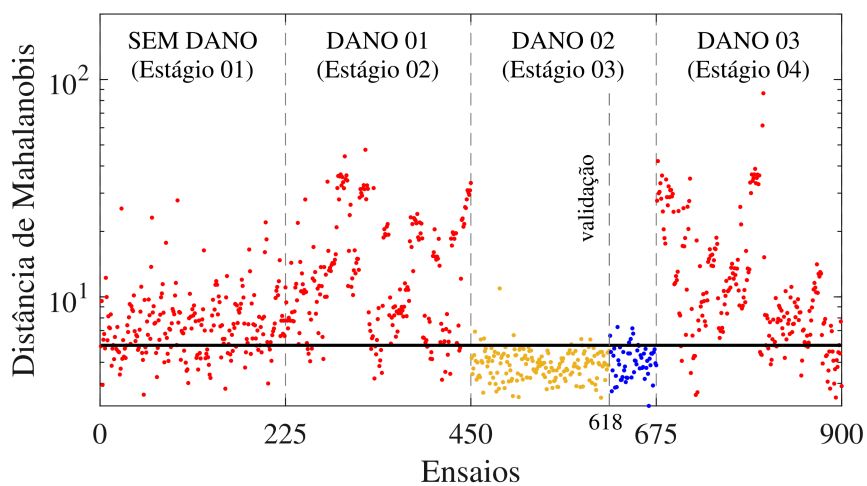


Figura 74 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

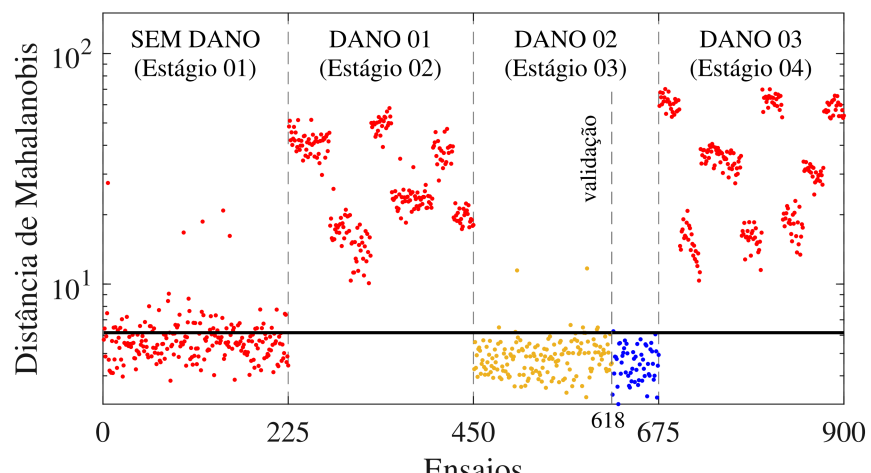
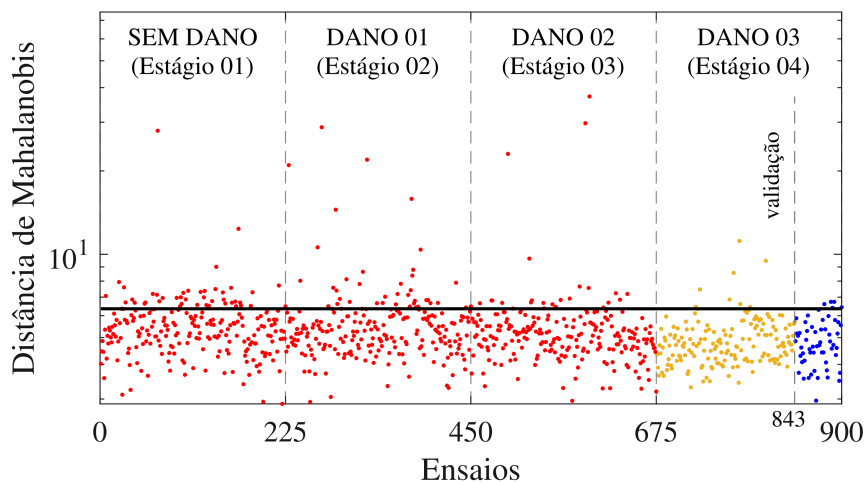
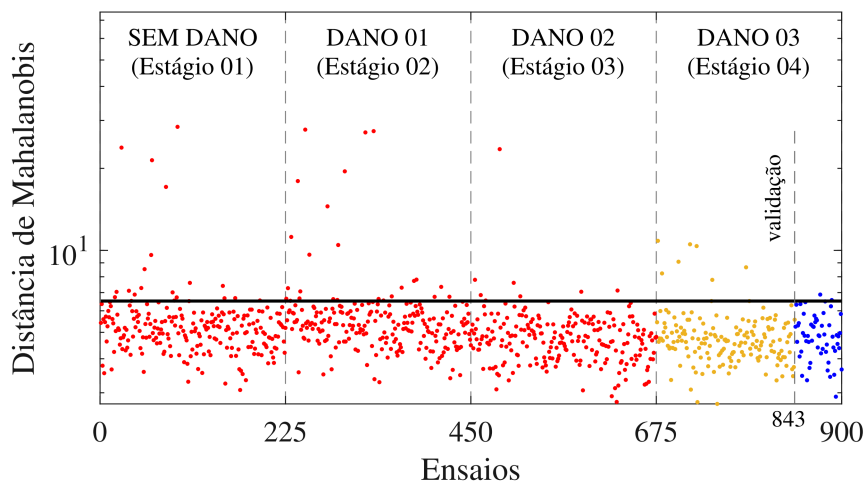


Figura 75 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

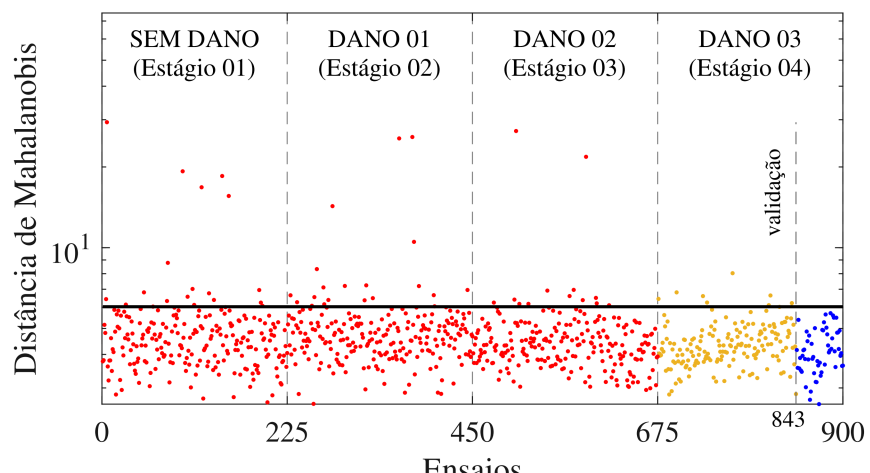


Figura 76 – Matriz de Confusão -

Dano Real	0	496 6,78%	74 1,01%	87,02% 12,98%
	1+2+3	2358 32,21%	4392 60,00%	65,07% 34,93%
		0	1+2+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	511 6,98%	59 0,81%	89,65% 10,35%
	1+2+3	1879 25,67%	4871 66,54%	72,16% 27,84%
		0	1+2+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0	533 7,28%	37 0,51%	93,51% 6,49%
	1+2+3	3462 47,30%	3288 44,92%	48,71% 51,29%
		0	1+2+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0	511 6,98%	
	1+2+3	2098 28,66%	
		0	
		Dano P	

(e) Acele

Fonte: Do Autor, 2024

Figura 77 – Matriz de Confusão -

Dano Real	1	527 7,20%	43 0,59%	92,46% 7,54%
	2+3	6019	731 10,83%	

Dano Real	1	535 7,31%	35 0,48%	93,86% 6,14%
	2+3	4580	2170 32,15%	

Figura 78 – Matriz de Confusão -

Dano Real	2	543 7,42%	27 0,37%	95,26% 4,74%
	0+1+3	1816 24,81%	4934 67,40%	73,10% 26,90%
		23,02% 76,98%	99,46% 0,54%	74,82% 25,18%
		2	0+1+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	2	544 7,43%	26 0,36%	95,44% 4,56%
	0+1+3	2624 35,85%	4126 56,37%	61,13% 38,87%
		17,17% 82,83%	99,37% 0,63%	63,80% 36,20%
		2	0+1+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	2	533 7,28%	37 0,51%	93,51% 6,49%
	0+1+3	1167 15,94%	5583 76,27%	82,71% 17,29%
		31,35% 68,65%	99,34% 0,66%	83,55% 16,45%
		2	0+1+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	2	522 7,13%		
	0+1+3	2097 28,65%		
		19,93% 80,07%		
		2		
		Dano P		

(e) Acele

Fonte: Do Autor, 2024

Figura 79 – Matriz de Confusão -

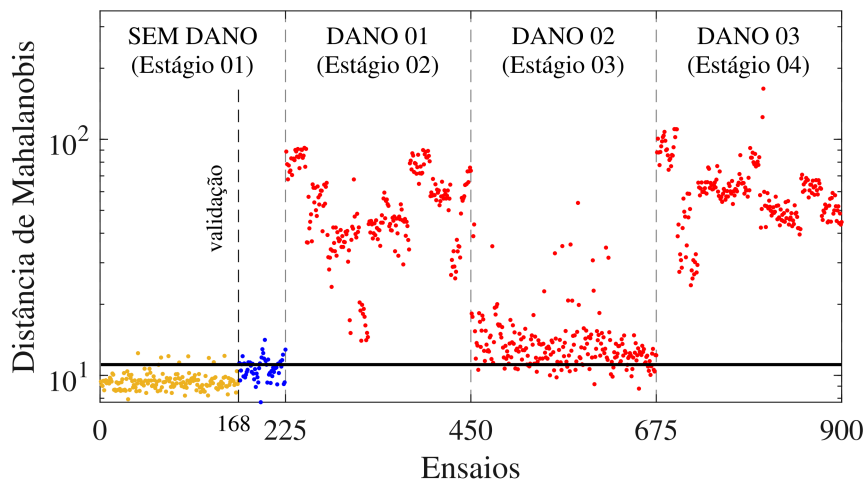
Dano Real	3	532 7,27%	38 0,52%	93,33% 6,67%
	1+2	5565	1185 17,56%	

Dano Real	3	556 7,60%	14 0,19%	97,54% 2,46%
	1+2	4431	2319 34,36%	

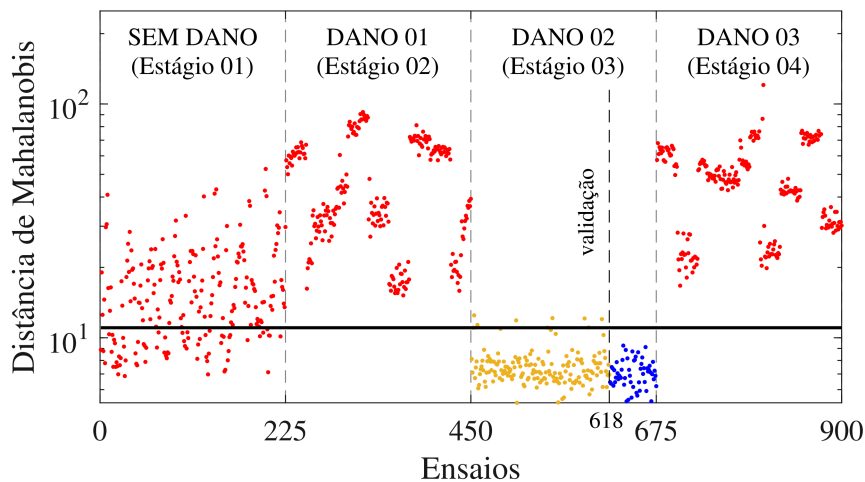
- Distância de Mahalanobis - Acelerômetros Compilados

A Figura 80 ilustra a medida da distância de Mahalanobis para a distribuição de referência dos cinco estágios de danos, utilizados nos dados compilados. Observa-se que este procedimento também se aplica à identificação de anomalias estruturais. Contudo, alguns resultados são melhores que a estratégia anterior.

Figura 80 – Distância de Mahalanobis - Acelerômetros Compilados



(a) Estágio 1



(b) Estágio 2

Figura 81 – Matriz de Confusão - Acelerôn

Dano Real	0	288	282	50,53%
		3,80%	3,72%	49,47%
1+2+3		505	6500	92,79%
		6,67%	85,81%	7,21%
		36,32%	95,84%	89,61%
		63,68%	4,16%	10,39%
		0	1+2+3	
		Dano Predito		

(a) Estágio 1

Dano Real	1	519	51	91,05%
		3,95%	0,39%	8,95%
0+2+3		6065	6500	51,73%
		46,17%	49,49%	48,27%
		7,88%	99,22%	53,44%
		92,12%	0,78%	46,56%
		1	0+2+3	
		Dano Predito		

(b) Estágio 2

Dano Real	3	462	108	81,05%
		3,70%	0,86%	18,95%
0+1+2		5432	6500	54,48%
		43,45%	51,99%	45,52%
		7,84%	98,37%	55,69%
		92,16%	1,63%	44,31%
		3	0+1+2	
		Dano Predito		

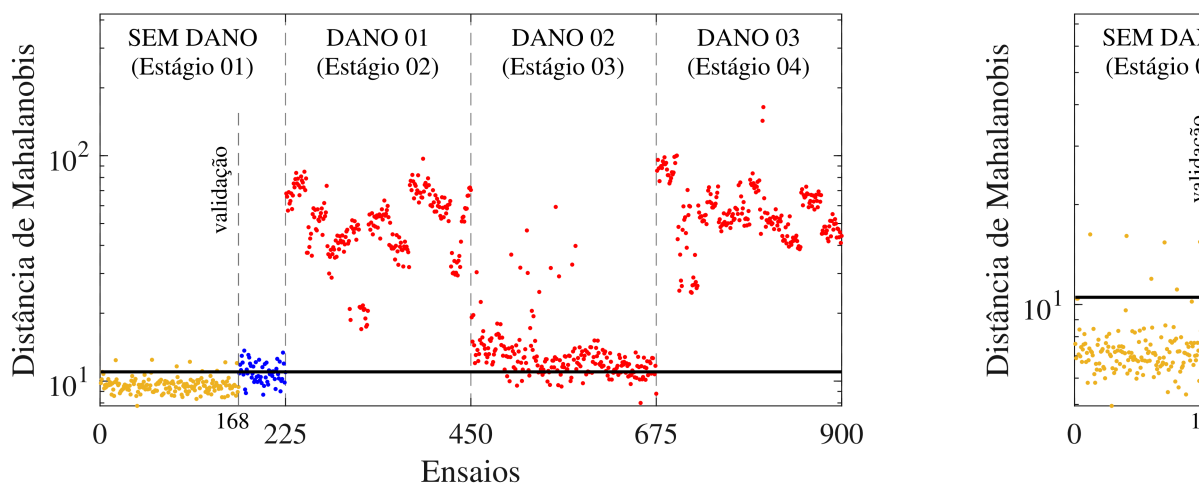
(d) Estágio 4

Fonte: Do Autor, 2024

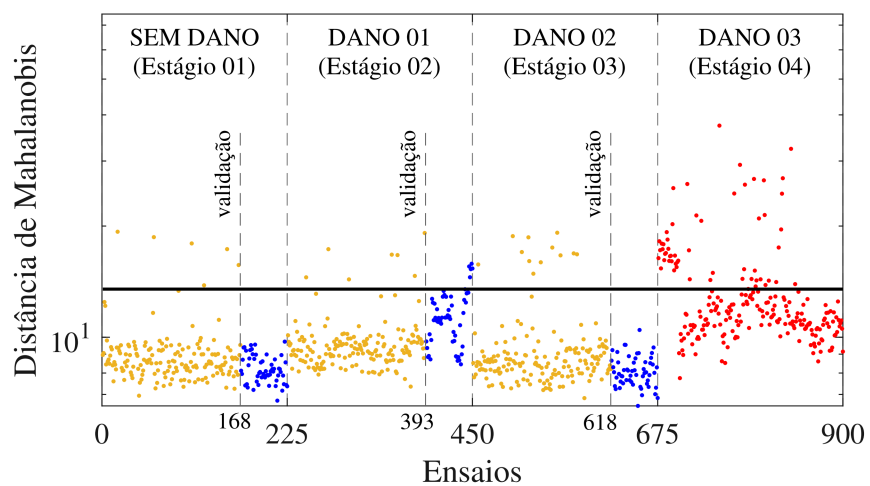
- Distância de Mahalanobis - Acrescentando Danos

A Figura 80 representa a avaliação da distância distribuição de referência dos cinco estágios de danos,

Figura 82 – Distância de Mahalanobis - Ac



(a) Estágio 1



(c) Estágio 3

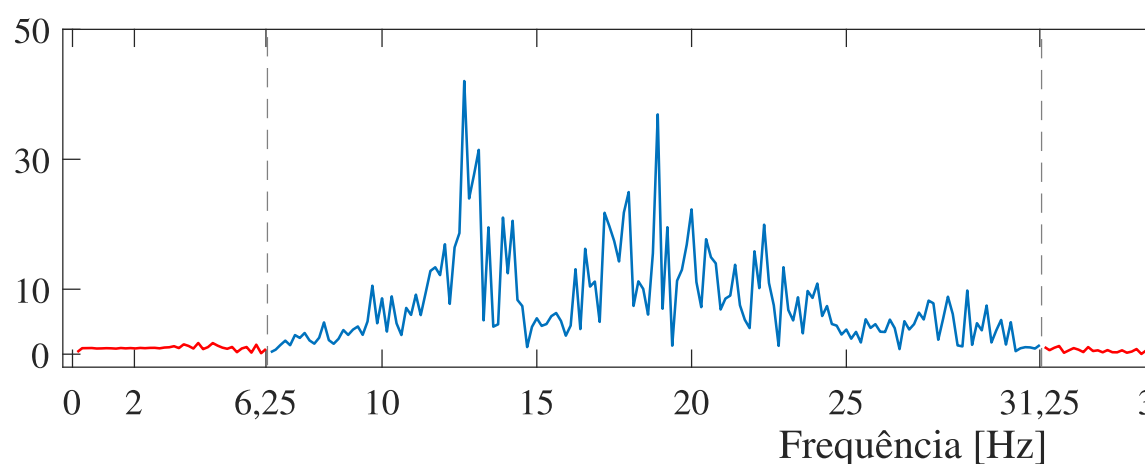
Fonte: Do Autor, 2024

A matriz de confusão para a estratégia do Modelo média da acurácia global oscilou entre 28,35% e 91,86%. Na abordagem, quando empregada nesta arquitetura especifico salvo o primeiro estágio, não exibe uma eficácia satisfatória

6.2 Domínio da Frequência

A Figura 84 exibe um sinal característico transformado no domínio da frequência. As frequências além de 31,25 Hz e inferiores a 6,25 Hz são excluídas devido a estruturas, levando à exclusão das respostas dinâmicas a essas frequências. É importante notar que esses sinais são normalizados no domínio do tempo pelo módulo de sua amplitude máxima. Em seguida, após a normalização, os dados são particionados entre os conjuntos de treinamento e teste, seguindo a metodologia de treinamento da rede neural.

Figura 84 – Gráfico no domínio da frequência de uma resposta



Fonte: Do Autor, 2024

6.2.1 Divisão dos dados

Devido à subdivisão do sinal, o processo de treinamento é realizado no domínio da frequência. No âmbito da frequência, a configuração para a divisão dos dados é a seguinte:

- **Modelo C:**

- Treinamento: Este modelo foi treinado com Estágio 1, 688 ensaios do Estágio 2 e 688 ensaios do Estágio 3.
- Validação: A avaliação foi feita a partir dos resultados do Estágio 1, 230 ensaios do Estágio 2 e 230 ensaios do Estágio 3.
- Teste: O teste do Modelo C usou os 918 ensaios do Estágio 4.

- **Modelo D:**

- Treinamento: Este modelo foi treinado com Estágio 1, 688 ensaios do Estágio 02, 688 ensaios do Estágio 3 e 688 ensaios do Estágio 4.
- Validação: A avaliação foi feita a partir dos resultados do Estágio 1, 230 ensaios do Estágio 2, 230 ensaios do Estágio 3 e 230 ensaios do Estágio 5.
- Teste: O teste do Modelo D é conforme cada estágio.

Figura 85 – Modelos de treinamento - Domínio

	SEM DANO (Estágio 01)		DANO 01 (Estágio 02)		DANO 02 (Estágio 03)
	TREINO	VALID.	TREINO	VALID.	TREINO
Modelo de Treinamento A	688	230	TESTE		TESTE
Modelo de Treinamento B	688	230	688	230	TESTE

estágios de danos. Esses modelos específicos serão empregados para a realização de alterações estruturais por meio do erro quadrático médio. Em relação ao Modelo D, semelhante ao Modelo E do estudo anterior, foi realizado um treinamento único que envolveu partes de todos os estágios.

6.2.2 Otimização dos parâmetros

A arquitetura da rede neural utilizada para o diagnóstico foi otimizada por meio de duas otimizações de parâmetros. A primeira análise focou na identificação da rede da qual apresentassem melhores resultados para a detecção de danos, isto é, a estratégia dos modelos A, B e C. Devido à complexidade dos modelos explorados, foram realizadas variações nas divisões do sinal de entrada e do espaço latente (2, 5, 10, 20, 25, 50 e 75). Além disso, foram testados filtros convolucionais e a dimensão do *Batch* (16, 72, 128, 256, respectivamente). Também foram alternadas as funções de ativação *Tangente Hiperbólica* e *Leaky Relu*. Por fim, a última variação testada foi o tamanho do filtro, com dimensões de 5, 50, 100, 150 e 200.

Tabela 10 – Análise Pórtico - Parâmetros Fixos

ANÁLISE PÓRTICO	
PARÂMETROS FIXOS	
Domínio de Treinamento	Frequência
Dimensão Sinal de Entrada	[160, 180]
Solucionador	Adabo
Análise da rede	Erro de Rede
Modelos de Treinamento	A, B, C
Nº de Estratégias para Detecção	1
Tamanho do Filtro	5, 50, 100, 150, 200

treinadas. Já a segunda estratégia realiza treinamento de parcelas dos estágios de danos, resultando em um total de

A segunda otimização teve como objetivo definir parâmetros aprimorados nas estratégias que envolvem a distância de Mahalanobis. Foram realizadas variações na dimensão do espaço latente, no número de filtros convolucionais (12, 16, 48) e na dimensão do *Batch Size*. As funções de ativação, que incluíram *Relu*, *Tangente Hiperbólica* e *Softmax*, foram testadas.

Após esse estudo preliminar, a otimização dos parâmetros foi conduzida, conforme detalhado na Tabela 11. Através deste processo foram identificados os parâmetros que resultaram nos melhores resultados no treinamento. Isso envolveu variações na dimensão do espaço latente, no número de filtros convolucionais e na função de ativação. Como resultado, todas as redes foram treinadas, incluindo todos os acelerômetros mencionados anteriormente, como as 3 estratégias que utilizam a distância de Mahalanobis.

– Análise Ponte Z24 - Parâmetros Variáveis

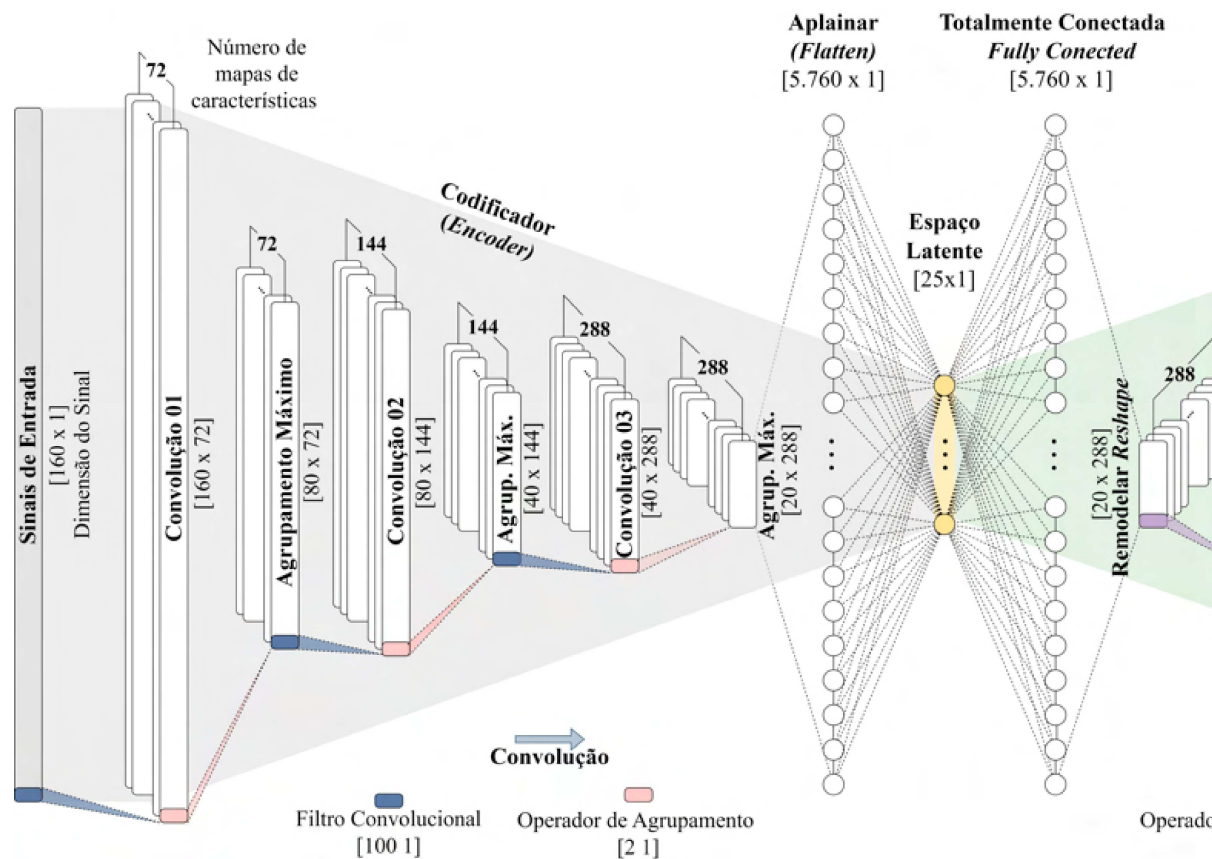
ANÁLISE Z-24												
PARÂMETROS VARIÁVEIS												
Ac_03	Ac_04	Ac_05	Ac_01	Ac_02	Ac_03	Ac_04	Ac_05	Ac_01	Ac_02	Ac_03	Ac_04	Ac_05
4					76					96		
5					15					15		
6					16					16		
REDES												
			2				3					
ReLU			ReLU				ReLU					
15	89,85	66,31	48,00	90,66	47,53	89,80	66,17	46,37	91,36	46,30	88,49	66,18
69	61,62	57,97	46,92	77,64	34,66	62,88	59,09	44,59	77,20	34,03	65,96	62,59
05	79,75	60,20	46,95	93,88	47,87	77,85	60,39	46,54	94,84	47,26	79,14	56,36
62	99,64	95,01	81,21	99,67	82,89	99,64	94,66	84,40	99,66	82,37	99,66	95,98
60	99,47	94,54	82,28	99,65	71,82	99,55	95,09	85,97	99,68	70,18	99,55	94,94
82	99,47	93,09	84,61	99,51	72,07	99,43	95,00	85,92	99,54	72,41	99,49	95,67
09	99,43	98,52	90,95	99,42	84,80	99,39	99,08	91,65	99,51	86,39	99,09	98,79
22					99,36					99,36		
15					99,10					99,28		
03					98,98					99,09		
04					99,01					99,10		
22					99,36					99,36		
75					98,78					98,81		
33					97,37					97,42		
4			5				6					
Hiperbólica			Tangente Hiperbólica				Tangente Hiperbólica					
42	85,59	56,86	39,86	92,51	53,87	84,52	53,73	39,17	92,47	55,92	80,91	55,12

Ac_03	Ac_04	Ac_05	Ac_01	Ac_02	Ac_03	Ac_04	Ac_05	Ac_01	Ac_02	Ac_03	Ac_04	Ac_05	
4					76					96			
5					15					15			
8					48					48			
REDES													
7						8							9
ReLU						ReLU							ReLU
18	96,54	75,48	65,59	97,66	60,64	97,41	76,88	61,23	97,15	60,08	96,78	74,84	
09	77,53	68,81	62,03	85,27	43,61	79,33	70,91	60,13	87,60	41,56	77,87	71,13	
49	79,24	66,61	50,19	95,82	50,20	79,99	67,54	49,28	95,46	50,39	82,82	69,40	
63	96,98	84,47	69,09	99,12	82,21	98,66	82,29	74,34	99,12	81,47	98,41	85,15	
43	98,81	87,34	69,76	99,62	69,96	99,27	87,31	73,17	99,61	66,61	98,46	89,63	
73	98,78	88,11	70,79	98,87	66,88	99,18	87,78	72,04	99,07	62,65	98,54	86,97	
69	98,51	97,95	82,03	99,59	82,45	98,43	98,45	86,30	99,59	80,93	98,42	98,49	
22					99,24					99,25			
35					99,45					99,38			
29					99,31					99,33			
25					99,20					99,11			
22					99,24					99,25			
80					98,81					98,78			
49					97,64					97,48			
0						11							12
Hiperbólica						Tangente Hiperbólica							Tangente Hiperbólica
41	92,09	61,27	53,37	98,10	68,71	91,81	61,53	52,47	97,25	65,99	89,31	61,32	
17	80,74	67,17	50,20	95,03	55,82	78,89	67,89	55,61	92,06	58,00	78,03	66,18	
16	85,95	73,03	45,66	96,45	55,71	82,11	70,63	45,78	96,79	54,25	86,19	69,86	
98	99,68	96,65	81,82	99,56	73,09	99,65	97,79	81,30	99,54	69,26	99,64	97,73	
51	99,57	93,10	81,01	99,64	64,83	99,63	95,85	81,51	99,63	61,60	99,67	94,74	

6.2.3 Arquitetura da Rede

Portanto, a primeira análise apresentou melhores resultados em termos de MSE utilizando uma divisão do sinal em 640, dimensão do *Batch* de 24, a função de ativação ReLU, tamanho de filtros de 72, dimensão do *Batch* de 24, a função de perda de 100. A figura 86 exemplifica a arquitetura da rede, evidenciando as três camadas convolucionais e as camadas totalmente conectadas da arquitetura.

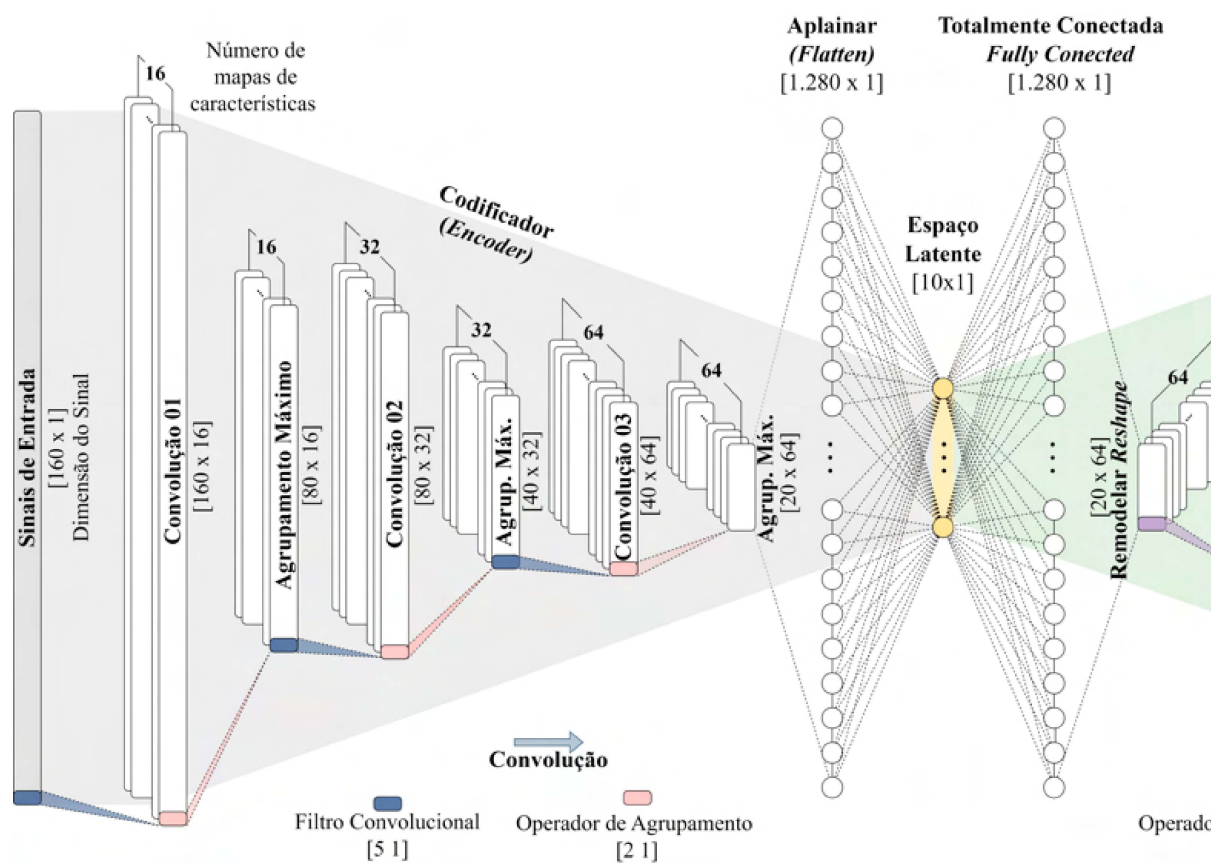
Figura 86 – Arquitetura autocodificada



Fonte: Do autor, 2024

A análise que resultou nos melhores desempenhos

Figura 87 – Arquitetura autocodificada



Fonte: Do autor, 2024

Tabela 12 – Arquitetura da

Camadas	Dimensão de Entrada	Dimensão de Saída
Entrada (Input)	160 x 1 x 1 x 1	-
Convolução - 01 (Convolution)	160 x 1 x 1 x 1	160 x 1 x 72/16 x 1
Agrupamento Máximo (Max Pooling)	160 x 1 x 72/16 x 1	80 x 1 x 128 x 1
Convolução - 02 (Convolution)	80 x 1 x 72/16 x 1	80 x 1 x 144/32 x 1
Agrupamento Máximo (Max Pooling)	80 x 1 x 144/32 x 1	40 x 1 x 144/32 x 1
Convolução - 03 (Convolution)	40 x 1 x 144/32 x 1	640 x 1 x 288/64 x 1

A tabela apresentada (Tabela 12) expõe as dimensões de cada camada do CAE, além do número de camadas convolucionais. Inicialmente, o sinal de entrada possui uma dimensão de 25 e passa por 5 etapas de processamento até alcançar uma representação de 10, dependendo da otimização realizada. Em seguida, a rede é transposta e iniciada após a camada de remodelação. Na parte superior da linha são exibidos as dimensões da rede referida e na parte inferior da linha está representado a arquitetura real.

6.2.4 Treinamento da rede

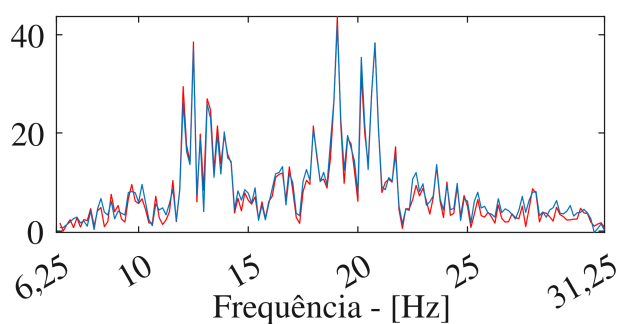
Com base na definição da divisão dos dados para a rede, é possível treinar a rede conforme as estratégias e modelos. A Estratégia 01 utilizou os modelos A, B e C com o objetivo de encontrar o melhor modelo. Já as Estratégias 02, 03 e 04 empregaram o modelo D.

6.2.5 Estratégias para detectar danos

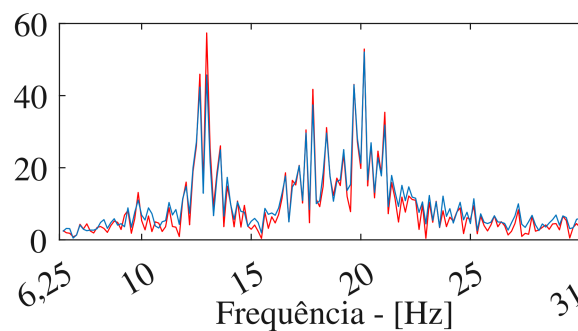
Visando detectar alterações estruturais no conjunto de dados de modo semelhante as quatro estratégias. Contudo, na Estratégia 01 e 02, estão sendo apresentados os resultados provenientes da primeira otimização e os resultados derivados da segunda otimização estão em Anexo. Já para a Estratégia 03 e 04, utilizou-se parâmetros obtido da segunda otimização, e os resultados referentes à primeira otimização estão em Anexo.

Resultados para a Estratégia por MSE - Estratégia 01

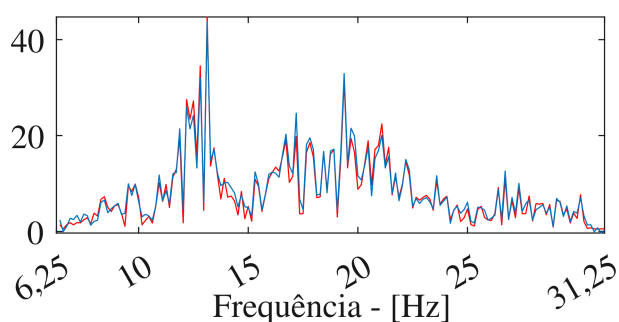
Figura 88 – Predição do Sinal - Primeira



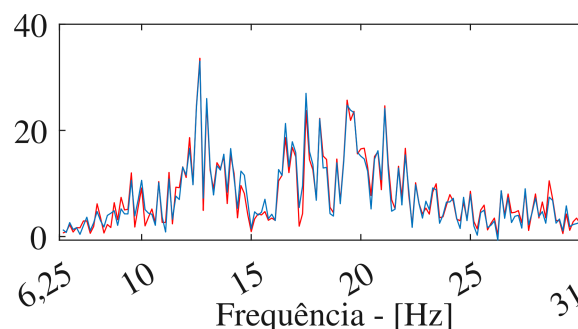
(a) Treino - Estágio 01



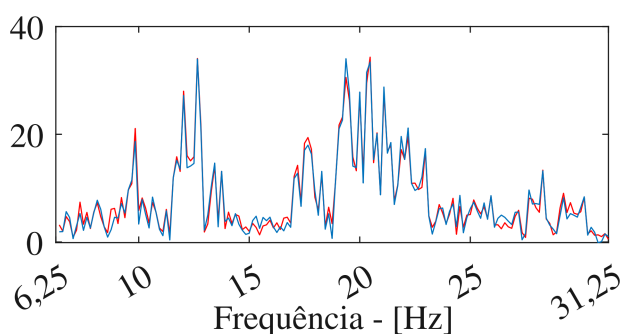
(b) Validação - Estágio 01



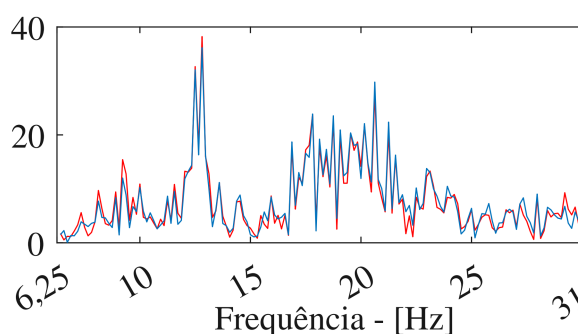
(d) Treino - Estágio 01+02



(e) Validação - Estágio 01+02



(g) Treino - Estágio 01+02+03



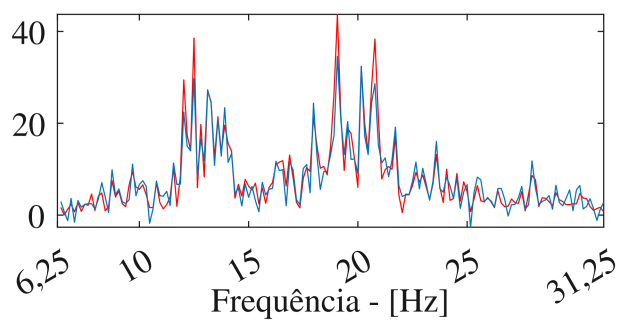
— Dado de entrada — Predito/Reconstru

(h) Validação - Estágio 01+02+03

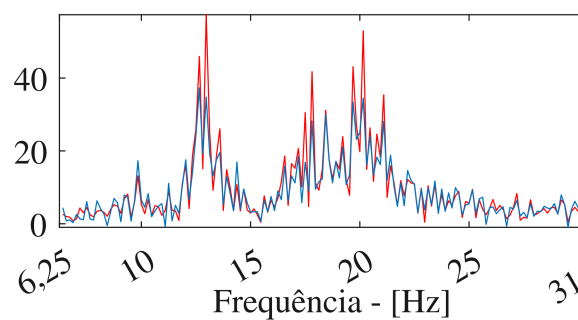
Fonte: Do Autor, 2024

Contudo, a segunda otimização reconstrói os sinais independentemente do estágio de dano. A expectativa é

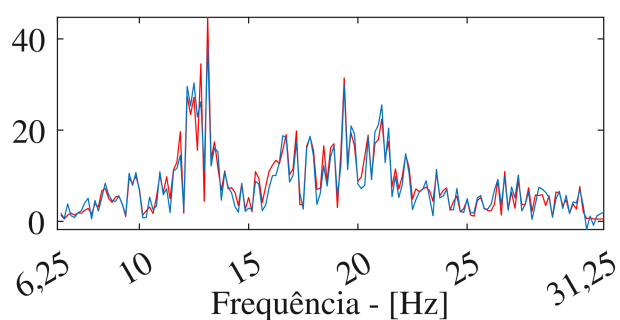
Figura 89 – Predição do Sinal - Segundo



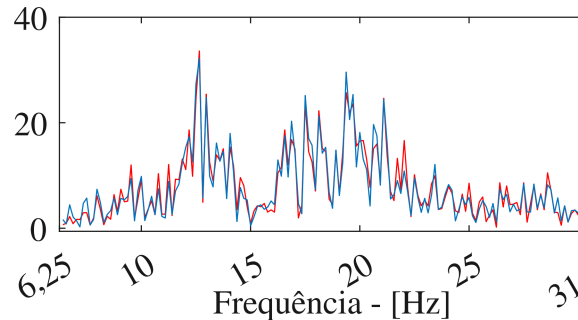
(a) Treino - Estágio 01



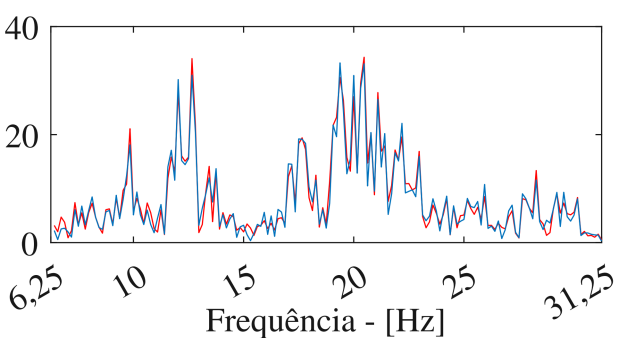
(b) Validação - Estágio 01



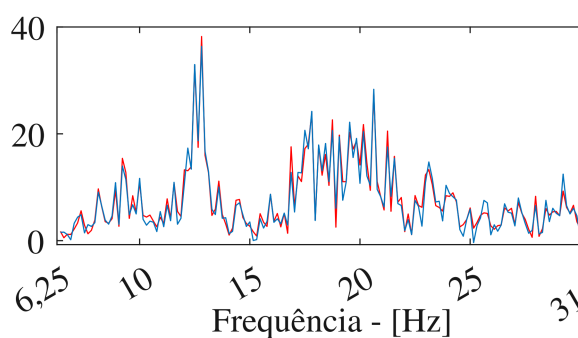
(d) Treino - Estágio 01+02



(e) Validação - Estágio 01+02



(g) Treino - Estágio 01+02+03



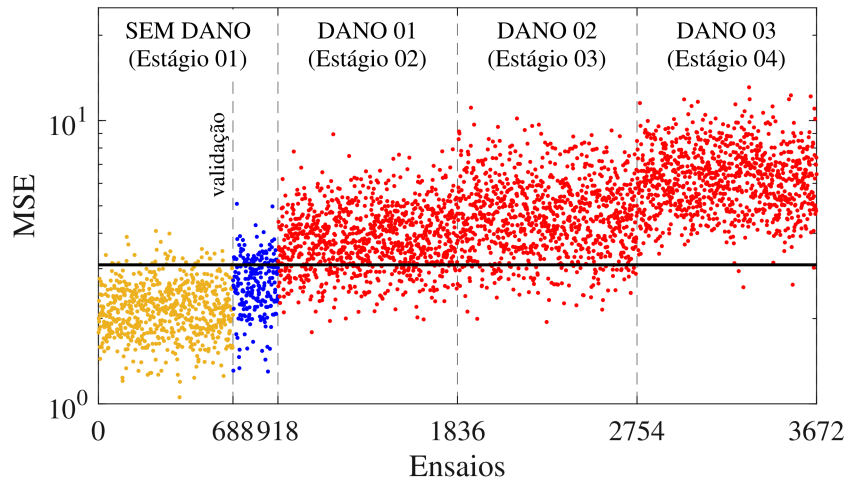
— Dado de entrada — Predito/Reconstru

(h) Validação - Estágio 01+02+03

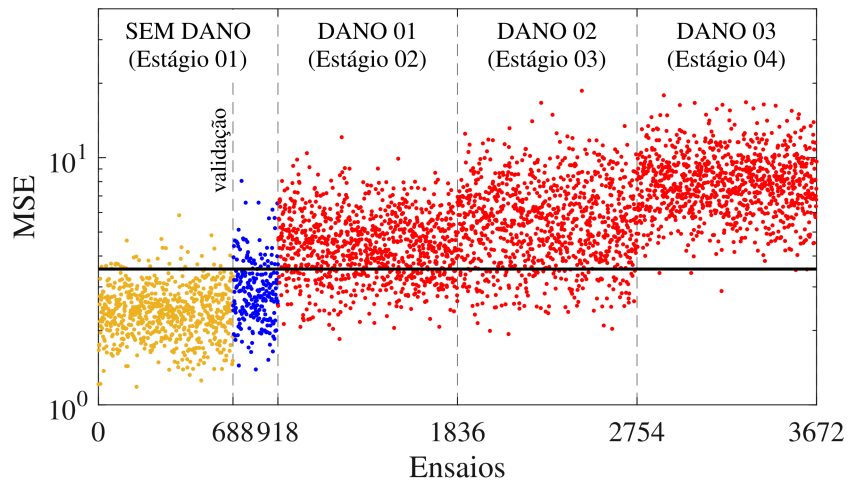
Fonte: Do Autor, 2024

- MSE:

Figura 90 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

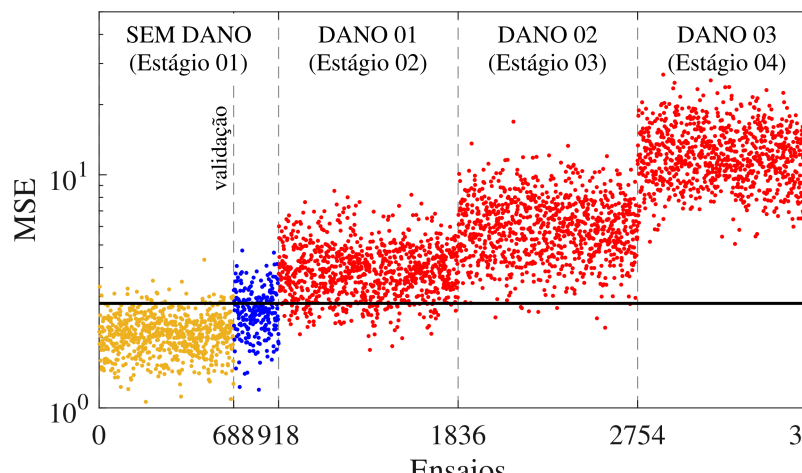
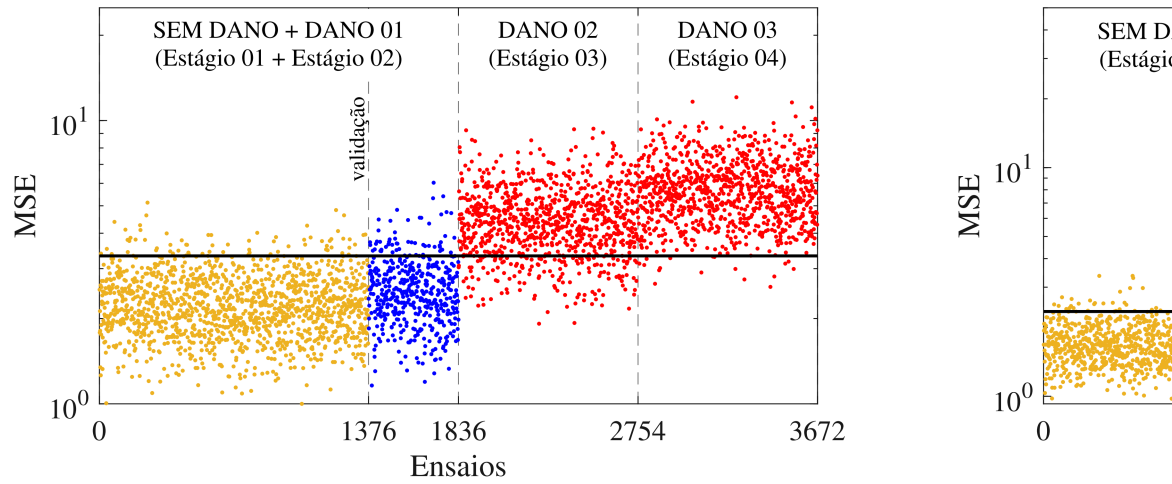
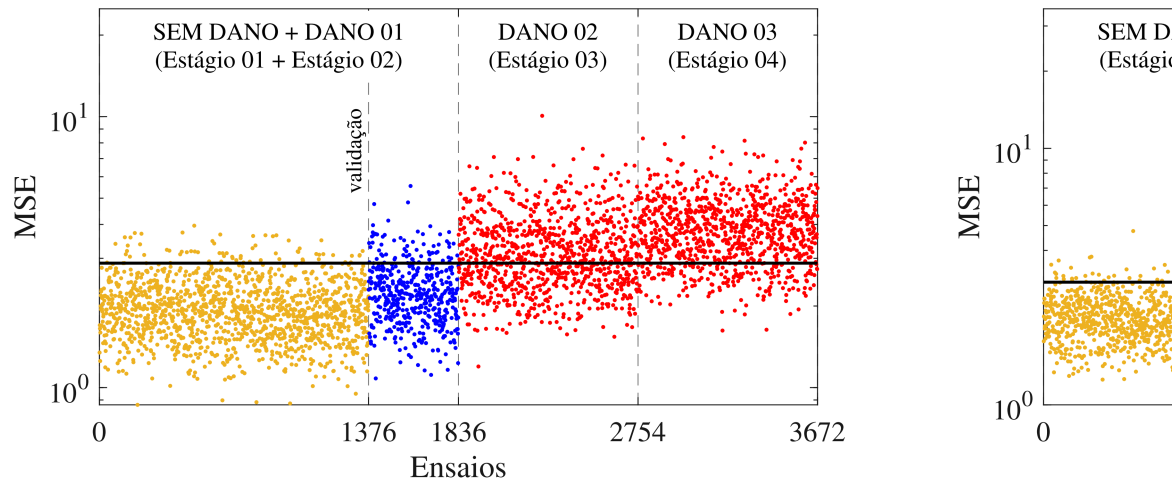


Figura 91 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

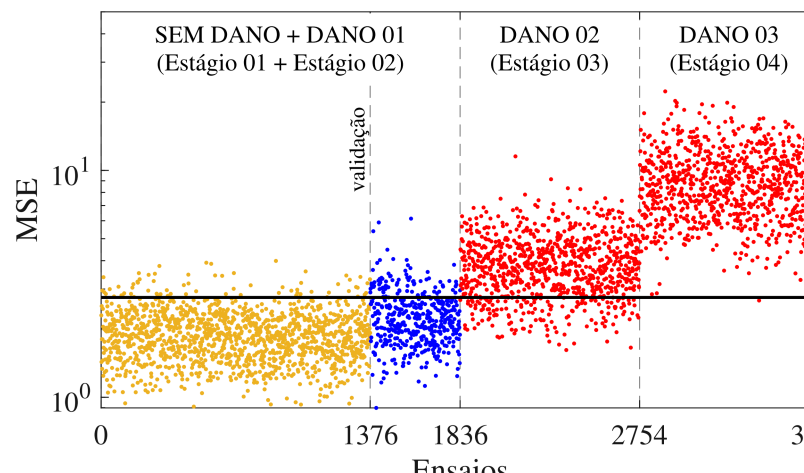
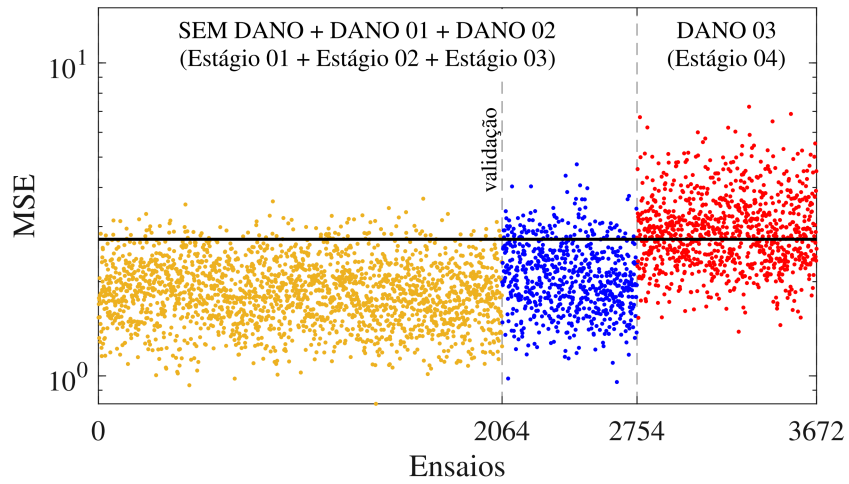
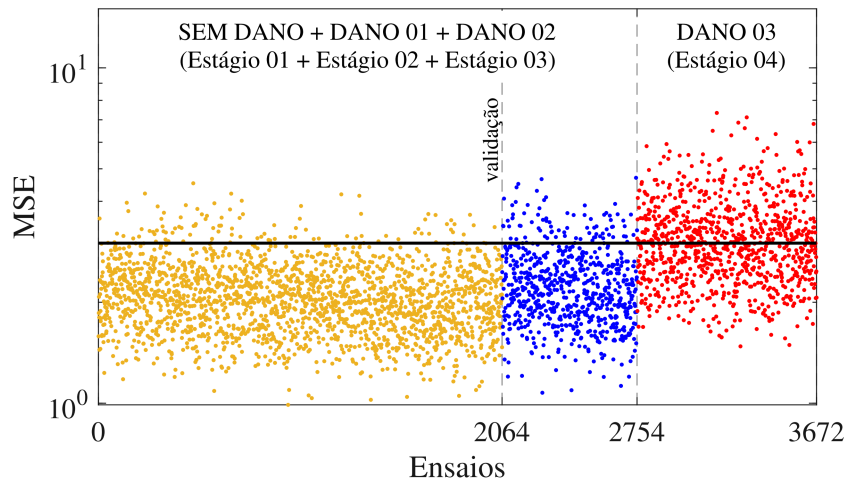


Figura 92 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

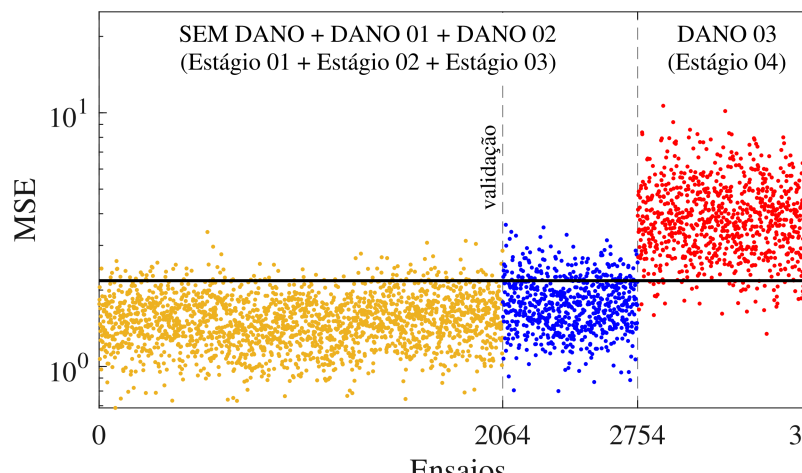


Figura 93 – Matriz de Confusão -

Dano Real	0	1680 5,63%	620 2,08%	73,04% 26,96%
	1+2+3	2910 9,75%	24630 82,54%	89,43% 10,57%
		36,60% 63,40%	97,54% 2,46%	88,17% 11,83%
		0	1+2+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	1640 5,50%	660 2,21%	71,30% 28,70%
	1+2+3	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	97,66% 2,34%	97,79% 2,21%
		0	1+2+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0	1560 5,23%	740 2,48%	67,83% 32,17%
	1+2+3	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	97,38% 2,62%	97,52% 2,48%
		0	1+2+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0	1500 5,03%
	1+2+3	1090 3,65%
		57,92% 42,08%
		0
		Dano P

(e) Acel

Fonte: Do Autor, 2024

Figura 94 – Matriz de Confusão -

Dano Real	0+1	4040 17,60%	560 2,44%	87,83% 12,17%
	3	1750	16610	90,47%

Dano Real	0+1	3330 14,50%	1270 5,53%	72,39% 27,61%
	3	0	18360	100,00%

Figura 95 – Matriz de Confusão -

Dano Real	0+1+2	5850 36,38%	1050 6,53%	84,78% 15,22%
	3	3910 24,32%	5270 32,77%	57,41% 42,59%
		59,94% 40,06%	83,39% 16,61%	69,15% 30,85%
		0+1+2	3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0+1+2	6680 41,54%	220 1,37%	96,81% 3,19%
	3	0 0,00%	9180 57,09%	100,00% 0,00%
		100,00% 0,00%	97,66% 2,34%	98,63% 1,37%
		0+1+2	3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0+1+2	6340 39,43%	560 3,48%	91,88% 8,12%
	3	30 0,19%	9150 56,90%	99,67% 0,33%
		99,53% 0,47%	94,23% 5,77%	96,33% 3,67%
		0+1+2	3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0+1+2	5530 34,39%	
	3	560 3,48%	
		90,80% 9,20%	
		0+1+2	Dano P

(e) Acel

Fonte: Do Autor, 2024

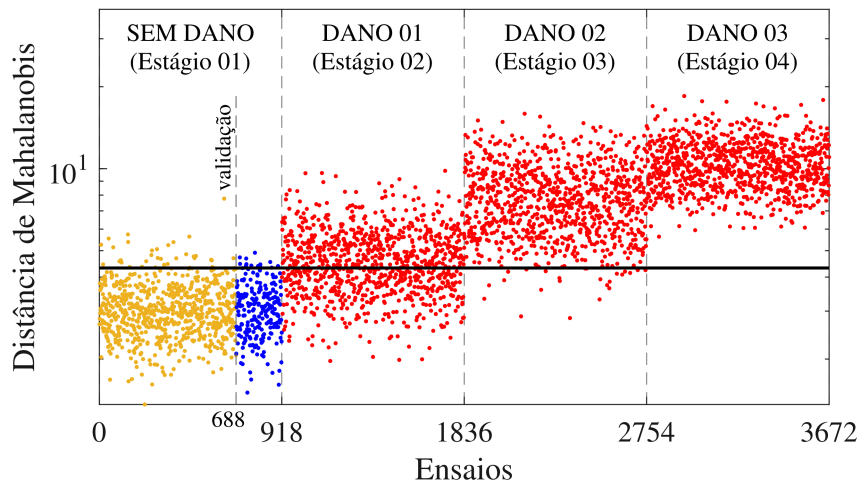
Resultados para Estratégia por Mahalanobis - Es

As estratégias 02, 03 e 04 do modelo de treinamento Mahalanobis, têm como objetivo diferenciar os estágios. Abaixo, são apresentados essas estratégias juntamente com

- Distância de Mahalanobis - Por Acelerômetro (Est

As Figuras 96, 97, 98 e 99 exibem a distância de Mahalanobis para a distribuição de referência para os cinco estágios de danos, SEM DANO (Estágio 01), DANO 01 (Estágio 02), DANO 02 (Estágio 03) e DANO 03 (Estágio 04). Foi possível identificar que esta estratégia alcançou resultados satisfatórios de detectar anomalias estruturais. No estágio 01, em 5 ensaios, observou-se a progressão dos danos, sendo neste primeiro estágio de dano.

Figura 96 – Distância de Mahalanobis



(a) Acelerômetro 1

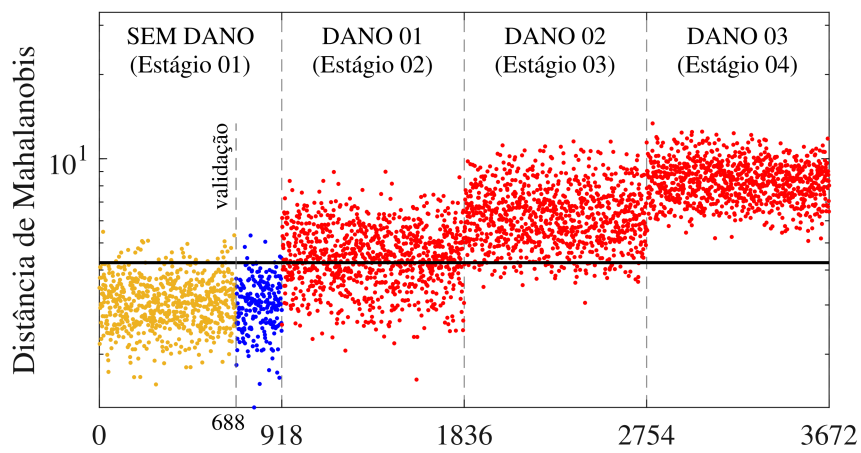
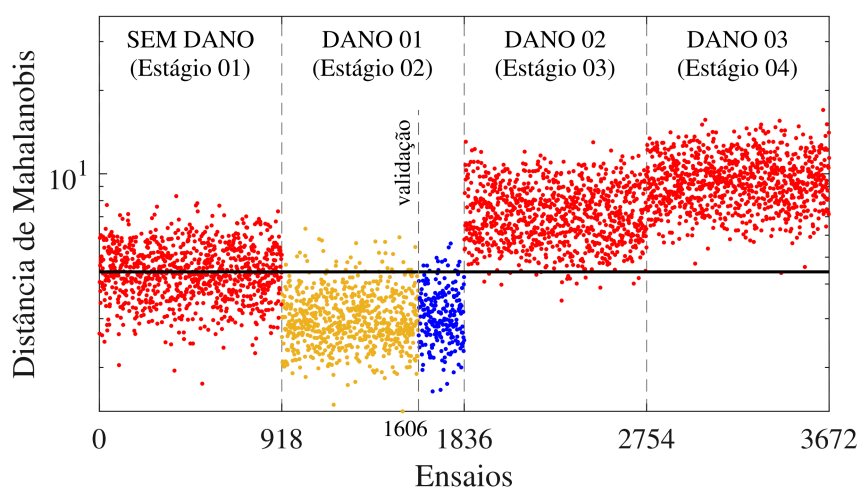
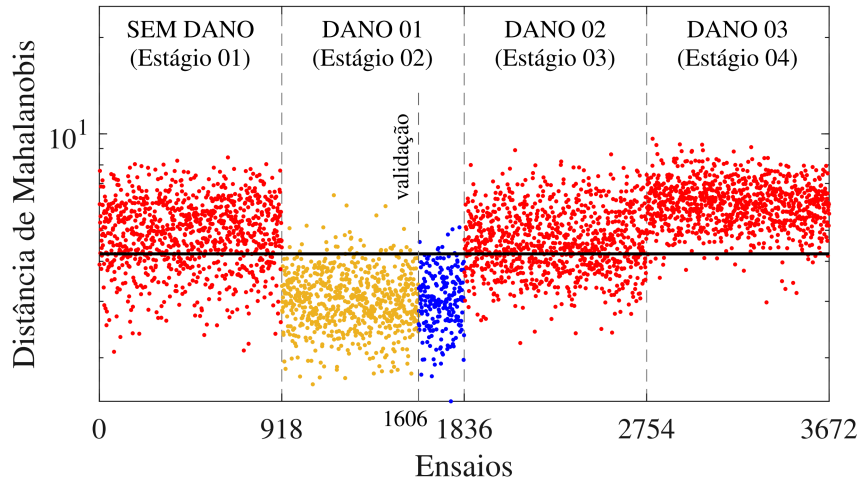


Figura 97 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

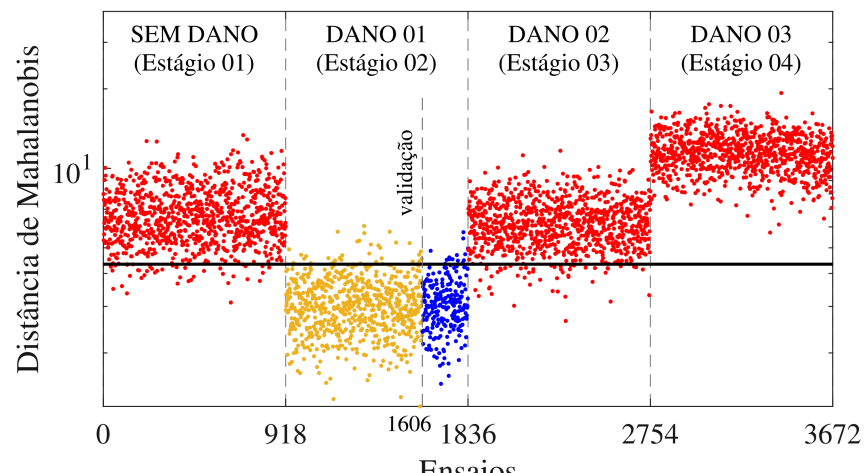
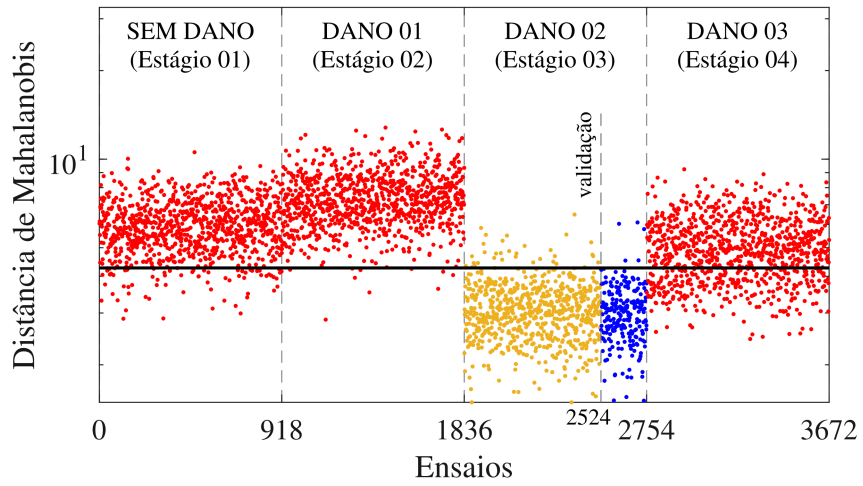
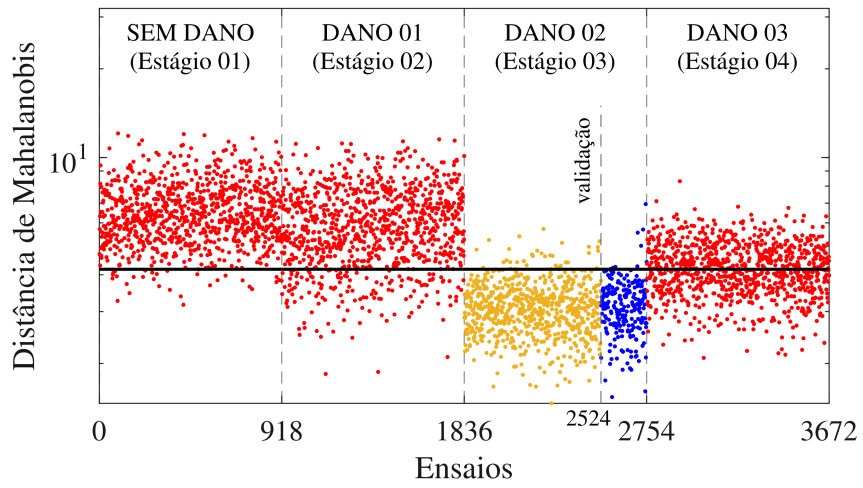


Figura 98 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

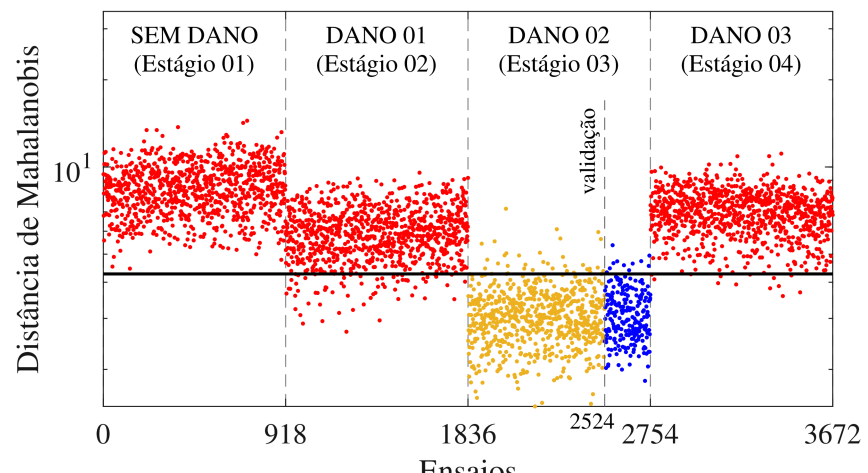
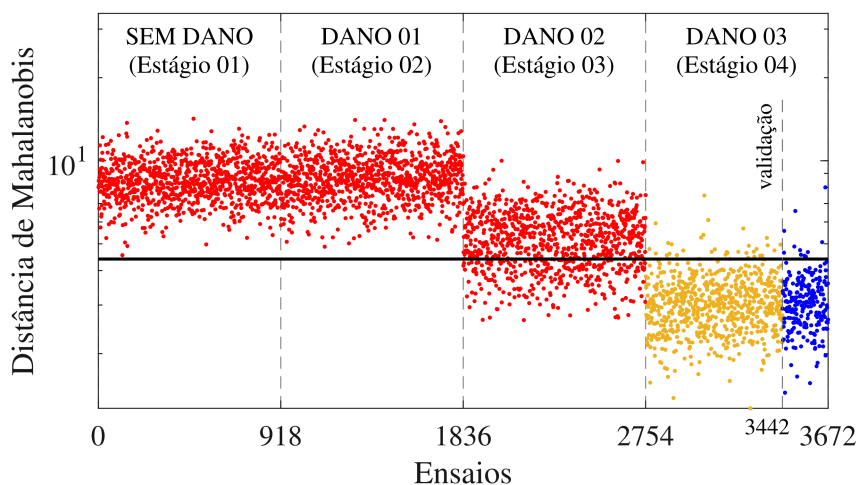
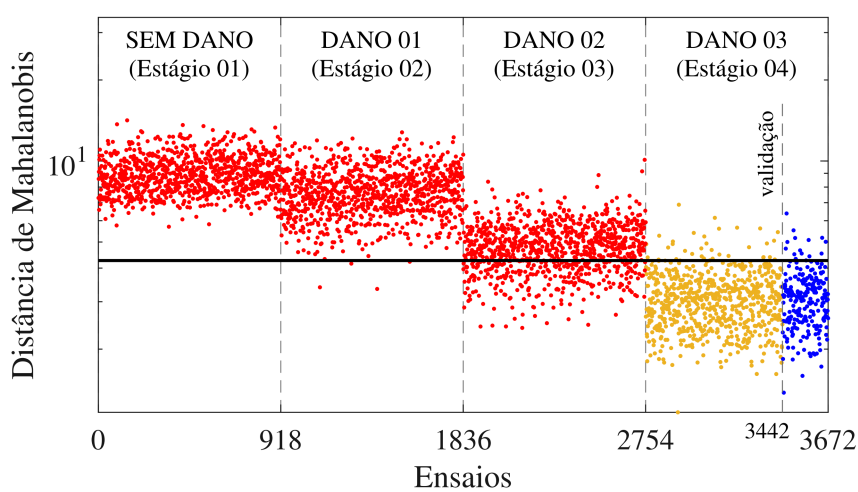


Figura 99 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

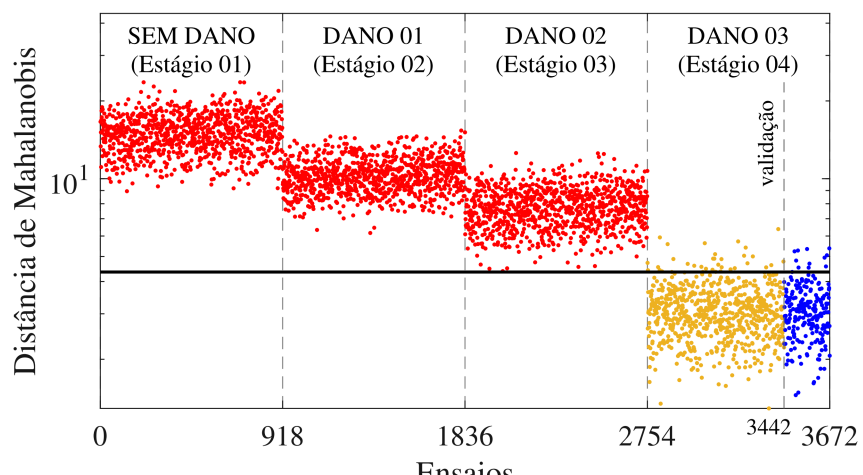


Figura 100 – Matriz de Confusão –

Dano Real	0	2212 7,41%	88 0,29%	96,17% 3,83%
	1+2+3	3842 12,88%	23698 79,42%	86,05% 13,95%
		36,54% 63,46%	99,63% 0,37%	86,83% 13,17%
		0	1+2+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	2172 7,28%	128 0,43%	94,43% 5,57%
	1+2+3	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	99,54% 0,46%	99,57% 0,43%
		0	1+2+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0	2200 7,37%	100 0,34%	95,65% 4,35%
	1+2+3	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	99,64% 0,36%	99,66% 0,34%
		0	1+2+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0	2168 7,27%
	1+2+3	651 2,18%
		76,91% 23,09%
		0
		Dano P

(e) Acele

Fonte: Do Autor, 2024

Figura 101 – Matriz de Confusão –

Dano Real	1	2149 7,20%	151 0,51%	93,43% 6,57%
	2+3	4080	23460	85,19%

Dano Real	1	2212 7,41%	88 0,29%	96,17% 3,83%
	2+3	0	27540	100,00%

Figura 102 – Matriz de Confusão -

Dano Real	2	2217 7,43%	83 0,28%	96,39% 3,61%
	0+1+3	3748 12,56%	23792 79,73%	86,39% 13,61%
		37,17% 62,83%	99,65% 0,35%	87,16% 12,84%
		2	0+1+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	2	2184 7,32%	116 0,39%	94,96% 5,04%
	0+1+3	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	99,58% 0,42%	99,61% 0,39%
		2	0+1+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	2	2114 7,08%	186 0,62%	91,91% 8,09%
	0+1+3	3 0,01%	27537 92,28%	99,99% 0,01%
		99,86% 0,14%	99,33% 0,67%	99,37% 0,63%
		2	0+1+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	2	2177 7,30%	
	0+1+3	623 2,09%	
		77,75% 22,25%	
		2	
		Dano P	

(e) Acele

Fonte: Do Autor, 2024

Figura 103 – Matriz de Confusão -

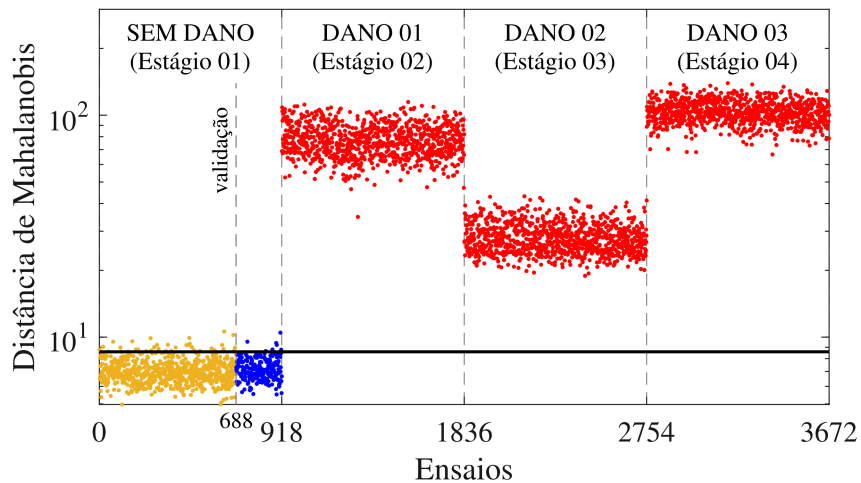
Dano Real	3	2156 7,23%	144 0,48%	93,74% 6,26%
	1+2	1772	25768	93,57%

Dano Real	3	2146 7,19%	154 0,52%	93,30% 6,70%
	1+2	0	27540	100,00%

- Distância de Mahalanobis - Acelerômetros Compil

A Figura 104 exibe a distância de Mahalanobis em referência dos cinco estágios de danos, utilizando os dados de treinamento. Ao examinar esta estratégia, torna-se claro que a distância de Mahalanobis é eficaz na detecção dos estágios de danos. Os dados de dano são consideravelmente acima do limiar, e os dados de validação são semelhante ao conjunto de treinamento.

Figura 104 – Distância de Mahalanobis - Acelerômetros Compil



(a) Estágio 1

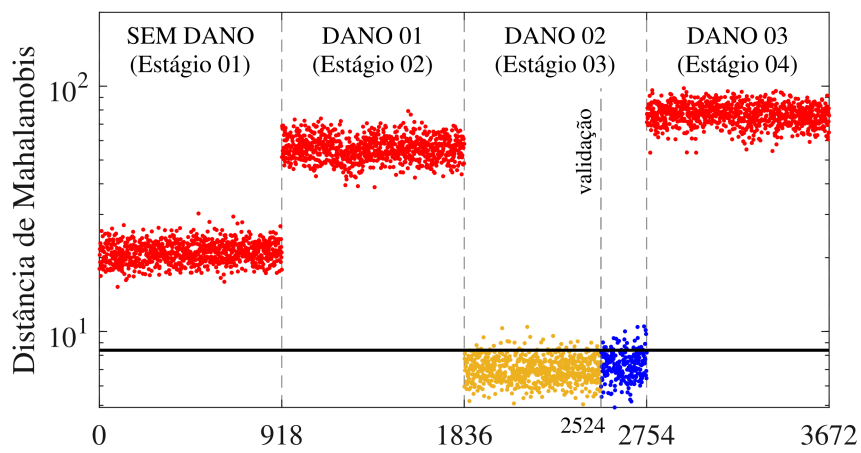


Figura 105 – Matriz de Confusão - Acelerô

Dano Real	0	2138 7,16%	162 0,54%	92,96% 7,04%
	1+2+3	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	99,42% 0,58%	99,46% 0,54%
		0	1+2+3	
		Dano Predito		

(a) Estágio 1

Dano Real	1	2051 6,87%	249 0,83%	89,17% 10,83%
	0+2+3	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	99,10% 0,90%	99,17% 0,83%
		1	0+2+3	
		Dano Predito		

(b) Estágio 2

Dano Real	3	2071 6,94%	229 0,77%	90,04% 9,96%
	0+1+2	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	99,18% 0,82%	99,23% 0,77%
		3	0+1+2	
		Dano Predito		

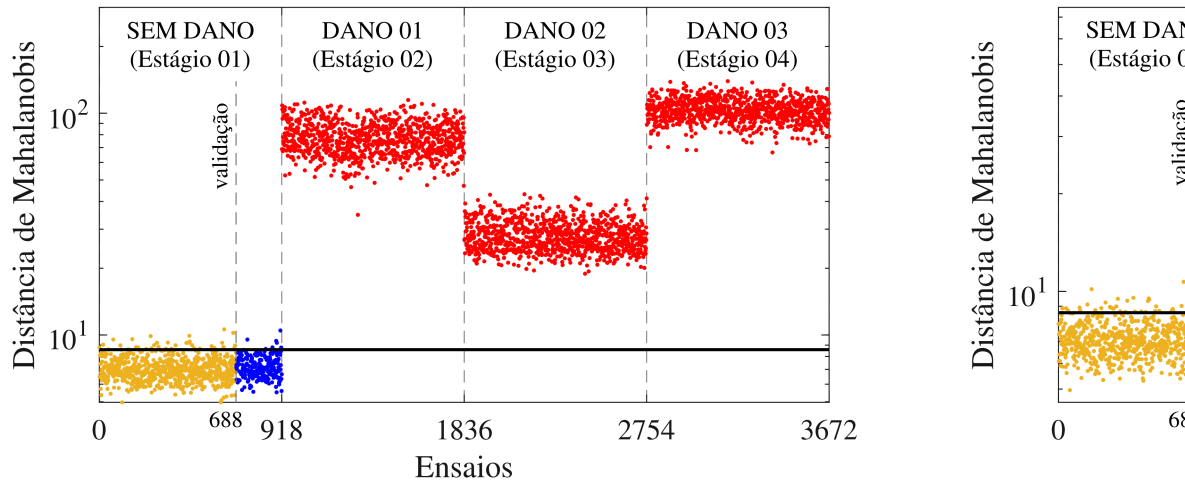
(d) Estágio 4

Fonte: Do Autor, 2024

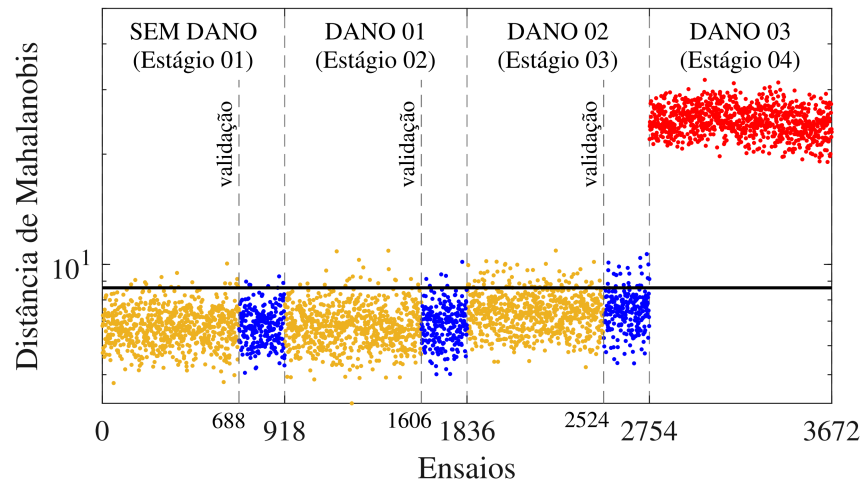
- Distância de Mahalanobis - Acrescentando Danos

Na Figura 106, está representada a distância de Mahalanobis. Ao avaliar esta estratégia, destaca-se a eficácia na detecção de danos, especialmente em situações de alta complexidade.

Figura 106 – Distância de Mahalanobis - A



(a) Estágio 1



(c) Estágio 3

Fonte: Do Autor, 2024

A figura 107 apresentam as matrizes de confusão para o Modelo E. Os valores da acurácia ficaram compreendidos entre 0,9 e 1,0. Na estratégia também não houve a presença de falsos negativos. A estratégia semelhante a estratégia 01 dos modelos A, B e C, contudo

7 CONCLUSÕES

O presente trabalho apresentou um estudo sobre o uso de um algoritmo CAE quando utilizado em uma estratégia SHM, de dano em um cenário controlado proveniente de um pylon UFJF, bem como quatro cenários de danos em uma estrutura devido a fatores externos da Ponte Z24, na Suíça. Para atingir esse objetivo, foram utilizados dados históricos de acelerações estruturais como dados de entrada. Este modelo foi desenvolvido com foco na reconstrução do espaço latente.

A abordagem proposta envolveu quatro estratégias de análise estruturais. A primeira estratégia baseou-se na avaliação dos dados, obtidos pela comparação entre as respostas medidas e as respostas pelo CAE. Essa análise permitiu identificar alterações no comportamento da estrutura. Além disso, foram empregadas outras três estratégias, as quais utilizaram o meio da distância de Mahalanobis nos espaços latentes.

O pórtico demonstrou resultados satisfatórios em comparação com os dados assim como em relação a estratégia empregadas. Em grande parte dos modelos excederam 95%. No entanto, é crucial destacar que surgiram em face dos resultados inferiores ao esperado.

Primeiramente, no processo de treinamento da rede, o critério de parada com base nas análises realizadas, foi não atingir o objetivo de treinamento, dada a quantidade de redes a serem treinadas. Uma estratégia eficaz na otimização do tempo, resultou na interrupção de

estratégias que exploram a distância de Mahalanobis na maioria dos casos, permaneceram superiores ou próximas além de identificar os danos, foi possível até mesmo monitorar ao longo do tempo.

É relevante destacar que houve uma considerável redução do treinamento de cada rede entre as estratégias. Quando o Quadrático Médio (MSE) a duração média foi de 5 minutos, para a demais estratégia esse tempo foi reduzido para 30 segundos. Este processo ressalta a importância de considerar não apenas a eficiência computacional ao selecionar a melhor, mas também a eficiência computacional ao selecionar a melhor para uma determinada aplicação.

Dessa forma, aprimorar a abordagem de treinamento e as técnicas para cada estratégia podem representar passos importantes para tornar as estratégias ainda mais robustas e confiáveis.

Por fim, algumas conclusões importantes sobre a eficácia das estratégias para identificar estágios de danos são fundamentais, destacando a importância dos sinais na duração de treinamento das redes. O domínio dos sinais mais ruidosos comparados ao domínio da frequência, sendo vantajoso utilizar sinais transformados. Além disso, a distância de Mahalanobis nos dados do espaço latente, durante o treinamento, otimiza de modo considerável o tempo de treinamento. A presença de mínimos ou nulos falsos positivos, bem como a eficácia da estratégia proposta por este estudo.

- Aplicação das métricas estudadas neste trabalho especialmente, em estruturas real e com influência

REFERÊNCIAS

- 1 CARDOSO, R.; CURY, A.; BARBOSA, F. Automatized damage detection strategy using raw dynamic measurements. **Engineering Structures**, v. 196, 2019. ISSN 109364.
- 2 FANTTI, B. **Ponte Rio-Niterói é totalmente liberada após batida de navio**. 2022. Disponível em: <<https://www1.folha.uol.com.br/ponte-rio-niteroi-e-totalmente-liberada-apos-batida-de-navio/>>. Acesso em: 29 de Agosto 2023.
- 3 USP. **Centro de Sismologia da Universidade de São Paulo**. Disponível em: <<https://moho.iag.usp.br/eq/latest>>. Acesso em: 25 de Agosto 2023.
- 4 DEFESA Civil do Estado de Santa Catarina. 2023. Disponível em: <<https://www.defesacivil.sc.gov.br/>>. Acesso em: 25 de Agosto 2023.
- 5 (ABNT). **NBR 15421:2023: Projeto de estruturas de concreto armado**. 2023.
- 6 (ABNT). **NBR 6123:1988: Forças devidas ao vento em edificações**. 1988.
- 7 (ABNT). **NBR 8681:2003: Ações e segurança nas estruturas de concreto armado**. [S.l.], 2003.
- 8 CURY, A.; CREMONA, C. Novelty detection based on machine learning applied to structural health monitoring. **Bridge Maintenance, Inspection and Rehabilitation (IABMAS'10)**, p. 172–182, 2010.
- 9 KITROEFF, N.; ABI-HABIB, M.; GLANZ, J.; LOPEZ, J.; ELLIOTT, E.; PEYTON, M.; CEGARRA, A. **Why the Mexico City train crash happened**. Disponível em: <<https://www.nytimes.com/interactive/2023/07/14/world/mexico-city-train-crash.html>>. Acesso em: 14 de julho 2023.

- 15 CACHOT, E.; VAYSSADE, T.; VIRLOGEUX, M.; SERVANT, C. The millau viaduct: Ten years of structural health monitoring. **Engineering International**, Technical Report, v. 25, n. 1, p. 1–10, 2021.
- 16 NUNES, L. A.; FINOTTI, R. P.; BARBOSA, F. S.; CURY, A. A. Deep learning-based strategy for structural damage detection. **Structural Health Monitoring**, v. 20, p. 18, 2021.
- 17 SHANG, Z.; SUN, L.; XIA, Y.; ZHANG, W. Vibration-based damage detection of bridges by deep convolutional denoising autoencoder. **Structural Health Monitoring**, SAGE, v. 20, p. 24, 2020.
- 18 RASTIN, Z.; AMIRI, G. G.; DARVISHAN, E. Unsupervised structural damage detection technique based on a deep convolutional autoencoder. **Salvatore Caddemi**, v. 2021, p. 11, 2021.
- 19 YESSOUFOU, F.; ZHU, J. Deep autoencoder model for structural damage detection subjected to a moving vehicle load under varying temperature. **Structural Health Monitoring**, Elsevier Ltd, v. 52, p. 752–767, 2023.
- 20 FINOTTI, R. P. **Inteligência artificial aplicada à detecção de danos em estruturas: detecção de alterações mecânico-estruturais por meio de redes neurais codificadoras esparsas para a caracterização de danos dinâmicos**. Tese (Doutorado) — Universidade Federal de Pernambuco, 2021.
- 21 FINOTTI, R. P.; GENTILE, C.; BARBOSA, F. S.; CURY, A. A. Deep learning-based damage detection based on sparse autoencoders and control charts. **Engineering and Mechanics**, ENGINEERING AND MECHANICS, v. 81(5), p. 1–10, 2021.
- 22 FINOTTI, R. P.; BARBOSA, F. S.; CURY, A. A.; PEREIRA, R. M. Experimental evaluation of structural changes using sparse autoencoders applied to dynamic responses. **Appl. Sci.**, MDPI, v. 11(24), 2021.
- 23 FINOTTI, R. P.; BARBOSA, F. S.; CURY, A. A.; PEREIRA, R. M. Deep learning-based structural damage detection using sparse autoencoders and control charts. **Structural Health Monitoring**, SAGE, v. 20, p. 24, 2020.

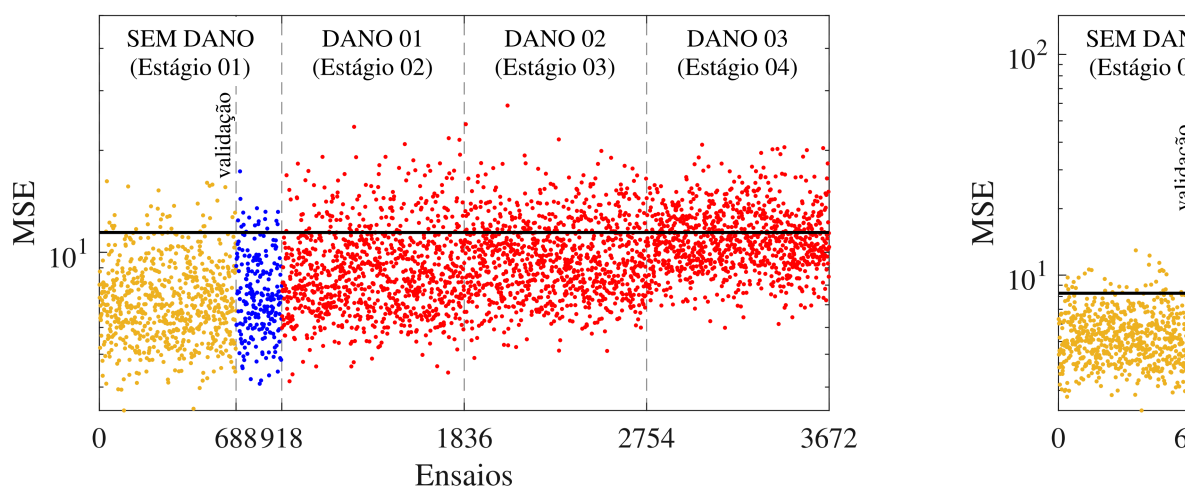
- 28 ZHANG, Y.; XIE, X.; LI, H.; ZHOU, B. An unsupervised identification method based on convolutional variational analysis. **Sensors**, MDPI, v. 22, p. 17, 2022.
- 29 LEE, K.; JEONG, S.; SIM, S.; SHIN, D. A novelty detection method for prestressed concrete bridges based on a convolutional autoencoder. **Sensors**, MDPI, v. 19(7), p. 1633, 2019.
- 30 FINOTTI, R. P.; SILVA, C. F.; OLIVEIRA, P. H. F.; FERREIRA, A. A.; SILVA, R. C. Novelty detection on a laboratory bridge using an unsupervised deep learning algorithm. **Latin American Journal of Structures**, 20(9), e512, 2023.
- 31 ROECK, G.; PEETERS, B.; MAECK, J. **Dynamic Analysis of Engineering Structures**. 2000. Computational Methods for Dynamic Structures IASS-IACM.
- 32 FARRAR, C. R.; WORDEN, K. **Structural health monitoring: a data-driven learning perspective**. [S.l.]: 2013 John Wiley & Sons, 2013.
- 33 PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, J. **Structural Systems: Fundamentals through Simulations**. [S.l.]: 1999.
- 34 SHARMA, A. **Understanding Activation Functions in Deep Learning**. Disponível em: <<https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9a1e1e1e1e1e>>. julho 2023.
- 35 JADON, S. **Introduction to Different Activation Functions in Deep Learning**. 2018. Disponível em: <<https://medium.com/@surya1999/survey-on-activation-functions-for-deep-learning-9689331e1e1e>>. 2023.

8 APÊNDICE A

Nesta seção, são apresentados os resultados obtidos com as variações dos parâmetros da arquitetura relacionados ao custo, especialmente ao domínio da frequência.

Os primeiros resultados exibidos são provenientes da primeira otimização com os parâmetros da segunda otimização. De acordo com a Figura 108, é possível visualizar que para este treinamento a rede conseguiu detectar alterações estruturais, contudo, para esta rede, a distância de Mahalanobis, geraram resultados satisfatórios.

Figura 108 – Erro Quadrático Médio



(a) Acelerômetro 1

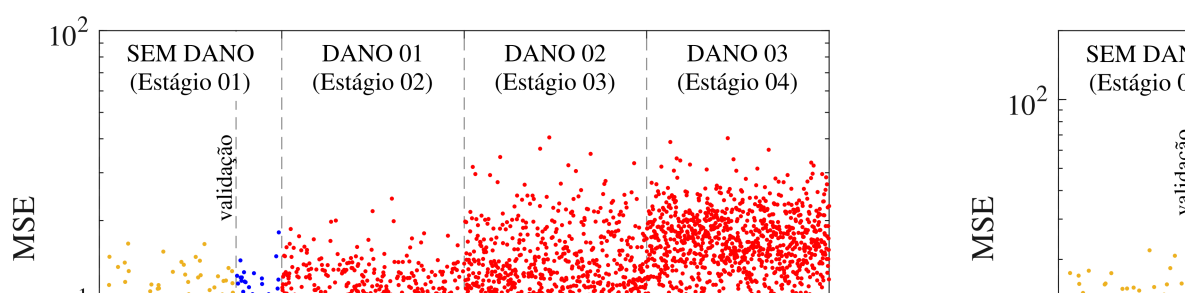
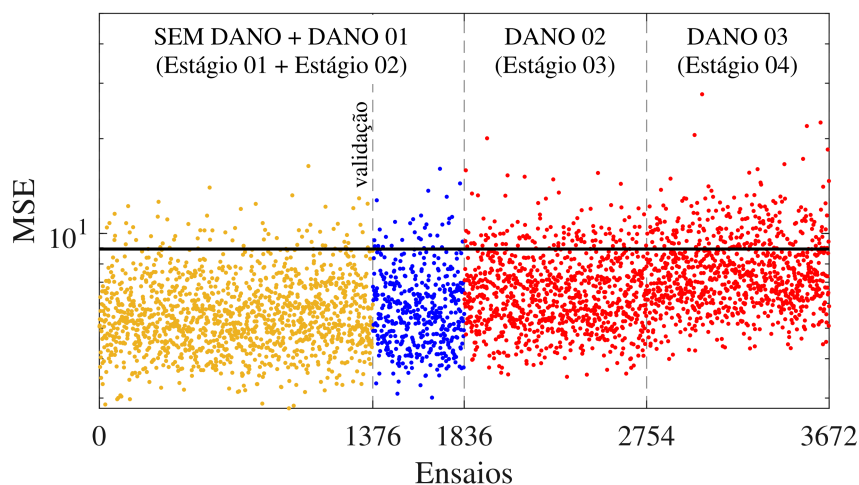
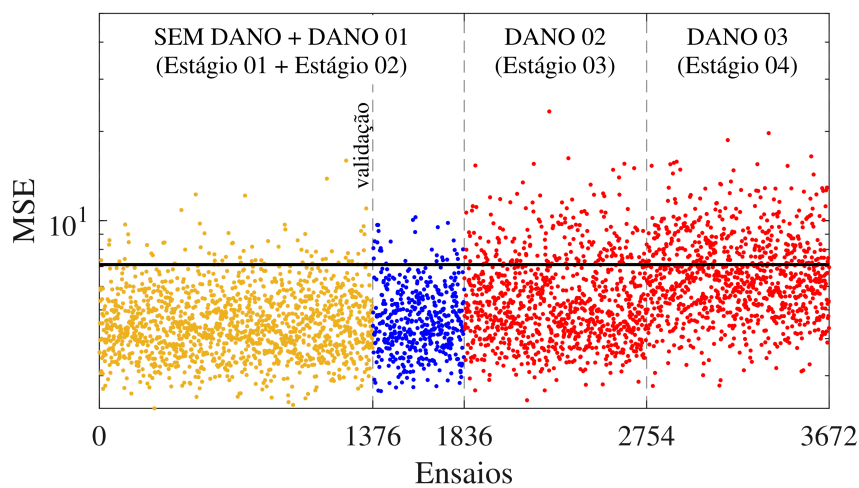


Figura 109 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

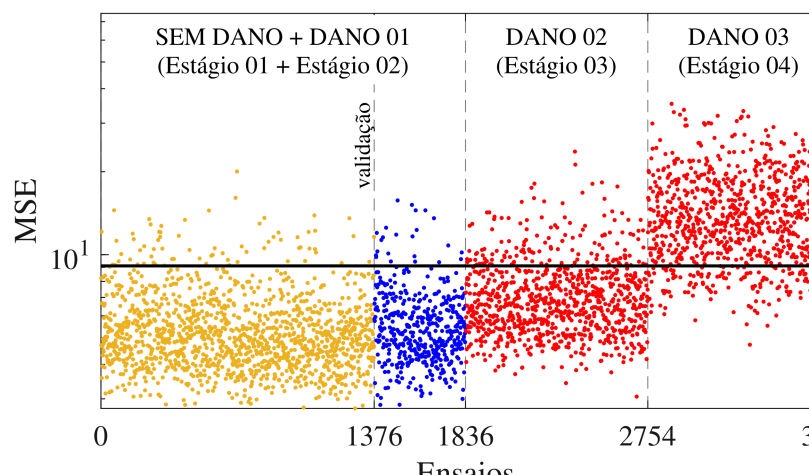
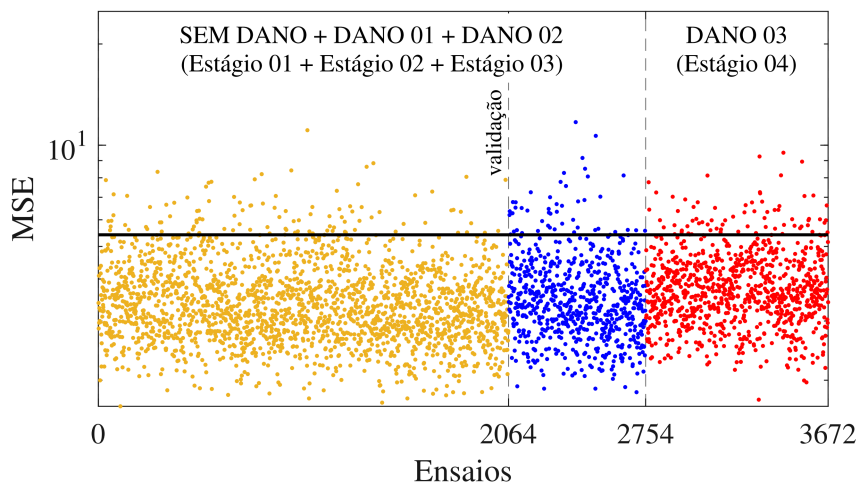
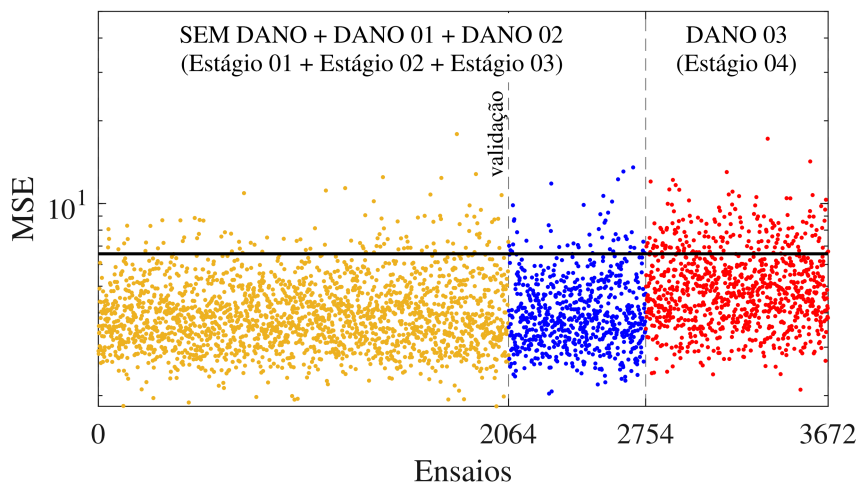


Figura 110 – Erro Quadrático Médio



(a) Acelerômetro 1



(c) Acelerômetro 3

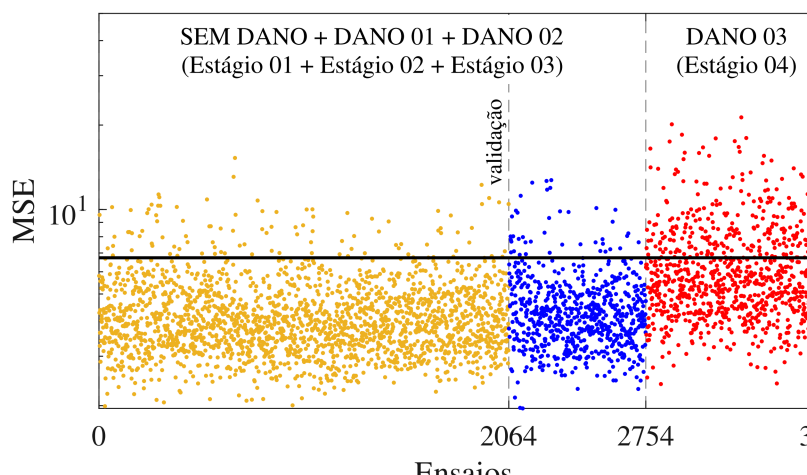


Figura 111 – Matriz de Confusão -

Dano Real	0	2139 7,17%	161 0,54%	93,00% 7,00%
	1+2+3	17992 60,29%	9548 32,00%	34,67% 65,33%
		0	1+2+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	2172 7,28%	128 0,43%	94,43% 5,57%
	1+2+3	2118 7,10%	25422 85,19%	92,31% 7,69%
		0	1+2+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0	2140 7,17%	160 0,54%	93,04% 6,96%
	1+2+3	5536 18,55%	22004 73,74%	79,90% 20,10%
		0	1+2+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0	2150 7,21%	
	1+2+3	13242 44,38%	
		0	
		Dano P	

(e) Acele

Fonte: Do Autor, 2024

Figura 112 – Matriz de Confusão -

Dano Real	0+1	4261 18,56%	339 1,48%	92,63% 7,37%
	3	13736	4624	25,19%

Dano Real	0+1	4327 18,85%	273 1,19%	94,07% 5,93%
	3	3848	14512	79,04%

Figura 113 – Matriz de Confusão –

Dano Real	0+1+2	6406 39,84%	494 3,07%	92,84% 7,16%
	3	8595 53,45%	585 3,64%	6,37% 93,63%
		42,70% 57,30%	54,22% 45,78%	43,48% 56,52%
		0+1+2	3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0+1+2	6499 40,42%	401 2,49%	94,19% 5,81%
	3	336 2,09%	8844 55,00%	96,34% 3,66%
		95,08% 4,92%	95,66% 4,34%	95,42% 4,58%
		0+1+2	3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0+1+2	6433 40,01%	467 2,90%	93,23% 6,77%
	3	2913 18,12%	6267 38,97%	68,27% 31,73%
		68,83% 31,17%	93,07% 6,93%	78,98% 21,02%
		0+1+2	3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0+1+2	6548 40,72%		
	3	5880 36,57%		
		52,69% 47,31%		
		0+1+2		
		Dano Predito		

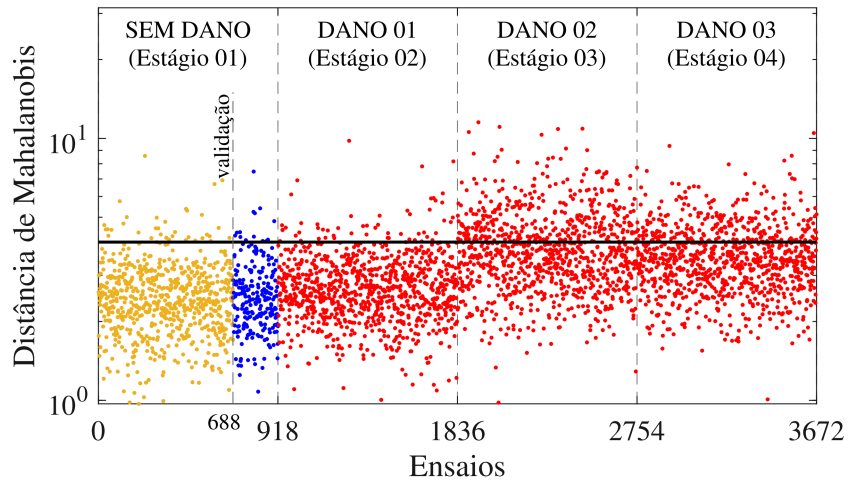
(e) Acelero

Fonte: Do Autor, 2024

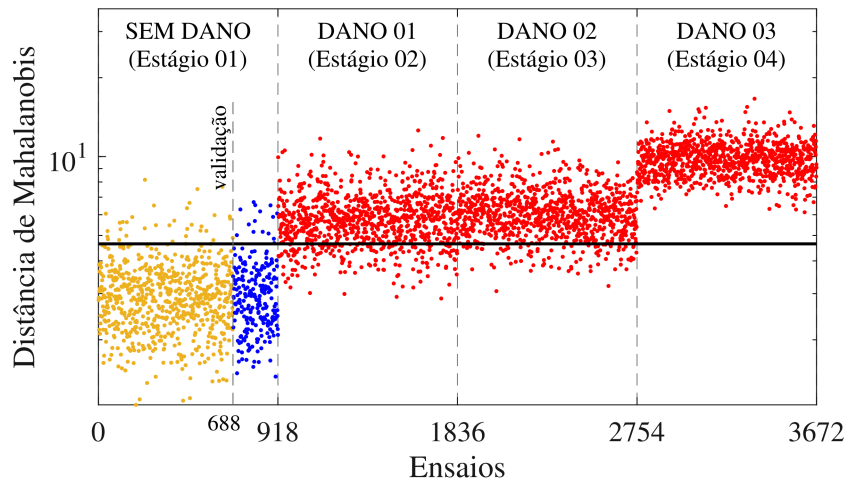
- Espaço Latente:

Nesta etapa, foram apresentados os resultados p
abrangem a distância de Mahalanobis, aplicando a prim

Figura 114 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

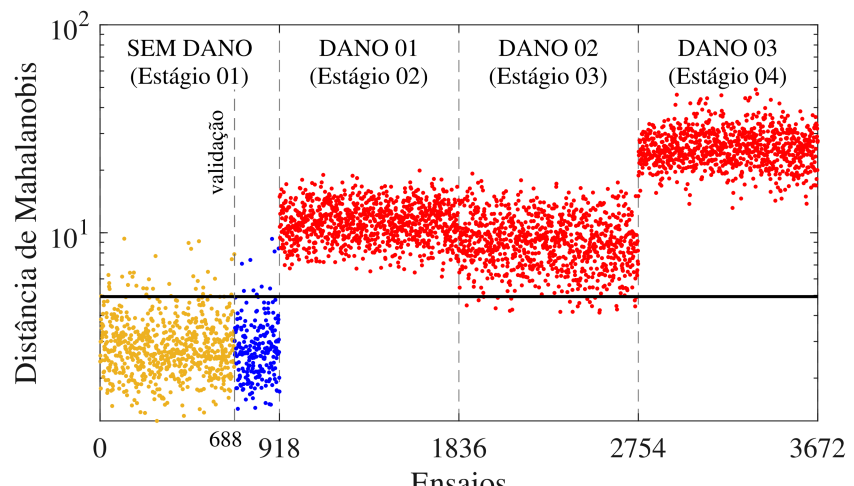
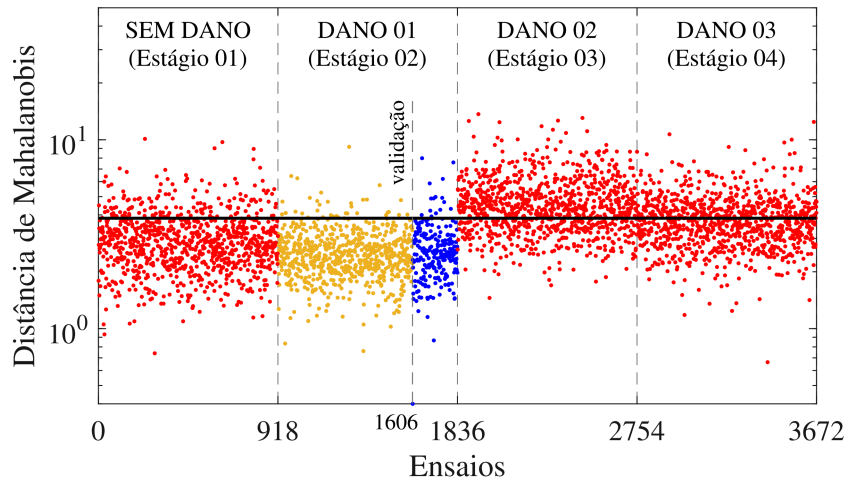
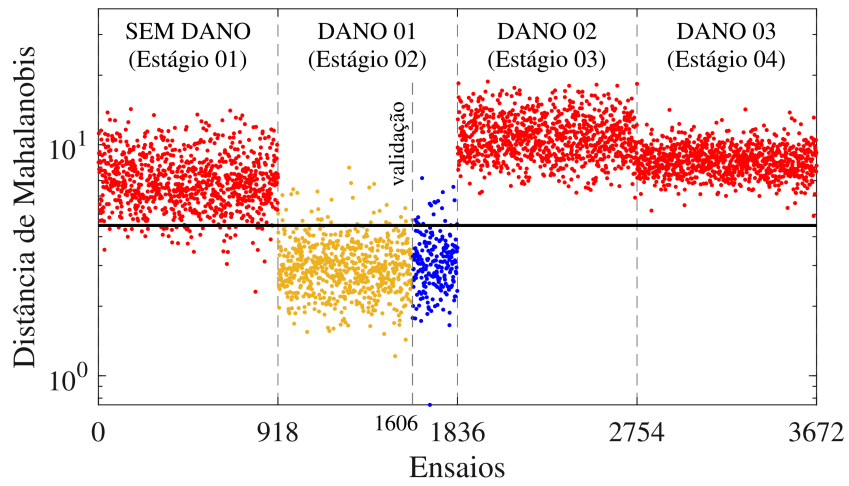


Figura 115 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

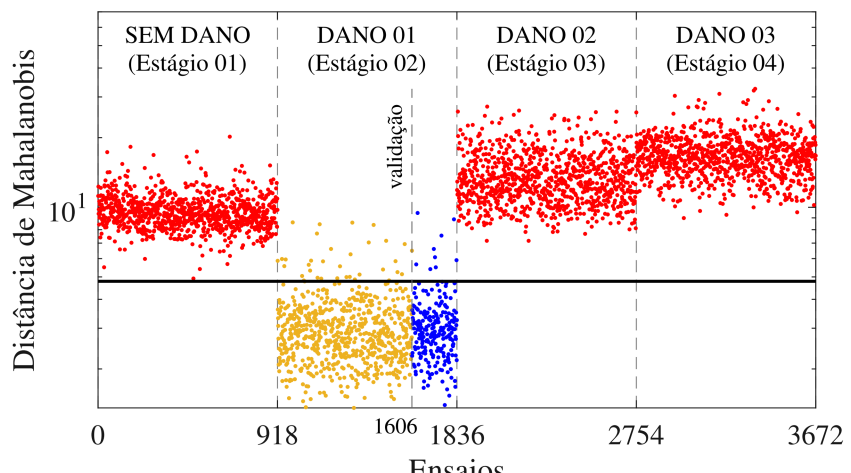
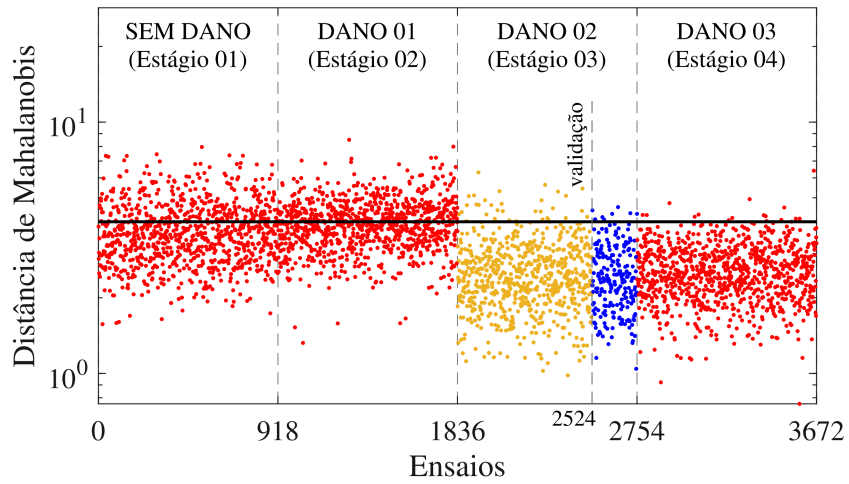
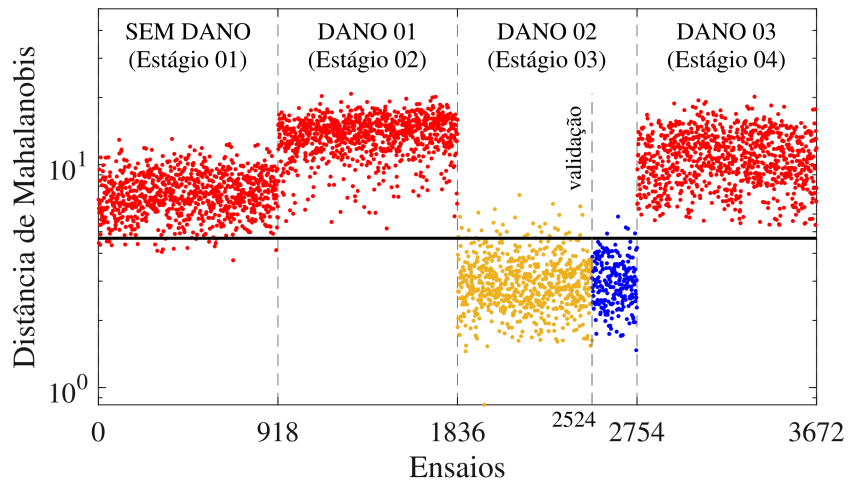


Figura 116 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

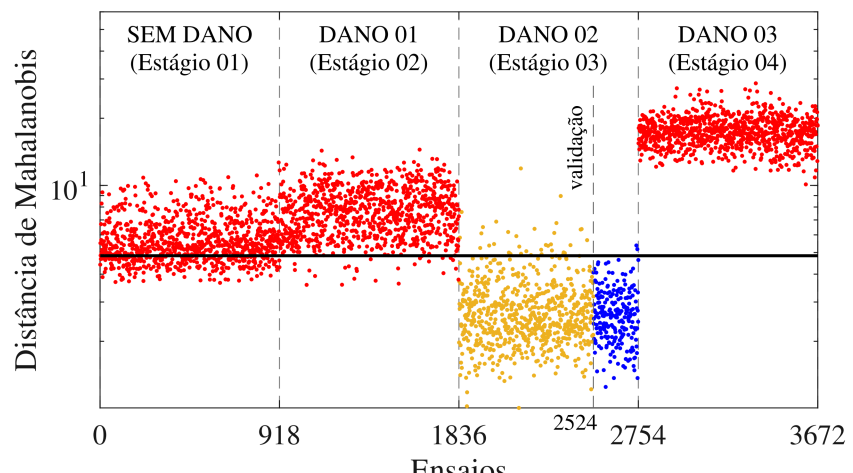
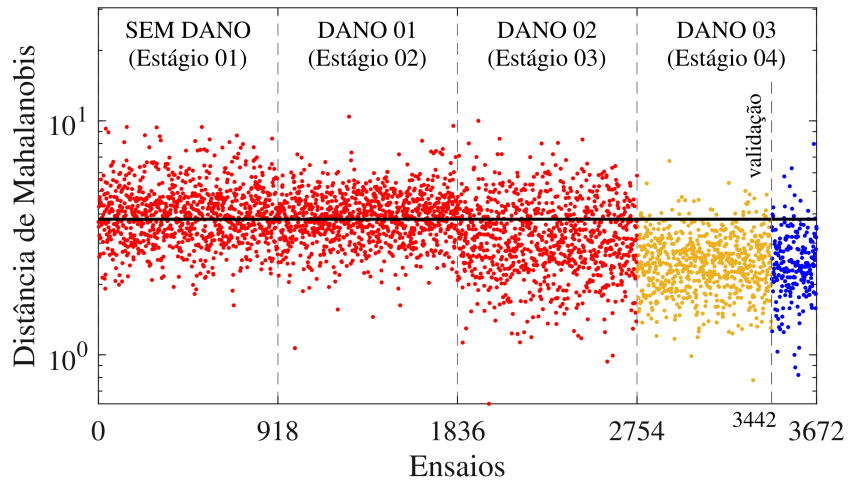
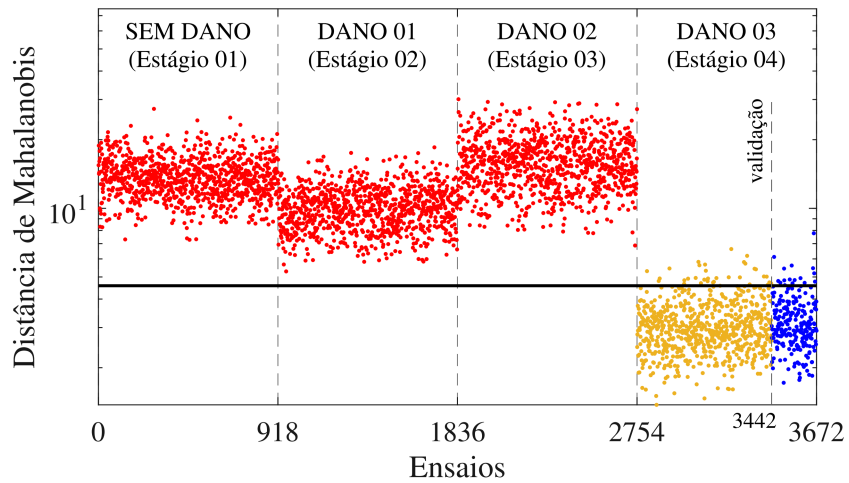


Figura 117 – Distância de Mahalanobis



(a) Acelerômetro 1



(c) Acelerômetro 3

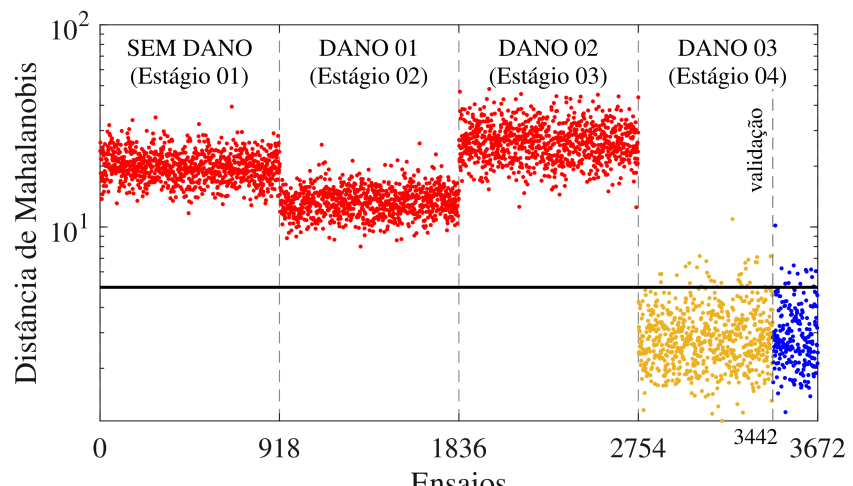


Figura 118 – Matriz de Confusão -

Dano Real	0	2160 7,24%	140 0,47%	93,91% 6,09%
	1+2+3	20930 70,14%	6610 22,15%	24,00% 76,00%
		9,35% 90,65%	97,93% 2,07%	29,39% 70,61%
		0	1+2+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	0	2160 7,24%	140 0,47%	93,91% 6,09%
	1+2+3	18860 63,20%	8680 29,09%	31,52% 68,48%
		10,28% 89,72%	98,41% 1,59%	36,33% 63,67%
		0	1+2+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	0	2190 7,34%	110 0,37%	95,22% 4,78%
	1+2+3	40 0,13%	27500 92,16%	99,85% 0,15%
		98,21% 1,79%	99,60% 0,40%	99,50% 0,50%
		0	1+2+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	0	2200 7,37%
	1+2+3	24 0,08%
		98,92% 1,08%
		0
		Dano P

(e) Acel

Fonte: Do Autor, 2024

Figura 119 – Matriz de Confusão -

Dano Real	1	2150 7,21%	150 0,50%	93,48% 6,52%
	1+3	15850	11690	42,45%

Dano Real	1	2180 7,31%	120 0,40%	94,78% 5,22%
	1+3	17890	9650	35,04%

Figura 120 – Matriz de Confusão -

Dano Real	2	2220 7,44%	80 0,27%	96,52% 3,48%
	0+1+3	20090 67,33%	7450 24,97%	27,05% 72,95%
		9,95% 90,05%	98,94% 1,06%	32,41% 67,59%
		2	0+1+3	
		Dano Predito		

(a) Acelerômetro 1

Dano Real	2	2230 7,47%	70 0,23%	96,96% 3,04%
	0+1+3	23820 79,83%	3720 12,47%	13,51% 86,49%
		8,56% 91,44%	98,15% 1,85%	19,94% 80,06%
		2	0+1+3	
		Dano Predito		

(b) Acelerômetro 2

Dano Real	2	2240 7,51%	60 0,20%	97,39% 2,61%
	0+1+3	110 0,37%	27430 91,92%	99,60% 0,40%
		95,32% 4,68%	99,78% 0,22%	99,43% 0,57%
		2	0+1+3	
		Dano Predito		

(d) Acelerômetro 4

Dano Real	2	2280 7,64%		
	0+1+3	2700 9,05%		
		45,78% 54,22%		
		2		
		Dano P		

(e) Acel

Fonte: Do Autor, 2024

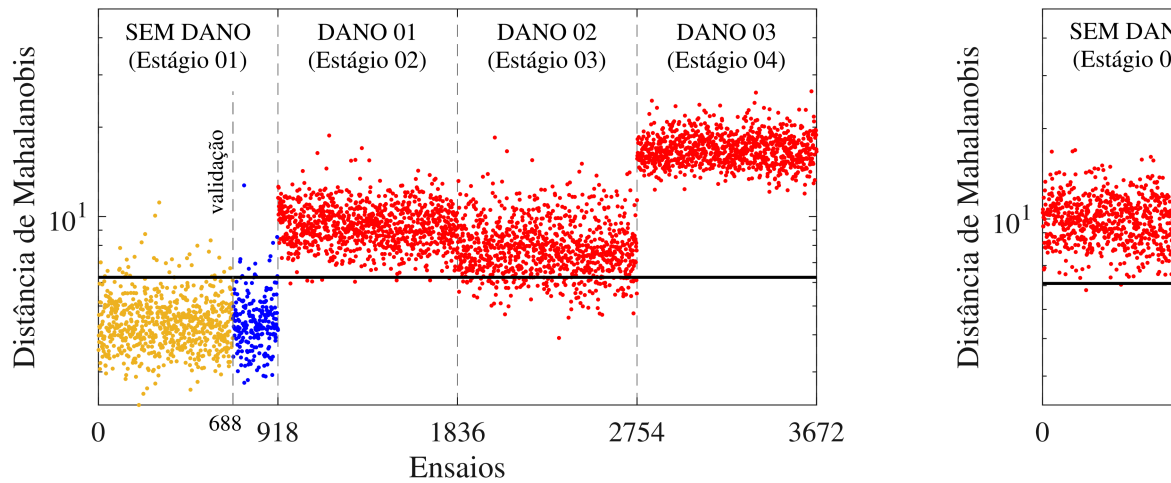
Figura 121 – Matriz de Confusão -

Dano Real	3	2130 7,14%	170 0,57%	92,61% 7,39%
	+2	14900	12640 45,90%	
		3	+2	
		Dano Real		

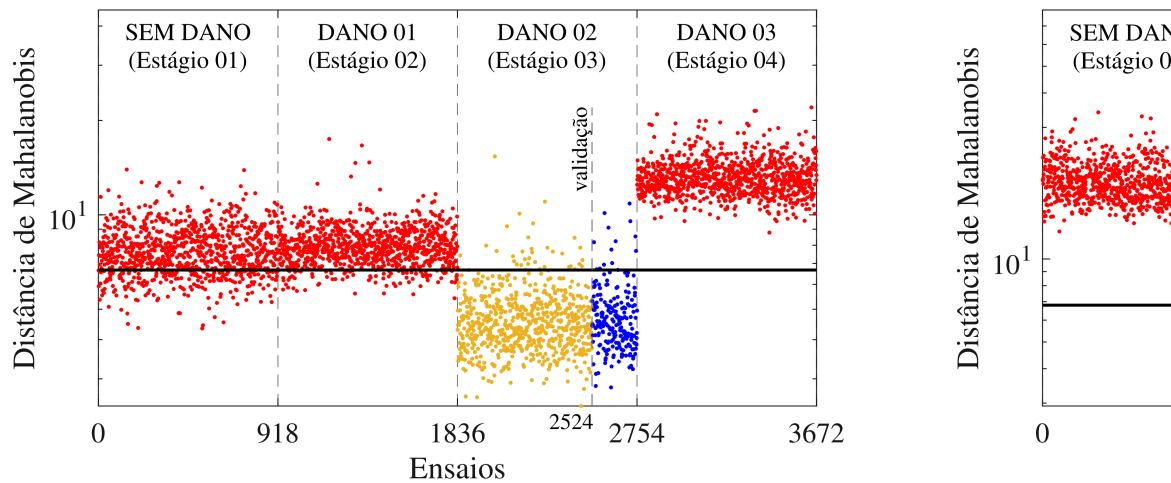
Dano Real	3	2120 7,10%	180 0,60%	92,17% 7,83%
	+2	16490	11050 40,12%	
		3	+2	
		Dano Real		

- Distância de Mahalanobis - Acelerômetros Compil

Figura 122 – Distância de Mahalanobis - Acel



(a) Estágio 1



(c) Estágio 3

● Distribuição de Referência ● Validação ● Teste —

Fonte: Do Autor, 2024

A figura 123 exibe as matrizes de confusão para

Figura 123 – Matriz de Confusão - Acelerô

Dano Real	0	2170 7,27%	130 0,44%	94,35% 5,65%
	1+2+3	1290 4,32%	26250 87,97%	95,32% 4,68%
		62,72% 37,28%	99,51% 0,49%	95,24% 4,76%
		0	1+2+3	
		Dano Predito		

(a) Estágio 1

Dano Real	1	2190 7,34%	110 0,37%	95,22% 4,78%
	0+2+3	110 0,37%	27430 91,92%	99,60% 0,40%
		95,22% 4,78%	99,60% 0,40%	99,26% 0,74%
		1	0+2+3	
		Dano Predito		

(b) Estágio 2

Dano Real	3	2130 7,14%	170 0,57%	92,61% 7,39%
	0+1+2	0 0,00%	27540 92,29%	100,00% 0,00%
		100,00% 0,00%	99,39% 0,61%	99,43% 0,57%
		3	0+1+2	
		Dano Predito		

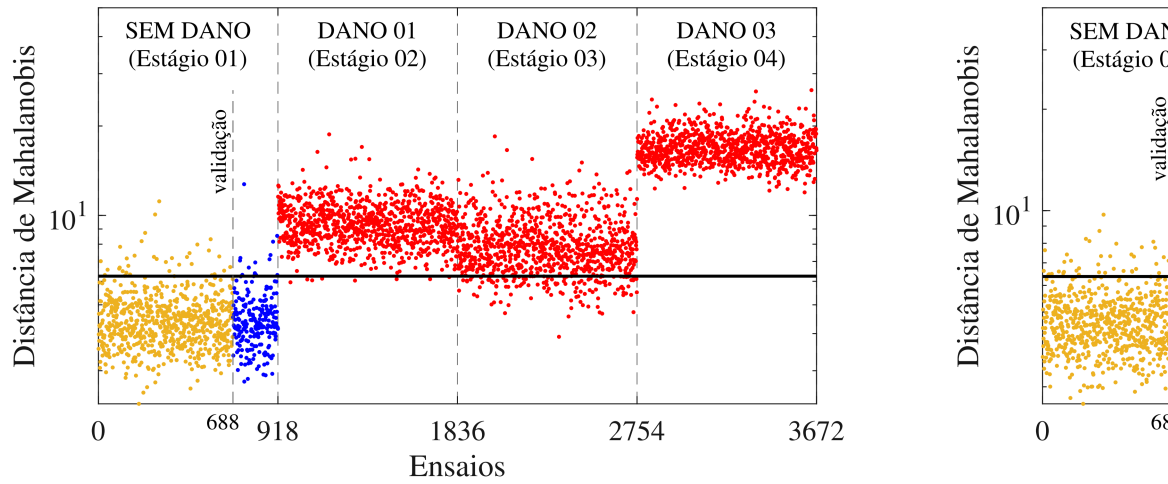
(d) Estágio 4

Fonte: Do Autor, 2024

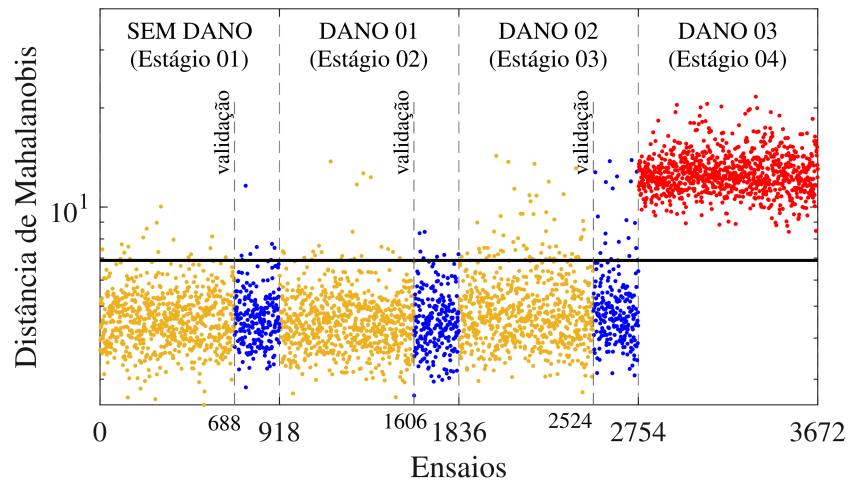
- Distância de Mahalanobis - Acrecendo Danos:

Na Figura 124, está representada a distância de M

Figura 124 – Distância de Mahalanobis -



(a) Estágio 1



(c) Estágio 3

Fonte: Do Autor, 2024

A figura 125 apresentam as matrizes de confusão do Modelo E. Os valores da acurácia ficaram compreendidos

Figura 125 – Matriz de Confusão - Acur

9 APÊNDICE B

O código elaborado no *Software MATLAB* para ser utilizado em ambas aplicações. O algoritmo foi dividido pela análise de treinamento e detecção dos danos, a seguir o primeiro código dos dados primeiramente no domínio do tempo, e em seguida:

- Função de importação dos dados no domínio do tempo

```
function
[inputSize,num_dano,num_col,num_pontos,num_ensaios] =
DT_loadingfilesPORTICO(num_ac,num_pontos,num_dano)
%% Importando e compilando os dados da estrutura
% Número total de ensaios
num_ensaios = num_col*num_dano;
% Carregando as matrizes de dados [4096(eixo x) x 1500]
for i = 0:num_dano-1
    for j = 1:num_col
        load(['H:\\Meu Drive\\Dados\\Pórtico Carga '
            'CARGA ',mat2str(i),'\\carga_',mat2str(j)'.mat'])
% Separando os sinais de cada acelerômetro
%{4(acelerômetros)}[1:num_pontos(eixo x) x 1500]
        for k = 1:num_ac
            V{k}(1:num_pontos,j+i*num_col) = v{k};
        end
    end
end
```

```

end
inputSize = size(V_norm{1}(:, :, 1, 1));
end

```

- Função de importação dos dados no domínio do tempo

```

function
[inputSize, num_dano, num_col, num_total_col, size_
num_ensaios, V_norm] = DT_loadingfilesZ24(num_ac
%% Importando e compilando os dados da estrutura
% Número de pontos ao longo do eixo X no domínio
num_pontos = 65530;
% Número total de ensaios sem divisão
num_total_col = num_dano*num_col;
% Novo eixo x de acordo com as divisões
size_signal = floor(num_pontos/num_div)*num_div;
% Novo número de ensaios resultante de um ensai
num__div = size_signal/num_div;
%Número total de ensaios por estágio
num_test = (num_col*num__div);
% Número total de ensaios com as divisões para
num_ensaios = size_signal/num_div*num_total_col;
for i = 1:num_dano
    for j = 1:num_col
        % Carregando matrizes de dados [(pontos eix

```

```

        for j = 1:num_ensaios
            aux = num_div*j;
            V_{k}(:,j) = V_{k}(aux-(num_div-1):aux);
        end
    end
end
clear V V_ aux
%% Normalizando no domínio do tempo
for j = 1:num_ensaios
    for i = 1:num_ac
        maxAbsVal(i,j) = max(max(abs(V_{i}(:,j))),[]);
        V_norm{i}(:,j) = V_{i}(:,j)/maxAbsVal(i,j);
    end
end
clear V_
%% Reorganização os dados no domínio do tempo -
%{5(accelerômetros)}[1:num_pontos(eixo x),1,1,num_ensaios];
for i = 1:length(V_norm)
    V_norm{i} = reshape(V_norm{i}, [], 1, 1, size(V_norm{i}, 5));
end
inputSize = size(V_norm{1}(:, :, 1, 1));
end

```

- Função de importação dos dados no domínio da fr

function

```

        end
    end
clear v
%% Normalização no domínio do tempo
for j = 1:num_ensaios
    for i = 1:num_ac
        maxAbsVal(i,j) = max(max(abs(V{i}(:,j)))
        V_n{i}(:,j) = V{i}(:,j)/maxAbsVal(i,j);
    end
end
clear V
%% Domínio da frequência
% Período em segundos
T = 4;
% Discretização domínio do tempo (eixo x)
Dc = num_pontos;
%Intervalo de frequência de aquisição
df=1/T;
% Eixo da frequência
f_=-Dc/2*df:df:(Dc/2*df)-0.001;
for i = 1:num_ac
    %Função no domínio da frequência
    xf{i} = fft(V_n{i});
    V_norm{i} = abs(xf{i}(1:ceil(Dc/2),:));
end

```

```

% Número de pontos ao longo do eixo X no domínio
num_pontos = 65530;
% Número total de ensaios sem divisão
num_total_col = num_dano*num_col;
% Novo eixo x de acordo com as divisões
size_signal = floor(num_pontos/num_div)*num_div;
% Novo número de ensaios resultante de um ensai
num__div = size_signal/num_div;
%Número total de ensaios por estágio
num_test = (num_col*num__div);
% Número total de ensaios com as divisões para
num_ensaios = size_signal/num_div*num_total_col;
for i = 1:num_dano
    for j = 1:num_col
        % Carregando matrizes de dados [(pontos eix
            load(['H:\\Meu Drive\\DADOS\\Z-24\\
                xc',mat2str(i),'_',mat2str(j)'],'.mat
        % Separando os sinais de cada acelerômetro
        {5(acelerômetros)}[1:num_pontos(eixo x) x 9
            for k = 1:num_ac
                V{k}(1:num_pontos,j+(i-1)*num_col)
            end
        end
    end
end
clear X

```

```

        maxAbsVal(i,j) = max(max(abs(V_{i}(:,j)
        V_norm{i}(:,j) = V_{i}(:,j)/maxAbsVal(
    end
end
clear V__
%% Domínio da frequência
% Período de aquisição de dados = 0,01s
% Frequência de aquisição = 100Hz
dt = 0.01;
% Discretização domínio do tempo (eixo x)
N = num_div;
% Período em segundo
T = dt*N;
% Eixo do tempo
time = 0:dt:N*dt;
% Intervalo de frequência de aquisição
df = 1/T;
% Eixo da frequência
f_ = -N/2*df:df:N/2*df;
f = f_(ceil(num_div/2):end);
for i = 1:num_ac
    %Função no domínio da frequência
    xf{i} = fft(V_n{i});
    V_norm{i} = abs(xf{i}(1:ceil(num_div/2),:))
end

```

estudo.

```
clc; clear;
%% Importando os dados da estrutura
% Número total de acelerômetros analisados
num_ac = 4 (Pórtico) ou 5 (Z24);
% Número total de estágios de dano analisados
num_dano = 5 (Pórtico) ou 4 (Z24);
% Número de colisões/Ensaio em cada estágio de
num_col = 300 (Pórtico) ou 9 (Z24);
% Número de pontos ao longo do eixo X no domínio
num_pontos = 2000 (Pórtico); ou
% Número de pontos para divisão do eixo x
num_div = 2560 (Z24);

%(PÓRTICO - TEMPO)
[inputSize,num_dano,num_col,num_pontos,num_ensaio] =
DT_loadingFilesPORTICO(num_ac,num_pontos,num_dano,num_col,
%(PÓRTICO - FREQUÊNCIA)
[inputSize,num_dano,num_col,num_pontos,num_ensaio] =
DF_loadingfilesPORTICO(num_ac,num_pontos,num_dano,num_col,

%(PÓRTICO - TEMPO)
[inputSize,num_dano,num_col,num_total_col,size_ensaios,V_norm] = DT_loadingFilesZ24(num_ac,num_dano,num_col,num_total_col,size_ensaios,V_norm)
```

```

num_test = 300;

%% Paramêtros dos dados de entrada (z24)
% Número de dados para treino da estrutura sem
num_train = floor(num_test*0.75);
% Número de dados para validação da estrutura s
num_validation = (num_col*num__div) - num_train;
% Número de dados de Treinamento
num_trainings = [num_train 2*num_train 3*num_tr
%Número de dados de Validação
num_validations =
[num_validation 2*num_validation 3*num_validati

%% Paramêtros das camadas (Define opções de tre
% Número de redes treinadas
num_netw_train = 10;
%Dimensão do Mini-Batch
miniBatchSize = 2;
%Número máximo de Épocas
maxEpochs = 100;
%Dimensão do Espaço Latente
num_latent = 5;
%Número de Filtros Convolucionais
num_filters = 48;

```



```

c = zeros(1,num_dano);
A = [];
B = [];
for j = 0:num_dano-1
    a(j+1) = (j*num_col); %Pontos iniciais de
    b(j+1) = (num_col*j+num_train); %Pontos finais de
    c(j+1) = ((j+1)*num_col); %Pontos finais de
    d = (a(j+1)+1):num_ensaios; %Vetor com os índices
    A = [A a(j+1)+1:b(j+1)]; %Vetor com os índices
    B = [B b(j+1)+1:c(j+1)]; %Vetor com os índices
    for i = 1:num_ac
        V_train{j+1,i} = V_norm{i}(:, :, :, A);
        V_validation{j+1,i} = V_norm{i}(:, :, :, d);
        V_dano{j+1,i} = V_norm{i}(:, :, :, d);
        % A cada iteração adiciona os dados de
        % de modo que o último contenha todos os dados
    end
end
clear A B j i d num_validation

%% Define a arquitetura da rede
% Criação de um gráfico de camadas
lgraph = layerGraph();

% Adiciona as ramificações das camadas

```

```

        "HasUnpoolingOutputs", true, "Padding", "same"
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    convolution2dLayer(filterSize, 4*num_filters
    batchNormalizationLayer("Name", "batchnorm_2")
    leakyReluLayer("Name", "relu_2")
    maxPooling2dLayer(poolingSize, "Name", "maxpo
        "HasUnpoolingOutputs", true, "Padding", "same"
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    flattenLayer("Name", "flatten")
    fullyConnectedLayer(num_latent, "Name", "fc")
    fullyConnectedLayer((320*4*num_filters), "Na
    reshapeLayer("Reshape")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    maxUnpooling2dLayer("Name", "maxunpool")
    leakyReluLayer("Name", "relu_3")
    batchNormalizationLayer("Name", "batchnorm_3")
    transposedConv2dLayer(filterSize, 2*num_filt
        "Cropping", "same")];
lgraph = addLayers(lgraph, tempLayers);

```

```

        "Cropping", "same")
        regressionLayer("Name", "regressionoutput")]
lgraph = addLayers(lgraph, tempLayers);

% Limpa as variáveis auxiliares
clear tempLayers;

% conecta as ramificações das camadas
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "maxpoolForUnpool
lgraph = connectLayers(lgraph, "Reshape", "maxunp
lgraph = connectLayers(lgraph, "transposed-conv"
lgraph = connectLayers(lgraph, "transposed-conv_

% Plotando as camadas
plot(lgraph);
analyzeNetwork(lgraph)

% Define as opções de treinamento

```

```

%% Treinamento da rede neural
autoenc = cell(1,num_netw_train);
for k = 1:num_netw_train
    for j = 0:num_dano-1
        for i = 1:num_ac
            fprintf('\n')
            disp(['REDE DE TREINAMENTO ',num2st
            disp(['CAE - Estágio de Treinamento
            disp(['-> AUTOENCODER - ACELERÔMETR
            autoenc{k}{j+1,i} =
            trainNetwork(V_train{j+1,i},V_train
            %{1 x num_netw_train}{num_dano x nu
        end
    end
end
% Salvando os dados de treinamento da rede
save('CAE-PORTICO-TIME-DEF','autoenc')
% Recarrega os dados da rede treinada
load('CAE-PORTICO-TIME-DEF')

%%%%%%%%%%%% Erro de Reconstrução %%%%%%%%%%%%%%
%% Extraindo os sinais preditos da rede
V_pred = cell(1,num_netw_train);
V_pred_train = cell(1,num_netw_train);
V_pred_validation = cell(1,num_netw_train):

```

```

        (autoenc{k}{j+1,i},V_validation{j+1,i}),
        V_pred_dano{k}{j+1,i} = reshape(pre
        (autoenc{k}{j+1,i},V_dano{j+2,i}), [
        V_pred{k}{j+1,i}(:,A) = V_pred_train
        V_pred{k}{j+1,i}(:,B) = V_pred_vali
        V_pred{k}{j+1,i}(:,C) = V_pred_dano

        end

    end

clear A B
end

% Salvando os sinais preditos
save('Dado-Predito','V_pred')
% Recarrega os sinais preditos
load('Dado-Predito')

%%%%%%%%%%%% Espaço Latente %%%%%%%%%%%%%%
%% Extraindo os parâmetros do espaço latente
esp_lat = cell(num_netw_train,num_ac);
for k = 1:num_netw_train
    for i = 1:num_ac
        esp_lat{k,i} = reshape(activations
            (autoenc{k}{5,i},V_norm{i},"fc"),num_la
            latent{k}(num_latent*i-(num_latent-1):n

    end
end

```

```

methods
    function layer = reshapeLayer(name)
        layer.Name = name;
    end
    function [Z] = predict(layer, X)
        Z = reshape(X, (inputSize(1)/8), 1, 4);
        Z = dlarray(Z, 'SSCB');
    end
end
end

function stop = RMSECallback(info)
    % Inicializar o sinalizador de parada como
    stop = false;
    % Define o erro de treinamento desejado
    if info.TrainingRMSE < 1.5
        stop = true;
        % Defina o sinalizador de parada como v
        % treinamento desejado for alcançado
    end
end
end

```