

João Gabriel da Costa de Souza Duarte

**Sobre o Monitoramento de Saúde Estrutural  
utilizando Análise Modal: uma abordagem de  
Aprendizado de Máquina e Aprendizado  
Profundo**

Maceió, AL

2022

João Gabriel da Costa de Souza Duarte

**Sobre o Monitoramento de Saúde Estrutural utilizando  
Análise Modal: uma abordagem de Aprendizado de  
Máquina e Aprendizado Profundo**

Trabalho de Conclusão de Curso submetido  
ao Colegiado do Curso de Engenharia Civil  
como parte dos requisitos necessários para  
obtenção do grau de Bacharel em Engenharia  
Civil.

Universidade Federal de Alagoas

Centro de Tecnologia

Curso de Engenharia Civil

Orientador: Prof. Dr. João Carlos Cordeiro Barbirato

Coorientador: M.S. João Gabriel Zago

Maceió, AL

2022

**Catálogo na Fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 – 1767

D812s Duarte, João Gabriel da Costa de Souza.  
Sobre o monitoramento de saúde estrutural utilizando análise modal : uma abordagem de aprendizado de máquina e aprendizado profundo / João Gabriel da Costa de Souza Duarte. – Maceió, 2022.  
103 f. : il., graf. e tabs. color.

Orientador: João Carlos Cordeiro Barbirato.

Co-orientador: João Gabriel Zago.

Monografia (Trabalho de conclusão de curso em Engenharia Civil) – Universidade Federal de Alagoas. Centro de Tecnologia. Maceió, 2022.

Bibliografia: f. 100-103.

1. Análise estrutural (Engenharia). 2. Análise modal. 3. Monitoramento de saúde estrutural. 4. Aprendizado do computador. 5. Aprendizado profundo. I. Título.

CDU: 624

*Este trabalho é dedicado àqueles  
que defendem a Liberdade  
de ser, de estar e de pensar.*

# Agradecimentos

Aos meus pais, por todo esforço feito e suporte dado durante anos para que eu chegasse até aqui. Sem eles eu não teria conseguido.

Ao Laboratório de Computação Científica e Visualização, em especial aos amigos Tiago Lôbo e Diogo Cintra, por todas oportunidades e conselhos.

Aos professores e amigos João Barbirato, Eduardo Toledo e Christopher Souza, por terem acreditado no meu potencial e me dado oportunidades. Certamente as experiências que tive com eles me moldaram para o que sou hoje.

À Americanas S.A., em especial ao Delermundo Filho, por terem acreditado no meu potencial e por possibilitarem que eu trabalhasse e estudasse ao mesmo tempo. Este trabalho foi construído com a ajuda das minhas experiências atuando como Cientista de Dados.

Ao meu coorientador e amigo João Gabriel Zago, por ter me mostrado o caminho das pedras no mundo do Aprendizado de Máquina no início de minha carreira. Nossas conversas sobre tecnologia e ciência me ajudaram a ver o mundo de uma forma diferente.

Ao Centro Acadêmico de Engenharia Civil, por ter contribuído com minha formação pessoal e me proporcionado experiências gratificantes e inesquecíveis ao representar os alunos do curso.

*"Per Aspera Ad Astra"*

# Resumo

À medida que a tecnologia avança, estruturas civis tornam-se mais leves e esbeltas. Com projetos estruturais mais ousados e complexos, os materiais utilizados são frequentemente submetidos a cargas dinâmicas advindas de rajadas de vento, abalos sísmicos e ocupação humana. Para assegurar um alto nível de confiança e segurança em estruturas sob tais condições, é necessário implementar um sistema de Monitoramento de Saúde Estrutural com a finalidade de verificar adequadamente a integridade estrutural e mitigar eventuais falhas. Um componente desse sistema de monitoramento é a detecção de dano estrutural por inspeção humana. Entretanto, esse processo é suscetível a erros devido ao próprio fator humano, como a inspeção, visual ou manual, de patologias. Utilizando modelos matemáticos computacionais, a localização e a quantificação do dano pode ser determinada por custosas simulações computacionais para obter resultados fidedignos à realidade, o que impossibilita o monitoramento em tempo-real. Devido aos fatores mencionados, técnicas de Aprendizado de Máquina e Aprendizado Profundo combinadas com Análise Modal vêm sendo utilizadas para a construção de um sistema de monitoramento confiável. Elas permitem detecção rápida e em tempo-real quando submetidas aos conjuntos de dados apropriados e à devida calibração de seus respectivos hiperparâmetros. Nesse sentido, esse trabalho de conclusão de curso tem como objetivo desenvolver uma ferramenta computacional capaz de executar a detecção, localização e quantificação de danos estruturais, baseando-se nos parâmetros modais da estrutura e utilizando Redes Neurais Artificiais e Florestas Aleatórias. Treliças e pórticos feitos em aço, danificados e intactos, foram simulados no programa implementado para determinar os parâmetros modais com a finalidade de criar os conjuntos de entrada para os modelos estatísticos. A capacidade de cada algoritmo de detectar, localizar e quantificar o dano foi estudada para encontrar um modelo que generalize a análise bidimensional de pórticos e treliças sob as condições de contorno utilizadas nas simulações numéricas. Para as tarefas de classificação e regressão, isto é, localização e quantificação do dano, respectivamente, os resultados indicam que as Florestas Aleatórias têm uma melhor performance quando comparadas às Redes Neurais Artificiais para ambos os modelos estruturais, tanto para as tarefas de regressão e de classificação.

**Palavras-chaves:** Análise Matricial de Estruturas. Análise Modal. Monitoramento de Saúde Estrutural. Aprendizado de Máquina. Aprendizado Profundo.

# Abstract

With technological advances, civil structures became taller, slender, and lighter. With bolder structural designs, newer and classical materials are often subject to dynamic loads arising from gusts of wind, earthquakes, and human occupation. To ensure reliability and safety in structures under environmental and occupational conditions, it is necessary to implement a Structural Health Monitoring system to verify the structural integrity and mitigate future failures. One component of the SHM is damage detection, localization, and measurement by human inspection. However, this process is susceptible to errors due to human presence itself, such as incorrect measurement or pathology identification. The damage quantification and localization can be determined by computational simulations, making it impossible for real-time monitoring due to the time required to execute such simulations and obtain answers close to reality. Because of mentioned factors, Machine Learning (ML) and Deep Learning (DL) techniques, combined with Modal Analysis, can be used for such tasks. They enable fast and real-time detection when subjected to the proper dataset and calibration of the model's hyper-parameters. Thus, this undergraduate thesis aims to develop a computational tool capable of performing damage detection, localization, and quantification based on structural modal parameters and using Artificial Neural Networks (ANN) and Random Forests (RF). Several damaged and undamaged steel trusses and frames were submitted to Modal Analysis and simulated in the implemented software to determine the modal parameters and create input datasets for the aforementioned statistical models. The damage detection capability of each model was analyzed to find a model that generalizes two-dimensional steel frames and trusses under the boundary conditions used in the numeric simulations. Results show that RF is better than ANN in both structural models in the classification and regression tasks.

**Key-words:** Matrix Analysis of Structures. Modal Analysis. Structural Health Monitoring. Machine Learning. Deep Learning.



# Lista de ilustrações

Figura 1 – Sistema com um grau de liberdade . . . . .	27
Figura 2 – Diagrama de Corpo Livre para Sistemas Amortecidos com um Grau de Liberdade . . . . .	27
Figura 3 – Resposta de um sistema sobreamortecido . . . . .	29
Figura 4 – Resposta de um sistema sub-amortecido . . . . .	31
Figura 5 – Resposta de um sistema não amortecido . . . . .	33
Figura 6 – Segmento de viga mostrando as forças e deslocamentos nas coordenadas nodais . . . . .	34
Figura 7 – Segmento de viga mostrando as deflexões e deformações lineares devido a um deslocamento unitário em cada uma das coordenadas nodais . . .	35
Figura 8 – Viga cantilever segmentada em 2 elementos de viga com coordenadas nodais numeradas . . . . .	40
Figura 9 – elemento de barra ilustrando forças nodais $f_i$ em coordenadas locais (x, y, z) e forças nodas $F_i$ em coordenadas globais (X, Y, Z) . . . . .	42
Figura 10 – Diagrama de como AM pode ajudar humanos a tomar decisões . . . .	51
Figura 11 – Conjunto de treinamento para aprendizado supervisionado de um algoritmo classificador de <i>e-mails spam</i> . . . . .	51
Figura 12 – Regressão em algoritmos supervisionados . . . . .	52
Figura 13 – Árvore de Decisão para o conjunto de treinamento <i>iris</i> . . . . .	53
Figura 14 – Fronteira de decisões do modelo <i>Iris</i> . . . . .	54
Figura 15 – Conjunto de dado quadrático e pseudo-aleatório para treinamento do modelo preditivo . . . . .	55
Figura 16 – Árvore de Decisão para a tarefa de regressão . . . . .	56
Figura 17 – Fronteiras de decisão na árvore criadas para a tarefa de regressão . . .	57
Figura 18 – Ilustração de como a escolha por uma classe é feita em Florestas Aleatórias	57
Figura 19 – Operações lógicas realizadas com neurônios artificiais . . . . .	58
Figura 20 – Rede Neural Artificial do tipo Perceptron Multicamadas . . . . .	60
Figura 21 – Funções de ativação utilizadas para o treinamento de redes neurais . .	61
Figura 22 – Arquitetura de uma Rede Neural Artificial para a tarefa de classificação	62
Figura 23 – Gradiente Descendente de uma função quadrática . . . . .	63
Figura 24 – Gradiente Descendente Estocástico . . . . .	63
Figura 25 – O PCA identifica o subespaço e projeta os dados sobre ele . . . . .	65
Figura 26 – O PCA identifica o hiperplano 2D e projeta o conjunto de dados 3D sobre ele . . . . .	66
Figura 27 – Fluxograma de execução do programa de análise matricial de estruturas	69
Figura 28 – Pórtico utilizado para a validação do módulo de análise estática . . . .	70

Figura 29 – Viga utilizada para a validação do módulo de análise modal . . . . .	71
Figura 30 – Pórtico utilizado por Rojas et al. (2004) . . . . .	72
Figura 31 – Fluxograma de análise dos resultados obtidos para os modelos estruturais de treliça e pórtico . . . . .	73
Figura 32 – Viga utilizada por Maia com 33 elementos idênticos e 34 nós. . . . .	74
Figura 33 – Treliça utilizada para realizar os estudos de localização e quantificação do dano estrutural. Ela detém 60 nós, 129 elementos com a mesma rigidez axial e 116 graus de liberdade . . . . .	76
Figura 34 – Funções de dano discretizadas em 200 pontos . . . . .	77
Figura 35 – Resultados das previsões para as três funções de dano . . . . .	83
Figura 36 – Resultados das previsões para as três funções de dano com a utilização do PCA na etapa de tratamento de dados . . . . .	84
Figura 37 – Resultados das previsões utilizando Redes Neurais Artificiais para as três funções de dano sem a aplicação do PCA na etapa de tratamento de dados . . . . .	86
Figura 38 – Resultados das previsões utilizando Redes Neurais Artificiais para as três funções de dano com a aplicação do PCA na etapa de tratamento de dados . . . . .	87
Figura 39 – Pórtico utilizado para realizar os estudos de localização e quantificação do dano estrutural. Ele detém 218 nós, 225 elementos com rigidez axial idêntica e 665 graus de liberdade . . . . .	88
Figura 40 – Resultados das previsões utilizando Florestas Aleatórias, sem a aplicação do PCA, para as três funções de dano propostas para o pórtico estudado neste trabalho . . . . .	93
Figura 41 – Resultados das previsões utilizando Florestas Aleatórias, com a aplicação do PCA, para as três funções de dano propostas para o pórtico estudado neste trabalho . . . . .	94
Figura 42 – Resultados das previsões utilizando Redes Neurais Artificiais, sem a aplicação do PCA, para as três funções de dano propostas para o pórtico estudado neste trabalho . . . . .	96
Figura 43 – Resultados das previsões utilizando Redes Neurais Artificiais, com a aplicação do PCA, para as três funções de dano propostas para o pórtico estudado neste trabalho . . . . .	97

# Lista de tabelas

Tabela 1 – Comparação de reações de apoio nos nós A e B entre o programa implementado e a ferramenta Ftool . . . . .	70
Tabela 2 – Comparação entre as frequências naturais obtidas pela implementação própria e a ferramenta Ansys . . . . .	71
Tabela 3 – Comparação entre as frequências naturais obtidas pelo programa implementado e o trabalho de Rojas et al. (2004) . . . . .	72
Tabela 4 – Cenários estudados neste trabalho . . . . .	74
Tabela 5 – Comparação entre os Coeficientes de Pearson obtidos por este trabalho e Maia (2006) . . . . .	75
Tabela 6 – Propriedades de todos os elementos estruturais que compõem a treliça plana . . . . .	76
Tabela 7 – Métricas obtidas após o treinamento do algoritmo de Florestas Aleatórias sem a utilização do PCA para as três funções de dano propostas neste trabalho . . . . .	79
Tabela 8 – Métricas obtidas após o treinamento do algoritmo de Florestas Aleatórias com a utilização do PCA para as três funções de dano propostas neste trabalho . . . . .	79
Tabela 9 – Configuração da rede neural utilizada na tarefa de classificação sem a utilização do PCA . . . . .	80
Tabela 10 – Métricas obtidas após o treinamento do algoritmo de Rede Neural Artificial sem a utilização do PCA para as três funções de dano propostas neste trabalho . . . . .	80
Tabela 11 – Métricas obtidas após o treinamento do algoritmo de Rede Neural Artificial com a utilização do PCA para as três funções de dano propostas neste trabalho . . . . .	81
Tabela 12 – Métricas coletadas, para a tarefa de regressão, após o treinamento do algoritmo de Florestas Aleatórias sem a utilização do PCA para as três funções de dano propostas neste trabalho . . . . .	83
Tabela 13 – Métricas coletadas, para a tarefa de regressão, após o treinamento do algoritmo de Florestas Aleatórias com a utilização do PCA para as três funções de dano propostas neste trabalho . . . . .	84
Tabela 14 – Métricas coletadas, para a tarefa de regressão, após o treinamento do algoritmo de Rede Neural Artificial sem a aplicação do PCA para as três funções de dano propostas neste trabalho . . . . .	85
Tabela 15 – Configuração da rede neural utilizada na tarefa de regressão sem a aplicação do PCA . . . . .	85

Tabela 16 – Métricas coletadas, para a tarefa de regressão, após o treinamento do algoritmo de Rede Neural Artificial com a aplicação do PCA para as três funções de dano propostas neste trabalho . . . . .	86
Tabela 17 – Propriedades físicas e geométricas de todos os elementos estruturais que compõe o pórtico plano estudado neste trabalho . . . . .	87
Tabela 18 – Métricas obtidas após o treinamento do algoritmo de Árvore Aleatórias sem a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho . . . . .	89
Tabela 19 – Métricas obtidas após o treinamento do algoritmo de Árvores Aleatórias com a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho . . . . .	90
Tabela 20 – Configuração da rede neural utilizada na tarefa de classificação sem a utilização do PCA para o pórtico estudado . . . . .	90
Tabela 21 – Métricas alcançadas após o treinamento do algoritmo de Redes Neurais Artificiais sem a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho . . . . .	91
Tabela 22 – Métricas alcançadas após o treinamento do algoritmo de Redes Neurais Artificiais com a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho . . . . .	91
Tabela 23 – Métricas obtidas, para a tarefa de regressão, após o treinamento do algoritmo de Florestas Aleatórias sem a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho . . .	92
Tabela 24 – Métricas alcançadas, para a tarefa de regressão, após o treinamento do algoritmo de Florestas Aleatórias com a aplicação do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho . . .	93
Tabela 25 – Configuração da rede neural utilizada na tarefa de regressão sem a utilização do PCA para o pórtico estudado . . . . .	95
Tabela 26 – Métricas alcançadas após o treinamento do algoritmo de Redes Neurais Artificiais, para a tarefa de regressão, sem a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho	95
Tabela 27 – Métricas obtidas após o treinamento do algoritmo de Redes Neurais Artificiais, para a tarefa de regressão, com a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho	96

# Lista de abreviaturas e siglas

AEC	<i>Autoenconders</i> Convolucionais
AM	Aprendizado de Máquina
AP	Aprendizado Profundo
CSP	Controle Estatístico de Processos
DCL	Diagrama de Corpo Livre
MSE	Erro Quadrático Médio
FA	Floresta Aleatória
GD	Gradiente Descendente
SGD	Gradiente Descendente Estocástico
MRD	Método da Rigidez Direta
MEF	Método dos Elementos Finitos
MEC	Método dos Elementos de Contorno
MC	Monitoramento de Condição
SHM	Monitoramento de Saúde Estrutural
NGL	Número de Graus de Liberdade
MLP	Perceptron de Multicamadas
PD	Prognóstico de Dano
RNA	Rede Neural Artificial
RNC	Rede Neural Convolucional
RNP	Rede Neural Profunda
SMUS	Sistema de Monitoramento de Uso e Saúde
TND	Testagem Não-Destrutiva
TLU	Unidade Lógica de Limiar

# Lista de símbolos

$[C]$	Matriz de Amortecimento da estrutura
$[X]$	Matriz de Características
$[F]$	Matriz de Flexibilidade da estrutura
$[F^D]$	Matriz de Flexibilidade da estrutura danificada
$[M]$	Matriz de Massa da estrutura
$[m]$	Matriz de Massa do elemento em coordenadas globais
$[\bar{m}]$	Matriz de Massa do elemento em coordenadas locais
$[W]_L^T$	Matriz de Pesos transposta
$[K]$	Matriz de Rigidez da estrutura
$[k]$	Matriz de Rigidez do elemento em coordenadas globais
$[\bar{k}]$	Matriz de Rigidez do elemento em coordenadas locais
$[T]$	Matriz de Rotação
$[\Omega]$	Matriz Diagonal de Frequências Naturais
$[I]$	Matriz Identidade
$[\Phi]$	Matriz Modal normalizada pela massa
$\{\ddot{x}\}$	Vetor Aceleração
$\{x\}$	Vetor Deslocamento
$\{y\}$	Vetor de valores analíticos
$\{\hat{y}\}$	Vetor de valores preditos
$\{F(t)\}$	Vetor Força no tempo
$\{\phi\}_i$	Vetor Modo de Vibração $i$
$\{\dot{x}\}$	Vetor Velocidade
$\ddot{x}$	Aceleração

$Ac$	Acurácia
$c$	Amortecimento
$A$	Área de Seção Transversal
$L$	Comprimento
$\rho$	Densidade
$x$	Deslocamento
$e$	Erro
$\omega$	Frequência Natural
$g$	Função de Ativação da camada $L$ da rede neural
$d_i$	Função Geradora de Dano $i$
$f_L$	Função de Mapeamento da camada $L$ da rede neural
$FN$	Número de Falsos Negativos
$FP$	Número de Falsos Positivos
$TN$	Número de Verdadeiros Negativos
$TP$	Número de Verdadeiros Positivos
$m$	Massa
$F1$	Média Harmônica
$E$	Módulo de Elasticidade Longitudinal
$I$	Momento de Inércia
$\nabla$	Operador Gradiente
$P$	Precisão
$R$	<i>Recall</i>
$k$	Rigidez
$\eta$	Taxa de Aprendizado da rede neural
$\alpha_i$	Valor máximo da coluna $j$ da Matriz de Variação de Flexibilidade
$b_L$	Viés da camada $L$ da rede neural
$\dot{x}$	Velocidade

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
<b>1.1</b>	<b>Comentários Iniciais</b>	<b>19</b>
<b>1.2</b>	<b>Objetivos</b>	<b>24</b>
1.2.1	Objetivos específicos	24
<b>1.3</b>	<b>Justificativa</b>	<b>24</b>
<b>1.4</b>	<b>Metodologia</b>	<b>25</b>
<b>2</b>	<b>DINÂMICA DAS ESTRUTURAS</b>	<b>26</b>
<b>2.1</b>	<b>Introdução</b>	<b>26</b>
<b>2.2</b>	<b>Graus de Liberdade</b>	<b>26</b>
<b>2.3</b>	<b>Sistemas com um Grau de Liberdade</b>	<b>27</b>
2.3.1	Princípio de D'Alembert	27
2.3.2	Equação Geral do Movimento	27
2.3.3	Amortecimento crítico	28
2.3.4	Sobreamortecimento	30
2.3.5	Subamortecimento	30
2.3.6	Ausência de amortecimento	31
<b>2.4</b>	<b>Sistemas com múltiplos Graus de Liberdade</b>	<b>33</b>
2.4.1	Equação Matricial do Movimento	33
<b>2.5</b>	<b>Análise dinâmica de estruturas via Método da Rigidez Direta</b>	<b>34</b>
2.5.1	Determinação dos coeficientes de rigidez devido às forças cortantes e fletoras	35
2.5.2	Determinação dos coeficientes de rigidez devido às forças axiais	38
2.5.3	Matriz de rigidez de um elemento de barra	39
2.5.4	Matriz de rigidez da estrutura	39
2.5.5	Matriz de massa da estrutura	41
2.5.6	Transformação de coordenadas	42
<b>2.6</b>	<b>Análise Modal</b>	<b>44</b>
2.6.1	Frequências Naturais e Modos de Vibração	45
2.6.2	Propriedade de Ortogonalidade dos Modos de Vibração	46
<b>2.7</b>	<b>Método da Variação da Flexibilidade Modal</b>	<b>46</b>
<b>3</b>	<b>SISTEMAS COMPUTACIONAIS INTELIGENTES</b>	<b>49</b>
<b>3.1</b>	<b>Introdução</b>	<b>49</b>
<b>3.2</b>	<b>Aprendizado de Máquina</b>	<b>49</b>
3.2.1	O que é Aprendizado de Máquina?	49
3.2.2	Por que utilizar Aprendizado de Máquina?	50



3.2.3	Aprendizado Supervisionado . . . . .	50
3.2.4	Árvores de Decisão . . . . .	52
3.2.4.1	Treinamento . . . . .	52
3.2.4.2	Realizando decisões . . . . .	52
3.2.4.3	Algoritmo de Treinamento CART . . . . .	53
3.2.4.4	Complexidade Computacional . . . . .	54
3.2.4.5	Regressão . . . . .	55
3.2.5	Florestas Aleatórias . . . . .	56
<b>3.3</b>	<b>Aprendizado Profundo . . . . .</b>	<b>58</b>
3.3.1	Redes Neurais Artificiais . . . . .	58
3.3.1.1	Operações Lógicas com Neurônios . . . . .	58
3.3.1.2	O Perceptron . . . . .	59
3.3.1.3	Perceptron Multicamadas e Retropropagação . . . . .	59
3.3.2	Funções de ativação . . . . .	60
3.3.2.1	Regressão utilizando MLPs . . . . .	61
3.3.2.2	Classificação utilizando MLPs . . . . .	61
3.3.3	Gradiente Descendente . . . . .	62
3.3.3.1	Gradiente Descendente Estocástico . . . . .	62
3.3.4	Treinamento de uma Rede Neural . . . . .	64
<b>3.4</b>	<b>Redução de Dimensionalidade . . . . .</b>	<b>64</b>
3.4.1	Análise de Componentes Principais . . . . .	65
<b>4</b>	<b>PROGRAMA COMPUTACIONAL DESENVOLVIDO PARA O MO- NITORAMENTO DE SAÚDE ESTRUTURAL . . . . .</b>	<b>67</b>
<b>4.1</b>	<b>Entrada de Dados . . . . .</b>	<b>67</b>
4.1.1	<i>Flags</i> . . . . .	68
<b>4.2</b>	<b>Fluxograma . . . . .</b>	<b>69</b>
<b>4.3</b>	<b>Validação . . . . .</b>	<b>69</b>
4.3.1	Análise Estática . . . . .	70
4.3.2	Análise Modal . . . . .	71
4.3.2.1	ANSYS . . . . .	71
4.3.2.2	Literatura . . . . .	71
<b>5</b>	<b>EXPERIMENTOS COMPUTACIONAIS E RESULTADOS . . . . .</b>	<b>73</b>
<b>5.1</b>	<b>Introdução . . . . .</b>	<b>73</b>
<b>5.2</b>	<b>Benchmarking . . . . .</b>	<b>74</b>
<b>5.3</b>	<b>Treliça Plana . . . . .</b>	<b>75</b>
5.3.1	Geração de dados para treinamento dos modelos estatísticos . . . . .	76
5.3.2	Localização do Dano . . . . .	78
5.3.2.1	Florestas Aleatórias . . . . .	78

5.3.2.1.1	Cenário 1 . . . . .	78
5.3.2.1.2	Cenário 2 . . . . .	79
5.3.2.2	Redes Neurais Artificiais . . . . .	80
5.3.2.2.1	Cenário 3 . . . . .	80
5.3.2.2.2	Cenário 4 . . . . .	81
5.3.3	Quantificação do Dano . . . . .	81
5.3.3.1	Florestas Aleatórias . . . . .	82
5.3.3.1.1	Cenário 5 . . . . .	82
5.3.3.1.2	Cenário 6 . . . . .	83
5.3.3.2	Redes Neurais Artificiais . . . . .	84
5.3.3.2.1	Cenário 7 . . . . .	84
5.3.3.2.2	Cenário 8 . . . . .	85
<b>5.4</b>	<b>Pórtico Plano . . . . .</b>	<b>87</b>
5.4.1	Geração de dados para treinamento dos modelos estatísticos . . . . .	88
5.4.2	Localização do Dano . . . . .	89
5.4.2.1	Florestas Aleatórias . . . . .	89
5.4.2.1.1	Cenário 1 . . . . .	89
5.4.2.1.2	Cenário 2 . . . . .	89
5.4.2.2	Redes Neurais Artificiais . . . . .	90
5.4.2.2.1	Cenário 3 . . . . .	90
5.4.2.2.2	Cenário 4 . . . . .	91
5.4.3	Quantificação do Dano . . . . .	92
5.4.3.1	Florestas Aleatórias . . . . .	92
5.4.3.1.1	Cenário 5 . . . . .	92
5.4.3.1.2	Cenário 6 . . . . .	93
5.4.3.2	Redes Neurais Artificiais . . . . .	94
5.4.3.2.1	Cenário 7 . . . . .	94
5.4.3.2.2	Cenário 8 . . . . .	95
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>98</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>100</b>

# 1 Introdução

## 1.1 Comentários Iniciais

Na contemporaneidade, os seres humanos são altamente dependentes de sistemas estruturais, elétricos e mecânicos, como aeronaves, pontes, usinas de geração de energia, fabricação de semicondutores e edifícios. Tais sistemas são complexos e geralmente não são facilmente substituídos devido a fatores econômicos, logísticos e construtivos. Logo, tecnologias de monitoramento tornam-se necessárias para a usabilidade manter-se segura e possibilitar o prolongamento da vida útil de serviço. Além disso, é possível aplicar essas tecnologias para o estudo de novos materiais cujo as propriedades físicas não são completamente conhecidas, a partir da análise de sua degradação a longo prazo.

Ainda, as técnicas de monitoramento podem ser utilizadas para o desenvolvimento otimizado de projetos estruturais, permitindo menor custo a partir do conhecimento do comportamento estrutural e mecânico a longo prazo para condições ocupacionais e ambientais previamente determinadas. Dessa forma, torna-se possível reduzir as margens de segurança e a quantidade de material necessário na execução do projeto.

Devido a importância econômica e social dos sistemas supracitados, o monitoramento é um fator imprescindível, pois a detecção de dano pode prevenir falhas e colapsos com grande antecedência, prevenindo catástrofes e reduzindo custos.

Detecção de dano envolve diversas ferramentas de monitoramento determinístico ou probabilístico: Monitoramento de Saúde Estrutural (SHM), Monitoramento de Condição (MC), Testagem Não-Destrutiva (TND), Sistema de Monitoramento de Uso e Saúde (SMUS), Controle Estatístico de Processos (CSP) e Prognóstico de Dano (PD). Este trabalho é focado na primeira, onde são estudados os métodos de detecção a partir das características modais da estrutura. Nesse sentido, a ideia fundamental de monitoramento utilizando vibrações é obter informações a respeito da resposta dinâmica da estrutura devido a carregamentos de serviço ou extraordinários. Os métodos de Monitoramento de Saúde Estrutural são categorizados da seguinte forma (RYTTER, 1993):

- Nível 1 - Detecção: fornece indicação que o dano está presente na estrutura;
- Nível 2 - Localização: oferece informação a respeito da localização do dano;
- Nível 3 - Quantificação: provê indicação da severidade do dano;
- Nível 4 - Consequência: fornece informação a respeito da atual condição de segurança da estrutura.

O termo monitoramento de saúde estrutural refere-se ao processo de detecção, localização e mensuração de dano em estruturas aeroespaciais, civis ou mecânicas. Esse processo envolve:

1. Observação do comportamento dinâmico da estrutura no decorrer do tempo;
2. Extração das características sensíveis a dano a partir dos dados coletados;
3. Análise, através de modelos estatísticos, das características modais para determinar o estado atual de saúde do sistema estudado.

Para o SHM de longo prazo, a saída desse processo é periodicamente atualizada com as informações referentes às características sensíveis ao dano que são modificadas com o envelhecimento e a degradação natural da estrutura devido às condições de carregamento e ao ambiente no qual a estrutura está inserida (FARRAR; WORDEN, 2013).

Diante de eventos extremos, como desastres naturais, o monitoramento de saúde estrutural pode ser utilizado para o rápido levantamento de informações da condição estrutural em tempo-real de uma construção civil, com a finalidade de prover informações confiáveis a respeito da performance do sistema durante o evento e posteriormente sobre a integridade estrutural, o que não seria possível em métodos convencionais, como a inspeção visual que depende exclusivamente da capacidade de detecção de um humano.

Há duas abordagens possíveis para a construção de um sistema de SHM (SVENDSEN et al., 2022):

1. Baseada em métodos numéricos, como Método de Elementos Finitos (MEF) ou Método de Elementos de Contorno (MEC);
2. Baseada em dados.

Na primeira abordagem, o modelo analítico é atualizado quando há uma mudança nas características modais, isto é, modos de vibração, frequências naturais e amortecimento dos nós analisados. Tal abordagem é computacionalmente custosa, devido ao extenso tempo de simulação necessário para obter uma resposta da localização e da severidade do dano fidedigno com a realidade.

Na segunda abordagem, conhecida também como problema inverso, tem-se o conhecimento prévio das frequências naturais e dos modos de vibração da estrutura e a partir de mudanças nos seus parâmetros modais, causadas pela degradação das propriedades físicas devido às ações já mencionadas, é possível detectar, localizar e quantificar o dano com o auxílio de modelos estatísticos. Em ambas abordagens, é possível determinar o dano em um elemento como a redução do módulo de rigidez longitudinal, massa linear ou área de

seção transversal, uma vez que as características dinâmicas estão intrinsecamente ligadas a essas propriedades.

Um sistema de monitoramento de saúde estrutural consiste em Acelerômetros ou *Strain Gauges* instalados na estrutura da estrutura, conectados através de cabos ou via *wireless* (LYNCH, 2017). Tais transdutores são dispostos em pontos estratégicos, de maneira a identificar respostas dinâmicas e características modais. Esses sensores podem ser colocados nos nós dos elementos estruturais, de maneira a simular os dados obtidos por modelos numéricos. Como o número de sensores implantados é diretamente proporcional à qualidade do monitoramento, busca-se instalar centenas de sensores por toda estrutura, com a finalidade de obter informações globais e locais. Essas informações coletadas são redirecionadas para algum banco de dados, que serão utilizadas para construir o histórico do comportamento estrutural.

Carrillo (2004) estudou os métodos dinâmicos de detecção de dano em vigas a partir da variação da flexibilidade aferida dinamicamente. Os métodos revisados por ele formam parte das técnicas de TND. Em seu trabalho, o problema de vibração inverso foi apresentado e a matriz de flexibilidade estática da estrutura foi determinada a partir dos parâmetros modais utilizando o MEF. Baseando-se nas respostas obtidas, foi possível identificar as áreas danificadas da estrutura.

Tan et al. (2017) apresentou uma técnica baseada em vibração. Utilizando apenas o primeiro modo de vibração da estrutura, conseguiu detectar, localizar e quantificar o dano em vigas de aço. No trabalho, foram criados dois casos de dano durante as simulações: único e múltiplo. Para o cenário de dano único, a energia modal de deformação baseada em um índice de dano ( $\beta$ ) foi utilizada na análise e a metodologia aplicada conseguiu alcançar o objetivo proposto. Para o caso de dano múltiplo, uma Rede Neural Artificial (RNA) foi utilizada, onde sua camada entrada incorporou o  $\beta$ . O dano foi representado como a diminuição da rigidez estrutural, e seus resultados demonstraram a viabilidade do método e a sua aplicação em prevenir a falha estrutural.

No trabalho de Gillich et al. (2022), foram propostos dois métodos estatísticos para detecção de dano utilizando os parâmetros modais da estrutura: RNN e Florestas Aleatórias (FA). Além disso, foi desenvolvido um banco de dados contendo cenários de dano para uma viga engastada-livre, com a finalidade de alimentar os modelos propostos. Em uma primeira abordagem, a localização do dano foi feita através do treinamento da rede neural, utilizando dados dos parâmetros modais da viga como um todo. Em seguida, para a etapa de quantificação do dano, foram utilizados os dados provenientes do elemento onde o dano foi encontrado na etapa anterior. Concluiu-se que para o cenário criado para o artigo, foi possível encontrar e mensurar o dano proposto, com uma acurácia de 99,4%, o que demonstra a robustez e a viabilidade da metodologia empregada.

Rastin, Amiri e Darvishan (2022) propuseram um novo método de Aprendizado

Profundo (AP) não-supervisionado para detecção de dano utilizando *Autoencoders* Convolucionais (AEC). O objetivo principal da metodologia empregada foi identificar e quantificar o dano estrutural utilizando uma rede neural AEC, que faz uso dos dados de vibração provenientes da estrutura sã para a fase de treinamento do modelo. Esse modelo foi escolhido devido a capacidade das camadas convolucionais de extrair características dos dados de entrada e devido às vantagens intrínsecas de algoritmos de aprendizado não-supervisionado *autoencoders*, pois não necessitam dos dados originados das estruturas danificadas no treinamento do modelo. Após aplicar a metodologia em exemplos numéricos, foi provada a eficiência do método proposto em aferir a saúde global da estrutura e a severidade do dano.

Em sua tese doutoral, Morales (2012) estudou duas formas de detecção de dano em estruturas a partir da utilização da resposta dinâmica e da aplicação de técnicas de aprendizado de máquina. Em um primeiro momento, um problema de otimização foi criado para a minimização da diferença entre os parâmetros dinâmicos experimentais provenientes da estrutura danificada e aqueles calculados pelo modelo matemático de elementos finitos. Em segundo lugar, ele utilizou redes neurais para mapear as características modais das estruturas danificadas e não-danificadas e verificar a existência do dano nos modelos estruturais analisados. Seus resultados mostram a habilidade das técnicas computacionais empregadas e sua viabilidade para os casos estudados.

Bakhary, Hao e Deeks (2007) propuseram uma abordagem estatística utilizando o Método de Rosenbluth verificado por simulações de Monte Carlo para levar em consideração as incertezas relacionadas a estimação dos parâmetros de rigidez da estrutura no momento do desenvolvimento do modelo de RNN. A probabilidade da existência de dano foi calculada baseada na função de probabilidade dos estados não-danificados e danificados do modelo numérico e laboratorial estudado. A presença de ruído e múltiplos casos de severidade de dano foram estudados no trabalho. Os resultados demonstraram que a presença do dano pode ser estimada por esse método probabilístico combinado com a RNN criada.

Yang et al. (2012) abordaram uma nova técnica de índice de dano, chamada *Stiffness Variation Index* (SVI) para detectar e mensurar dano em vigas bi-apoiadas. Na técnica, através dos deslocamentos e frequências modais extraídos de ensaios dinâmicos, a curvatura modal é obtida e as rigidezes referentes aos momentos fletores são calculadas diretamente na técnica apresentada. Comparando a rigidez calculada com a do projeto original da estrutura, a localidade e a severidade do dano pode ser determinada. Os resultados demonstram a viabilidade da técnica ao mensurar diretamente o parâmetro físico relacionado à rigidez com a presença do dano.

Janaa et al. (2022) mostraram um novo modelo de AP para sistemas lineares com parâmetros que não variam no tempo. O modelo é capaz de identificar a presença e o tipo de

falha nos dados provenientes de sensores defeituosos e a respectiva localização dos sensores. Em seguida, a metodologia reconstrói o dano para o estado correto através da aplicação de Redes Neurais Convolucionais (RNC) e AEC utilizando dados numéricos e experimentais. Foi demonstrada a eficiência do modelo de AP proposto para indicar os sensores defeituosos em um modelo sintético de pórtico e em um modelo experimental de arco de ponte. Além disso, os resultados alcançados mostraram uma acurácia de aproximadamente 100% na localização dos sensores defeituosos, 98.7% de acurácia na detecção do tipo da falha, e mais do que 99% de acurácia na reconstrução dos dados.

Feng e Feng (2018) realizaram uma revisão bibliográfica sobre a utilização de Visão Computacional para o Monitoramento de Saúde Estrutural (SHM). Devido aos grandes e rápidos avanços da área, o sensor de visão baseado em câmera emergiu como alternativa promissora para os sensores de contato convencionais utilizados para a detecção da resposta dinâmica da estrutura. Vantagens significativas do sensor de visão computacional incluem o baixo custo de aplicação, fácil instalação e operação, e flexibilidade para extrair as informações buscadas em qualquer ponto da estrutura a partir de um vídeo. A partir da revisão realizada no trabalho, os autores concluíram que a tecnologia pode ser aplicada futuramente para a detecção de dano e identificação de características modais.

Farrar e Worden (2010) apontaram os desafios para o SHM. O primeiro desafio apontado no trabalho é o fato do dano ser um fenômeno local e que pode ser insignificante para alterações das baixas frequências globais da estrutura. Dessa forma, alguns métodos de detecção não são suscetíveis a pequenas variações dos modos de vibração e das frequências naturais, como o Método da Rigidez Direta. O segundo ponto apontado é que na maioria dos casos, a escolha das características modais e a identificação do dano deve ser feito de maneira não-supervisionada, mas não são todas as estruturas que detêm dados apropriados para essa metodologia. O terceiro fator analisado é a garantia do funcionamento do sistema de sensores. Assegurar que eles estão funcionando corretamente é primordial para o monitoramento, pois se existe a possibilidade dos sensores apresentarem dano e informações imprecisas, é o caso de monitorar os sensores por si só, o que acaba sendo contraprodutivo.

Júnior (2020) estudou a detecção de danos em vigas utilizando quatro métodos de identificação: 1) Diferença da Curvatura, 2) Identificador Baseado em Dano dos Modos de Vibração, 3) Mudança da Energia de Deformação Modal e 4) Mudança da Energia de Deformação Reformulada. Além disso, ele propôs uma mudança nos dois primeiros métodos. Os resultados obtidos indicam que o método da Mudança da Energia de Deformação Modal detectou os danos corretamente, apesar da sua difícil aplicação prática. O método da Mudança da Energia de Deformação Modal Reformulada demonstrou-se promissor e os métodos modificados propostos apresentaram melhores resultados em relação aos seus pares originais.

Por fim, Falcão (2017) estudou a aplicação de Máquinas de Vetor de Suporte para a detecção de dano em vigas de concreto armado. Segundo o autor, os melhores resultados foram alcançados quando o kernel *Radial Basis Function* (RBF) foi utilizado durante o treinamento do algoritmo.

## 1.2 Objetivos

Este trabalho tem como objetivo principal o desenvolvimento de um programa computacional capaz de efetuar Análise Modal com a finalidade de fornecer dados para os algoritmos de Aprendizado de Máquina e Aprendizado Profundo, que são utilizados para detectar, localizar e mensurar o dano em elementos estruturais de pórtico e treliça feitos em aço.

### 1.2.1 Objetivos específicos

- Estudar o estado-da-arte do problema de detecção de dano resolvido a partir de mudanças nos parâmetros modais da estrutura;
- Criar um programa computacional em Python utilizando o Método da Rigidez Direta que seja capaz de realizar Análise Modal sem amortecimento;
- Validar o programa implementado com os *softwares* comerciais Ftool e Ansys;
- Gerar modelos danificados de treliças planas e pórticos planos a partir de dano único, diferentes severidades e aplicado a diferentes elementos da estrutura;
- Armazenar as diferenças da matriz de flexibilidade modal entre o modelo intacto e os modelos danificados previamente simulados;
- A partir dos dados armazenados, alimentar os modelos de Floresta Aleatória e Rede Neural Artificial propostos. Em seguida, treiná-los e avaliá-los por métricas estatísticas;
- Contribuir com a disseminação do monitoramento de saúde estrutural no meio acadêmico brasileiro, visto os impactos positivos que são proporcionados.

## 1.3 Justificativa

Considerando o atual cenário das construções brasileiras, o histórico de abandono de prédios públicos e a importância que a comunidade acadêmica internacional tem dado para o tema SHM, torna-se imprescindível que técnicas de monitoramento estrutural sejam



implantadas em território nacional, visando economizar recursos financeiros, prevenir acidentes e falhas estruturais, otimizar o agendamento de manutenções e prolongar a vida útil de serviço das estruturas. Além disso, é notável o acanhado acervo de conteúdo acadêmico em português abrangendo o tema proposto por este trabalho. Logo, a disseminação e produção de conhecimento na área de Engenharia Civil torna-se necessária devido aos fatores supracitados.

## 1.4 Metodologia

O trabalho proposto foi desenvolvido de forma numérica através do desenvolvimento e implementação de um programa computacional, que fez uso modelos estruturais lineares, isotrópicos, elásticos e homogêneos, baseados na Análise Matricial de Estruturas Planas. O programa desenvolvido foi validado nos pares comerciais Ftool e Ansys Workbench Student Version para as análises estática e modal, respectivamente.

A partir de uma estrutura intacta, foram criados diferentes cenários de dano único, modelados numericamente por meio de uma redução do Módulo de Elasticidade Longitudinal dos elementos danificados. As características modais da estrutura sã e de cada um dos seus modelos danificados foram obtidas via Análise Modal, em seguida armazenadas e comparadas por meio do Método da Variação da Flexibilidade Modal, proposto por Carrillo (2004). Utilizou-se modelos estatísticos para detectar, localizar e quantificar o dano nos elementos que apresentaram degradação da integridade estrutural.

Realizou-se um *benchmark* com o trabalho de Maia (2006), tendo como finalidade validar a metodologia adotada e verificar as métricas de performance. Em seguida, diferentes modelos de pórticos e treliças foram estudados para atestar a viabilidade da utilização de técnicas de Florestas Aleatórias e Redes Neurais Artificiais no processo de monitoramento de estruturas planas com muitos graus de liberdade, testando e verificando as métricas Acurácia, Precisão, F1 Score e Recall para as tarefas de classificação, e as métricas MAPE,  $R^2$  e Similaridade por Cosseno para as tarefas de regressão.

Os modelos de treliça e pórtico foram escolhidos por causa da facilidade de modelagem computacional. Devido ao fato da treliça ter uma configuração geométrica mais complexa do que o pórtico, espera-se que os algoritmos estatísticos tenham mais dificuldade de generalizá-la. Além disso, escolheu-se as Florestas Aleatórias e as Redes Neurais Artificiais como algoritmos de inteligência computacional com a finalidade de comparar o poder de generalização de um algoritmo menos sofisticado com um mais sofisticado. O custo computacional de ambos também foi estudado, onde foi medido o tempo de treinamento e previsão dos algoritmos.

## 2 Dinâmica das Estruturas

### 2.1 Introdução

A análise e o projeto de sistemas estruturais para resistir aos mais variados efeitos que variam no tempo requerem idealizações conceituais e simplificações no que diz respeito ao sistema físico real. Tal idealização é conhecida como modelo matemático. Essas simplificações são categorizadas em (PAZ; KIM, 2019):

1. Assume que o material é elástico, linear, isotrópico e homogêneo;
2. Considera que as forças são aplicadas em pontos geométricos, sendo elas pontuais, constantes ou periódicas;
3. Assume que elementos unidimensionais constituem vigas, pórticos e treliças. Para placas, elementos bidimensionais são adotados com uma espessura pequena. É importante considerar que estruturas contínuas são discretizadas em elementos, especificando nós e direções para os deslocamentos através de coordenadas.

### 2.2 Graus de Liberdade

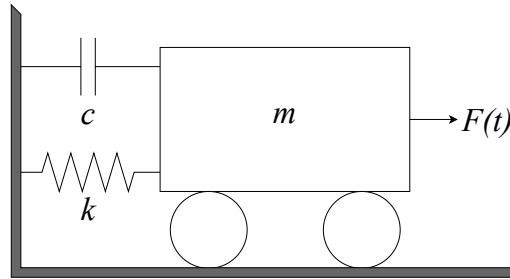
Em análise dinâmica de estruturas, o número de coordenadas independentes necessárias para especificar as configurações de um sistema estrutural em qualquer tempo é referida como o Número de Graus de Liberdade (NGL). Em um contexto geral, uma estrutura contínua tem infinitos graus de liberdade. Isso é completamente impraticável no contexto de simulações devido ao alto custo computacional e da complexidade da análise. Nesse sentido, adota-se modelos matemáticos para realizar a idealização do meio contínuo, de maneira que seja possível discretizá-lo em um número finito de graus de liberdade, o que possibilita a análise numérica.

A Figura 1 mostra uma configuração possível para sistemas de um grau de liberdade, onde o elemento  $m$  representa a massa e a inércia da estrutura, o elemento  $k$  representa a força elástica restauradora e a energia potencial armazenada na estrutura, o elemento  $c$  representa as características friccionais e de dissipação de energia do elemento estrutural, e  $F(t)$  representa a força externa de excitação do sistema em função do tempo.

Certamente não existem modelos numéricos completamente fidedignos com a realidade, como apontou Kassimali (2012) em seu livro. Dessa forma, tais modelos podem providenciar informações aproximadas a respeito do real comportamento da estrutura.

Entretanto, de um ponto de vista prático, as informações obtidas através deles são de suma importância para o entendimento do comportamento dinâmico do sistema físico real.

Figura 1 – Sistema com um grau de liberdade



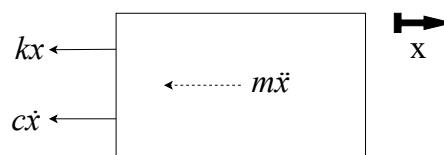
Adaptado de: Clough e Penzien (2003)

## 2.3 Sistemas com um Grau de Liberdade

### 2.3.1 Princípio de D'Alembert

O princípio de d'Alembert afirma que um sistema tende a ficar em equilíbrio dinâmico quando uma força externa inercial fictícia é aplicada. Ao analisar o Diagrama de Corpo Livre (DCL) referente ao sistema apresentado na Figura 1, é possível construir a Figura 2 após aplicar uma força inercial fictícia e anular a excitação externa, isto é, possibilitando ao sistema oscilar em vibração livre. Esta força é igual à massa multiplicada pela aceleração, e deve ser sempre inversa à direção positiva do eixo de referência.

Figura 2 – Diagrama de Corpo Livre para Sistemas Amortecidos com um Grau de Liberdade



Fonte: Autor (2022)

### 2.3.2 Equação Geral do Movimento

Aplicando a Segunda Lei de Newton no DCL mostrado na Figura 2 e utilizando o Princípio de D'Alembert, obtém-se a equação diferencial do movimento:

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (2.1)$$

cuja solução é da forma:

$$x = Ce^{pt} \quad (2.2)$$

Substituindo a Equação 2.2 na Equação 2.1, tem-se:

$$mCp^2e^{pt} + cCpe^{pt} + kCe^{pt} = 0 \quad (2.3)$$

Eliminando os termos comuns, obtém-se a equação característica do sistema:

$$mp^2 + cp + k = 0 \quad (2.4)$$

em que as suas raízes são definidas da seguinte forma:

$$p_i = \frac{-c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}} \quad (2.5)$$

A solução da Equação 2.1 é dada pela superposição de duas possíveis soluções, logo:

$$x(t) = C_1e^{p_1t} + C_2e^{p_2t} \quad (2.6)$$

sendo  $C_1$  e  $C_2$  as constantes de integração determinadas pelas condições de contorno do sistema.

A Equação 2.6 detém 3 cenários possíveis, que dependem da expressão dentro do radical da Equação 2.5:

1. A quantidade no radical é igual a zero, significando que o sistema está no estado de amortecimento crítico;
2. A expressão do radical é maior do que zero, o que significa um estado de sobre-amortecimento do sistema oscilador;
3. A quantidade abaixo da raiz quadrada é menor do que zero, logo o sistema está no estado de sub-amortecimento.

### 2.3.3 Amortecimento crítico

Para um sistema que está oscilando em estado de amortecimento crítico ( $c = c_{cr}$ ), tem-se:

$$\left(\frac{c_{cr}}{2m}\right)^2 - \frac{k}{m} = 0 \quad (2.7)$$

então:

$$c_{cr} = 2\sqrt{km} \quad (2.8)$$

Sendo a Frequência Natural de um sistema de um Grau de Liberdade sem amortecimento dada por:

$$\omega = \sqrt{\frac{k}{m}} \quad (2.9)$$

Logo, a Equação 2.8 torna-se:

$$c_{cr} = 2m\omega \quad (2.10)$$

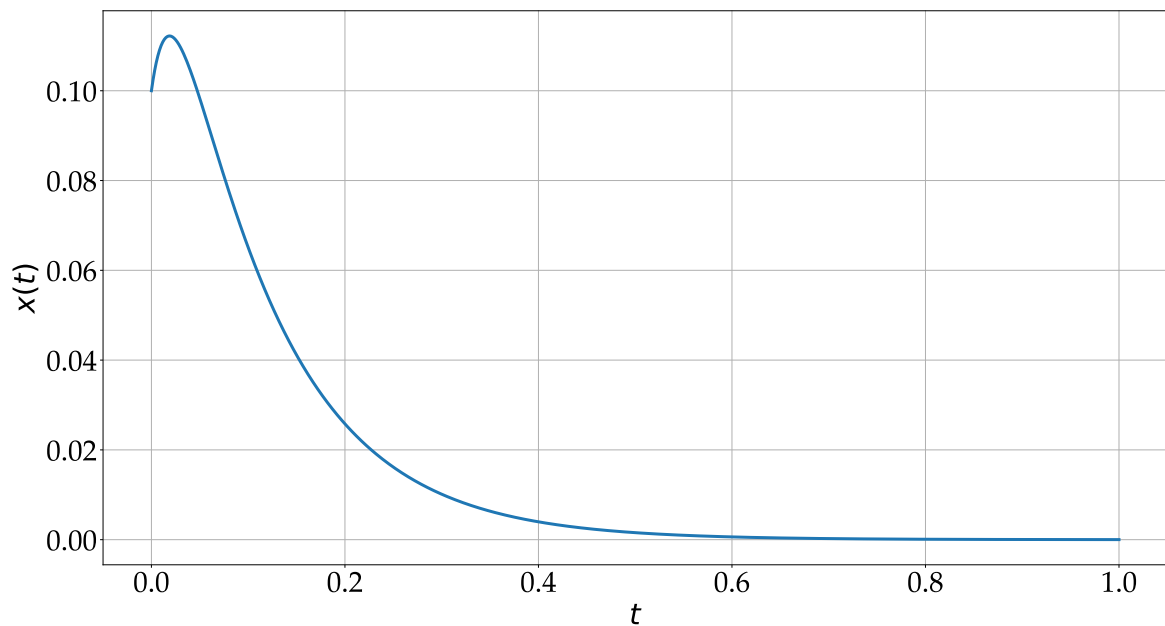
No estado de amortecimento crítico, as raiz da equação características são iguais:

$$p_1 = p_2 = \frac{-c_{cr}}{2m} \quad (2.11)$$

Finalmente, a solução de  $x(t)$  é dada:

$$x(t) = (C_1 + C_2t)e^{\left(\frac{c_{cr}}{2m}\right)t} \quad (2.12)$$

Figura 3 – Resposta de um sistema sobreamortecido



Fonte: Autor (2022)

### 2.3.4 Sobreamortecimento

Um sistema estar em estado de sobreamortecimento significa:

$$c > c_{cr} \quad (2.13)$$

Dessa forma, a quantidade abaixo do radical da Equação 2.5 é positiva. Logo,  $p_1 \neq p_2$ . A Figura 3 mostra a resposta de um sistema em estado de amortecimento crítico. A resposta para o estado de sobreamortecimento é semelhante à de um sistema em amortecimento crítico, mudando apenas a taxa em que o deslocamento do sistema vai à zero.

### 2.3.5 Subamortecimento

Quando o valor do coeficiente de amortecimento é menor do que o valor do amortecimento crítico, diz-se que o sistema é sub-amortecido e as raízes da equação Equação 2.5 são complexas. O que leva à:

$$p_n = \frac{-c}{2m} \pm i \sqrt{\frac{k}{m} - \left(\frac{c}{2m}\right)^2} \quad (2.14)$$

sendo  $i = \sqrt{-1}$  a unidade imaginária.

Para solucionar equações com raízes complexas, utiliza-se as expressões de Euler que relacionam funções trigonométricas e exponenciais:

$$e^{\pm ix} = \cos x \pm i \sin x \quad (2.15)$$

Fazendo uso das raízes  $p_1$  e  $p_2$  da Equação 2.14 e da expressão descrita pela Equação 2.15 na Equação 2.6, tem-se a solução geral para sistemas subamortecidos:

$$x(t) = e^{-\left(\frac{c}{2m}\right)t} (A \cos \omega_d t + B \sin \omega_d t) \quad (2.16)$$

em que  $A$  e  $B$  são as constantes de integração definidas pelas condições de contorno do sistema e  $\omega_d$  é a frequência de amortecimento, dada por:

$$\omega_d = \sqrt{\frac{k}{m} - \left(\frac{c}{2m}\right)^2} \quad (2.17)$$

ou

$$\omega_d = \omega \sqrt{1 - \xi^2} \quad (2.18)$$

sendo  $\xi$  a taxa de amortecimento do sistema, definida como:

$$\xi = \frac{c}{c_{cr}} \quad (2.19)$$

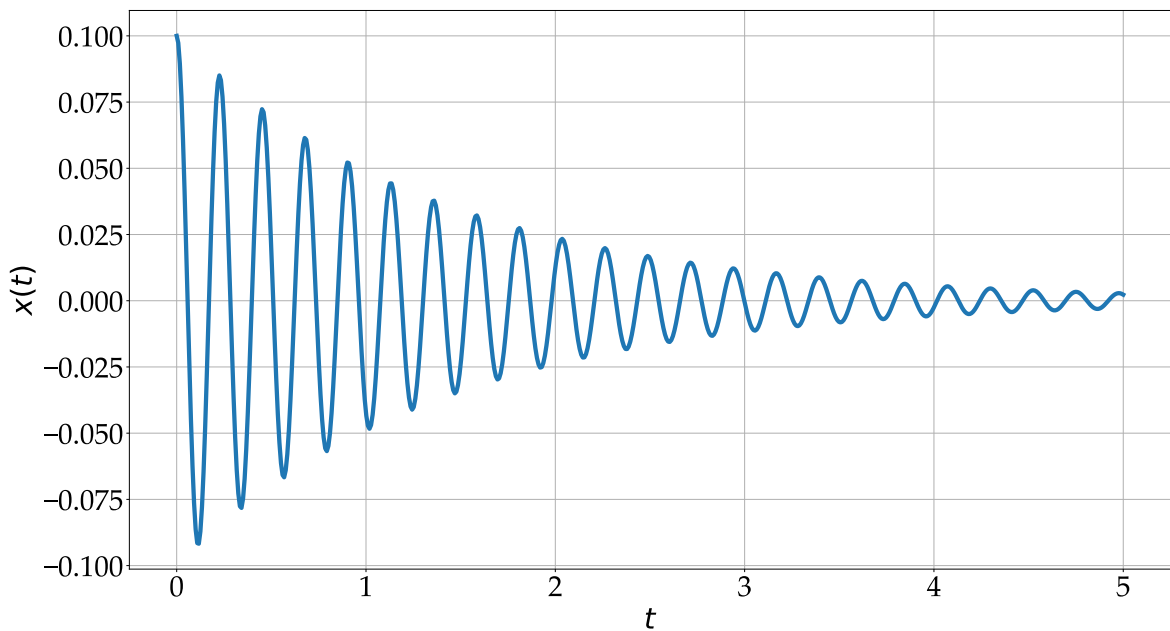
Finalmente, quando as condições iniciais de deslocamento e velocidade,  $x_0$  e  $v_0$  são introduzidas, as constantes de integração podem ser definidas na Equação 2.16, levando à:

$$x(t) = e^{-\xi\omega t} \left( x_0 \cos \omega_d t + \frac{v_0 + x_0 \xi \omega}{\omega_d} \sin \omega_d t \right) \quad (2.20)$$

Um histórico gráfico da resposta de um sistema subamortecido com deslocamento inicial  $x_0$  e zero velocidade, isto é,  $v_0 = 0$ , é mostrado na Figura 4. Perceptivelmente que o movimento é oscilatório, mas as amplitudes são distintas a cada período, o que implica na dissipação de energia do sistema proveniente do amortecimento. Define-se esse intervalo como período de amortecimento:

$$T_d = \frac{2\pi}{\omega_d} = \frac{2\pi}{\omega\sqrt{1-\xi^2}} \quad (2.21)$$

Figura 4 – Resposta de um sistema sub-amortecido



Fonte: Autor (2022)

### 2.3.6 Ausência de amortecimento

Note que se  $c = 0$  na Equação 2.1, o sistema não terá amortecimento, o que implica em um estado de vibração livre. Dessa forma, a equação geral do movimento torna-se:

$$m\ddot{x} + kx = 0 \quad (2.22)$$

Sistemas sem amortecimento são frequentemente estudados e utilizados, pois a partir dessa configuração é possível efetuar a Análise Modal.

A solução da equação Equação 2.22 é da forma:

$$x(t) = A \cos \omega t \quad (2.23)$$

ou

$$x(t) = B \sin \omega t \quad (2.24)$$

sendo A e B as constantes de integração dependentes das condições de contorno do sistema em vibração livre. Como a Equação 2.22 é diferencial, homogênea e linear, a superposição de duas soluções também é uma solução. Dessa forma, a solução geral da equação do movimento para um sistema em vibração livre é definida sendo:

$$x(t) = A \cos \omega t + B \sin \omega t \quad (2.25)$$

Derivando a Equação 2.25, obtém-se a velocidade do sistema em função do tempo:

$$\dot{x}(t) = v(t) = -A\omega \sin \omega t + B\omega \cos \omega t \quad (2.26)$$

Com o tempo inicial igual a zero, o deslocamento inicial sendo  $x_0$  e a velocidade inicial descrita  $v_0$ , é possível determinar as constantes A e B. Logo:

$$A = x_0 \quad (2.27)$$

e

$$B = \frac{v_0}{\omega} \quad (2.28)$$

Finalmente, substituindo as constantes de integração na Equação 2.22, é possível deduzir a equação geral do movimento de um sistema oscilador simples como função do tempo:

$$x(t) = x_0 \cos \omega t + \frac{v_0}{\omega} \sin \omega t \quad (2.29)$$

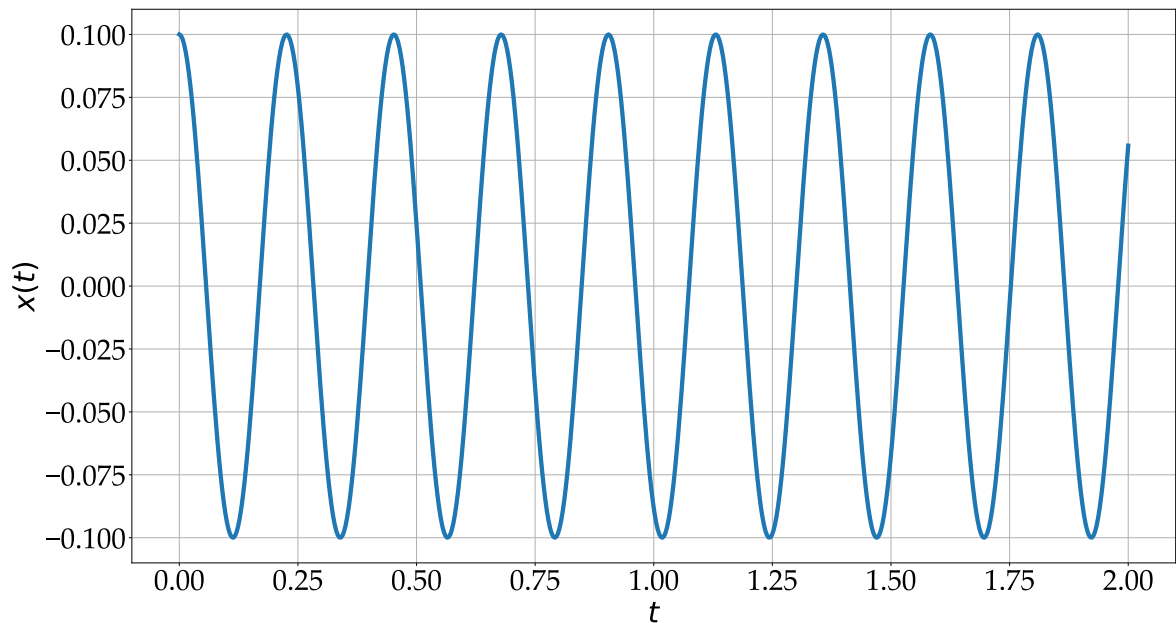
Visto que  $c = 0$ , logo  $\xi = 0$ , então o período de um oscilador simples é dado a partir da Equação 2.21:



$$T_d = \frac{2\pi}{\omega\sqrt{1-\xi^2}} = \frac{2\pi}{\omega\sqrt{1-0^2}} \implies T = \frac{2\pi}{\omega} \quad (2.30)$$

A Figura 5 mostra a resposta de um oscilador simples, também conhecido como harmônico devido a sua relação com as funções trigonométricas e a periodicidade delas.

Figura 5 – Resposta de um sistema não amortecido



Fonte: Autor (2022)

## 2.4 Sistemas com múltiplos Graus de Liberdade

### 2.4.1 Equação Matricial do Movimento

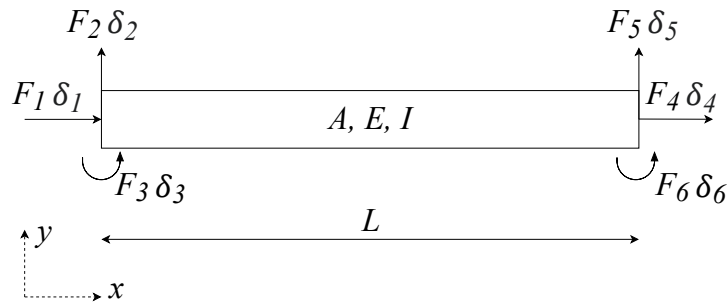
Quando análises de sistemas complexos são requeridas, como um pórtico ou uma treliça, é necessário discretizar a estrutura em mais nós e elementos, inevitavelmente aumentando o número de GDL, o que também aumenta a fidelidade da aproximação do modelo matemático proposto. Dessa forma, as equações diferenciais apresentadas para sistemas com um grau de liberdade não são suficientes para determinar a resposta dinâmica. Segundo Soriano (2014), define-se a equação de movimento em sua forma matricial para um sistema com múltiplos GDL como:

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{F(t)\} \quad (2.31)$$

## 2.5 Análise dinâmica de estruturas via Método da Rigidez Direta

A análise dinâmica de estruturas utilizando o Método da Rigidez Direta permite o cálculo do vetor de forças e das matrizes de rigidez, massa e amortecimento, possibilitando a resolução do sistema matricial mostrado na Equação 2.31 para estruturas formadas por elementos lineares, unidimensionais e com pequenos deslocamentos. Considerando o elemento de barra mostrando na Figura 6, é possível determinar as forças  $F_1$ ,  $F_2$ ,  $F_4$  e  $F_5$ , os momentos  $F_3$  e  $F_6$ , os deslocamentos lineares  $\delta_1$ ,  $\delta_2$ ,  $\delta_4$  e  $\delta_5$ , e os deslocamentos angulares  $\delta_3$  e  $\delta_6$  nas coordenadas nodais dispostas nas extremidades do elemento da viga.

Figura 6 – Segmento de viga mostrando as forças e deslocamentos nas coordenadas nodais



Fonte: Autor (2022)

A equação diferencial que descreve um pequeno deslocamento transversal na viga é dada por:

$$EI \frac{d^2 u}{dx^2} = M(x) \quad (2.32)$$

cujos  $M(x)$  é o momento fletor na seção  $x$  da viga e  $u$  o deslocamento transversal.

Para um elemento uniforme de viga, a Equação 2.32 pode ser escrita da forma:

$$EI \frac{d^4 u}{dx^4} = p(x) \quad (2.33)$$

desde

$$\frac{dM(x)}{dx} = V(x) \quad (2.34)$$

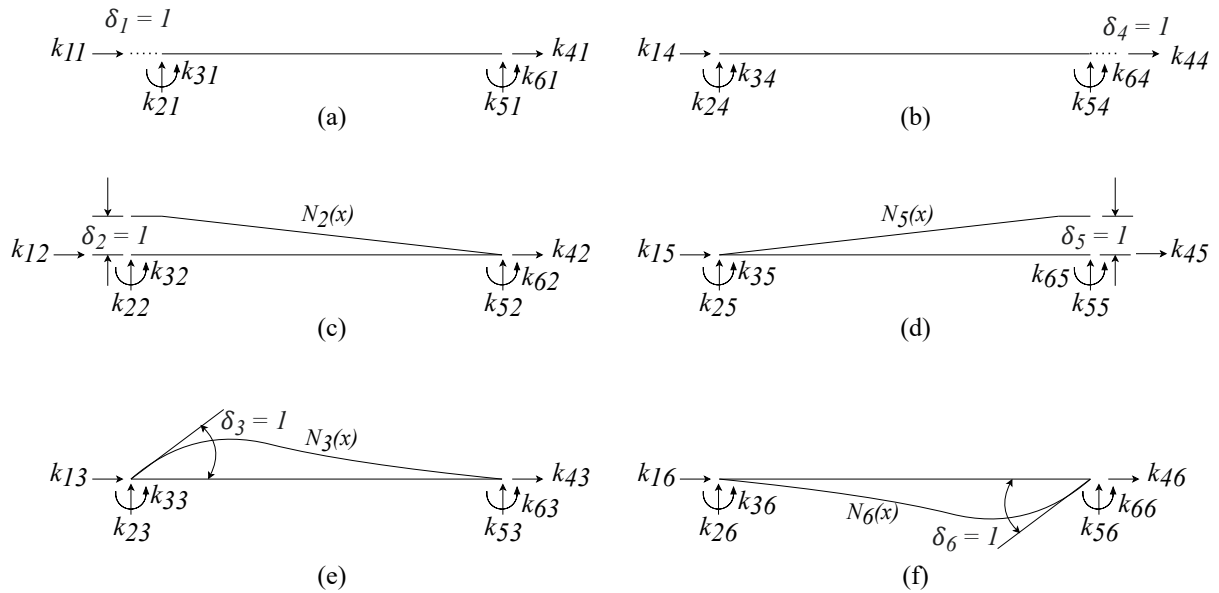
e

$$\frac{dV(x)}{dx} = p(x) \quad (2.35)$$

em que  $p(x)$  é a carga transversal por unidade de comprimento atuante sobre a viga e no sentido positivo do eixo  $y$ . Além disso,  $V(x)$  é a força cisalhante atuando na seção transversal.

Considerando-se  $k_{ij}$  como o coeficiente de rigidez, isto é, a força que atua na coordenada nodal  $i$  devido a um deslocamento unitário na coordenada  $j$  enquanto em todas as outras coordenadas nodais são mantidas com deslocamento zero. A Figura 7 mostra as configurações deformadas da viga e seus respectivos coeficientes de rigidez  $k$  devido a deslocamentos  $u$  unitários em cada uma das 6 coordenadas nodais.

Figura 7 – Segmento de viga mostrando as deflexões e deformações lineares devido a um deslocamento unitário em cada uma das coordenadas nodais



Fonte: Autor (2022)

### 2.5.1 Determinação dos coeficientes de rigidez devido às forças cortantes e fletoras

Para determinar os coeficientes  $k_{ij}$  mostrados na Figura 7 devido às parcelas de momento fletor e forças cortantes, toma-se  $p(x) = 0$ , uma vez que não há carregamentos externos na configuração mostrada na Figura 6, exceto as forças  $F_2, F_3, F_5$  e  $F_6$ . Nesse sentido, a Equação 2.33 é reduzida para:

$$\frac{d^4 u}{dx^4} = 0 \tag{2.36}$$

Realizando sucessivas integrações, chega-se à:

$$u = \frac{1}{6}C_1x^3 + \frac{1}{2}C_2x^2 + C_3x + C_4 \quad (2.37)$$

sendo  $C_1$ ,  $C_2$ ,  $C_3$  e  $C_4$  as constantes de integração determinada pelas condições de contorno.

Por exemplo, para a função de forma  $N_2(x)$  mostrada na Figura 7c, as condições de contorno são:

$$x = 0 \implies u(0) = 1 \implies \frac{du(0)}{dx} = 0 \quad (2.38)$$

$$x = L \implies u(L) = 0 \implies \frac{du(L)}{dx} = 0 \quad (2.39)$$

De maneira subsequente, é possível obter todas as constantes de integração da Equação 2.37. Logo, a função de forma  $N_2$  é dada por:

$$N_2(x) = 1 - 3\left(\frac{x}{L}\right)^2 + 2\left(\frac{x}{L}\right)^3 \quad (2.40)$$

sendo  $N_2(x)$  utilizado ao invés de  $u(x)$  para corresponder à condição  $\delta_2 = 1$  imposta no elemento de barra. Realizando procedimentos análogos para a determinação de  $N_3(x)$ ,  $N_5(x)$  e  $N_6(x)$ , obtém-se as seguintes equações:

$$N_3(x) = x\left(1 - \frac{x}{L}\right)^2 \quad (2.41)$$

$$N_5(x) = 3\left(\frac{x}{L}\right)^2 - 2\left(\frac{x}{L}\right)^3 \quad (2.42)$$

$$N_6(x) = \frac{x^2}{L}\left(\frac{x}{L} - 1\right) \quad (2.43)$$

Sendo assim, a deflexão total  $u(x)$  na seção  $x$  devido a deslocamentos arbitrários nas coordenadas nodais do elemento de barra é dada pela superposição das funções de forma  $N_2(x)$ ,  $N_3(x)$ ,  $N_5(x)$  e  $N_6(x)$ :

$$u(x) = N_2(x)\delta_2 + N_3(x)\delta_3 + N_5(x)\delta_5 + N_6(x)\delta_6 \quad (2.44)$$

As funções de forma  $N_2(x)$ ,  $N_3(x)$ ,  $N_5(x)$  e  $N_6(x)$  e os deslocamentos unitários nas coordenadas nodais da viga podem ser utilizados para determinar os coeficientes de rigidez com o auxílio do Princípio dos Trabalhos Virtuais. Quando aplica-se tal princípio para um sistema elástico em equilíbrio, como o elemento de barra discutido, o trabalho realizado pelas forças externas é igual ao trabalho realizado pelas forças internas durante

um deslocamento virtual. Por exemplo, o trabalho externo  $W_e$  é definido pela equação abaixo:

$$W_e = k_{23}\delta_2 \quad (2.45)$$

Este trabalho é igual ao trabalho executado pela força elástica, representada pelo coeficiente  $k_{23}$ , durante o deslocamento virtual  $\delta_2$ . Considerando o trabalho performedo pelo momento fletor, adquire-se:

$$W = \int_0^L M(x)d\theta \quad (2.46)$$

em que  $M(x)$  é o momento fletor atuante na seção  $x$  e  $d\theta$  é o deslocamento angular infinitesimal na mesma seção.

Para o deslocamento virtual em consideração, a deflexão transversal da viga é dada pela Equação 2.41, que está diretamente relacionada com o momento fletor através da equação diferencial Equação 2.32. Tomando a segunda derivada de  $N_3(x)$  e realizando a substituição na equação diferencial, tem-se:

$$EI \frac{d^2 N_3(x)}{dx^2} = EIN_3'' = M(x) \quad (2.47)$$

A deflexão angular infinitesimal  $d\theta$  produzida durante o deslocamento virtual tem relação com a deflexão transversal resultante  $N_2(x)$ . Logo:

$$\frac{d\theta}{dx} = \frac{d^2 N_2(x)}{dx^2} = EIN_2'' \quad (2.48)$$

Equacionado o trabalho externo  $W_e$  com o deslocamento interno  $W$  e aplicando as duas equações acima, o coeficiente de rigidez torna-se:

$$k_{23} = EI \int_0^L N_2''(x)N_3''(x)dx \quad (2.49)$$

Em termos gerais, o coeficiente de rigidez  $k_{ij}$  para um elemento de barra pode ser expresso como:

$$k_{ij} = EI \int_0^L N_i''(x)N_j''(x)dx \quad (2.50)$$

É conveniente dispor os coeficientes  $k_{ij}$  em formato matricial:

$$[k] = \begin{bmatrix} k_{22} & k_{23} & k_{25} & k_{26} \\ k_{32} & k_{33} & k_{35} & k_{36} \\ k_{52} & k_{53} & k_{55} & k_{56} \\ k_{62} & k_{63} & k_{65} & k_{66} \end{bmatrix} = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \quad (2.51)$$

Fazendo uso da Lei de Hooke em sua forma matricial, é possível encontrar as forças  $F_2$ ,  $F_3$ ,  $F_5$  e  $F_6$  a partir dos deslocamentos  $\delta_2$ ,  $\delta_3$ ,  $\delta_5$  e  $\delta_6$ , onde:

$$\begin{bmatrix} F_2 \\ F_3 \\ F_5 \\ F_6 \end{bmatrix} = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \begin{bmatrix} \delta_2 \\ \delta_3 \\ \delta_5 \\ \delta_6 \end{bmatrix} \quad (2.52)$$

## 2.5.2 Determinação dos coeficientes de rigidez devido às forças axiais

A inclusão de forças axiais na matriz de rigidez de um elemento de barra exige a determinação dos coeficientes de rigidez devidos aos esforços axiais. Ao observar a Figura 6 como referência, nota-se que as forças axiais atuantes são  $F_1$  e  $F_4$ , que produzem deslocamentos axiais  $\delta_1$  e  $\delta_4$  nos nós do elemento. Para uma viga prismática e uniforme, aplica-se a Lei de Hook com a finalidade de obter a relação entre deslocamentos e forças. Nesse sentido, o deslocamento  $\delta_1$  produzido por uma força  $F_1$  atuando no nó 1 enquanto todos os demais deslocamentos são mantidos em zero é descrito por:

$$\delta_1 = \frac{F_1 L}{AE} \quad (2.53)$$

ou

$$k_{11} = \frac{F_1}{\delta_1} = \frac{AE}{L} \quad (2.54)$$

O equilíbrio de um segmento de viga requer uma força  $k_{21}$  de mesma magnitude e sentido contrário à  $k_{11}$ , logo:

$$k_{21} = -k_{11} = -\frac{AE}{L} \quad (2.55)$$

De maneira análoga, define-se:

$$k_{22} = \frac{AE}{L} \quad (2.56)$$

e

$$k_{12} = -\frac{AE}{L} \quad (2.57)$$

Como todos os demais carregamentos externos são nulos, os demais coeficientes de rigidez também são. Além disso, é conveniente mostrar os coeficientes de rigidez devido às forças axiais em forma matricial, como:

$$\begin{bmatrix} F_1 \\ F_4 \end{bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_4 \end{bmatrix} \quad (2.58)$$

### 2.5.3 Matriz de rigidez de um elemento de barra

Para criar a matriz de rigidez para um elemento de barra submetido a efeitos cortantes, fletores e axiais simultaneamente, combina-se a matriz de rigidez produzida por efeitos fletores e cortantes com a matriz de rigidez produzida por efeitos axiais (CRAIG; KURDILA, 2006). Sendo assim, o sistema matricial torna-se:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix} = \frac{EI}{L^3} \begin{bmatrix} \frac{AL^2}{I} & 0 & 0 & -\frac{AL^2}{I} & 0 & 0 \\ 0 & 12 & 6L & 0 & -12 & 6L \\ 0 & 6L & 4L^2 & 0 & -6L & 2L^2 \\ -\frac{AL^2}{I} & 0 & 0 & \frac{AL^2}{I} & 0 & 0 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ 0 & 6L & 2L^2 & 0 & -6L & 4L^2 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{bmatrix} \quad (2.59)$$

Ou na forma condensada:

$$\{F\} = [k]\{\delta\} \quad (2.60)$$

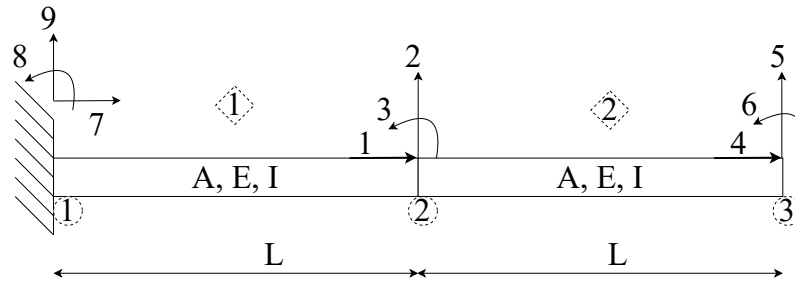
### 2.5.4 Matriz de rigidez da estrutura

A matriz de rigidez da estrutura pode ser determinada a partir da utilização do MRD. São passos do método:

1. Dividir a estrutura em elementos de viga;
2. Identificar os nós entre os elementos e numerar consecutivamente as coordenadas nodais que não estão com movimentações restringidas;
3. Enumerar os nós restringidos;
4. Obter sistematicamente a matriz de rigidez de cada elemento que compõe a estrutura;

5. Unir as contribuições nodais de rigidez que cada elemento fornece para a estrutura, respeitando o sistema de coordenadas nodais e suas respectivas restrições de deslocamento.

Figura 8 – Viga cantilever segmentada em 2 elementos de viga com coordenadas nodais numeradas



Fonte: Autor (2022)

O termo "método de rigidez direta" está ligado diretamente à forma em que a construção da matriz de rigidez da estrutura acontece. O coeficiente de rigidez  $k_{ij}$  do sistema pode ser obtido somando todas as contribuições nodais correspondentes às coordenadas nodais livres em questão. Tomando como exemplo o coeficiente  $k_{33}$ , é necessário adicionar as contribuições dos coeficientes de rigidez dos elementos 1 e 2 mostrados na Figura 8. Procedendo com o exemplo mostrado na Figura 8, obtém-se as matrizes de rigidez de cada um dos elementos que compõe a viga cantilever:

$$[k^{(1)}] = \begin{bmatrix} k_{77} & k_{78} & k_{79} & k_{71} & k_{72} & k_{73} \\ k_{87} & k_{88} & k_{89} & k_{81} & k_{82} & k_{83} \\ k_{97} & k_{98} & k_{99} & k_{91} & k_{92} & k_{93} \\ k_{17} & k_{18} & k_{19} & k_{11} & k_{12} & k_{13} \\ k_{27} & k_{28} & k_{29} & k_{21} & k_{22} & k_{23} \\ k_{37} & k_{38} & k_{39} & k_{31} & k_{32} & k_{33} \end{bmatrix} \quad (2.61)$$

$$[k^{(2)}] = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \quad (2.62)$$

Note que os índices de cada  $k_{ij}$  são referentes às coordenadas nodais e aos respectivos efeitos provocados pelas demais coordenadas, seguindo a definição: a força que atua na



coordenada nodal  $i$  devido a um deslocamento unitário na coordenada  $j$  enquanto em todas as outras coordenadas nodais são mantidas com deslocamento zero. Sendo assim, a matriz de rigidez da estrutura é encontrada a partir da soma das contribuições nodais de cada elemento em suas respectivas coordenadas, desconsiderando as coordenadas nodais restringidas pelo engaste. Dessa forma, tem-se a matriz de rigidez da estrutura como:

$$[K] = \begin{bmatrix} k_{11}^{(1)} + k_{11}^{(2)} & k_{12}^{(1)} + k_{12}^{(2)} & k_{13}^{(1)} + k_{13}^{(2)} & k_{14}^{(2)} & k_{15}^{(2)} & k_{16}^{(2)} \\ k_{21}^{(1)} + k_{21}^{(2)} & k_{22}^{(1)} + k_{22}^{(2)} & k_{23}^{(1)} + k_{23}^{(2)} & k_{24}^{(2)} & k_{25}^{(2)} & k_{26}^{(2)} \\ k_{31}^{(1)} + k_{31}^{(2)} & k_{32}^{(1)} + k_{32}^{(2)} & k_{33}^{(1)} + k_{33}^{(2)} & k_{34}^{(2)} & k_{35}^{(2)} & k_{36}^{(2)} \\ k_{41}^{(2)} & k_{42}^{(2)} & k_{43}^{(2)} & k_{44}^{(2)} & k_{45}^{(2)} & k_{46}^{(2)} \\ k_{51}^{(2)} & k_{52}^{(2)} & k_{53}^{(2)} & k_{54}^{(2)} & k_{55}^{(2)} & k_{56}^{(2)} \\ k_{61}^{(2)} & k_{62}^{(2)} & k_{63}^{(2)} & k_{64}^{(2)} & k_{65}^{(2)} & k_{66}^{(2)} \end{bmatrix} \quad (2.63)$$

Sendo assim, a matriz de rigidez da estrutura relaciona as forças e os deslocamentos nas coordenadas nodais globais da mesma forma que a matriz de rigidez do elemento relaciona as forças e os deslocamentos nas coordenadas nodais locais.

### 2.5.5 Matriz de massa da estrutura

É possível obter os coeficientes de massa, correspondentes às coordenadas nodais de um elemento de barra submetida a efeitos axiais, cortantes e fletores, de uma maneira similar à determinação dos coeficientes de rigidez. Primeiramente, define-se os coeficientes  $m_{ij}$  como a força nodal na coordenada  $i$  devido a uma aceleração unitária na coordenada nodal  $j$  enquanto todas as outras coordenadas permanecem com a aceleração igual a zero.

Logo, os coeficientes de massa suscetíveis aos esforços cortantes e fletores é definido da seguinte forma:

$$m_{ij} = \int_0^L \bar{m}(x) N_i(x) N_j(x) dx \quad (2.64)$$

sendo  $\bar{m}(x)$  a função que descreve a distribuição de massa por unidade de comprimento.

Para os efeitos axiais, os coeficientes de massa são obtidos como:

$$m_{ij} = \int_0^L \bar{m}(x) u_i(x) u_j(x) dx \quad (2.65)$$

em que  $u_i(x)$  e  $u_j(x)$  são as funções que descrevem o deslocamento da seção transversal  $x$  da viga para um dado deslocamento nodal.

Dessa forma, a matriz consistente de massa para o elemento de barra mostrando na Figura 6 com a massa distribuída igualmente nos nós é escrita da seguinte forma (RAO, 2018):

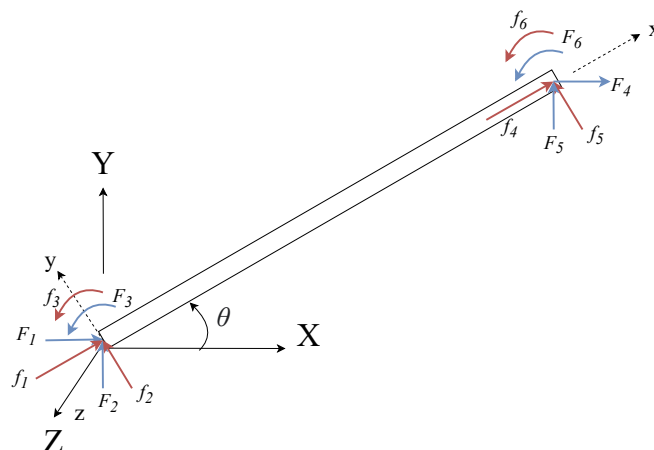
$$[M] = \frac{\rho AL}{420} \begin{bmatrix} 140 & 0 & 0 & 70 & 0 & 0 \\ 0 & 156 & 22L & 0 & 54 & -13L \\ 0 & 22L & 4L^2 & 0 & 13L & -3L^2 \\ 70 & 0 & 0 & 140 & 0 & 0 \\ 0 & 54 & 13L & 0 & 156 & -22L \\ 0 & -13L & -3L^2 & 0 & -22L & 4L^2 \end{bmatrix} \quad (2.66)$$

A montagem da matriz de massa da estrutura é similar à montagem da matriz de rigidez da estrutura, pois utiliza-se da mesma estratégia de somar as contribuições das coordenadas nodais que estão sem restrição de deslocamento.

### 2.5.6 Transformação de coordenadas

As matrizes de rigidez e de massa do elemento de barra têm como referência as coordenadas nodais fixadas nos nós deste elemento. Tais coordenadas são chamadas de *locais*, enquanto as coordenadas referentes a estrutura como um todo são chamadas de *globais*. A Figura 9 mostra um elemento de barra com  $f_1, \dots, f_6$  forças referenciadas no sistema local de coordenadas  $(x, y, z)$  e  $F_1, \dots, F_6$  forças referenciadas no sistema global de coordenadas  $(X, Y, Z)$ .

Figura 9 – elemento de barra ilustrando forças nodais  $f_i$  em coordenadas locais  $(x, y, z)$  e forças nodas  $F_i$  em coordenadas globais  $(X, Y, Z)$



Fonte: Autor (2022)

O objetivo da transformação de coordenadas é rotacionar matrizes referenciadas em coordenadas locais para matrizes referenciadas em coordenadas globais. Essa mudança é necessária para que todos os elementos da estrutura estejam no mesmo sistema de referências. Dessa forma, as matrizes de rigidez e massa, originalmente em coordenadas

locais, tornam-se compatíveis para a montagem de um sistema de matrizes que representam o comportamento da estrutura em sua totalidade.

Sendo assim, a rotação das forças mostradas na Figura 9 são realizadas como:

$$f_1 = F_1 \cos(\theta) + F_2 \sin(\theta) \quad (2.67)$$

$$f_2 = -F_1 \sin(\theta) + F_2 \cos(\theta) \quad (2.68)$$

$$f_3 = F_3 \quad (2.69)$$

$$f_4 = F_4 \cos(\theta) + F_5 \sin(\theta) \quad (2.70)$$

$$f_5 = -F_4 \sin(\theta) + F_5 \cos(\theta) \quad (2.71)$$

$$f_6 = F_6 \quad (2.72)$$

É conveniente reorganizar as equações acima em notação matricial, logo:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix} \quad (2.73)$$

ou na formatação condensada:

$$\{f\} = [T]\{F\} \quad (2.74)$$

Nesse sentido, torna-se trivial transitar entre os sistemas de coordenadas. Ao se repetir o mesmo procedimento anterior, é obtida uma relação entre os deslocamentos nodais nas coordenadas locais e coordenadas globais:

$$\begin{bmatrix} \bar{\delta}_1 \\ \bar{\delta}_2 \\ \bar{\delta}_3 \\ \bar{\delta}_4 \\ \bar{\delta}_5 \\ \bar{\delta}_6 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{bmatrix} \quad (2.75)$$

ou em notação matricial:

$$\{\bar{\delta}\} = [T]\{\delta\} \quad (2.76)$$

Utilizando a Lei de Hook em sua notação matricial, para o elemento de barra em coordenadas locais, tem-se:

$$[f] = [\bar{k}][\bar{\delta}] \implies [T]\{F\} = [\bar{k}][T]\{\delta\} \quad (2.77)$$

então

$$\{F\} = [T]^{-1}[\bar{k}][T]\{\delta\} \quad (2.78)$$

ou

$$\{F\} = [k]\{\delta\} \quad (2.79)$$

em que

$$[k] = [T]^T[\bar{k}][T] \quad (2.80)$$

é a matriz de rigidez de um elemento de barra referenciada no sistema de coordenadas globais. Note que devido à ortogonalidade da matriz de rotação, obtém-se  $[T]^{-1} = [T]^T$ . É possível obter a matriz de massa de maneira semelhante à mostrada na Equação 2.80:

$$[m] = [T]^T[\bar{m}][T] \quad (2.81)$$

## 2.6 Análise Modal

Quando o amortecimento é desprezado, o vetor de carregamentos é considerado nulo e o sistema é colocado em movimento devido às condições iniciais de deslocamento e velocidade, implica que o sistema está em vibração livre não amortecida, tornando-se possível realizar a Análise Modal com o objetivo de descobrir suas Frequências Naturais e seus Modos de Vibração (BRASIL; SILVA, 2015).

### 2.6.1 Frequências Naturais e Modos de Vibração

O problema de vibração livre sem amortecimento é regido pela Equação 2.82.

$$[M]\{\ddot{x}\} + [K]\{x\} = 0 \quad (2.82)$$

sua solução é da forma:

$$\{x\} = \{a\} \sin(\omega t - \alpha) \quad (2.83)$$

sendo  $\alpha$  o ângulo de fase e o vetor  $\{a\}$  contendo as amplitudes de movimento das coordenadas nodais. Substituindo a Equação 2.83 na Equação 2.82, obtém-se:

$$-\omega^2[M]\{a\} \sin(\omega t - \alpha) + [K]\{a\} \sin(\omega t - \alpha) = 0 \quad (2.84)$$

simplificando os termos comuns, tem-se:

$$[[K] - \omega^2[M]]\{a\} = 0 \quad (2.85)$$

A formulação descrita pela Equação 2.85 é um problema matemático conhecido como problema de autovalores e autovetores. Para uma solução não trivial, o determinante da matriz  $[[K] - \omega^2[M]]$  deve ser igual a zero.

$$\det([[K] - \omega^2[M]]) = 0 \quad (2.86)$$

resultando em uma função polinomial de grau  $n$  na variável  $\omega^2$ , onde  $n$  é igual ao número de GDL da estrutura. A função polinomial é conhecida como a equação característica do sistema.  $n$  soluções  $w_i$  são reais e positivas e são as Frequências Naturais do sistema estrutural. É costumeiro adotar  $w_1$  como a menor frequência natural, e  $w_n$  a maior. Para cada valor  $w^2$ , soluciona-se a Equação 2.85, encontrando os respectivos autovetores, ou modos de vibração  $\{a\}_i$ , ao arbitrar uma de suas componentes. Uma forma de realizar isso é fazer a primeira coordenada de cada modo ser unitária. Com esse valor admitido, as outras coordenadas poderão ser determinadas univocamente. Sendo assim, é possível determinar os  $n$  modos de vibração e colocá-los numa matriz modal  $n \times n$ , cuja as colunas são os  $n$  modos de vibração livre, não amortecidos, normalizados:

$$[\Phi] = [\{a_1\} \quad \{a_2\} \quad \cdots \quad \{a_n\}] = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{bmatrix} \quad (2.87)$$

### 2.6.2 Propriedade de Ortogonalidade dos Modos de Vibração

Os modos de vibração possuem a propriedade de serem ortogonais com relação às matrizes de rigidez e de massa. Logo:

$$\phi_{ij} = \frac{a_{ij}}{\sqrt{\{a\}_i^T [M] \{a\}_j}} \quad (2.88)$$

cujo  $\phi_{ij}$  é o componente  $i$  normalizado do modo de vibração  $j$ . Para autovetores normalizados, a condição de ortogonalidade é:

$$\{\phi\}_i^T [M] \{\phi\}_j = \begin{cases} 0 & \text{para } i \neq j \\ 1 & \text{para } i = j \end{cases} \quad (2.89)$$

Outra condição de ortogonalidade pode ser obtida através da Equação 2.85 e do modo vibração normalizado  $j$  como:

$$[K] \{\phi\}_j = \omega_j^2 [M] \{\phi\}_j \quad (2.90)$$

Pré-multiplicando a Equação 2.90 por  $\{\phi\}_i^T$  obtém-se outra condição de ortogonalidade:

$$\{\phi\}_i^T [K] \{\phi\}_j = \begin{cases} 0 & \text{para } i \neq j \\ w_j^2 & \text{para } i = j \end{cases} \quad (2.91)$$

Com a finalidade de testar as condições de ortogonalidade mencionadas acima, faz-se:

$$[\Phi]^T [M] [\Phi] = [I] \quad (2.92)$$

e

$$[\Phi]^T [K] [\Phi] = [\Omega] \quad (2.93)$$

sendo  $[I]$  a matriz identidade e  $[\Omega]$  a matriz diagonal contendo os autovalores  $w_i^2$  do sistema.

## 2.7 Método da Variação da Flexibilidade Modal

A teoria do Modelos Incompletos de Estruturas Dinâmicas foi apresentada por Berman e Flannelly (1971). No trabalho, os autores mostraram a dedução da matriz de

flexibilidade modal a partir das características modais da estrutura, isto é, a partir dos modos de vibração e das frequências naturais.

Pandey e Biswas (1994) constataram que o dano em estruturas altera suas propriedades modais, como os modos de vibração e as frequências naturais. Além disso, concluíram que a deterioração estrutural afeta a rigidez, flexibilidade, massa e o amortecimento. No trabalho, os pesquisadores propuseram uma nova abordagem para o método apresentado por Berman e Flannelly (1971) e analisaram que variações na matriz de flexibilidade podem ser utilizadas para detectar e localizar dano em estruturas. Carrillo (2004) em sua tese doutoral fez uso da formulação proposta por Pandey e Biswas (1994), e comparou o método proposto por eles com outras técnicas analíticas, também baseadas em MEF.

A Equação 2.90 pode ser reescrita como (CARRILLO, 2004):

$$[K][\Phi] = [M][\Phi][\Omega] \quad (2.94)$$

Pré-multiplicando a equação acima por  $[K]^{-1} = [F]$  e pós-multiplicando por  $[\Omega]^{-1}$ , obtém-se:

$$[\Phi][\Omega]^{-1} = [F][M][\Phi] \quad (2.95)$$

descrevendo o problema de autovalores e autovetores em termos de flexibilidade.

A Equação 2.92 pode ser escrita como:

$$[\Phi]^T[M] = [\Phi]^{-1} \quad (2.96)$$

ou

$$[M][\Phi] = [\Phi]^{T-1} \quad (2.97)$$

pois devido ao fato de que a matriz de formas modais normalizada pela massa para um sistema com  $n$  GDL consiste em  $n$  vetores modais linearmente independentes, a matriz em questão é não-singular e pode ser invertida.

Pós-multiplicando a Equação 2.94 por  $[\Phi]^{-1}$  e utilizando a propriedade descrita pela Equação 2.96, tem-se:

$$[K] = [M][\Phi][\Omega][\Phi]^T[M] = [M]\left(\sum_{i=1}^{GDL} \omega_i^2 \{\phi\}_i \{\phi\}_i^T\right)[M] \quad (2.98)$$

em que a matriz de rigidez é expandida em termos de autovalores e autovetores. Substituindo a Equação 2.96 na Equação 2.95 e pós-multiplicando por  $[\Phi]^T$ , adquire-se:

$$[F] = [\Phi][\Omega]^{-1}[\Phi]^T = \sum_{i=1}^{GDL} \frac{1}{w_i^2} \{\phi\}_i \{\phi\}_i^T \quad (2.99)$$

em que a matriz de flexibilidade é expandida em termos de autovalores e autovetores.

Analisando a Equação 2.98, nota-se que a contribuição modal para a matriz de rigidez aumenta à medida que as frequências naturais aumentam. Para obter uma estimativa acurada da rigidez, são necessários muitos modos de vibração em frequências elevadas. Isso torna-se inviável para experimentos práticos, pois necessita-se de uma quantidade massiva de energia para excitar estruturas em frequências elevadas com a finalidade de medir os modos de vibração para alta-frequência.

Por outro lado, observa-se que a aproximação da matriz de flexibilidade descrita pela Equação 2.99 rapidamente converge, isto é, as contribuições modais para a flexibilidade decrescem à medida que as frequências aumentam. Sendo assim, com algumas frequências baixas, é possível obter uma boa estimativa da matriz de flexibilidade, o que viabiliza a aplicação prática da técnica.

Como a presença de dano reduz a rigidez de uma estrutura, isso inevitavelmente produz um aumento na flexibilidade dela. Nesse sentido, é possível utilizar a variação da matriz de flexibilidade obtida através dos parâmetros modais para detectar, localizar e mensurar dano. Logo, a Equação 2.100 traduz a variação da matriz de flexibilidade modal ( $[\Delta F]$ ) entre um estado intacto e um estado danificado de uma mesma estrutura.

$$[\Delta F] = [F] - [F^D] \quad (2.100)$$

sendo cada coluna da matriz  $[\Delta F]$  representa a variação das flexibilidades nodais devido a uma força unitária no respectivo grau de liberdade. Para cada grau de liberdade  $j$ , define-se  $\bar{\xi}_j$  como o valor máximo absoluto dos elementos correspondentes àquela coluna. Isto é:

$$\bar{\xi}_j = \max([\Delta F_j]) \quad (2.101)$$

Para  $k$  casos de estruturas danificadas, é conveniente dispor todos os coeficientes  $\bar{\xi}_j$  em uma matriz, cuja as linhas representam o  $k$ -ésimo caso da estrutura danificada e as colunas descrevem a  $j$ -ésima variação máxima da flexibilidade modal no grau de liberdade em questão:

$$[\Xi] = \begin{bmatrix} \{\xi_1\} \\ \{\xi_2\} \\ \vdots \\ \{\xi_k\} \end{bmatrix} = \begin{bmatrix} \bar{\xi}_{11} & \bar{\xi}_{12} & \cdots & \bar{\xi}_{1j} \\ \bar{\xi}_{21} & \bar{\xi}_{22} & \cdots & \bar{\xi}_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\xi}_{k1} & \bar{\xi}_{k2} & \cdots & \bar{\xi}_{kj} \end{bmatrix} \quad (2.102)$$



## 3 Sistemas Computacionais Inteligentes

### 3.1 Introdução

Hinton, Osindero e Teh (2006) mostraram como treinar uma rede neural profunda capaz de reconhecer dígitos escritos à mão com uma precisão acima de 96%. Os autores batizaram essa técnica como "Aprendizado Profundo". Naquela época era considerado impossível treinar uma rede neural profunda, então grande parte dos pesquisadores abandonaram essa ideia por volta dos anos 90. Entretanto, o trabalho publicado pelos autores reviveu o interesse da comunidade acadêmica e rapidamente novos artigos foram desenvolvidos, mostrando que o Aprendizado Profundo era capaz de alcançar lugares que as técnicas de Aprendizado de Máquina não conseguiram.

Atualmente, modelos estatísticos de AP e AM conquistaram as mais variadas indústrias e são peças fundamentais no funcionamento do mundo contemporâneo. A título de exemplificação, tais técnicas são aplicadas em:

1. Classificação de resultados de buscas na internet;
2. Reconhecimento de fala;
3. Recomendação de produtos;
4. Veículos Autônomos;
5. Previsão de Demanda.

### 3.2 Aprendizado de Máquina

Quando a maioria das pessoas ouvem o termo "Aprendizado de Máquina", elas imaginam um robô aprendendo a fazer algo. Mas o termo não está limitado a isso, ele abrange muitas áreas do conhecimento de diferentes maneiras. O AM foi ganhando espaço na indústria no início dos anos 90, quando o filtro de *spam* foi criado. Essencialmente, o filtro separava *e-mails* de conteúdo duvidoso daqueles que realmente eram confiáveis, ou seja, realizava uma tarefa de classificação.

#### 3.2.1 O que é Aprendizado de Máquina?

Aprendizado de Máquina é a ciência de programar computadores para que eles possam aprender a partir de dados (GÉRON, 2019). Tomando como exemplo o filtro

de *spam* citado anteriormente, ele é um programa que é capaz de aprender a partir de exemplos de *e-mails spam* e *e-mails* confiáveis e classificar novos *e-mails*. Esse conjunto de dados utilizado para o treino do programa (modelo) é chamado de conjunto de treinamento. Cada exemplo do conjunto de treino é chamado de instância. Neste caso, a tarefa T é rotular novos e-mails, isto é, classificar se eles são *spam* ou não. A experiência E é adquirida a partir do treinamento e a métrica M precisa ser definida para medir a capacidade de classificação do modelo. Por exemplo, é possível utilizar a razão de *e-mails* classificados corretamente para verificar a performance do modelo criado. Essa métrica de performance é chamada de acurácia.

### 3.2.2 Por que utilizar Aprendizado de Máquina?

Algoritmos de AM são utilizados na identificação de padrões e assim na generalização da tarefa desempenhada pelo modelo em questão. Caso uma nova instância seja adicionada ao conjunto de dados original, o modelo irá interpretá-la corretamente, sem a necessidade de criar uma nova regra específica.

Em certos cenários, um humano é capaz de identificar se um *e-mail* é maligno ou não ao analisar palavras tendenciosas ou que se refiram a temas delicados. Mas diante da vastidão de *e-mails* que são trocados diariamente, seria completamente impossível verificar todos e classificá-los. Nesse sentido, técnicas de AM podem ser utilizadas para a automação de atividades complexas e/ou extensas que humanos não conseguiriam ou levariam muito tempo para executar.

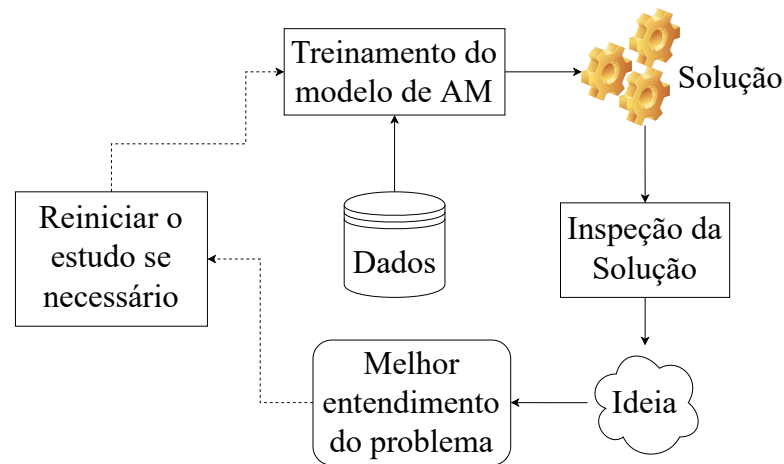
Finalmente, algoritmos de AM podem ajudar os seres humanos a aprenderem a partir dos resultados promovidos pelo modelo. Diversas empresas utilizam de seus modelos estatísticos para auxiliar os humanos a tomarem decisões de negócio. A Figura 10 ilustra o processo de tomada de decisão feita por humanos baseado em conclusões originadas de modelos de AM. O humano atua na etapa da "ideia", interpretando os resultados advindos do modelo, cabendo a ele realizar as ponderações necessárias.

### 3.2.3 Aprendizado Supervisionado

Em algoritmos de aprendizado supervisionado, o conjunto de dados para treinamento inclui as soluções desejadas, chamadas de rótulos. A Figura 11 exemplifica a ideia de um algoritmo supervisionado para detecção de *e-mails spam*.

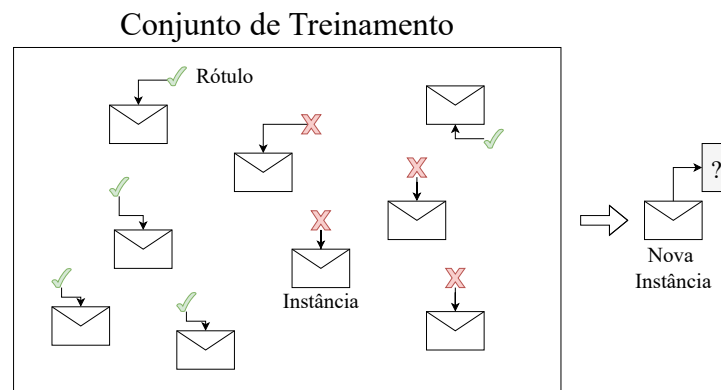
Uma típica tarefa de algoritmos de aprendizado supervisionado é a classificação. O filtro de *spam* é um ótimo exemplo de um algoritmo classificador, pois ele é treinando para detectar a classe, *spam* ou *ham*, de uma nova instância.

Figura 10 – Diagrama de como AM pode ajudar humanos a tomar decisões



Fonte: Autor (2022)

Figura 11 – Conjunto de treinamento para aprendizado supervisionado de um algoritmo classificador de *e-mails spam*



Fonte: Autor (2022)

Outra tarefa comum para esse tipo de algoritmo é a regressão, que consiste em realizar a predição de um valor numérico. Para realizar o treinamento de um *regressor*, é necessário fornecer exemplos numéricos, incluindo o rótulo associado. Por exemplo, para um sistema que prevê o preço de casas, é necessário fornecer, como conjunto de treinamento, diversos valores numéricos referentes às características que influenciam o preço do imóvel. Neste caso, o conjunto de treinamento seria formado por características como a área da residência, o número de quartos e banheiros, bairro, cidade, país e o próprio preço (rótulo) do imóvel.

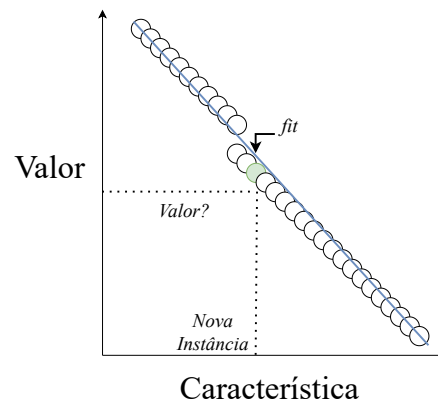
A Figura 12 mostra a tarefa de regressão executada em um conjunto de treinamento com apenas uma característica numérica. Para encontrar a reta *fit*, o modelo proposto

realiza uma regressão linear, descrita como:

$$\hat{y} = \theta_0 + \theta_1 x_1 \quad (3.1)$$

sendo  $\hat{y}$  o valor predito,  $x_1$  a característica,  $\theta_0$  o viés ou intercepto da reta, e  $\theta_1$  o peso ou coeficiente angular da reta.

Figura 12 – Regressão em algoritmos supervisionados



Fonte: Autor (2022)

### 3.2.4 Árvores de Decisão

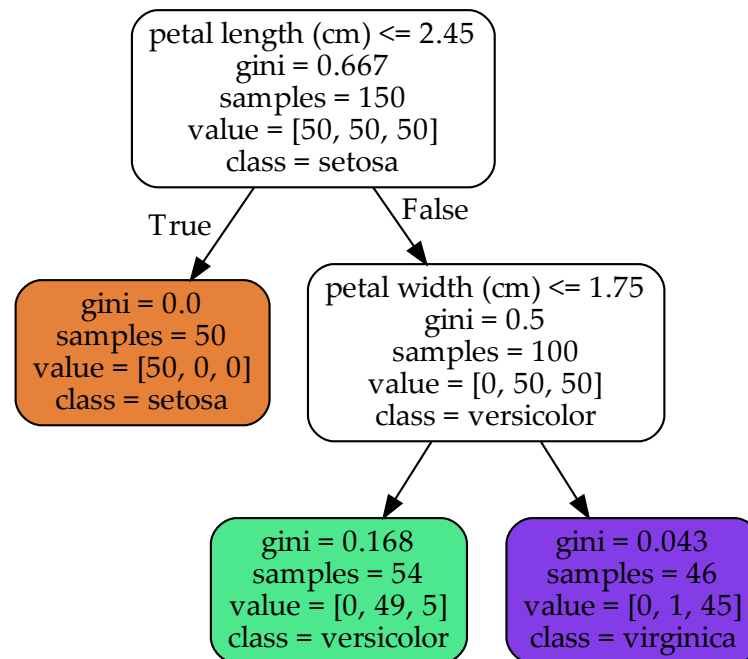
Árvores de Decisão são algoritmos de AM extremamente versáteis e capazes de realizar tarefas de classificação e regressão em complexos conjuntos de dados. Elas também são componentes fundamentais de Florestas Aleatórias, outro algoritmo de AM.

#### 3.2.4.1 Treinamento

Para compreender as Árvores de Decisão, é conveniente observar como o algoritmo é treinado. A Figura 13 ilustra as ramificações da árvore treinada e suas respectivas fronteiras de decisões para o *iris dataset* (ZHONG; FUKUSHIMA, 2007).

#### 3.2.4.2 Realizando decisões

Supondo que o algoritmo recebe uma nova instância de flor de iris, ele deve classificá-la em sua respectiva classe. O processo de classificação inicia no nó raiz (profundidade 0): o nó questiona se o comprimento da pétala é menor do que 2,45 cm. Se sim, move-se para o nó da esquerda (profundidade 1); caso contrário, para a direita. O processo de classificação continua até a instância estar em um nó "folha", isto é, o último nó de uma ramificação. O mesmo algoritmo de tomada de decisão de acordo com as características da classe é realizado para qualquer caso de nova instância.

Figura 13 – Árvore de Decisão para o conjunto de treinamento *iris*

Fonte: Autor (2022)

O atributo *samples* do nó representa a quantidade de instâncias treinadas que se aplicam àquela classe. O atributo *value* descreve quanta instâncias de treinamento de cada classe estão naquele nó. Finalmente, o atributo *gini* quantifica a "impureza" do nó, ou seja, um nó puro tem  $gini = 0$  se todas as instâncias de treinamento são aplicáveis àquele nó. Por exemplo, como a profundidade 1 à esquerda detém apenas flores do tipo Iris-Setosa, logo seu *gini score* é igual a zero.

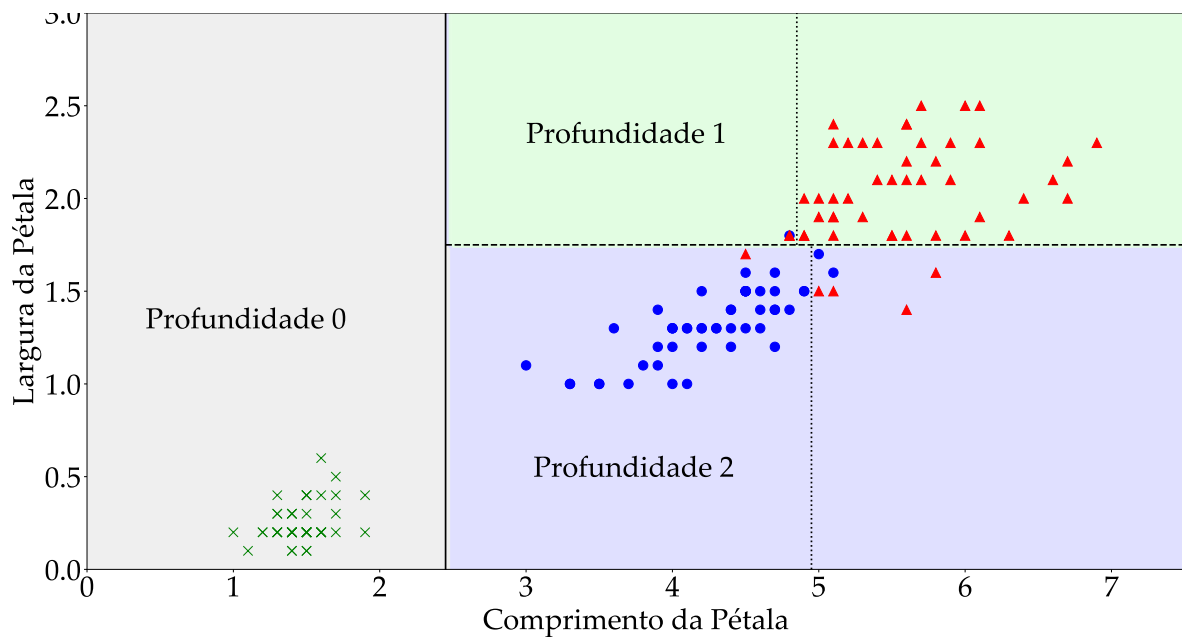
A equação abaixo mostra como o algoritmo realiza o cálculo do *gini score* do  $i$ -ésimo nó (SURYAKANTHI, 2020):

$$G_i = 1 - \sum_{k=1}^n p_{ik}^2 \quad (3.2)$$

em que  $p_{ik}$  é a razão de instâncias da classe  $k$  entre as instâncias treinadas no nó  $i$ .

### 3.2.4.3 Algoritmo de Treinamento CART

O algoritmo de treinamento *Classification And Regression Tree* (CART) é utilizado para treinar Árvores de Decisão. Ele é fundamentado de uma maneira muito simples: o algoritmo primeiramente divide o conjunto de treinamento em dois subconjuntos utilizando uma única característica  $k$  e um limiar  $t_k$  (para o exemplo abordado, comprimento  $\leq 2,45$  cm). A procura do par  $(k, t_k)$  produz os subconjuntos mais puros, ponderados pelo

Figura 14 – Fronteira de decisões do modelo *Iris*

Fonte: Adaptado de Géron (2019)

respectivo tamanho. Nesse sentido, a função de custo que minimiza o algoritmo é dada por:

$$J(k, t_k) = \frac{m_e}{m} G_e + \frac{m_d}{m} G_d \quad (3.3)$$

sendo  $G_{e/d}$  as quantificações de impureza e  $m_{e/d}$  são o número de instâncias em cada subconjunto.

A divisão de (sub)conjuntos acontece de maneira recursiva, parando quando o algoritmo atinge a profundidade desejada ou quando não consegue realizar mais nenhuma divisão que irá reduzir a impureza.

#### 3.2.4.4 Complexidade Computacional

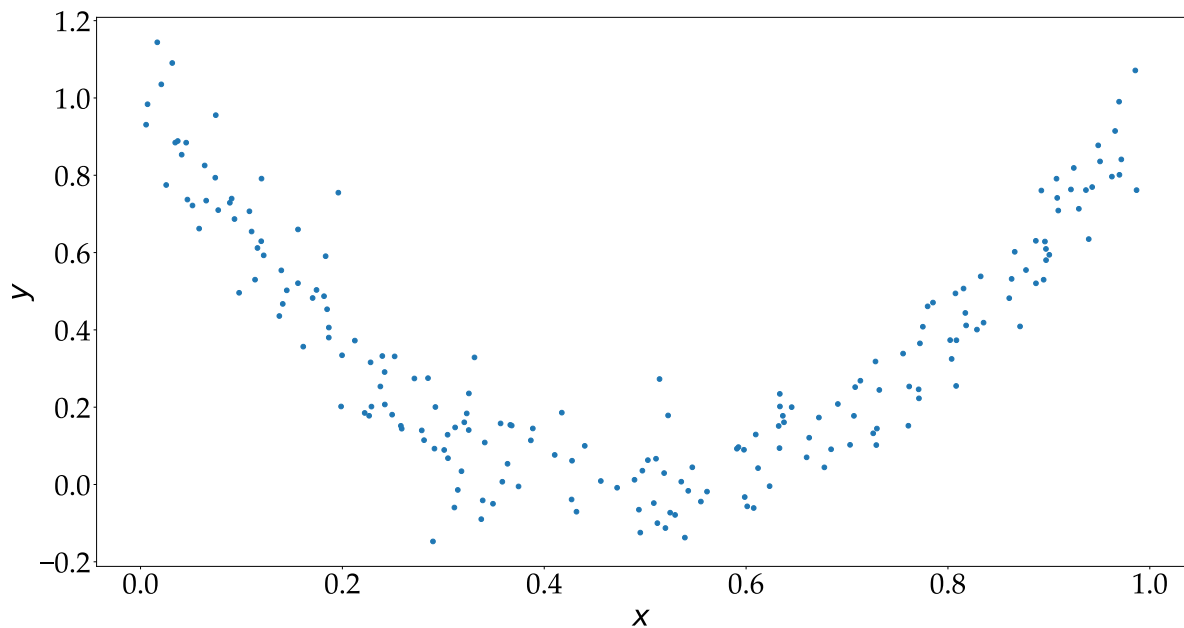
Realizar previsões requer atravessar toda a árvore, do nó raiz até a última folha. Árvores de Decisões são, geralmente, balanceadas, então percorrer uma árvore por completo tem complexidade  $O(\log_2(m))$ . Como cada nó requisita a avaliação do valor de uma das características do conjunto de treinamento, a complexidade computacional se mantém a mesma independentemente do número de características. Sendo assim, previsões são extremamente rápidas, até mesmo quando está lidando com grandes conjuntos de treinamento.

### 3.2.4.5 Regressão

Árvores de Decisão também são capazes de realizar a tarefa de regressão, como mencionado anteriormente. Para exemplificar essa habilidade, utiliza-se o conjunto de treinamento mostrado na Figura 15.

A árvore mostrada na Figura 16 é muito similar à mostrada na Figura 13. A principal diferença é que ao invés de estar prevendo uma classe para cada nó, ela prevê um valor. Supondo que uma nova instância tem como valor  $x = 0,6$  e percorrendo todos os nós da árvore de regressão, percebe-se que o nó folha escolhido pelo algoritmo é  $value = 0,1106$ . O valor da previsão é a média entre as 110 instâncias de treinamento associadas ao nó em questão.

Figura 15 – Conjunto de dado quadrático e pseudo-aleatório para treinamento do modelo preditivo



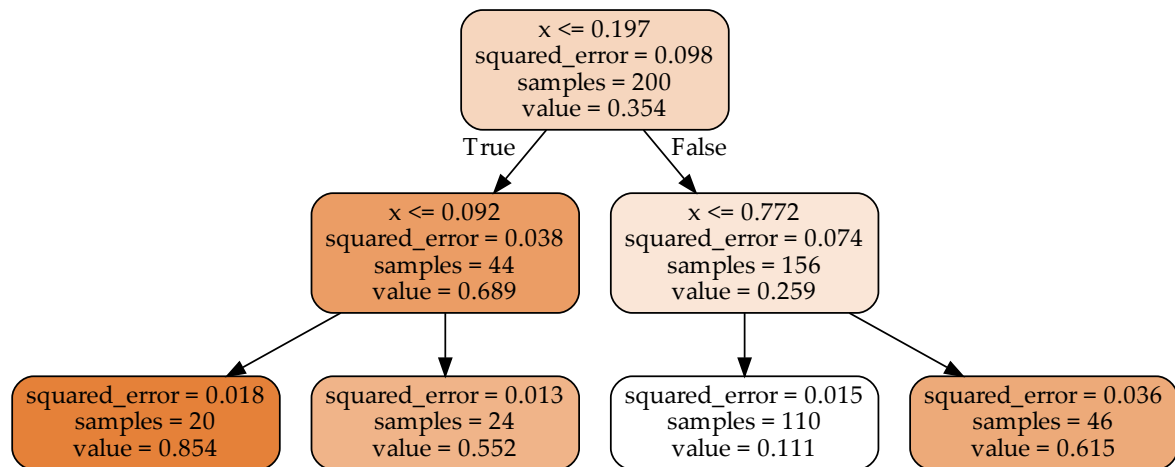
Fonte: Autor (2022)

O algoritmo CART funciona essencialmente da mesma forma para as tarefas de classificação e regressão, exceto que ao invés de dividir o conjunto de treinamento em subconjuntos com a finalidade de diminuir a impureza no momento de classificar, ele divide com a finalidade de diminuir o Erro Quadrático Médio (MSE) na regressão.

A Equação 3.4 mostra a função de custo para regressão utilizada pelo algoritmo CART.

$$J(k, t_k) = \frac{m_e}{m} MSE_e + \frac{m_d}{m} MSE_r \quad (3.4)$$

Figura 16 – Árvore de Decisão para a tarefa de regressão



Fonte: Autor (2022)

sendo

$$MSE_{nó} = \sum_{i \in nó} (\hat{y}_{nó} - \hat{y}^{(i)})^2 \quad (3.5)$$

$$\hat{y}_{nó} = \frac{1}{m_{nó}} \sum_{i \in nó} y^{(i)} \quad (3.6)$$

A Figura 17 mostra como as fronteiras de decisões são definidas no conjunto de treinamento mostrado na Figura 15.

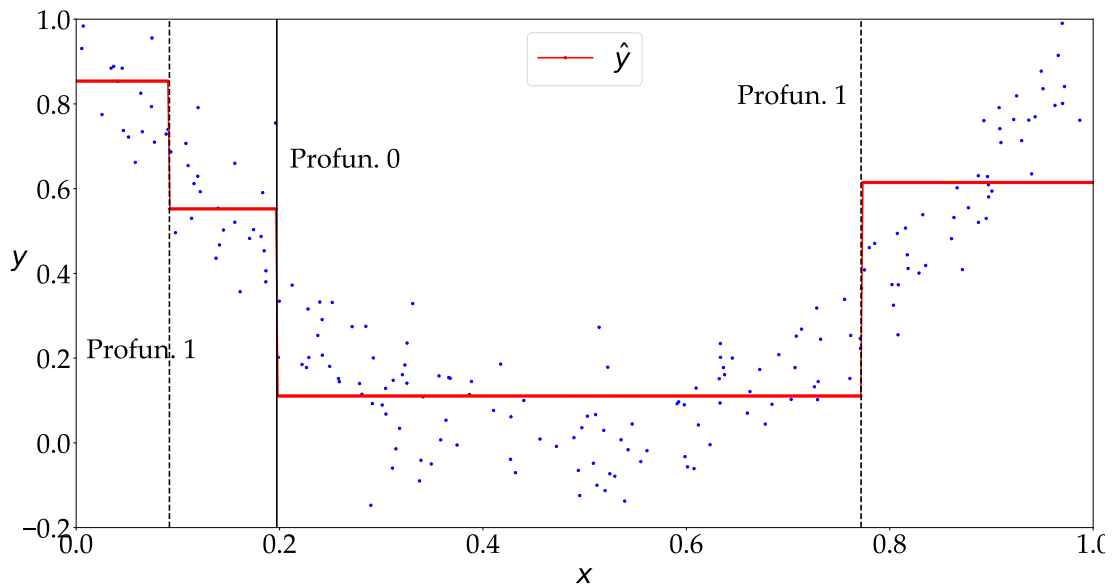
### 3.2.5 Florestas Aleatórias

Florestas Aleatórias é uma técnica de AM que faz uso de diversas Árvores de Decisões (HO, 1995). Árvores de Decisões são modelos extremamente atrativos devido a sua performance computacional. Entretanto, elas não podem ser cultivadas até uma complexidade arbitrária devido a perda de precisão da generalização das novas instâncias. A limitação na complexidade geralmente significa uma precisão abaixo do ideal no conjunto de treinamento.

Uma alternativa para essa limitação é utilizar Florestas Aleatórias, pois são capazes de utilizar os princípios estocásticos para se expandir com a finalidade de aumentar a acurácia em instâncias de treinamento e novas. A essência da técnica é construir múltiplas Árvores de Decisão em subdomínios selecionados aleatoriamente a partir do domínio de características. Tal propriedade permite que as Florestas tenham uma capacidade



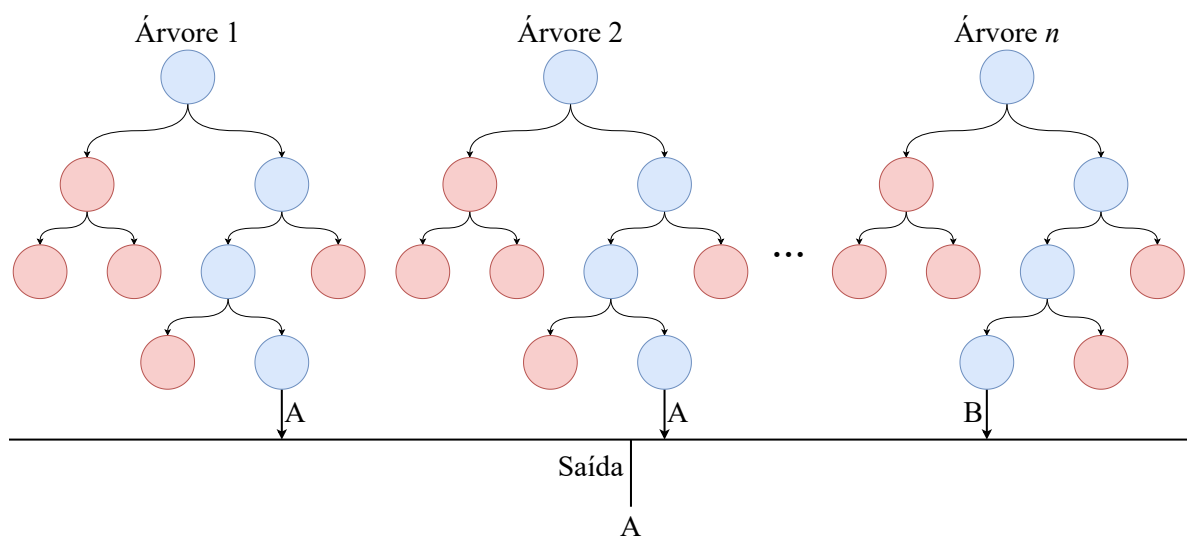
Figura 17 – Fronteiras de decisão na árvore criadas para a tarefa de regressão



Fonte: Adaptado de Géron (2019)

maior de generalização, pois suas árvores foram testadas em diferentes cenários distintos e escolheram as melhores características localmente. Para tarefas de classificação, a saída do modelo é a classe selecionada pelo maior número de árvores. Para tarefas de regressão, a média das previsões de cada árvore é retornada como saída. A Figura 18 ilustra o processo de decisão de uma Floresta Aleatória.

Figura 18 – Ilustração de como a escolha por uma classe é feita em Florestas Aleatórias



Fonte: Autor (2022)

### 3.3 Aprendizado Profundo

Redes Neurais Artificiais compõem o cerne do AP. Elas são versáteis, poderosas e escaláveis, o que faz delas ideais para tarefas complexas de aprendizagem, como classificar bilhões de imagens, recomendar os melhores vídeos para usuários, reconhecimento de fala e inteligência artificial.

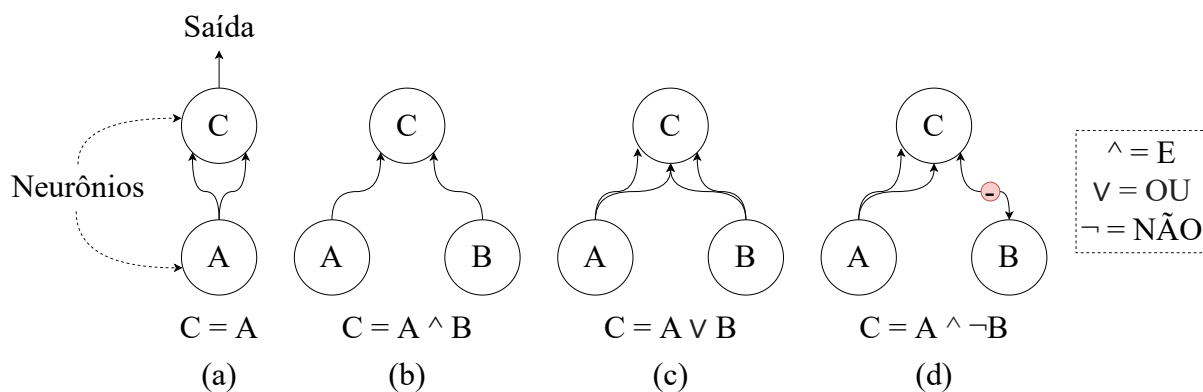
#### 3.3.1 Redes Neurais Artificiais

As primeiras concepções de RNAs datam do ano 1943, quando (MCCULLOCH; PITTS, 1943) apresentaram um modelo computacional simplificado de como possivelmente os neurônios biológicos trabalham juntos em animais para executar cálculos complexos utilizando lógica proposicional.

Nos anos 60, o repentino sucesso das primeiras redes neurais permitiu que a comunidade científica imaginasse que máquinas inteligentes logo estariam entre os humanos. Entretanto, devido às limitações computacionais da época e dos anos seguintes, as aplicações que utilizavam esses algoritmos caíram por terra e as redes voltaram para o esquecimento.

##### 3.3.1.1 Operações Lógicas com Neurônios

Warren McCulloch e Walter Pitts propuseram um modelo computacional de um neurônio biológico: o neurônio artificial, capaz de realizar operações binárias. Um neurônio artificial simplesmente ativa sua saída quando uma ou mais entradas são ativadas. Com esse modelo, os pesquisadores mostraram que era possível executar qualquer operação lógica desejada. A Figura 19 mostra algumas configurações de neurônios artificiais sendo utilizados para executar operações lógicas.



Fonte: Adaptado de Géron (2019)

Assumindo que um neurônio artificial é ativado quando pelo menos duas entradas são ativadas. A partir disso, conclui-se:

- Em (a) está sendo representada a função identidade: a saída é igual a entrada;
- Em (b) está sendo representada a operação E. C somente será ativado se A e B forem ativados;
- (c) mostra a operação OU. Se A ou B (ou ambos) forem ativados, C será ativado;
- Finalmente, (d) ilustra uma composição de operações: E e NÃO. C somente será ativado se o neurônio A for ativado e se B estiver desativado. Se B estiver ativado, logo C não estará.

### 3.3.1.2 O Perceptron

O Perceptron é a arquitetura mais simples de uma RNA. Ele é baseado em um neurônio artificial chamado Unidade Lógica de Limiars (TLU). Nele, as entradas e as saídas são números e não valores binários. Além disso, cada entrada está associada a um peso. A TLU computa a soma ponderada das entradas, como mostra a Equação 3.7.

$$z = w_1x_1 + \dots + w_nx_n = \{x\}^T\{w\} \quad (3.7)$$

Em seguida, aplica-se uma função de ativação  $f$  na combinação linear, gerando uma saída descrita pela Equação 3.8.

$$h_w(y) = f(z) \quad (3.8)$$

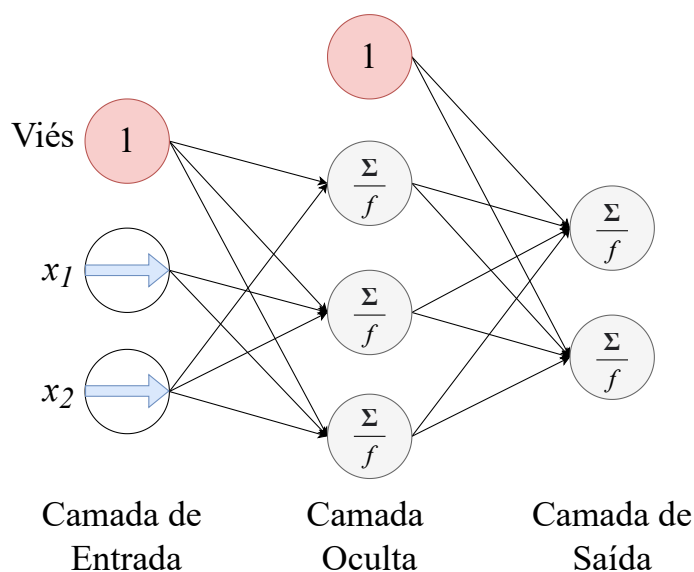
Uma TLU pode ser utilizada para realizar uma simples classificação binária linear. Ela computa a combinação linear das entradas e se o resultado for maior do que um limiar, a entrada é classificada como positiva ou negativa. Entretanto, os Perceptrons não fornecem a probabilidade de pertencer a uma determinada classe.

Minsky e Papert (2017) salientaram algumas limitações dos Perceptrons, em particular a incapacidade deles em solucionar alguns problemas triviais de classificação, como o OU EXCLUSIVO. Entretanto, para ultrapassar essa limitação, é possível reunir diversos Perceptrons. A RNA resultante é chamada de Perceptron Multicamadas.

### 3.3.1.3 Perceptron Multicamadas e Retropropagação

Uma RNA do tipo MLP é composta de uma camada de entrada, uma ou mais camadas ocultas, e uma camada de saída. Cada camada, exceto a de saída, inclui um neurônio de viés. A Figura 20 mostra a ilustração de uma MLP.

Figura 20 – Rede Neural Artificial do tipo Perceptron Multicamadas



Fonte: Autor (2022)

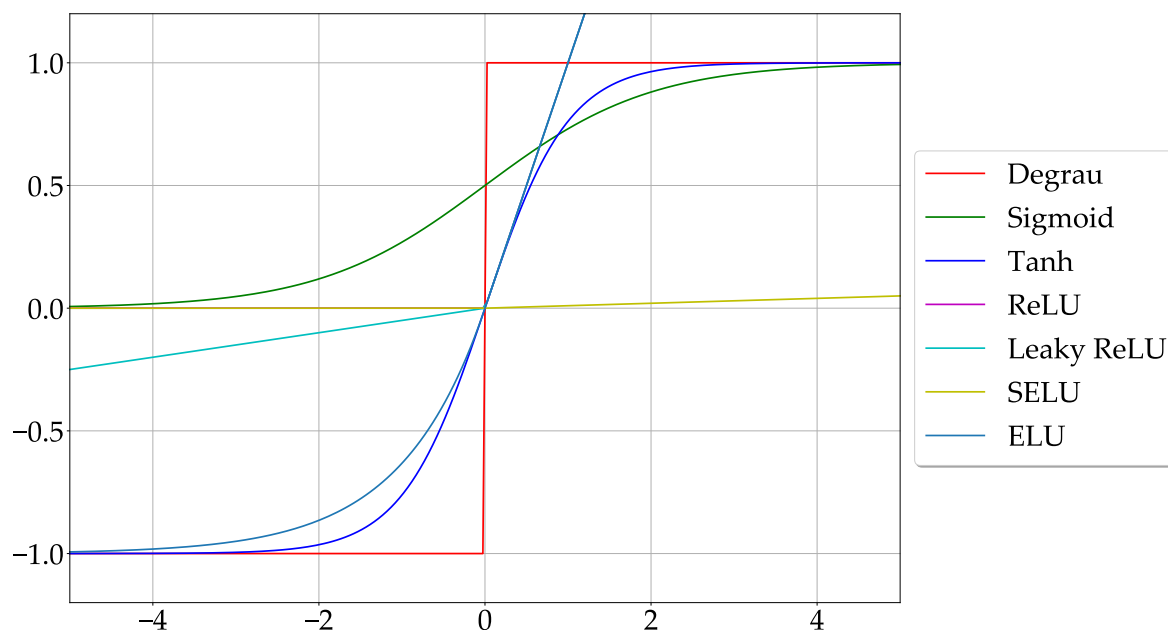
Quando uma RNA contém múltiplas camadas ocultas, ela é chamada de Rede Neural Profunda (RNP). O campo de AP estuda as mais variadas arquiteturas de RNP. Entretanto, muitos autores falam que redes neurais do tipo MLP também estão inclusas no AP. Este trabalho trata MLPs como pertencentes a área de aprendizado profundo.

Por muitos anos, pesquisadores tiveram dificuldades para treinar MLPs. Todavia, Rumelhart e McClelland (1987) publicaram um trabalho introduzindo o algoritmo de retropropagação de erro para o treinamento de redes neurais, que é utilizado até hoje. O algoritmo desenvolvido faz uso do Gradiente Descendente e durante duas passagens pela rede, uma da camada de entrada para a camada de saída e outra o inverso, computa o gradiente do erro com relação a cada parâmetro do modelo. Dessa forma, o algoritmo é capaz de encontrar cada peso e viés que devem ser ajustados para que o erro diminua. O processo acontece até a convergência da rede neural. Para que o treino de uma RNA não falhe, é importante que todos os pesos das camadas ocultas sejam iniciados aleatoriamente.

### 3.3.2 Funções de ativação

Para que o Gradiente Descendente funcione, as funções de ativação devem ser deriváveis em seus domínios. Isto é, elas não devem possuir descontinuidades ou mudanças bruscas. Devido às funções de ativação não-lineares, as redes neurais dos mais variados tipos podem resolver problemas extremamente complexos. A Figura 21 mostra algumas das funções de ativação mais utilizadas.

Figura 21 – Funções de ativação utilizadas para o treinamento de redes neurais



Fonte: Autor (2022)

### 3.3.2.1 Regressão utilizando MLPs

Quando MLPs são utilizadas para realizar a tarefa de regressão, a quantidade de neurônios na camada de saída é exatamente igual ao número de valores que são previstos. Por exemplo, se o valor a ser previsto é o preço de casas, utiliza-se apenas um neurônio na camada de saída da rede. Para localizar o centro de uma imagem bidimensional, 2 neurônios são necessários, um para cada coordenada.

Em geral, quando se está construindo uma rede neural para regressão, a função de ativação depende do tipo da saída que está sendo prevista. Logo, quando se está tentando prever valores somente positivos, é comum utilizar a função de ativação ReLU; se a previsão está sendo para valores entre 0 e 1, utiliza-se a função logística; ou se a previsão está sendo para valores entre -1 e 1, a função tangente hiperbólica é uma possível escolha. A Figura 20 ilustra um exemplo de uma rede neural sendo utilizada para regressão múltipla.

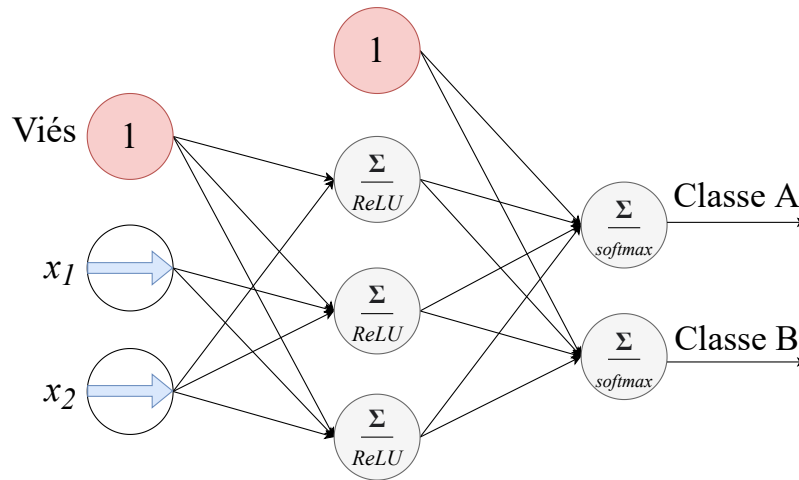
### 3.3.2.2 Classificação utilizando MLPs

Redes neurais do tipo MLP também podem ser utilizadas para tarefas de classificação. Para um problema de classificação binária, é necessário utilizar apenas, na camada de saída, um único neurônio com a função logística como função de ativação. Quando a classificação envolve multiclasse na camada de saída, o número de neurônios a ser utilizados é igual ao número de classes do conjunto de treinamento.

Uma rede treinada para identificar dígitos de 0 a 9 terá 10 neurônios de saída,

um para cada classe. Todos os neurônios utilizarão a função de ativação *softmax*. Ela irá garantir que a soma de todas as probabilidades, para todas as classes, estará entre 0 e 1. A Figura 22 mostra a arquitetura de uma MLP para classificação multiclasse.

Figura 22 – Arquitetura de uma Rede Neural Artificial para a tarefa de classificação



Fonte: Autor (2022)

### 3.3.3 Gradiente Descendente

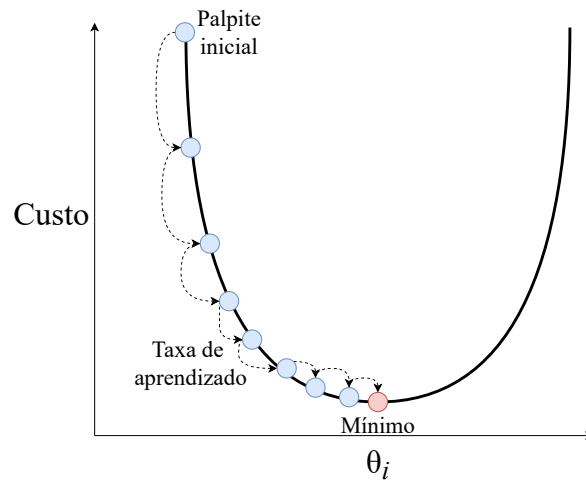
O Gradiente Descendente (GD) é um algoritmo de otimização capaz de encontrar soluções ótimas para uma grande gama de problemas. A ideia geral do algoritmo é ajustar todos os parâmetros de maneira iterativa, com a finalidade de minimizar a função de custo. O algoritmo quantifica o gradiente local da função de erro com relação a um peso  $\theta_i$  e prossegue na direção decrescente. Uma vez que o gradiente da função é zero, o algoritmo chegou ao mínimo. A Figura 23 mostra como a técnica realiza a otimização da função de erro.

Um importante parâmetro do algoritmo é a taxa de aprendizado. Se a taxa de aprendizado for muito pequena, o algoritmo precisará de muitas iterações para poder convergir. Se a taxa for muito grande, ele irá falhar em minimizar a função de custo. Se a função de erro possuir descontinuidades, platôs, mínimos locais ou mínimo global, será difícil para o GD convergir. Nesse sentido, é ideal que as funções de erro sejam convexas, como é o caso da função MSE.

#### 3.3.3.1 Gradiente Descendente Estocástico

O principal problema do GD é o fato dele usar todo o conjunto de treinamento para calcular os gradientes a cada iteração, o que torna a sua performance computacional

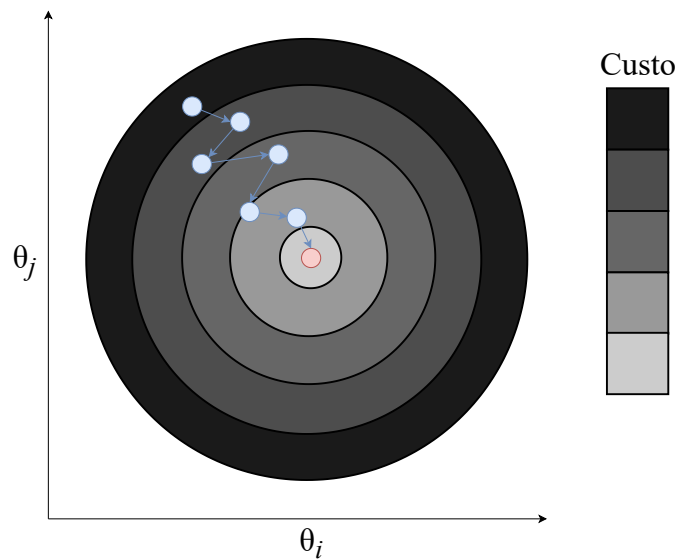
Figura 23 – Gradiente Descendente de uma função quadrática



Fonte: Autor (2022)

bastante indesejável. Tal fato impossibilita o treino de redes neurais com um conjunto de treinamento complexo e/ou grande.

Figura 24 – Gradiente Descendente Estocástico



Fonte: Autor (2022)

O Gradiente Descendente Estocástico (SGD) realiza a escolha de uma instância aleatória do conjunto de treinamento e calcula o gradiente baseado naquela única instância. Isso torna o algoritmo muito mais rápido (SONG; CHAUDHURI; SARWATE, 2013), o que torna possível treinar redes neurais em grandes conjuntos de dados. Em contrapartida,

devido à natureza estocástica, o SGD é mais irregular do que seu primo, o que é ilustrado na Figura 24.

Quando a função de custo é muito irregular, o SGD é capaz de escapar de um mínimo local, algo que dificilmente o GD conseguiria. Entretanto, também é possível que o SGD nunca encontre o mínimo da função.

### 3.3.4 Treinamento de uma Rede Neural

A Equação 3.9 define a função de custo utilizada para treinar modelos de AP, onde  $\{Y\}_i$  é o vetor de valores observados e  $\{\dot{Y}\}_i$  o vetor de valores previstos.

$$E = \frac{1}{n} \sum_{i=1}^n (\{Y\}_i - \{\dot{Y}\}_i)^2 \quad (3.9)$$

A Equação 3.10 descreve a função generalizada de uma RNA para  $L$  camadas aplicadas às características  $X_i$

$$\{\dot{Y}\} = F(\{X_i\}) = f_L(f_{L-1}(\dots(f_1(\{X_i\})))) \quad (3.10)$$

em que a função de mapeamento de uma camada  $L$  é dada por:

$$f_i(\{X\}_i) = g(\{W\}_{i,l}^T \{X\}_i + b_{i,l}) \quad (3.11)$$

sendo  $g$  a função de ativação da camada em questão,  $\{W\}_{i,l}$  é o  $i$ -ésimo vetor peso da camada  $L$  e  $b_{i,l}$  é o viés. Sendo  $l \in \{1, \dots, L\}$ .

Para minimizar a função de custo, o SGD seleciona uma instância aleatória para treinar e calcula o respectivo gradiente a cada iteração do algoritmo de treinamento de uma rede neural. Sendo assim, tem-se a equação de atualização dos pesos da camada  $l$  da rede:

$$\{W\}_{i+1,l}^T = \{W\}_{i,l}^T - \eta \nabla E \quad (3.12)$$

cujo  $\eta$  é igual à taxa de aprendizado adotada para realizar o treinamento e  $\nabla E$  é o gradiente da função de erro.

## 3.4 Redução de Dimensionalidade

Quando modelos de AP e AM são submetidos a um conjunto de treinamento com grande dimensão, isto é, com diversas características, a performance computacional tende a ser precária. Esse fato se acentua quando redes neurais com muitos neurônios e muitas camadas são treinadas em conjuntos de dados complexos. Quando se está realizando o



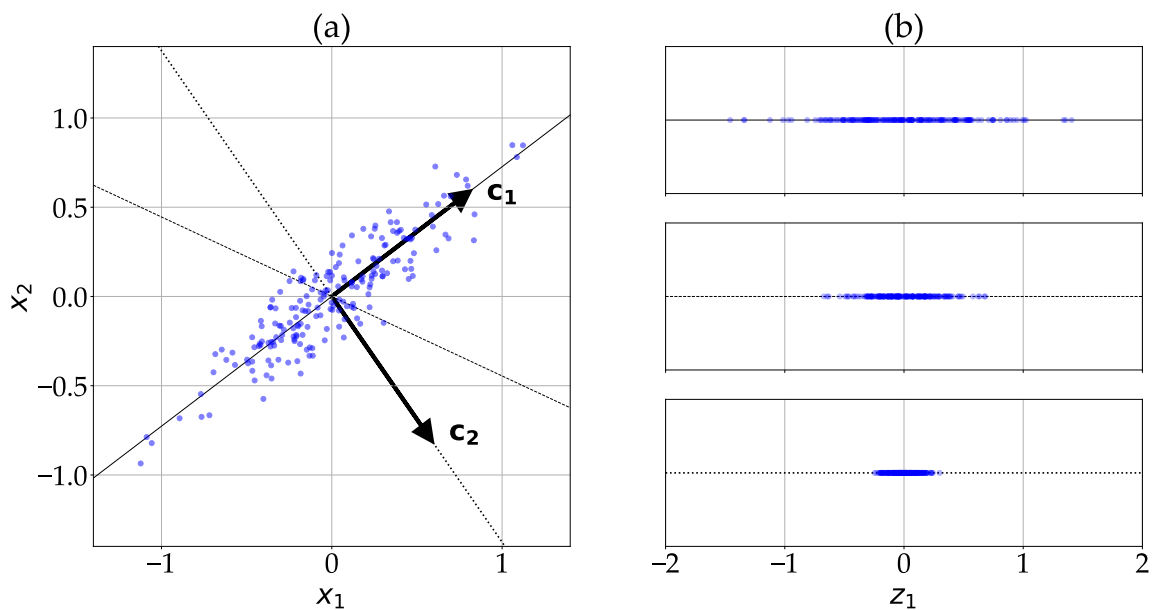
monitoramento estrutural de estruturas com muitos GDL, torna-se inviável utilizar o conjunto de treinamento por completo, pois a velocidade do treino será prejudicada pela grande dimensão dos dados.

Nesse sentido, torna-se necessário realizar a redução de dimensionalidade do conjunto de treinamento. Há diversas técnicas capazes de reduzir a dimensionalidade, mas um dos principais é o algoritmo Análise de Componentes Principais (PCA) (F.R.S., 1901).

### 3.4.1 Análise de Componentes Principais

O PCA projeta o conjunto de dados no hiperplano mais próximo encontrado à eles, como mostrado na Figura 25. O conjunto de dados 2D mostrado em (a) foi projetado sobre a linha 1D mostrado em (b).

Figura 25 – O PCA identifica o subespaço e projeta os dados sobre ele



Fonte: Géron (2019)

Como é possível ver, a projeção sobre a linha sólida mantém a variância máxima dos dados, enquanto sobre as linhas pontilhada e tracejada o mesmo não acontece. Logo, é intuitivo que a projeção escolhida seja a que mantém a maior variância possível, uma vez que a perda de dados é a menor entre os três casos mostrados.

Para encontrar os componentes principais de um conjunto de dados com  $n$  dimensões, o PCA utiliza a técnica de fatoração matricial chamada Decomposição em Valores Singulares (SVD), que decompõe o conjunto de dados em uma multiplicação de matrizes do tipo  $[U][\Sigma][V]^T$ , onde  $[V]$  contém todos os componentes principais.

$$V = [\{c\}_1^T \quad \{c\}_2^T \quad \cdots \quad \{c\}_n^T] \quad (3.13)$$

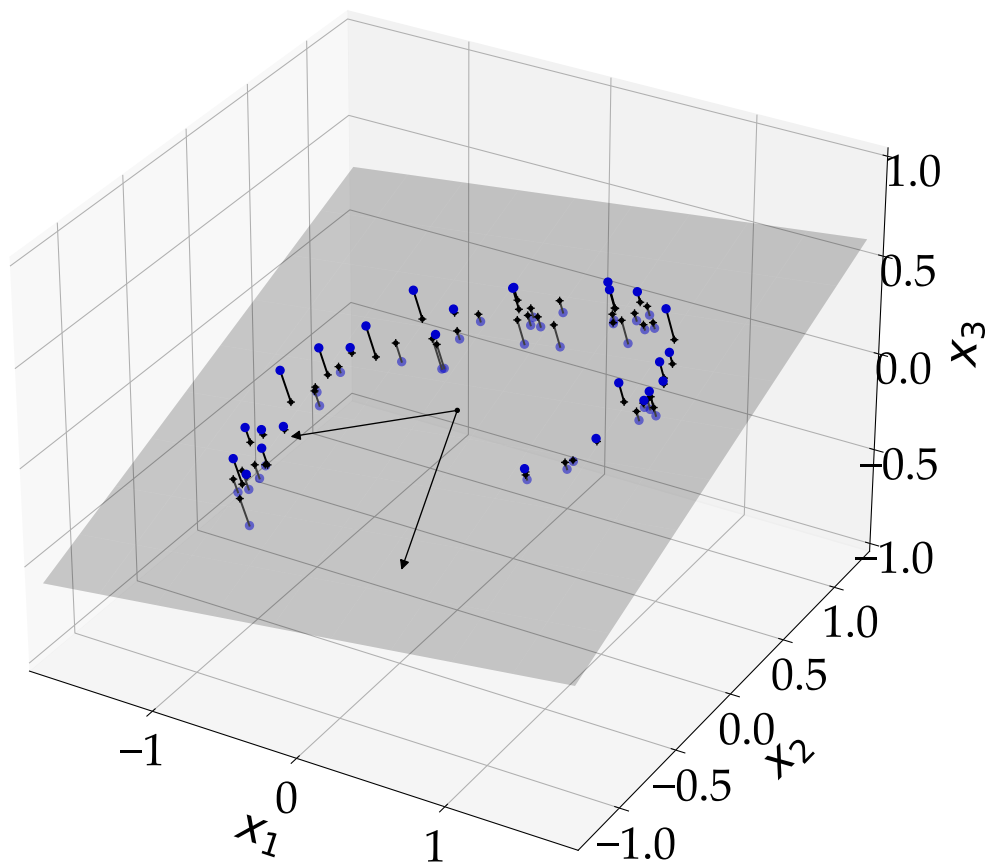
Após encontrar todos os componentes principais, a técnica projeta o conjunto de dados sobre o hiperplano definido pelos  $d$  primeiros componentes principais. O cálculo da projeção é feito de acordo com a equação abaixo:

$$[X]_{d-proj} = [X][W]_d \quad (3.14)$$

sendo  $[X]$  a matriz de características e  $[W]_d$  a matriz que contém os  $d$  primeiros componentes principais.

A Figura 26 mostra um conjunto de dados 3D e o respectivo hiperplano 2D encontrado pelo PCA, de forma que a variância seja a máxima possível.

Figura 26 – O PCA identifica o hiperplano 2D e projeta o conjunto de dados 3D sobre ele



Fonte: Géron (2019)

## 4 Programa Computacional desenvolvido para o Monitoramento de Saúde Estrutural

Para a criação do programa computacional proposto e utilizado para gerar os futuros resultados deste trabalho, utilizou-se Programação Orientada a Objetos (POO), a linguagem computacional Python (versão 3.7.9), as bibliotecas Numpy (versão 1.21.5), Scipy (versão 1.5.4) e Pandas (1.1.4). Para a criação dos modelos estatísticos, foram utilizadas as bibliotecas Scikit-Learn (versão 1.0.1) e TensorFlow (versão 2.8.0). O código-fonte está disponível na íntegra no endereço [github/gabrielogia/matrix-analysis-of-structures](https://github.com/gabrielogia/matrix-analysis-of-structures).

O escopo da implementação está limitado à análise linear 2D de treliças, vigas e pórticos, nos regimes estático e dinâmico.

### 4.1 Entrada de Dados

Abaixo encontra-se o fragmento da formatação de entrada que o programa aceita para o módulo dinâmico.

```
#MODEL
frame

#TYPE
dynamic

#DELTATIME
0.01

#TIME
5

#NODES
1,0,0,1,1,1,0,0,0,0,0,0
2,1,0,0,0,0,0,0,0,0,0,0
3,2,0,0,0,0,0,0,0,0,0,0

#ELEMENTS
1,1,2,1,1,1,1,local,0,0
2,2,3,1,1,1,1,local,0,0
```

### 4.1.1 *Flags*

#### #MODEL

Define o tipo de modelo a ser utilizado pelo programa. Os modelos disponíveis são: *frame*, *beam* e *truss*.

#### #TYPE

Define o tipo de análise a ser feita pelo programa. As opções disponíveis são: *dynamic* e *static*.

#### #DELTATIME

Define o incremento de tempo a ser utilizado no algoritmo de Newmark para a obtenção das respostas dinâmicas da estrutura. O tipo é *float*.

#### #TIME

Define o tempo de simulação do programa. O tipo é *int*.

#### #NODES

Define os nós que compõem o modelo estrutural. A ordem da informação referente ao nó é: id, posição  $x$ , posição  $y$ , restrição de deslocamento na direção  $x$ , restrição de deslocamento na direção  $y$ , restrição de rotação na direção  $z$ , constante elástica da mola colocada na direção  $x$ , constante elástica da mola colocada na direção  $y$ , constante elástica da mola colocada na direção  $z$ , força nodal atuante na direção  $x$ , força nodal atuante na direção  $y$  e força nodal atuante na direção  $z$ .

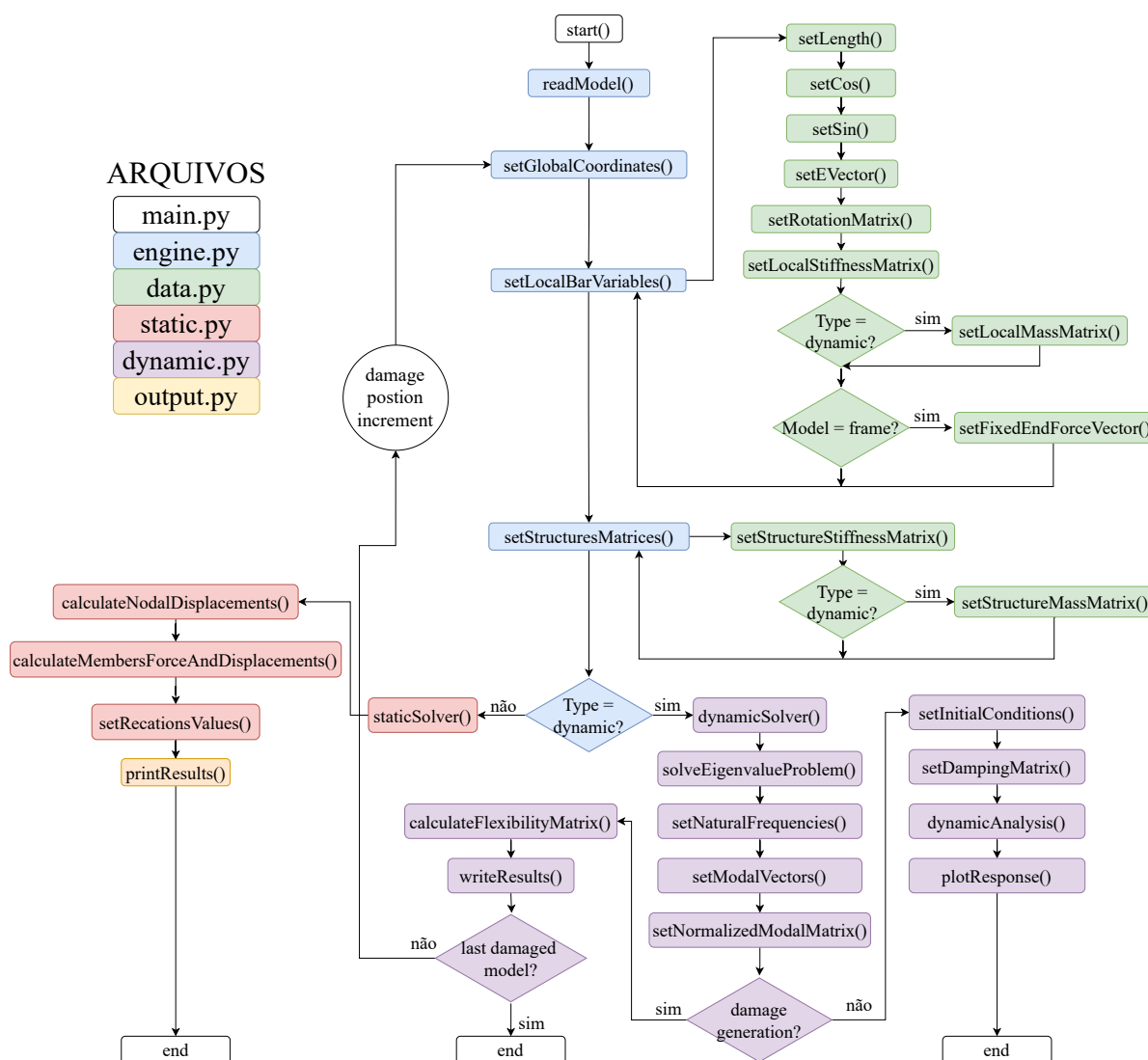
#### #ELEMENTS

Define os elementos que compõem o modelo estrutural. A ordem da informação referente ao elemento é: id, id do nó inicial, id do nó final, módulo de elasticidade do material, área da seção transversal, momento de inércia, densidade volumétrica do material, carga distribuída atuante na direção  $x$  e carga distribuída atuante na direção  $y$ .

## 4.2 Fluxograma

A Figura 27 ilustra o fluxograma de execução do programa desenvolvido. Quando a *branch* "damage\_generation" está selecionada, o programa deve gerar arquivos de saída no formato .csv para o modelo intacto e para cada um dos modelos danificados da estrutura. As saídas foram unificadas em um único arquivo, que é utilizado para realizar o treinamento dos modelos estatísticos.

Figura 27 – Fluxograma de execução do programa de análise matricial de estruturas



Fonte: Autor (2022)

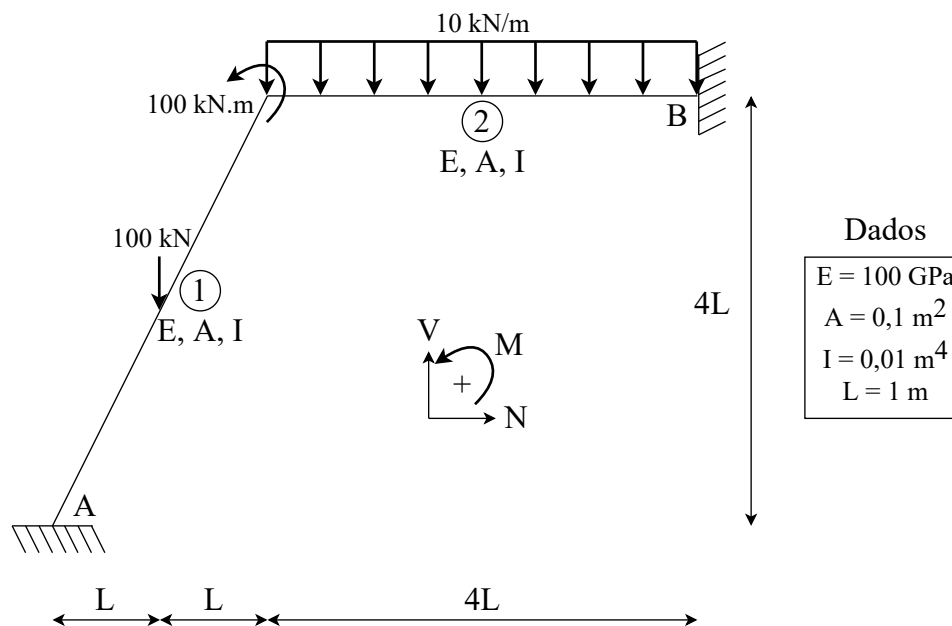
## 4.3 Validação

Utilizou-se as ferramentas computacionais Ftool © e Ansys Student © para validar os módulos de análise estática e modal, respectivamente.

### 4.3.1 Análise Estática

O modelo estrutural mostrado na Figura 28 foi utilizado para validar a análise estática. A Tabela 1 mostra a comparação de resultados obtidos entre o programa desenvolvido para este trabalho e a ferramenta Ftool.

Figura 28 – Pórtico utilizado para a validação do módulo de análise estática



Fonte: Autor (2022)

Tabela 1 – Comparação de reações de apoio nos nós A e B entre o programa implementado e a ferramenta Ftool

Reações de Apoio	Implementação Própria	Ftool
Axial em A (kN)	17,32	17,32
Cortante em A (kN)	135,37	135,37
Momento Fletor em A (kN.m)	61,48	61,48
Axial em B (kN)	-17,32	-17,32
Cortante em B (kN.m)	4,63	4,63
Cortante em B (kN.m)	1,47	1,47

Fonte: Autor (2022)

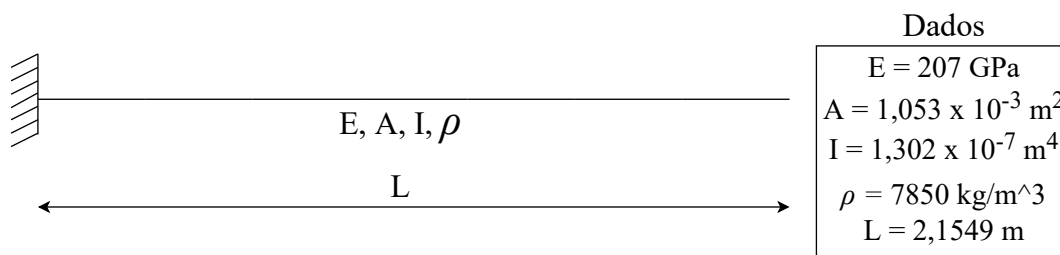
Como é possível observar na tabela acima, todas as reações de apoio foram idênticas. Logo, o programa implementado foi validado com sucesso.

## 4.3.2 Análise Modal

### 4.3.2.1 ANSYS

O Viga Cantilever mostrada na Figura 29 foi utilizado para validar a análise modal. A Tabela 2 mostra a comparação de resultados obtidos entre o programa desenvolvido para este trabalho e o *software* Ansys.

Figura 29 – Viga utilizada para a validação do módulo de análise modal



Fonte: Autor (2022)

Tabela 2 – Comparação entre as frequências naturais obtidas pela implementação própria e a ferramenta Ansys

Programa	Frequência Natural (Hz)					
	1	2	3	4	5	6
Implementação Própria	6,88	43,12	120,75	236,61	391,14	584,35
Ansys	6,89	43,10	120,39	235,10	386,94	574,95

Fonte: Autor (2022)

Observa-se na tabela acima pequenas divergências entre os valores de frequências naturais obtidos para os dois programas. Utilizando a Equação 4.1 pra calcular o Erro Percentual Absoluto Médio (MAPE) para as 6 observações, obtém-se um valor de 6,42%. É perceptível notar pequenas diferenças nas frequências naturais da tabela acima. Isso ocorreu devido ao truncamento de ponto flutuante no momento da leitura dos dados de entrada do programa implementado neste trabalho.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - P_i}{A_i} \right| \quad (4.1)$$

em que  $n$  é o número de observações,  $P_i$  e  $A_i$  são as  $i$ -ésimas frequências naturais obtidas através da implementação própria e do *software* Ansys, respectivamente.

### 4.3.2.2 Literatura

Além da validação via Ansys, o programa implementado neste trabalho também foi validado a partir da literatura. Para tal, utilizou-se o pórtico mostrado na Figura 30,

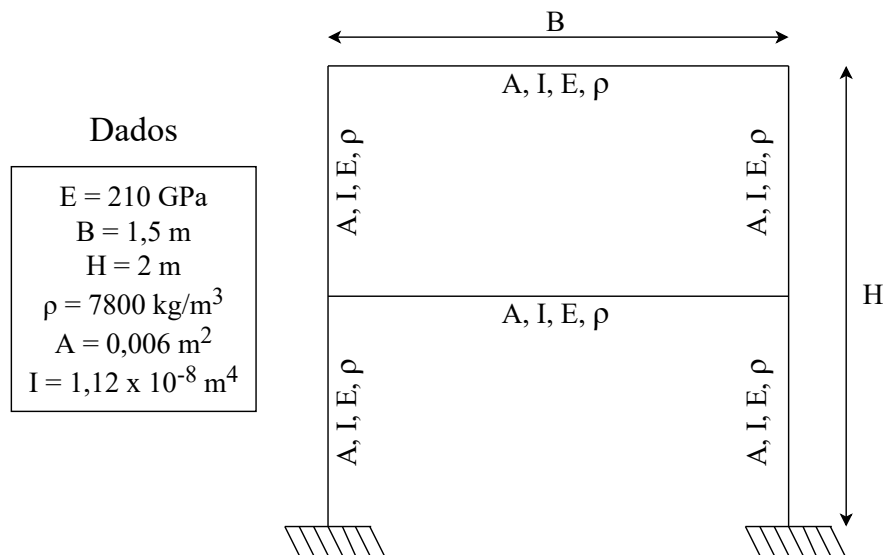
Tabela 3 – Comparação entre as frequências naturais obtidas pelo programa implementado e o trabalho de Rojas et al. (2004)

Cenário	Frequência Natural (Hz)					
	1	2	3	4	5	6
Programa Implementado	4,24	15,07	22,74	28,30	51,85	59,64
Rojas et al. (2004)	4,24	15,07	22,74	28,30	51,85	59,64

Fonte: Autor (2022)

retirado diretamente do trabalho de Rojas et al. (2004). A Tabela 3 descreve o comparativo entre as frequências naturais obtidas em ambos os trabalhos.

Figura 30 – Pórtico utilizado por Rojas et al. (2004)



Fonte: Autor (2022)

Ao analisar a tabela acima, percebe-se que o programa desenvolvido obtém os mesmos resultados que os autores obtiveram.

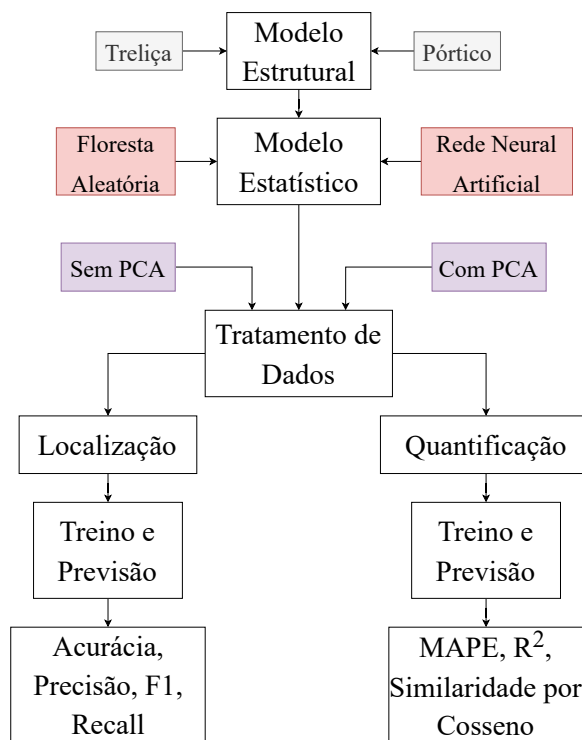


## 5 Experimentos Computacionais e Resultados

### 5.1 Introdução

Nesta seção são apresentados exemplos numéricos submetidos aos modelos estatísticos de Florestas Aleatórias e de Redes Neurais Artificiais, no intuito de comprovar a eficácia da formulação apresentada neste trabalho e a viabilidade da utilização de sistemas inteligentes computacionais para realizar a detecção de dano em estruturas. Inicialmente é mostrado um *benchmarking* realizado com o trabalho de Maia (2006) com o intuito de validar a abordagem deste trabalho a partir da análise do Coeficiente de Pearson. Em seguida, são percorridos exemplos de uma treliça e de um pórtico, onde é realizada uma análise de performance computacional e de performance estatística, em cenários onde houve e não houve a utilização da Análise de Componentes Principais. Todos os conjuntos de treinamento foram gerados a partir do Método da Variação da Flexibilidade Modal, seguindo a formulação já abordada anteriormente.

Figura 31 – Fluxograma de análise dos resultados obtidos para os modelos estruturais de treliça e pórtico



Fonte: Autor (2022)

Para realizar a comparação entre os dois modelos estruturais, dividiu-se a análise

Tabela 4 – Cenários estudados neste trabalho

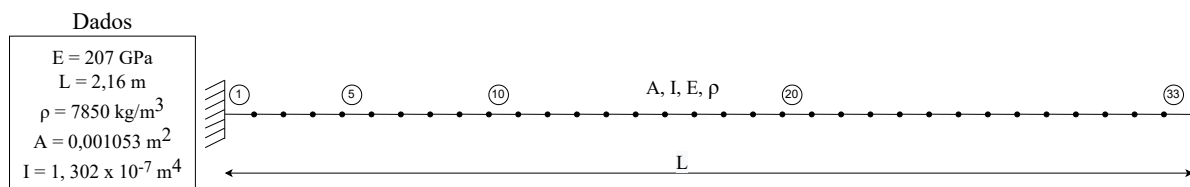
Cenário	Modelo Estrutural	Modelo Estatístico	Utilização do PCA	Tarefa SHM
1	Treliça ou Pórtico	Florestas Aleatórias	Não	Localização
2	Treliça ou Pórtico	Florestas Aleatórias	Sim	Localização
3	Treliça ou Pórtico	Redes Neurais Artificiais	Não	Localização
4	Treliça ou Pórtico	Redes Neurais Artificiais	Sim	Localização
5	Treliça ou Pórtico	Florestas Aleatórias	Não	Quantificação
6	Treliça ou Pórtico	Florestas Aleatórias	Sim	Quantificação
7	Treliça ou Pórtico	Redes Neurais Artificiais	Não	Quantificação
8	Treliça ou Pórtico	Redes Neurais Artificiais	Sim	Quantificação

nas etapas mostradas na Figura 31. Dessa forma, para cada modelo estrutural estudado, utilizou-se ambos os modelos estatísticos e ambos os métodos de tratamentos de dados, sendo gerada oito análises distintas para cada modelo estrutural. A Tabela 4 abaixo ilustra todas os cenários estudados.

## 5.2 Benchmarking

O *benchmark* consistiu na realização um estudo comparativo entre os resultados obtidos por Maia (2006) e pelo programa desenvolvido neste trabalho no que tange a localização de dano em uma Viga Cantilever. Maia simulou diversas redes neurais com diversos conjuntos de treinamento originados de uma Análise Modal. Para o trabalho de conclusão de curso, simulou-se diferentes cenários de dano, onde todos os elementos da viga mostrada na Figura 32 foram danificados com uma severidade de 0,01 até 0,99, gerando 3300 modelos estruturais danificados e 1 modelo intacto, isto é, sem a presença de dano em qualquer elemento. A Equação 5.1 define o novo Módulo de Elasticidade Longitudinal do elemento  $i$  em questão após a aplicação do dano  $j$ .

Figura 32 – Viga utilizada por Maia com 33 elementos idênticos e 34 nós.



Fonte: Autor (2022)

$$E_i = E_0 D_j \quad D_j \in [0, 01, \dots, 0, 99] \quad (5.1)$$

Tabela 5 – Comparação entre os Coeficientes de Pearson obtidos por este trabalho e Maia (2006)

Métrica	Maia					Este Trabalho
	$ANN_1$	$ANN_2$	$ANN_3$	$ANN_4$	$ANN_5$	FA
Coefficiente de Pearson	0,9970	0,9444	0,9806	0,9631	0,9877	0,9999

Após a etapa de geração dos modelos danificados, treinou-se um algoritmo de Florestas Aleatórias para realizar a localização do dano, isto é, realizar uma tarefa de classificação supervisionada. O vetor de características representado na Equação 5.2 foi utilizado para treinar o modelo estatístico supracitado, em que  $k$  varia de 1 até 3300,  $\Delta\omega$  é a variação da frequência natural entre o modelo danificado  $k$  e o modelo intacto, e  $\xi$  é a variação máxima da flexibilidade modal para o grau de liberdade em questão.

$$\{\Xi\}_k = \{\Delta\omega_1, \Delta\omega_2, \dots, \Delta\omega_{10}, \xi_1, \xi_2, \dots, \xi_{66}\} \quad (5.2)$$

A Tabela 5 mostra os resultados obtidos pelo *benchmark*, em que a métrica utilizada foi a Correlação de Pearson, definida pela Equação 5.3.

$$Pearson = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum_{i=1}^n (x_i - \bar{x})^2)} \sqrt{(\sum_{i=1}^n (y_i - \bar{y})^2)}} \quad (5.3)$$

Tratando-se de uma tarefa de classificação, Maia mediu a correlação existente entre os elementos que foram classificados corretamente como danificado e intacto. Quanto mais próximo de 1 é o coeficiente, melhor é o resultado da classificação. Dessa forma, nota-se que o algoritmo de Florestas Aleatórias implementado e treinado para avaliar a capacidade de localização de dano é eficaz e ligeiramente superior às Redes Neurais Artificiais implementadas no trabalho de Maia. Finalmente, pode-se concluir que as Florestas Aleatórias podem ser utilizadas para detectar o dano em estruturas lineares, elásticas, isotrópicas, homogêneas e sem vibração forçada, assim como as Redes Neurais Artificiais.

### 5.3 Treliça Plana

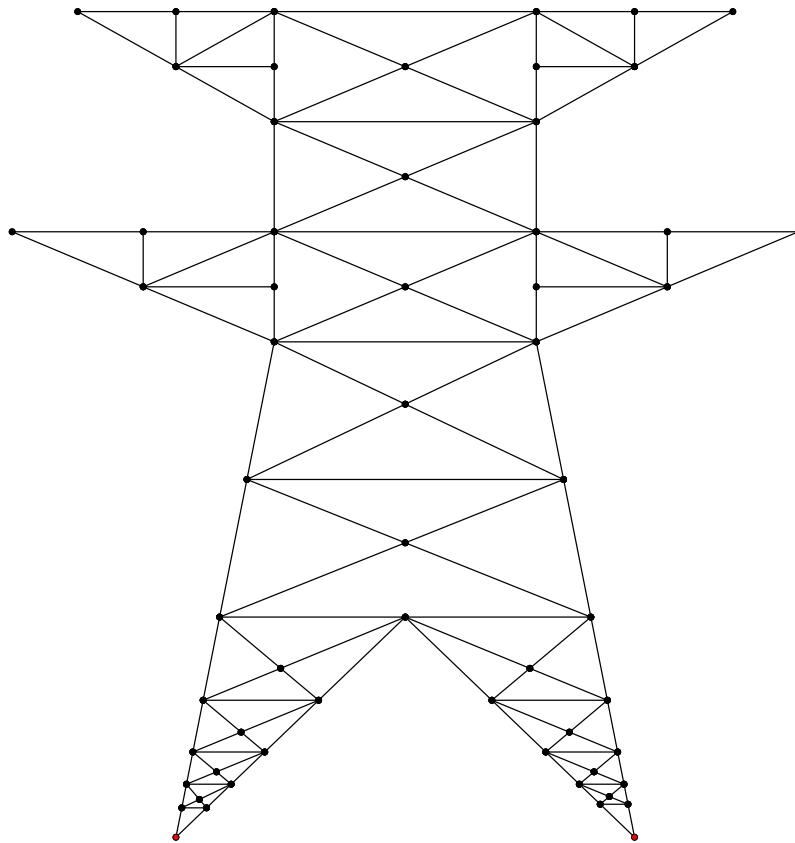
O modelo estrutural plano de treliça, ilustrado na Figura 33, foi utilizado para realizar os estudos de localização e quantificação de dano. Os nós são rotulados e a Tabela 6 mostra as propriedades físicas e geométricas de todos os elementos. Os nós em vermelho são aqueles que tiveram seus deslocamentos restringidos nas três direções do plano cartesiano.

Tabela 6 – Propriedades de todos os elementos estruturais que compõem a treliça plana

Propriedade	Valor
Módulo de Elasticidade [E]	210 <i>GPa</i>
Área de Seção Transversal [A]	$3,1414 * 10^{-2} m^2$
Momento de Inércia [I]	$7,8529 * 10^{-4} m^4$
Densidade [ $\rho$ ]	$7850 kg/m^3$

Fonte: Autor (2022)

Figura 33 – Treliça utilizada para realizar os estudos de localização e quantificação do dano estrutural. Ela detém 60 nós, 129 elementos com a mesma rigidez axial e 116 graus de liberdade



Fonte: Autor (2022)

### 5.3.1 Geração de dados para treinamento dos modelos estatísticos

Diferentemente da estratégia adotada na realização do *benchmarking*, todos os elementos da treliça foram submetidos a dano estrutural de acordo com as funções definidas abaixo, discretizadas em 200 pontos. Assim como no estudo comparativo, o dano foi modelado como a redução do Módulo de Elasticidade Longitudinal. A Figura 34 mostra

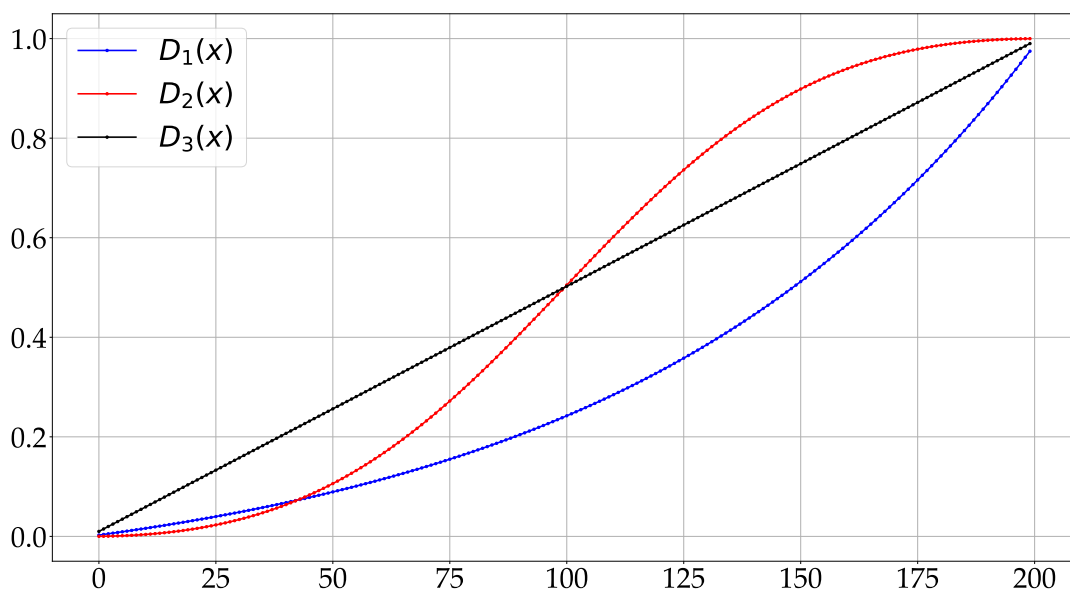
a distribuição do dano de acordo com o valor adotado no domínio da função. Para todas as funções, o domínio varia de 0,01 a 0,99.

$$D_1(x) = \frac{(10^x - 1)}{9} \quad (5.4)$$

$$D_2(x) = \frac{1}{1 + \left(\frac{3x}{x-1}\right)^{-2}} \quad (5.5)$$

$$D_3(x) = x \quad (5.6)$$

Figura 34 – Funções de dano discretizadas em 200 pontos



Fonte: Autor (2022)

Sendo assim, três conjuntos de dados foram gerados, um para cada função de dano, onde cada conjunto de dados contém 25.800 instâncias, pois, para cada elemento da treliça, aplicou-se uma severidade de dano por vez, enquanto os demais elementos permaneceram intactos. Utilizou-se as três funções para entender como os algoritmos de Florestas Aleatórias e Redes Neurais Artificiais se adaptam e como as métricas estatísticas são impactadas pela variação do dano.

A semelhança com funções de ativação de Redes Neurais Artificiais foi o principal motivo para as três funções  $D_1(x)$ ,  $D_2(x)$  e  $D_3(x)$  terem sido escolhidas com a finalidade de modelar o dano aplicado como redução do Módulo de Elasticidade do elemento estrutural. Em todos os casos, 90% do conjunto de dados foi utilizado para treinamento, enquanto o restante foi utilizado para a validação do modelo.

### 5.3.2 Localização do Dano

Como já mencionado neste trabalho, a etapa de localização de dano em estruturas é uma tarefa de classificação supervisionada, que pode ser feita através de qualquer algoritmo classificador. Este processo de classificação é definido pela Equação 5.7, em que  $n$  é o número de características ou a dimensão do vetor de características  $X_i$ ,  $m$  é a dimensão das possíveis classes e  $y$  representa o elemento associado com a maior probabilidade de ter dano.

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m \mid y = f(X_i) \quad (5.7)$$

A Equação 5.8 define o  $k$ -ésimo vetor instância utilizado para o treinamento dos modelos estatísticos, em que  $\Delta\omega$  representa a variação das frequências naturais entre o modelo estrutural com um único elemento danificado e o modelo intacto,  $\xi$  é a máxima variação da flexibilidade modal para o grau de liberdade em questão. Neste caso, o alvo da classificação foi o elemento danificado.

$$\{\Xi\}_k = \{\Delta\omega_1, \Delta\omega_2, \dots, \Delta\omega_{10}, \xi_1, \xi_2, \dots, \xi_{116}\} \quad (5.8)$$

#### 5.3.2.1 Florestas Aleatórias

##### 5.3.2.1.1 Cenário 1

Após a finalização do treinamento das Florestas Aleatórias, coletou-se as métricas de classificação: Acurácia (A), Precisão (P), F1 Score (F1) e Recall (R), definidas pelas equações abaixo. Nessas equações, TN significa a quantidade de verdadeiros negativos, TP a quantidade verdadeiros positivos, FN significa falsos negativos e, finalmente, FP são falsos positivos.

$$A = \frac{TN + TP}{TN + TP + FN + FP} \quad (5.9)$$

$$P = \frac{TP}{TP + FP} \quad (5.10)$$

$$R = \frac{TP}{TP + FN} \quad (5.11)$$

$$F1 = 2 * \frac{P * R}{P + R} \quad (5.12)$$

A tabela abaixo mostra as métricas obtidas após a tarefa de classificação utilizando o algoritmo de Florestas Aleatórias, aplicando as três funções de dano individualmente, e sem utilizar o PCA em nenhum cenário.

Tabela 7 – Métricas obtidas após o treinamento do algoritmo de Florestas Aleatórias sem a utilização do PCA para as três funções de dano propostas neste trabalho

<b>Métrica</b>	$D_1(x)$	$D_2(x)$	$D_3(x)$
Acurácia (%)	97,326	99,516	99,806
Precisão (%)	99,291	99,715	99,851
Recall (%)	97,385	99,535	99,992
F1 Score (%)	98,124	99,590	99,811
Tempo de execução	33,7s	38,4s	39,5s

Fonte: Autor (2022)

Ao analisar os resultados acima, é possível afirmar que as Florestas Aleatórias performaram muito bem. Devido a linearidade física do problema dinâmico, esse modelo estatístico consegue se adaptar facilmente.

### 5.3.2.1.2 Cenário 2

A Tabela 8 mostra o cenário supracitado, onde diferencia-se apenas com a utilização do PCA para a etapa de tratamento de dados. Como já explicado anteriormente, o propósito de utilizar o PCA é utilizar as características mais relevantes do conjunto de treinamento, de maneira que seja possível ter um ganho no tempo de execução do treinamento e da previsão. Em todos os exemplos discutidos neste trabalho, configurou-se o PCA para manter 95% da variância dos dados.

Tabela 8 – Métricas obtidas após o treinamento do algoritmo de Florestas Aleatórias com a utilização do PCA para as três funções de dano propostas neste trabalho

<b>Métrica</b>	$D_1(x)$	$D_2(x)$	$D_3(x)$
Acurácia (%)	49,748	96,512	98,760
Precisão (%)	91,046	98,602	99,269
Recall (%)	50,656	96,554	98,822
F1 Score (%)	60,357	97,325	98,902
Tempo de execução	28,7s	24,3s	24,4s
Nº de Componentes Principais	19	14	20

Fonte: Autor (2022)

Analisando os resultados da tabela acima, pode-se afirmar que a utilização do PCA desencadeou uma diminuição não significativa no tempo de execução do treinamento e da previsão. Além disso, onerou as métricas estatísticas, principalmente para o caso da função de dano  $D_1(x)$ . Nesse sentido, é correto afirmar que a utilização do PCA piorou

os resultados obtidos pelo modelo de Florestas Aleatórias para as três funções de dano aplicadas ao elemento da treliça.

### 5.3.2.2 Redes Neurais Artificiais

#### 5.3.2.2.1 Cenário 3

Finalizado o treinamento das Redes Neurais Artificiais, coletou-se as mesmas métricas de classificação que foram utilizadas anteriormente. A Tabela 10 mostra as métricas obtidas após a tarefa de classificação, aplicando as três funções de dano individualmente mencionadas anteriormente, e sem a utilização do PCA. A Tabela 9 mostra a configuração da Rede Neural utilizada para a tarefa de localização de dano nos três conjuntos de dados. Usou-se a função de perda *Categorical Crossentropy*, o otimizador SGD, taxa de aprendizado igual a 0,0001 e 1000 épocas para a realização do treinamento da rede. Os hiper-parâmetros de todas as redes utilizadas neste trabalho foram definidos através de testes manuais.

Tabela 9 – Configuração da rede neural utilizada na tarefa de classificação sem a utilização do PCA

Camada	Função de Ativação	Neurônios	Inicialização do Kernel
Entrada	-	126	Padrão
1	LeakyReLU ( $\alpha = 0,5$ )	80	He Normal
2	LeakyReLU ( $\alpha = 0,5$ )	80	He Normal
3	LeakyReLU ( $\alpha = 0,5$ )	80	He Normal
Saída	Softmax	129	Padrão

Fonte: Autor (2022)

Tabela 10 – Métricas obtidas após o treinamento do algoritmo de Rede Neural Artificial sem a utilização do PCA para as três funções de dano propostas neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
Acurácia (%)	95,349	79,225	97,481
Precisão (%)	96,652	88,174	98,188
Recall (%)	95,524	79,752	97,556
F1 Score (%)	95,587	81,911	97,684
Tempo de execução	16min e 25s	17min e 34s	13min e 31s

Fonte: Autor (2022)

Analisando a Tabela 10, nota-se que há uma perda significativa de performance computacional e estatística quando comparadas a Tabela 7, principalmente no cenário da função de dano  $D_2(x)$ . O modelo de Redes Neurais Artificiais não conseguiu generalizar o problema com a mesma eficiência das Florestas Aleatórias, e devido a isso, degradou-se. Isso pode ter ocorrido devido ao fato das Redes Neurais Artificiais se adaptarem com



mais facilidade a problemas não-lineares, o que não é o caso para os exemplos abordados neste capítulo. Além disso, deve existir uma configuração de hiper-parâmetros da rede em questão que forneça resultados melhores, porém a otimização deles é uma tarefa árdua e não é objetivo deste trabalho.

#### 5.3.2.2.2 Cenário 4

Por outro lado, a Tabela 11 mostra o cenário onde utilizou-se o PCA na etapa de tratamento de dados. Além disso, alterou-se a configuração da rede neural mostrada na Tabela 9 e passou-se a utilizar 20 neurônios nas três camadas ocultas, exceto para o cenário  $D_2(x)$ , por ser o mais suscetível a erros, que continuou utilizando 80 neurônios em todas as camadas ocultas.

Tabela 11 – Métricas obtidas após o treinamento do algoritmo de Rede Neural Artificial com a utilização do PCA para as três funções de dano propostas neste trabalho

<b>Métrica</b>	$D_1(x)$	$D_2(x)$	$D_3(x)$
Acurácia (%)	94,264	94,225	96,047
Precisão (%)	95,371	95,371	97,184
Recall (%)	94,487	94,487	96,101
F1 Score (%)	94,523	94,523	96,414
Tempo de execução	12min e 39s	14min e 42s	12min e 4s
Nº de Componentes Principais	19	14	20

Fonte: Autor (2022)

Nota-se com facilidade que a utilização do PCA não onera as métricas estatísticas e melhora o tempo de treinamento e previsão do algoritmo. Para o caso  $D_2(x)$ , melhorou significativamente todas as métricas propostas. Além disso, percebe-se que a redução de dimensionalidade, proveniente da aplicação do PCA, surte um efeito benéfico na etapa de treinamento e previsão, isto é, o Tempo de execução, das Redes Neurais Artificiais. Porém, tal fato não ocorre com as Florestas Aleatórias.

### 5.3.3 Quantificação do Dano

Como já mencionado neste trabalho, a etapa de quantificação de dano em estruturas é uma tarefa de regressão supervisionada, que pode ser feita através de qualquer algoritmo de previsão. Este processo de regressão é definido pela Equação 5.13, em que  $k$  é o número de características ou a dimensão do vetor de  $X_j$  e  $y$  representa o dano previsto para o elemento de entrada.

$$f : \mathbb{R}^k \rightarrow \mathbb{R} \mid y = f(X_j) \quad (5.13)$$

A Equação 5.14 define o k-ésimo vetor instância utilizado para o treinamento dos modelos estatísticos, em que  $\Delta\omega$  representa a variação das frequências naturais entre o modelo estrutural com um único elemento danificado e o modelo intacto,  $\xi$  é a máxima variação da flexibilidade modal para o grau de liberdade em questão e  $elem\_dmg$  é o elemento estrutural danificado. Neste caso, o alvo da regressão foi a severidade do dano aplicado ao elemento estrutural.

$$\{\Xi\}_k = \{\Delta\omega_1, \Delta\omega_2, \dots, \Delta\omega_{10}, \xi_1, \xi_2, \dots, \xi_{116}, elem\_dmg\} \quad (5.14)$$

### 5.3.3.1 Florestas Aleatórias

#### 5.3.3.1.1 Cenário 5

Para avaliar a performance estatística das Florestas Aleatórias na etapa de quantificação do dano, fez-se uso das seguintes métricas estatísticas: Erro Percentual Absoluto Médio (MAPE), Coeficiente de Determinação ( $R^2$ ) e Similaridade por Cosseno, definidas de acordo com as equações abaixo.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - P_i}{A_i} \right| \quad (5.15)$$

em que  $n$  é o número de observações,  $P_i$  é o dano previsto,  $A_i$  o dano analítico utilizado para gerar o i-ésimo modelo danificado em questão. Quanto mais próximo de 0, melhor é o resultado da previsão.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (5.16)$$

sendo  $SS_{res}$  o somatório dos quadrados dos residuais e  $SS_{tot}$  a variância. Quanto mais próximo de 1, melhor é o resultado da previsão.

$$Similaridade = \cos(\theta) = \frac{\langle P, A \rangle}{\|P\| \|A\|} \quad (5.17)$$

no qual  $P$  e  $A$  são os vetores que guardam os danos previstos e os danos utilizados para gerar os modelos danificados, respectivamente. Quanto mais próximo de 1, melhor é o resultado da previsão, indicando que os vetores são geometricamente paralelos.

A Tabela 12 mostra as métricas obtidas após a tarefa de regressão utilizando o algoritmo de Florestas Aleatórias, aplicando as três funções de dano individualmente, e sem utilizar o PCA em nenhum cenário.

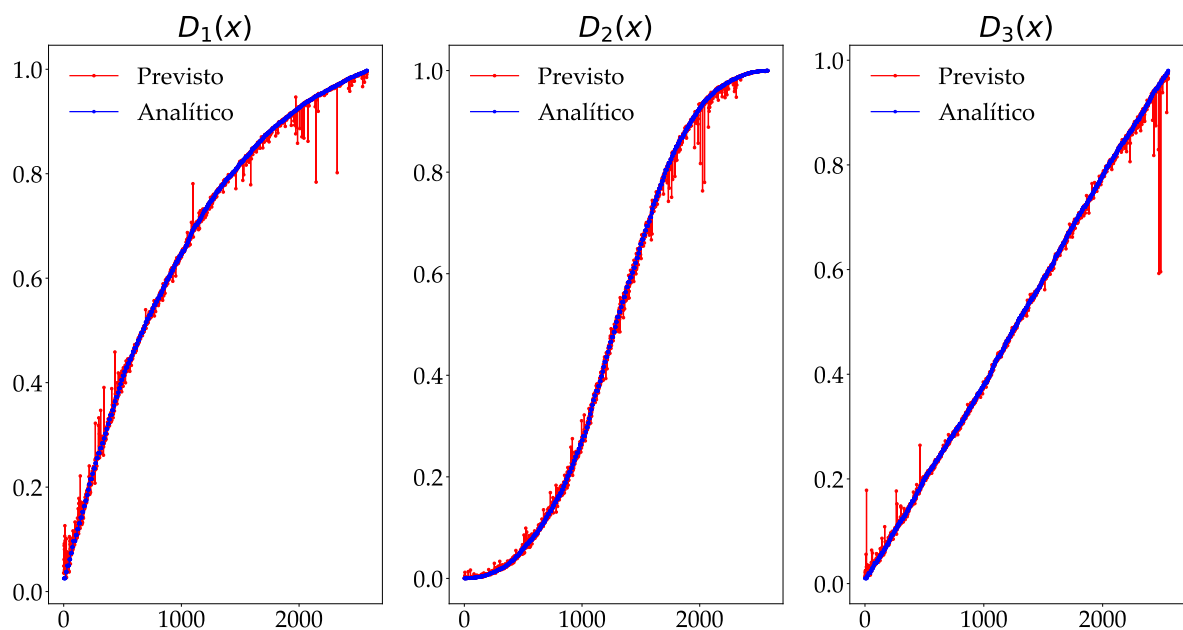
Dados os valores apresentados na Tabela 12, é correto afirmar que as Florestas Aleatórias performaram de maneira satisfatória na tarefa de quantificação do dano, com-

Tabela 12 – Métricas coletadas, para a tarefa de regressão, após o treinamento do algoritmo de Florestas Aleatórias sem a utilização do PCA para as três funções de dano propostas neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
MAPE (%)	2,52	16,80	2,98
$R^2$	0,9986	0,9994	0,9982
Similaridade	0,9999	0,9999	0,9997
Tempo de execução	37s	41,3s	38,8s

Fonte: Autor (2022)

Figura 35 – Resultados das previsões para as três funções de dano



Fonte: Autor (2022)

provando que elas podem ser utilizadas para tal atividade. A Figura 35 fornece maior embasamento, de maneira visual, para tal afirmação.

### 5.3.3.1.2 Cenário 6

A Tabela 13 mostra as métricas obtidas para a tarefa de regressão utilizando o algoritmo de Florestas Aleatórias, aplicando as três funções de dano individualmente, e aplicando-se o PCA em todos os cenários na etapa de tratamento de dados.

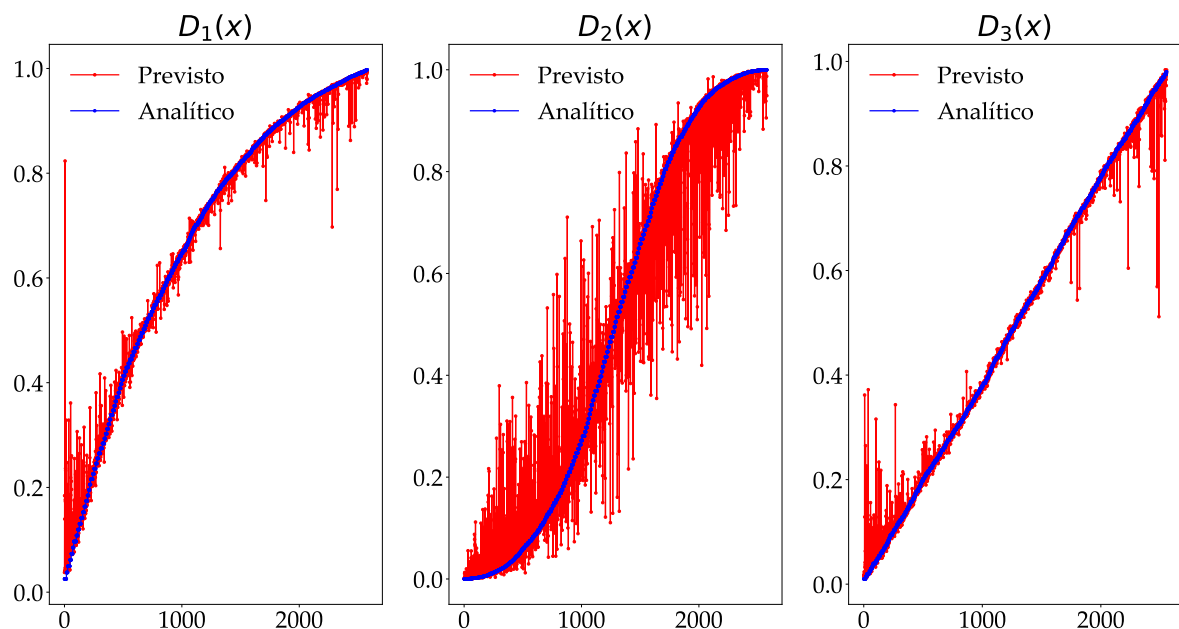
Dados os valores mostrados na Tabela 13, é correto afirmar que as Florestas Aleatórias perderam performance estatística ao se diminuir o número de dimensões do problema. Mesmo que o tempo de treinamento e previsão tenha sido reduzido de maneira vertiginosa,

Tabela 13 – Métricas coletadas, para a tarefa de regressão, após o treinamento do algoritmo de Florestas Aleatórias com a utilização do PCA para as três funções de dano propostas neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
MAPE (%)	8,82	147,72	11,74
$R^2$	0,9877	0,9424	0,9915
Similaridade	0,9895	0,9999	0,9989
Tempo de execução	5,66s	4,89s	6,21s
Nº de Componentes Principais	19	14	20

Fonte: Autor (2022)

Figura 36 – Resultados das previsões para as três funções de dano com a utilização do PCA na etapa de tratamento de dados



Fonte: Autor (2022)

as métricas estatísticas foram oneradas, principalmente no cenário  $D_2(x)$ . A Figura 36 ilustra a comparação entre os valores previstos e os valores analíticos.

### 5.3.3.2 Redes Neurais Artificiais

#### 5.3.3.2.1 Cenário 7

Finalizado o treinamento das Redes Neurais Artificiais, coletou-se as mesmas métricas de regressão que foram utilizadas anteriormente para o estudo das Florestas Aleatórias. A Tabela 14 mostra as métricas obtidas após a tarefa de quantificação de dano,

aplicando as três funções de dano individualmente mencionadas anteriormente, e sem a aplicação do PCA.

Tabela 14 – Métricas coletadas, para a tarefa de regressão, após o treinamento do algoritmo de Rede Neural Artificial sem a aplicação do PCA para as três funções de dano propostas neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
MAPE (%)	14,48	577,34	18,57
$R^2$	0,9578	0,9197	0,9721
Similaridade	0,9972	0,9853	0,9970
Tempo de execução	11min e 22s	11min e 4s	11min e 41s

Fonte: Autor (2022)

A Tabela 15 mostra a configuração da Rede Neural utilizada para a tarefa de localização de dano nos três conjuntos de dados. Usou-se a função de perda *Mean Square Error*, o otimizador SGD, taxa de aprendizado igual a 0,0001 e 1000 épocas para a realização do treinamento da rede.

Tabela 15 – Configuração da rede neural utilizada na tarefa de regressão sem a aplicação do PCA

Camada	Função de Ativação	Neurônios	Inicialização do Kernel
Entrada	-	127	Padrão
1	LeakyReLU ( $\alpha = 0,5$ )	80	He Normal
2	LeakyReLU ( $\alpha = 0,5$ )	80	He Normal
3	LeakyReLU ( $\alpha = 0,5$ )	80	He Normal
Saída	LeakyReLU ( $\alpha = 0,5$ )	1	Padrão

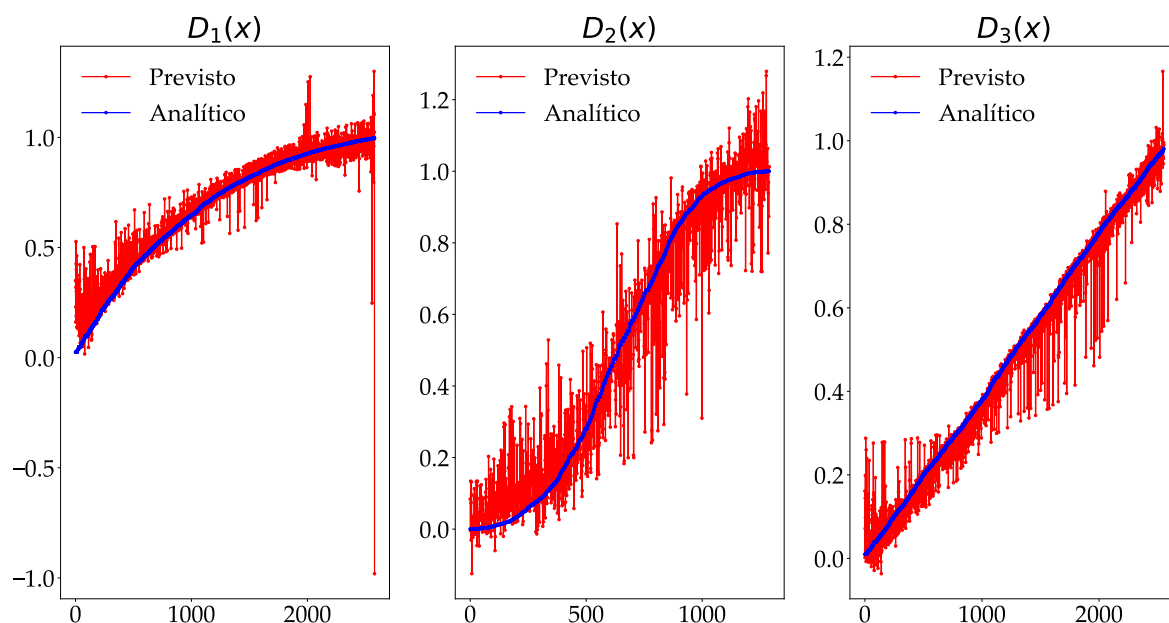
Fonte: Autor (2022)

Avaliando os resultados mostrados na Tabela 13, pode-se afirmar que o algoritmo não se adaptou tão bem quanto as Florestas Aleatórias para o problema de quantificação do dano, principalmente no cenário  $D_2(x)$ , onde o erro tornou-se gigantesco. Além disso, o tempo de execução do treinamento e da previsão foi altamente superior quando comparado ao de seu respectivo concorrente para o mesmo conjunto de dados, em todos três os cenários. Dessa forma, é correto afirmar que aplicação de Redes Neurais Artificiais, sem a utilização do PCA, onera as métricas da tarefa de regressão para esse conjunto de dados. A Figura 37 mostra o comparativo entre as previsões realizadas pelas RNNs e os valores analíticos. Devido ao grande erro apresentado pelo cenário  $D_2(x)$ , foi necessário aumentar o conjunto de treinamento para 95%, ao invés de 90%.

#### 5.3.3.2.2 Cenário 8

Por outro lado, a Tabela 16 mostra o cenário onde utilizou-se o PCA na etapa de tratamento de dados. Além disso, fez-se uso da configuração de rede neural mostrada na

Figura 37 – Resultados das previsões utilizando Redes Neurais Artificiais para as três funções de dano sem a aplicação do PCA na etapa de tratamento de dados



Fonte: Autor (2022)

tabela Tabela 15.

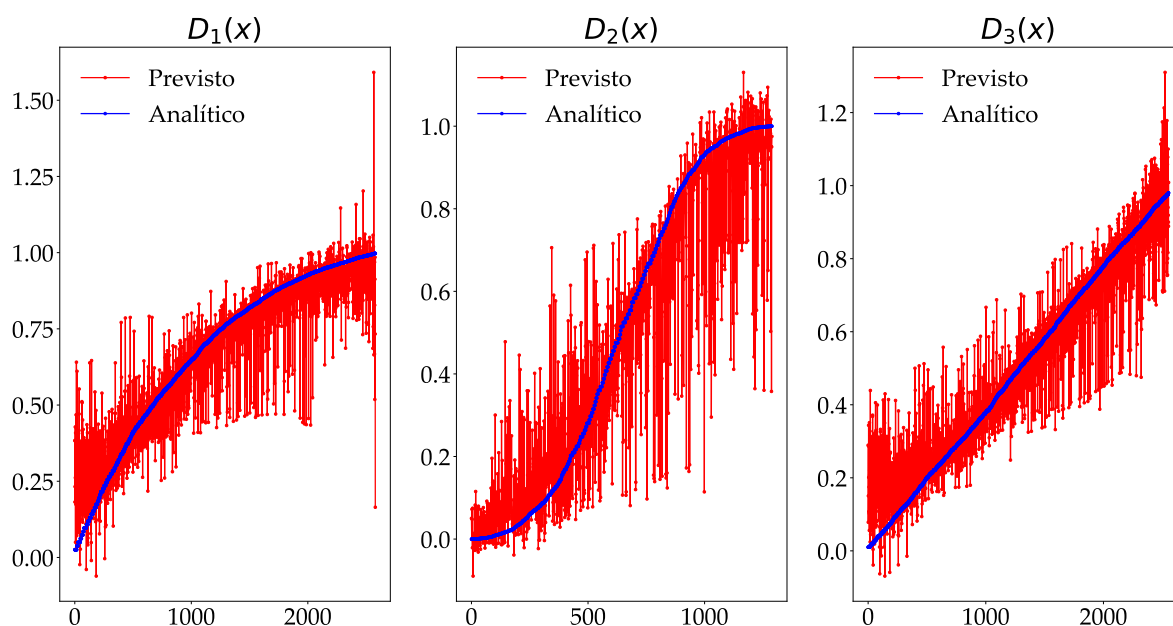
Tabela 16 – Métricas coletadas, para a tarefa de regressão, após o treinamento do algoritmo de Rede Neural Artificial com a aplicação do PCA para as três funções de dano propostas neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
MAPE (%)	29,02	951,85	38,49
$R^2$	0,8377	0,8037	0,8464
Similaridade	0,9893	0,9637	0,9806
Tempo de execução	12min e 54s	14min e 44s	12min e 13s
Nº de Componentes Principais	19	14	20

Fonte: Autor (2022)

Ao realizar a análise da tabela acima, é perceptível que houve uma grande deterioração das métricas de performance em comparação ao cenário de aplicação das Redes Neurais Artificiais para a tarefa de quantificação do dano sem a utilização do PCA. Além disso, em todos os cenários, houve um acréscimo do tempo de execução do treinamento e da previsão, o que sinaliza uma dificuldade da rede em ser treinada nesse conjunto de dados com baixa dimensão. Sendo assim, é correto afirmar que a utilização do PCA onerou todos os três cenários. A Figura 38 ilustra a comparação entre as previsões e os valores analíticos.

Figura 38 – Resultados das previsões utilizando Redes Neurais Artificiais para as três funções de dano com a aplicação do PCA na etapa de tratamento de dados



Fonte: Autor (2022)

## 5.4 Pórtico Plano

O modelo estrutural plano de pórtico, mostrado pela Figura 39, foi utilizado para realizar os estudos de localização e quantificação de dano. Os nós são rígidos e a Tabela 17 mostra as propriedades físicas e geométricas de todos os elementos.

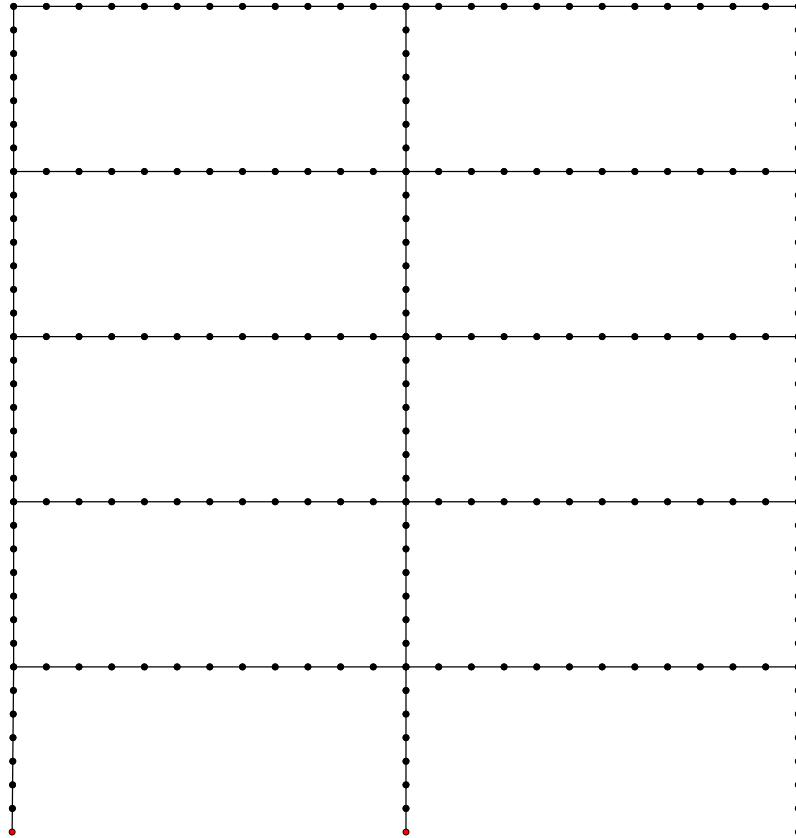
Tabela 17 – Propriedades físicas e geométricas de todos os elementos estruturais que compõe o pórtico plano estudado neste trabalho

Propriedade	Valor
Módulo de Elasticidade [E]	210 <i>GPa</i>
Área de Seção Transversal Reta [A]	0,1125 <i>m</i> <sup>2</sup>
Momento de Inércia (I)	5,8594 <sup>-4</sup> <i>m</i> <sup>4</sup>
Densidade ( $\rho$ )	7850 <i>kg/m</i> <sup>3</sup>

Fonte: Autor (2022)

Assim como no caso da treliça da seção anterior, os nós em vermelho são aqueles nós que foram travados nos três deslocamentos possíveis, isto é, translação em X, translação em Y e rotação em Z, tomando como base as direções clássicas dos eixos cartesianos.

Figura 39 – Pórtico utilizado para realizar os estudos de localização e quantificação do dano estrutural. Ele detém 218 nós, 225 elementos com rigidez axial idêntica e 665 graus de liberdade



Fonte: Autor (2022)

#### 5.4.1 Geração de dados para treinamento dos modelos estatísticos

Para a geração dos modelos danificados de pórtico, aplicou-se a mesma metodologia utilizada na seção anterior. Dessa forma, utilizou-se as funções de dano apresentadas na Figura 34, de maneira que três conjuntos de dados foram gerados, um para cada função de dano, onde cada conjunto de dados contém 45.000 instâncias, pois, para cada elemento do pórtico, aplicou-se uma severidade de dano por vez, enquanto os demais elementos permaneceram intactos.

Fez-se uso das três funções de dano para entender como os algoritmos de Florestas Aleatórias e Redes Neurais Artificiais se adaptam a esse modelo estrutural e como as métricas estatísticas são impactadas pela variação do dano. Em todos os casos, 90% do conjunto de dados foi utilizado para treinamento, enquanto o restante do conjunto foi utilizado para a validação do modelo.



## 5.4.2 Localização do Dano

Define-se o problema de localização de dano segundo a Equação 5.7. Dessa forma, a Equação 5.18 define o  $k$ -ésimo vetor instância utilizado para o treinamento dos modelos estatísticos, em que  $\Delta\omega$  representa a variação das frequências naturais entre o modelo estrutural com um único elemento danificado e o modelo intacto,  $\xi$  é a máxima variação da flexibilidade modal para o grau de liberdade em questão. Neste caso, o alvo da classificação foi o elemento danificado.

$$\{\Xi\}_k = \{\Delta\omega_1, \Delta\omega_2, \dots, \Delta\omega_{10}, \xi_1, \xi_2, \dots, \xi_{665}\} \quad (5.18)$$

### 5.4.2.1 Florestas Aleatórias

#### 5.4.2.1.1 Cenário 1

Finalizado o treinamento das Florestas Aleatórias, coletou-se as mesmas métricas de classificação que foram utilizadas na seção anterior, isto é, Equação 5.9, Equação 5.10, Equação 5.11 e Equação 5.12. A Tabela 18 mostra as métricas obtidas após a tarefa de classificação, aplicando as três funções de dano individualmente mencionadas anteriormente, e sem a utilização do PCA.

Tabela 18 – Métricas obtidas após o treinamento do algoritmo de Árvore Aleatórias sem a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
Acurácia (%)	99,011	96,332	99,733
Precisão (%)	99,664	98,520	99,837
Recall (%)	98,999	96,316	99,705
F1 Score (%)	99,011	97,698	99,741
Tempo de execução	6min e 30s	5min e 29s	6min e 32s

Fonte: Autor (2022)

Ao avaliar a tabela acima, é notável que o algoritmo de Florestas Aleatórias performou de maneira extremamente satisfatória, generalizando o problema para os três casos de dano e demonstrando que ele pode ser utilizado para a localização de dano em estruturas do tipo pórtico, assim como foi para a treliça da seção anterior.

#### 5.4.2.1.2 Cenário 2

Por outro lado, a Tabela 19 lista os resultados obtidos para o cenário onde se utilizou o PCA na etapa de tratamento de dados. Percebe-se que houve uma redução do tempo necessário para treinamento e a realização da previsão devido a redução brusca

de dimensionalidade. Além disso, as métricas estatísticas permaneceram praticamente sem alteração, o que ilustra mais uma vantagem da aplicação da Análise de Componentes Principais para este cenário.

Tabela 19 – Métricas obtidas após o treinamento do algoritmo de Árvores Aleatórias com a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
Acurácia (%)	98,833	97,289	99,556
Precisão (%)	99,606	99,583	99,756
Recall (%)	98,812	97,276	99,535
F1 Score (%)	99,108	98,248	99,601
Tempo de execução	1min e 39s	1min e 39s	1min e 49s
Nº de Componentes Principais	60	88	57

Fonte: Autor (2022)

#### 5.4.2.2 Redes Neurais Artificiais

##### 5.4.2.2.1 Cenário 3

A Tabela 20 mostra a configuração da rede neural utilizada para a tarefa de localização de dano nos três conjuntos de dados. Usou-se a função de perda *Categorical Crossentropy*, o otimizador SGD, taxa de aprendizado igual a 0,0001 e 100 épocas para a realização do treinamento da rede. Para o cenário  $D_2(x)$ , utilizou-se 300 épocas.

Tabela 20 – Configuração da rede neural utilizada na tarefa de classificação sem a utilização do PCA para o pórtico estudado

Camada	Função de Ativação	Neurônios	Inicialização do Kernel
Entrada	-	655	Padrão
1	LeakyReLU ( $\alpha = 0,5$ )	20	He Normal
2	LeakyReLU ( $\alpha = 0,5$ )	20	He Normal
3	LeakyReLU ( $\alpha = 0,5$ )	20	He Normal
Saída	Softmax	225	Padrão

Fonte: Autor (2022)

Finalizado o treinamento das Redes Neurais Artificiais, coletou-se as mesmas métricas de classificação que foram utilizadas anteriormente. A Tabela 21 mostra as métricas obtidas após a tarefa de classificação, aplicando as três funções de dano individualmente mencionadas anteriormente, e sem a utilização do PCA.

Analisando a Tabela 21, nota-se que não há uma perda significativa de performances computacional e estatística quando comparadas a Tabela 18. O modelo de Rede Neural conseguiu generalizar o problema com a eficiência pareada à das Florestas Aleatórias,

Tabela 21 – Métricas alcançadas após o treinamento do algoritmo de Redes Neurais Artificiais sem a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho

<b>Métrica</b>	$D_1(x)$	$D_2(x)$	$D_3(x)$
Acurácia (%)	98,578	97,289	98,844
Precisão (%)	98,657	96,583	98,891
Recall (%)	98,525	90,276	98,800
F1 Score (%)	98,539	92,248	98,775
Tempo de execução	2min e 47s	9min e 43s	2min e 52s

Fonte: Autor (2022)

principalmente para o caso  $D_2(x)$ , o que não havia acontecido quando estudou-se o modelo de treliça. Nota-se também que os algoritmos conseguiram convergir para ótimos resultados sem a necessidade de longos períodos de treinamento, o que não é trivial de entender, pois o conjunto de treinamento aqui é maior quando comparado ao conjunto de treinamento utilizado para os cenários onde analisou-se a treliça em aço.

#### 5.4.2.2.2 Cenário 4

A Tabela 22 mostra as métricas obtidas para o caso onde se aplicou o PCA na etapa de tratamento de dados.

Tabela 22 – Métricas alcançadas após o treinamento do algoritmo de Redes Neurais Artificiais com a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho

<b>Métrica</b>	$D_1(x)$	$D_2(x)$	$D_3(x)$
Acurácia (%)	98,467	92,244	98,489
Precisão (%)	98,662	96,362	98,634
Recall (%)	98,455	92,216	98,408
F1 Score (%)	98,484	93,662	98,442
Tempo de execução	2min e 42s	8min e 27s	2min e 32s
Nº de Componentes Principais	60	88	57

Fonte: Autor (2022)

É perceptível que a aplicação do PCA rendeu uma leve diminuição do tempo de execução do treinamento e da previsão das Redes Neurais Artificiais. Além disso, a diminuição da dimensionalidade não afetou significativamente as métricas estatísticas, o que pode apontar um bom comportamento dos dados.

### 5.4.3 Quantificação do Dano

Como já mencionado neste trabalho, a etapa de quantificação de dano em estruturas é uma tarefa de regressão supervisionada, que pode ser feita através de qualquer algoritmo de previsão. Este processo de regressão é definido pela Equação 5.13.

A Equação 5.19 define o  $k$ -ésimo vetor instância utilizado para o treinamento dos modelos estatísticos, em que  $\Delta\omega$  representa a variação das frequências naturais entre o modelo estrutural com um único elemento danificado e o modelo intacto,  $\xi$  é a máxima variação da flexibilidade modal para o grau de liberdade em questão e  $elem\_dmg$  é o elemento estrutural danificado. Neste caso, o alvo da regressão foi a severidade do dano aplicado ao elemento estrutural.

$$\{\Xi\}_k = \{\Delta\omega_1, \Delta\omega_2, \dots, \Delta\omega_{10}, \xi_1, \xi_2, \dots, \xi_{665}, elem\_dmg\} \quad (5.19)$$

#### 5.4.3.1 Florestas Aleatórias

##### 5.4.3.1.1 Cenário 5

Para avaliar a performance estatística das Florestas Aleatórias na etapa de quantificação do dano no pórtico proposto, fez-se uso das métricas de erro utilizadas na seção anterior, isto é, Equação 5.15, Equação 5.16, e Equação 5.17.

A Tabela 23 mostra as métricas obtidas após a tarefa de regressão utilizando o algoritmo de Florestas Aleatórias, aplicando as três funções de dano individualmente, e sem utilizar o PCA em nenhum cenário.

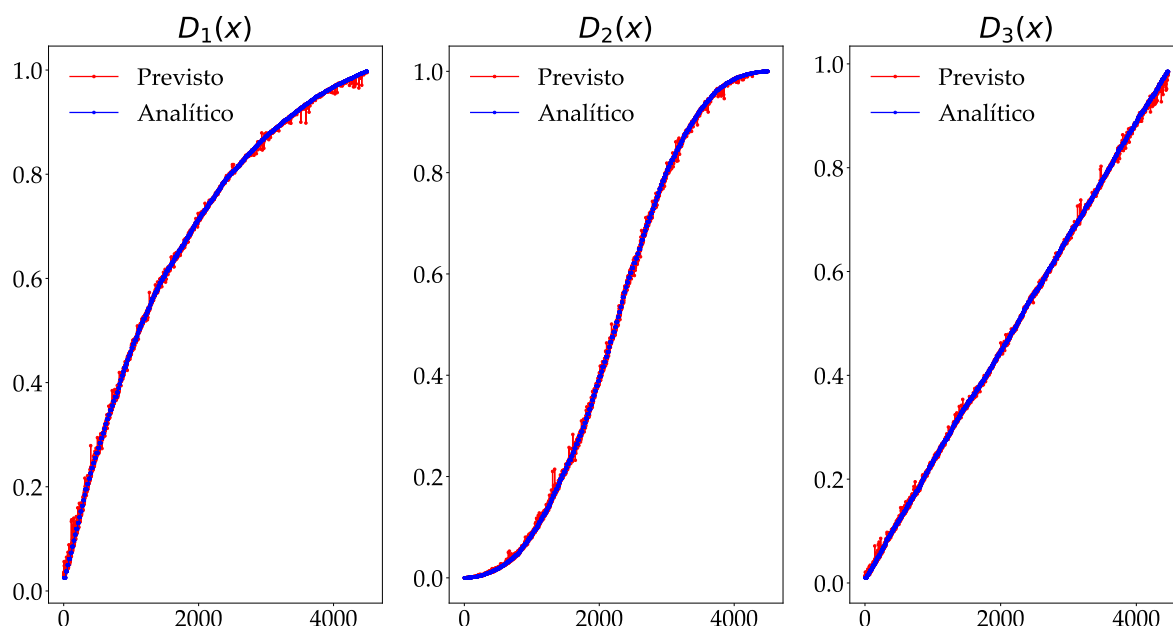
Tabela 23 – Métricas obtidas, para a tarefa de regressão, após o treinamento do algoritmo de Florestas Aleatórias sem a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
MAPE (%)	1,07	2,58	1,33
$R^2$	0,9997	0,9998	0,9998
Similaridade	0,9999	0,9999	0,9999
Tempo de execução	5min e 55s	8min e 15s	6min e 9s

Fonte: Autor (2022)

Percebe-se, ao analisar a tabela acima, que o algoritmo foi bem sucedido em prever o dano nos elementos estruturais do pórtico. Todas as métricas demonstram a boa capacidade das Florestas Aleatórias em se adaptar ao problema com múltiplos graus de liberdade. A Figura 40 ilustra o comparativo entre as previsões e os danos analíticos utilizados para gerar os modelos danificados. Nota-se com facilidade a boa qualidade das previsões realizadas para os três cenários.

Figura 40 – Resultados das previsões utilizando Florestas Aleatórias, sem a aplicação do PCA, para as três funções de dano propostas para o pórtico estudado neste trabalho



Fonte: Autor (2022)

#### 5.4.3.1.2 Cenário 6

Ao analisar a Tabela 24, nota-se que a aplicação do PCA não onerou o processo de previsão e manteve todas as métricas de maneira similar às mostradas na Tabela 23. Além disso, o uso da técnica promoveu uma redução significativa no tempo de execução do treinamento e da previsão, mostrando que a Análise de Componentes Principais reduziu intensamente o número de dimensões do problema. Dessa forma, é correto afirmar que as Florestas Aleatórias conseguem se adaptar ao problema com dimensionalidade reduzida sem perder capacidade de realizar previsões corretamente.

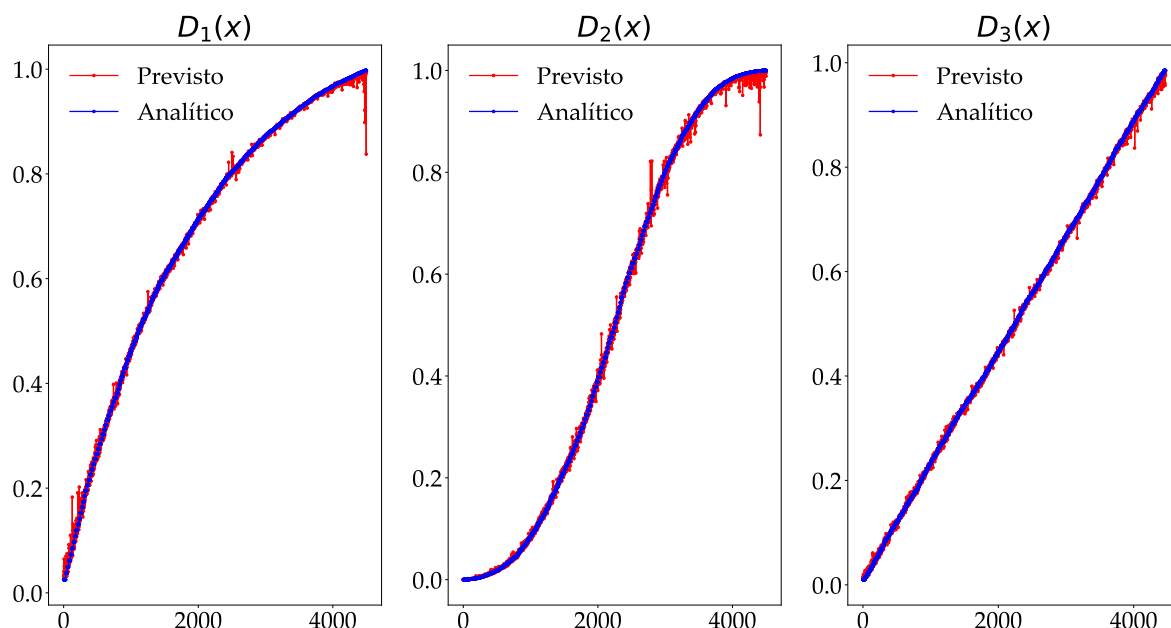
Tabela 24 – Métricas alcançadas, para a tarefa de regressão, após o treinamento do algoritmo de Florestas Aleatórias com a aplicação do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
MAPE (%)	1,40	3,79	1,60
$R^2$	0,9995	0,9997	0,9998
Similaridade	0,9999	0,9999	0,9999
Tempo de execução	32,2s	1min	33,6s
Nº de Componentes Principais	60	88	57

Fonte: Autor (2022)

Por fim, a Figura 41 ilustra o comparativo entre as previsões realizadas e os danos analíticos utilizados para a geração dos modelos danificados, onde em todos os cenários aplicou-se o PCA.

Figura 41 – Resultados das previsões utilizando Florestas Aleatórias, com a aplicação do PCA, para as três funções de dano propostas para o pórtico estudado neste trabalho



Fonte: Autor (2022)

#### 5.4.3.2 Redes Neurais Artificiais

##### 5.4.3.2.1 Cenário 7

A Tabela 25 mostra a configuração da rede neural utilizada para a tarefa de localização de dano nos três conjuntos de dados. Usou-se a função de perda *Mean Square Error*, o otimizador SGD, taxa de aprendizado igual a 0,0001 e 100 épocas para a realização do treinamento da rede. Para o cenário  $D_2(x)$ , utilizou-se 80 neurônios nas três camadas ocultas e 300 épocas.

Finalizado o treinamento das Redes Neurais Artificiais, coletou-se as mesmas métricas de regressão que foram utilizadas anteriormente. A Tabela 26 mostra as métricas obtidas após a tarefa de classificação, aplicando as três funções de dano individualmente mencionadas anteriormente, e sem a utilização do PCA.

Ao analisar a Tabela 26, nota-se que houve uma ligeira degradação das métricas nos cenários  $D_1(x)$  e  $D_3(x)$  quando comparado a Tabela 23, mas o erro encontra-se em valores aceitáveis, isto é, abaixo de 25%. Tal fato não ocorre para o caso  $D_2(x)$ , que apresentou

Tabela 25 – Configuração da rede neural utilizada na tarefa de regressão sem a utilização do PCA para o pórtico estudado

Camada	Função de Ativação	Neurônios	Inicialização do Kernel
Entrada	-	656	Padrão
1	LeakyReLU ( $\alpha = 0,5$ )	20	He Normal
2	LeakyReLU ( $\alpha = 0,5$ )	20	He Normal
3	LeakyReLU ( $\alpha = 0,5$ )	20	He Normal
Saída	LeakyReLU ( $\alpha = 0,5$ )	1	Padrão

Fonte: Autor (2022)

Tabela 26 – Métricas alcançadas após o treinamento do algoritmo de Redes Neurais Artificiais, para a tarefa de regressão, sem a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
MAPE (%)	3,23	62,19	7,26
$R^2$	0,9995	0,9992	0,9939
Similaridade	0,9997	0,9998	0,9992
Tempo de execução	2min e 37s	9min e 49s	2min e 27s

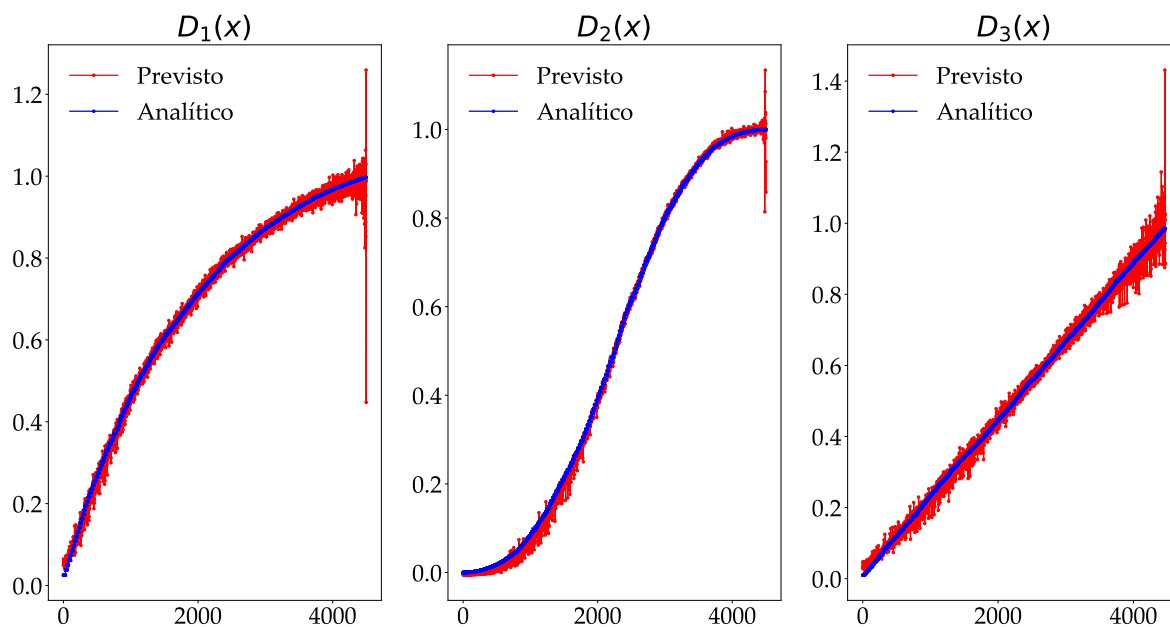
Fonte: Autor (2022)

um grande aumento do MAPE. Sendo assim, é correto afirmar que as Redes Neurais Artificiais não performaram tão bem quanto as Florestas Aleatórias, apesar de conseguirem quantificar o dano estrutural. Além disso, as RNNs apresentaram uma redução no tempo de treinamento e previsão para os cenários  $D_1(x)$  e  $D_3(x)$ . A Figura 42 abaixo mostra a comparação entre as previsões e os danos analíticos.

#### 5.4.3.2.2 Cenário 8

Após a aplicar o PCA na etapa de tratamento de dados, obtiveram-se os resultados mostrados na Tabela 27. Percebe-se uma leve degradação das métricas para os cenários  $D_1(x)$  e  $D_3(x)$ . Para eles, a Análise Principal de Componentes não obteve ganho relevante no tempo de execução do treinamento e da previsão. Para o caso  $D_3(x)$ , nota-se um aumento do MAPE e uma redução do tempo de execução do treinamento e previsão, o que indica que a rede neural não conseguiu se adaptar para o caso onde se reduziu a dimensão do problema. O MAPE demonstrou-se extremamente alto, o que impossibilita o seu uso em aplicações práticas. A Figura 43 mostra o comparativo entre as previsões e os danos analíticos.

Figura 42 – Resultados das previsões utilizando Redes Neurais Artificiais, sem a aplicação do PCA, para as três funções de dano propostas para o pórtico estudado neste trabalho



Fonte: Autor (2022)

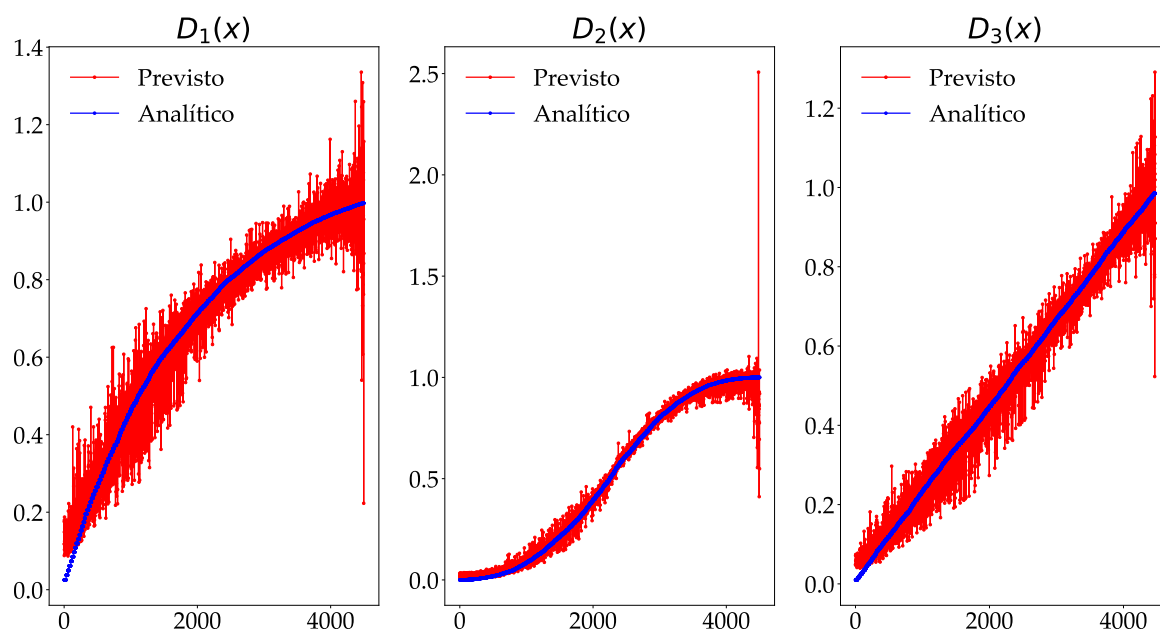
Tabela 27 – Métricas obtidas após o treinamento do algoritmo de Redes Neurais Artificiais, para a tarefa de regressão, com a utilização do PCA para as três funções de dano propostas para o pórtico estudado neste trabalho

Métrica	$D_1(x)$	$D_2(x)$	$D_3(x)$
MAPE (%)	13,82	280,01	17,35
$R^2$	0,9650	0,9887	0,9725
Similaridade	0,9976	0,9979	0,9966
Tempo de execução	2min e 19s	7min e 21s	2min e 21s
Nº de Componentes Principais	60	88	57

Fonte: Autor (2022)



Figura 43 – Resultados das previsões utilizando Redes Neurais Artificiais, com a aplicação do PCA, para as três funções de dano propostas para o pórtico estudado neste trabalho



Fonte: Autor (2022)

## 6 Conclusão

Monitoramento de Saúde Estrutural é o ramo da Engenharia Estrutural que une a análise Dinâmica de Estruturas com Sistemas Computacionais Inteligentes. A implementação de um sistema de monitoramento em tempo-real é benéfico e acarreta na mitigação de eventuais falhas estruturais, na otimização de recursos financeiros e, principalmente, no prolongamento da vida útil da estrutura, pois a integridade do sistema estrutural passa a ser conhecida em tempo-real. A combinação do Método da Variação da Flexibilidade Modal com técnicas estatísticas capazes de realizar previsões corretas é extremamente válido para alcançar o objetivo de manter em segurança os usuários de uma determinada estrutura. Nesse sentido, estudou-se a aplicação do Método da Variação da Flexibilidade Modal para a geração de dados sintéticos, que serviram para o treinamento de algoritmos de Florestas Aleatórias e de Redes Neurais Artificiais.

A partir do programa de Análise Dinâmica de Estruturas implementado neste trabalho, observou-se que as Florestas Aleatórias conseguiram ótima performance em todos os cenários de teste para as tarefas de localização e quantificação do dano, principalmente quando não se aplicou a Análise de Componentes Principais para reduzir a dimensionalidade do conjunto de treinamento.

Além disso, avaliou-se o poder de adaptabilidade das Redes Neurais Artificiais, que se demonstraram ser ferramentas viáveis para o Monitoramento de Saúde Estrutural, mas com performance inferior às Florestas Aleatórias. Como modelos de Aprendizado Profundo, as redes devem ter seus hiper-parâmetros otimizados para que a sua capacidade máxima de generalização seja alcançada. Entretanto, como não era a ênfase deste trabalho, optou-se por utilizar configurações mais simples, limitando-se a três camadas ocultas, por exemplo. Notou-se também que as Redes Neurais Artificiais enfrentaram dificuldades em generalizar conjuntos de treinamento que tinham pouca dimensão, principalmente para a função de dano  $D_2(x)$  no estudo do modelo de treliça. Em adição, o uso da Análise de Componentes Principais demonstrou-se ser eficaz na redução de dimensionalidade, que promoveu a redução do tempo de treinamento e de previsão em alguns casos de teste detalhados no capítulo anterior. Apesar de onerar as métricas estatísticas em alguns casos, ainda é uma ferramenta importante para um monitoramento rápido.

O fato das Florestas Aleatórias performarem melhor do que as Redes Neurais Artificiais prova que os algoritmos clássicos podem performar igualmente ou superior a algoritmos de Aprendizado Profundo, apesar das redes estarem nos holofotes do mundo da tecnologia. Além disso, as Florestas Aleatórias demonstraram uma explicabilidade melhor do que as Redes Neurais Artificiais, que a depender de sua complexidade, se tornam

modelos "caixa preta".

Por fim, é correto concluir que o Monitoramento de Saúde Estrutural baseado em Estatística Inferencial Computacional pode trazer diversas vantagens para a infraestrutura que está sendo analisada. Além disso, abre-se o leque de possibilidades em aplicações de Cidades Inteligentes, algo que está sendo cada vez mais disseminado no mundo real. Como os seres humanos geram dados a todo instante e cada vez mais, algoritmos de Aprendizado de Máquina e sistemas inteligentes terão sua demanda aumentada nos próximos anos.

Abaixo encontram-se alguns pontos de melhoria no software desenvolvido e propostas para trabalhos futuros:

1. Desenvolver um módulo de Análise Dinâmica de Estruturas 3D e submeter o modelo tridimensional a detecção de dano;
2. Implementar novos algoritmos de Aprendizado de Máquina para realizar a localização e quantificação de dano, como: Regressão Linear e Máquinas de Vetor de Suporte;
3. Implementar novas configurações de Redes Neurais Artificiais e otimizar seus hiperparâmetros com a finalidade de obter melhores resultados;
4. Estudar e criar algoritmos de detecção de anomalia para a localização de dano estrutural;
5. Utilizar Redes Neurais Convolucionais para construir um sistema de Monitoramento de Saúde Estrutural;
6. Utilizar outros métodos para gerar novos conjuntos de treinamento para os algoritmos estatísticos, como os Métodos Baseados na Flexibilidade Medida Dinamicamente e Métodos Energéticos.
7. Testar a capacidade de localização e quantificação de dano de algoritmos computacionais inteligentes submetidos a modelos estruturais com mais de um elemento danificado simultaneamente.

## Referências

- BAKHARY, N.; HAO, H.; DEEKS, A. J. Damage detection using artificial neural network with consideration of uncertainties. *Engineering Structures*, n. 29, p. 2806–2815, 2007. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0141029607000417>>. Acesso em: 14 fev. 2022. Citado na página 22.
- BERMAN, A.; FLANNELLY, W. G. Theory of incomplete models of dynamic structures. *American Institute of Aeronautics and Astronautics Journal*, v. 9, n. 8, 1971. Disponível em: <<https://doi.org/10.2514/3.49950>>. Acesso em: 02 jun. 2022. Citado 2 vezes nas páginas 46 e 47.
- BRASIL, R. M. L. R. F.; SILVA, M. A. *Introdução à dinâmica das estruturas para a engenharia civil*. 2. ed. Rua Pedroso Alvarenga, 1245, 4º Andar, 04531-934, São Paulo - SP, Brasil: Editora Edgard Blücher Ltda., 2015. Citado na página 44.
- CARRILLO, O. J. B. *Detecção de dano a partir da resposta dinâmica da estrutura: estudo analítico com aplicação a estruturas do tipo viga*. Tese (Doutorado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, São Paulo, Brasil, 2004. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/18/18134/tde-05042016-135235/pt-br.php>>. Acesso em: 25 mar. 2022. Citado 3 vezes nas páginas 21, 25 e 47.
- CLOUGH, R. W.; PENZIEN, J. *Dynamics of Structures*. 3. ed. 1995 University Ave. Berkeley, CA 94704, USA: Computers & Structures, Inc, 2003. Citado na página 27.
- CRAIG, R. R.; KURDILA, A. J. *Fundamentals of Structural Dynamics*. 2. ed. Hoboken. New Jersey. USA: John Wiley and Sons. Inc., 2006. Citado na página 39.
- FALCÃO, V. B. *Aplicação de metodologias de reconhecimento de padrões para detecção de dano em vigas de concreto*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Alagoas, 2017. Disponível em: <[https://sucupira.capes.gov.br/sucupira/public/consultas/coleta/trabalhoConclusao/viewTrabalhoConclusao.jsf?popup=true&id\\_trabalho=5554250#](https://sucupira.capes.gov.br/sucupira/public/consultas/coleta/trabalhoConclusao/viewTrabalhoConclusao.jsf?popup=true&id_trabalho=5554250#)>. Citado na página 24.
- FARRAR, C.; WORDEN, K. *Structural Health Monitoring: a Machine Learning Perspective*. 1. ed. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: John Wiley & Sons Ltd, 2013. Citado na página 20.
- FARRAR, C. R.; WORDEN, K. An introduction to structural health monitoring. In: \_\_\_\_\_. *New Trends in Vibration Based Structural Health Monitoring*. Springer Vienna, 2010. p. 1–17. Disponível em: <[https://doi.org/10.1007/978-3-7091-0399-9\\_1](https://doi.org/10.1007/978-3-7091-0399-9_1)>. Acesso em: 2 jan. 2012. Citado na página 23.
- FENG, D.; FENG, M. Q. Computer vision for shm of civil infrastructure: From dynamic response measurement to damage detection – a review. *Engineering Structures*, v. 156, p. 105–117, 2018. Disponível em: <<https://doi.org/10.1016/j.engstruct.2017.11.018>>. Acesso em: 25 mai. 2022. Citado na página 23.

- F.R.S., K. P. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor & Francis, v. 2, n. 11, p. 559–572, 1901. Disponível em: <<https://doi.org/10.1080/14786440109462720>>. Citado na página 65.
- GILLICH, N. et al. Beam damage assessment using natural frequency shift and machine learning. *Sensors*, v. 22, n. 1118, 2022. Disponível em: <<https://www.mdpi.com/1424-8220/22/3/1118>>. Acesso em: 17 jan. 2022. Citado na página 21.
- GÉRON, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2. ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472.: O'Reilly Media, Inc., 2019. Citado 6 vezes nas páginas 49, 54, 57, 58, 65 e 66.
- HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural Computation*, v. 18, p. 1527–1554, 2006. Disponível em: <[10.1162/neco.2006.18.7.1527](https://doi.org/10.1162/neco.2006.18.7.1527)>. Acesso em: 05 jun. 2022. Citado na página 49.
- HO, T. K. Random decision forests. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. [S.l.: s.n.], 1995. v. 1, p. 278–282 vol.1. Acesso em: 05 jun. 2022. Citado na página 56.
- JANAA, D. et al. Cnn and convolutional autoencoder (cae) based real-time sensor fault detection, localization, and correction. *Mechanical Systems and Signal Processing*, v. 169, n. 108723, 2022. Disponível em: <<https://doi.org/10.1016/j.ymssp.2021.108723>>. Acesso em: 15 mar. 2022. Citado na página 22.
- JÚNIOR, W. J. de L. *Identificação de danos em vigas via análise modal*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Alagoas, 2020. Citado na página 23.
- KASSIMALI, A. *Matrix Analysis of Structures*. 2. ed. 200 First Stamford Place, Suite 400 Stamford, CT 06902 USA: Cengage Learning, 2012. Citado na página 26.
- LYNCH, J. An overview of wireless structural health monitoring for civil structures. *Philosophical Transactions of the Royal Society A*, n. 365, p. 345–372, 2017. Disponível em: <<https://royalsocietypublishing.org>>. Acesso em: 23 mar. 2022. Citado na página 21.
- MAIA, R. A. *Detecção de dano estrutural em estruturas planas utilizando redes neurais*. Dissertação (Dissertação de Mestrado) — Universidade de Brasília, 2006. Disponível em: <<http://www.repositorio.ufal.br/jspui/handle/riufal/6804>>. Citado 5 vezes nas páginas 11, 25, 73, 74 e 75.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics volume*, v. 5, p. 115–133, 1943. Disponível em: <<https://doi.org/10.1007/BF02478259>>. Acesso em: 05 jun. 2022. Citado na página 58.
- MINSKY, M.; PAPER, S. A. *Perceptrons: An Introduction to Computational Geometry*. 1. ed. The MIT Press, 2017. Disponível em: <<https://doi.org/10.7551/mitpress/11301.001.0001>>. Citado na página 59.

- MORALES, J. D. V. *Detecção de dano em estruturas via inteligência computacional e análise dinâmica*. Tese (Doutorado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, São Paulo, Brasil, 2012. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/18/18134/tde-11122012-082530/pt-br.php>>. Acesso em: 27 abr. 2022. Citado na página 22.
- PANDEY, A. K.; BISWAS, M. Damage detection in structures using changes in flexibility. *Journal of Sound and Vibration*, v. 169, n. 1, p. 3–17, 1994. Disponível em: <<https://doi.org/10.1006/jsvi.1994.1002>>. Acesso em: 02 jun. 2022. Citado na página 47.
- PAZ, M.; KIM, Y. H. *Structural Dynamics: Theory and Computation*. 5. ed. Gewerbestrasse 11, 6330 Cham, Switzerland: Springer Nature Switzerland AG, 2019. Citado na página 26.
- RAO, S. S. *Mechanical Vibrations*. 6. ed. Harlow CM17 9NA United Kingdom: Pearson Education, Inc., 2018. Citado na página 41.
- RASTIN, Z.; AMIRI, G.; DARVISHAN, E. Unsupervised structural damage detection technique based on a deep convolutional autoencoder. *Shock and Vibration*, v. 2021, n. 6658575, 2022. Disponível em: <<https://www.hindawi.com/journals/sv/2021/6658575/>>. Acesso em: 25 mai. 2022. Citado na página 21.
- ROJAS, J. et al. Identification of external loads in mechanical systems through heuristic-based optimization methods and dynamic responses. *Latin American Journal of Solids and Structures*, v. 1, p. 297–318, 01 2004. Citado 3 vezes nas páginas 10, 11 e 72.
- RUMELHART, D. E.; MCCLELLAND, J. L. Learning internal representations by error propagation. In: \_\_\_\_\_. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. [S.l.: s.n.], 1987. v. 1, p. 318–362. Citado na página 60.
- RYTTER, A. *Vibration Based Inspection of Civil Engineering Structures*. Tese (Doutorado) — Dept. of Building Technology and Structural Engineering, Aalborg University, Aalborg, Dinamarca, 1993. Disponível em: <<https://vbn.aau.dk/en/publications/vibrational-based-inspection-of-civil-engineering-structures>>. Acesso em: 25 mar. 2022. Citado na página 19.
- SONG, S.; CHAUDHURI, K.; SARWATE, A. Stochastic gradient descent with differentially private updates. In: . [S.l.: s.n.], 2013. p. 245–248. ISBN 978-1-4799-0248-4. Citado na página 63.
- SORIANO, H. L. *Introdução à dinâmica das estruturas*. 1. ed. Rua Quintana, 753 – 8º andar, 04569-011 – Brooklin – São Paulo – SP – Brasil: Elsevier Editora Ltda., 2014. Citado na página 33.
- SURYAKANTHI, T. Evaluating the impact of gini index and information gain on classification using decision tree classifier algorithm\*. *International Journal of Advanced Computer Science and Applications*, v. 11, n. 2, 2020. Disponível em: <<http://dx.doi.org/10.14569/IJACSA.2020.0110277>>. Acesso em: 05 jun. 2022. Citado na página 53.
- SVENDSEN, B. et al. A data-based structural health monitoring approach for damage detection in steel bridges using experimental data. *Journal of Civil Structural Health Monitoring*, v. 1, n. 12, p. 101–115, 2022. Disponível em:

<<https://link.springer.com/article/10.1007/s13349-021-00530-8>>. Acesso em: 05 fev. 2022. Citado na página 20.

TAN, Z. et al. Detecting damage in steel beams using modal strain energy based damage index and artificial neural network. *Engineering Failure Analysis*, n. 79, p. 253–262, 2017. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S1350630717300523>>. Acesso em: 17 jan. 2022. Citado na página 21.

YANG, Y. et al. An improved direct stiffness calculation method for damage detection of beam structures. *Structural Control and Health Monitoring*, v. 20, n. 5, p. 835–851, 2012. Disponível em: <<https://doi.org/10.1002/stc.1503>>. Acesso em: 15 fev. 2022. Citado na página 22.

ZHONG, P.; FUKUSHIMA, M. Regularized nonsmooth newton method for multi-class support vector machines. *Optimization Methods and Software*, v. 22, n. 1, p. 225–236, 2007. Disponível em: <<https://doi.org/10.1080/10556780600834745>>. Acesso em: 05 jun. 2022. Citado na página 52.