

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA AEROESPACIAL

ANA CAROLINA RIBEIRO ALBINO

SISTEMA DE APRENDIZADO DE MÁQUINA COMO APOIO AO MONITORAMENTO
DE SAÚDE ESTRUTURAL

Joinville
2022

ANA CAROLINA RIBEIRO ALBINO

SISTEMA DE APRENDIZADO DE MÁQUINA COMO APOIO AO MONITORAMENTO
DE SAÚDE ESTRUTURAL

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia Aeroespacial, no curso Engenharia Aeroespacial da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Dr. Gian Ricardo Berkenbrock

Coorientador: Dr. Carlos Eduardo de Souza

Joinville
2022

RESUMO

A manutenção de aeronaves é uma atividade essencial para garantir o correto funcionamento de todos os sistemas que tornam sua operação possível. Desde o surgimento da aviação comercial, a checagem dos componentes é estruturada em intervalos dentro e fora de fase, que tem como base a data de entrada da aeronave em operação (T-0), tal planejamento tem gastos expressivos, que podem chegar a 15% do custo operacional (VIEIRA; LOURES, 2016), seu rígido controle não impede que componentes falhem e causem acidentes (SU; YE, 2009). O monitoramento da saúde estrutural (MSE) é uma vertente do condition based maintenance (CBM), que torna a manutenção embasada na mudança dos parâmetros físicos da estrutura e não por tempo de voo. A aplicação do MSE traz consigo potencial redução de custo de manutenção, podendo chegar a 30% (DONG; KIM, 2018), além de aumentar a confiabilidade dos sistemas. Ao considerar esse cenário potencial de evolução, o presente trabalho utiliza dados, de saúde estrutural, medidos pela universidade holandesa TuDelft, para criar um modelo de aprendizado de máquina (AP), por meio da criação de um multi layer perceptron, e prever o surgimento de falhas críticas, classificando a estrutura em saudável ou danificada. Devido à complexidade em analisar o padrão das séries temporais do experimento, o aprendizado de máquina, por meio de redes neurais de classificação, são uma opção para dar robustez ao processo. Por conta da localização espacial dos sensores é possível realizar a triangulação dos resultados, para localizar a falha e eliminar falsos positivos. A acurácia geral do código é de 90 %.

Palavras-chave: Manutenção de aeronaves. Aprendizado de máquina. Monitoramento da saúde estrutural. Causa raiz. Rede neural.

ABSTRACT

Aircraft maintenance is an essential activity to ensure the safety of all systems, that make its operation possible. Since the beginning of commercial aviation, it is structured at intervals, that can be in or out of phase, both have the maintenance start date (T-0) in common, which is related to when the aircraft has entered into service. This process has expressive costs, up to 15% of operational amount (VIEIRA; LOURES, 2016), this rigid control does not avoid accidents from happening (SU; YE, 2009). The structure health monitoring (SHM) is a variation of the condition based maintenance (CBM), both indicate the importance to do the maintenance whenever the structure has a change in its physical parameters. The SHM has a potential for cost reduction, up to 30% (DONG; KIM, 2018), and also increase system reliability. Considering this scenario, and its evolutionary potential, this project uses data of structural health, measured by the Dutch university, TuDelft, to create a machine learning model (ML), by a multi layer perceptron, and predict the beginning of critical fault, classifying the structure in healthy or damaged. Due to time series analysis, to identify a fault pattern, the classification neural network is a great alternative to give robustness for this process. Due to spatial location of sensors, it is possible to detect the location of the fault and reduce false positive results. The overall accuracy of the code is 90%.

Keywords: Aircraft maintenance. Machine learning. Structure health monitoring. Root cause. Neural network.

AGRADECIMENTOS

Agradeço o corpo docente dessa instituição e pelos profissionais dedicados que cruzaram meu caminho e tanto me ensinaram, à todas as pessoas que me guiaram a ser um ser humano melhor, sem dúvida, esse conhecimento é o mais valioso de toda a minha jornada.

Sou grata aos meus pais por essa oportunidade, e a todas as pessoas que ajudaram ela a se concretizar. Deixo meu muito obrigada a todas as bolsas recebidas e a todos os professores que me escolheram para seus projetos, em especial Victor Simões Barbosa, pelo período como sua monitora, Andréa Holz Pfutzenreuter, pelo projeto de pesquisa me abriu tantas portas, inclusive com experiências em outras instituições de ensino e congressos.

Tantos outros professores participaram de minha formação técnica, deixo meus mais sinceros agradecimentos ao Rafael Gigena Cuenca, por ter me acompanhado nas mais diversas atividades e por ser um exímio orientador de iniciação científica, equipe de competição e ombro amigo. Gratulo meu orientador de TCC, Gian Ricardo Berkenbrock, por aceitar minhas ideias com tanta prontidão, fazer eu me esforçar cada vez mais e evoluir com a construção desse trabalho.

O início da minha vida profissional não poderia ter sido em melhor ambiente, muito obrigada à Caroline Bertelli e Graziella Martins, por todos os ensinamentos técnicos e pessoais, todos os projetos a mim concedidos foram incríveis e sempre lembrarei dessa equipe com muito carinho. Obrigada a Embraer, pela oportunidade de aprofundar meus conhecimentos sobre aviação, agradeço aos meus parceiros de estágio Júlia e Leo, por suportar todas as minhas iniciativas e à Juliana por torná-las palpáveis, obrigada aos três por toda a proteção e carinho.

Aos meus amigos, me faltam palavras para explicar o quão especiais e importantes eles são para mim, agradeço a todos por cederem morada em suas vidas. Lídia, minha irmã de outra mãe, cuja família considero como minha, sem todos os seus encorajamentos e ajudas em momentos difíceis, nada disso seria possível ou teria o mesmo valor. Agradeço a Stella, por ser minha guardiã e fiel companheira. Caio, Larissa, Sheron, Júlia, Melissa, Djulia, Isabelle, Gabriela, Joana, Vinicius, Luiz, Arthur e Andrea obrigada por me apoiarem, aliviarem momentos difíceis e darem motivos para risos e compartilharem aventuras.

Às minhas amigas do ensino médio, obrigada pela amizade duradoura e por toda evolução compartilhada. Júlia Hass, Júlia Schimidt, Caren, Duda, Celine, Helo, Nathalia, Ana, Gabi e Kiki agradeço por todas as intervenções de moda, aceitação, carinho e incentivo para uma vida mais leve e divertida. Desejo a todos que leem esse texto, encontrar pessoas tão incríveis quanto essas, em momentos tão fortunitos quanto os que tive, caso você tenha essa mesma sorte, crio que possa se dizer feliz.

Menor do que o meu sonho, eu
não posso ser

Lindolf Bell

LISTA DE SIGLAS

GAMA	General Aviation Manufactures Association
FAA	Federal Air Administration
EASA	European Union Safety Agency
MSE	Monitoramento da saúde estrutural
AP	Aprendizado de máquina
SAE	Society of Automotive Engineers
ROAAS	Runway Overrun and Awareness Alerting
EDA	Exploratory data analysis
CBM	Condition based maintenance
SHM	Structure health monitoring
MLP	Multi layer perceptron
IIR	Resposta ao impulso de duração infinita

LISTA DE SÍMBOLOS

E	Módulo de elasticidade
f	Tensão de ruptura
u	Tensão de escoamento
σ	Tensão
S_{ut}	Tensão útil
$S_{e'}$	Tensão para limite de fadiga
m_t	Total de anomalias em conjunto de dados
m_p	Quantidade de dados classificados como anomalia
m_g	Dados anomálos classificados

LISTA DE ILUSTRAÇÕES

Figura 1 – Padrões de falha observados ao longo do ciclo de vida da aeronave	15
Figura 2 – Gráfico de tensão por deformação	18
Figura 3 – Curva de Wöhler	18
Figura 4 – Exemplificação do método: a. pitch-cath. b. pulse-echo.	20
Figura 5 – Taxonomia de métodos ativos de abordagem MSE	21
Figura 6 – Representação esquemática rede neural	24
Figura 7 – a. Malha de sensores usada. b. Resultado das medições obtidas . .	26
Figura 8 – a. Imagem do corpo de prova e evolução dimensional da trinca b. Evolução de trinca por mapa de calor	27
Figura 9 – Fotografia do corpo de prova	30
Figura 10 – Desenho tridimensional do corpo de prova	30
Figura 11 – Localização da região de impacto	31
Figura 12 – Localização dos sensores piezoelétricos	31
Figura 13 – Função de excitação	32
Figura 14 – Estruturação dos dados	33
Figura 15 – Formato rede neural	34
Figura 16 – Fluxograma do código	36
Figura 17 – Comparação de dados, estrutura saudável e com danos	38
Figura 18 – Evolução média global ao longo do experimento	38
Figura 19 – Evolução média individual ao longo do experimento	39
Figura 20 – Evolução média individual ao longo do experimento	39
Figura 21 – Evolução média individual ao longo do experimento	40
Figura 22 – Função de ruído adicionado aos dados da eq. 7	41
Figura 23 – Função de ruído adicionado aos dados da eq. 8	41
Figura 24 – Função de ruído adicionado aos dados	42

LISTA DE TABELAS

Tabela 1 – Tipos de checagem de acordo com a quantidade de ciclos de voo para aeronaves comerciais	16
Tabela 2 – Tipos de ondas elásticas	22
Tabela 3 – Comparação entre ensaios estruturais não destrutivos e monitoramento da saúde estrutural	27
Tabela 4 – Configuração de teste e força de impacto	30
Tabela 5 – Tabela com hiperparâmetros da rede neural	35
Tabela 6 – Critérios para triangulação de resultado	36
Tabela 7 – Parâmetros de qualidade	40
Tabela 8 – Parâmetros de qualidade - após adição de ruído nos dados	42
Tabela 9 – Parâmetros de qualidade - após adição de ruído aleatório nos dados	42

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	13
1.1.1	Objetivo Geral	13
1.1.2	Objetivos Específicos	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	ESCOPO DA MANUTENÇÃO CLÁSSICA	14
2.1.1	Modos de falha comuns às aeronaves	17
2.1.2	Fadiga de metais	17
2.2	Monitoramento da saúde estrutural	19
2.2.1	Ondas Lamb	21
2.3	Aprendizado de máquina	22
2.3.1	Redes neurais	23
2.3.2	Séries de dados temporais	24
2.3.3	Análise de anomalias em séries temporais	24
2.4	Trabalhos Relacionados	25
2.5	Considerações Finais	27
3	MÉTODO	29
3.1	Conjunto de dados	29
3.2	Pré-processamento	32
3.3	Processamento	33
3.4	Pós-processamento	36
4	RESULTADOS	37
4.1	Análise exploratória de dados	37
4.2	Treinamento da rede	40
4.3	Predição de novos resultados	41
4.4	Discussões	43
5	CONCLUSÕES	44
	REFERÊNCIAS	46
	APÊNDICE A	48
	APÊNDICE B	52

1 INTRODUÇÃO

A aviação em escala comercial surgiu em meados dos anos 30, desde então figura entre os principais modais de transportes, destacando-se pela eficiência e integridade com a qual as cargas são transportadas. Por conta desse desempenho, 275 transplantes de órgão, de longa distância, são realizados todos os anos no Brasil, segundo a FAB (2021).

O setor aeronáutico não se limita a aeronave, sendo sua evolução e complexidade tecnológica, apenas um dos inúmeros pilares necessários a estruturação segura e eficiente desse meio de transporte. Sendo inerente uma rede de acompanhamento e checagem em sua operação, a exemplo das torres de controle, responsáveis por direcionar as aeronaves em situações normais e críticas e as atividades de manutenção, responsáveis por garantir o correto funcionamento de todos os sistemas da aeronave e corrigir preventivamente falhas, mitigando a chance de eventos graves ou catastróficos.

O aumento da frota global demanda que todos esses pilares evoluam concomitantemente aos avanços técnicos dos sistemas embarcados. Segundo relatório da GAMA (2019), durante a última década a frota de aeronaves aumentou em 29% em relação a anterior, e segundo o levantamento Boeing (2022), essa demanda deve aumentar em 85% só na América Latina, sendo o Brasil responsável por 30% desse aumento.

O incremento na frota global se traduz em medidas concretas de modernização focadas em aumentar a segurança e eficiência de operação. Por exemplo, nos últimos anos a *Federal Air Administration* (FAA) e outras entidades reguladoras, como a *European Union Aviation Safety Agency* (EASA), tornaram obrigatório a instalação do *Automatic Dependent Surveillance – Broadcast* (ADS-B out), responsável por transmitir a posição da aeronave com maior precisão, permitindo um menor espaçamento no espaço aéreo, ajudando a suprir esse aumento de demanda.

Devido ao escopo robusto que checar todos os sistemas de forma periódica tem, o custo com manutenção corresponde de 12 a 15% do total de operação de vida da aeronave (VIEIRA; LOURES, 2016). Tendo como referência a entrada da aeronave em operação, chamado de data de início de manutenção (T-0). Ela pode ser de fase ou fora de fase, dependendo se sua ocorrência está em consonância as revisões periódicas da aeronave.

Quando o avião está impossibilitado de voar pela falha crítica de algum

componente, é conhecido como *aircraft on ground* (AOG). Na tentativa de diminuir sua ocorrência, sem prejudicar a confiabilidade do serviço prestado, surgem várias correntes de estudo (SARANGA; KUMAR, 2006), entre as quais o monitoramento da saúde estrutural (MSE) (DONG; KIM, 2018).

MSE é uma área multidisciplinar focada em entender como usar técnicas para prever quando uma estrutura pode apresentar falha (SU; YE, 2009). De acordo com Toh e Park (2020), uma das possibilidades é a análise do padrão de vibração gerado pela estrutura. A principal ideia, desse campo de estudo, é migrar a manutenção feita em intervalos periódicos para o chamado *conditional based maintenance* (CBM).

O padrão vibracional pode ser obtido por meio de ondas Lamb, ondas mecânicas capazes de percorrer longas distâncias e terem algumas características modificadas ao passarem por alguma descontinuidade estrutural. Por exemplo, para a asa uma das fraturas comuns são trincas (MEGSON, 2017), difíceis de identificar, mas com o modelo MSE seria possível localizá-las, por meio de acompanhamento não destrutivo.

A demanda para desenvolvimento de novas formas de CBM fomentou pesquisas como o Remap (REMAP, 2020), que em junção ao Advisory Council for Aviation Research and Innovation in Europe (ACARE)(WASSINK, 2018), estimam que, ao tornar o MSE uma ferramenta auxiliar na manutenção das aeronaves. Segundo ReMap (2020), espera-se economizar 700 milhões de euros por ano apenas na Europa, evitando um número considerável de paradas não programadas de aeronaves, ocasionadas por falhas fora dos períodos de manutenção programada.

O MSE gera uma quantidade massiva de dados para análise, sendo necessário, não apenas estudar sua forma de captação, mas também uma forma robusta de análise e predição. Por conta dessa complexidade, segundo Toh e Park (2020), o aprendizado de máquina (AP) pode ser associado ao MSE pela capacidade de prever cenários complexos com acurácia. O AP surgiu na década de 50 e desde então vem ganhando força como uma das principais ferramentas de inteligência artificial para ciência de dados (TOH; PARK, 2020).

A EASA vê no aprendizado de máquina, a ferramenta necessária para garantir a robustez que sistemas de tomada de decisão auxiliares ao piloto e à equipe de manutenção (EASA, 2021). A Sociedade dos engenheiros automotivos (SEA) criou o comitê G-34 focado em criar regulações e testes para a aplicação de inteligência artificial e aprendizado de máquina nos processos aeronáuticos.

Atualmente, algumas soluções já fazem a aplicação dessas ferramentas em caráter auxiliar, como no Runway Overrun and Awareness Alerting System (ROAAS), sistema do avião executivo Phenom 300E da EMBRAER, que indica ao piloto quando é desejável arremeter durante o processo de pouso.

Considerando tais possibilidades, visa-se conduzir um estudo exploratório

de MSE aplicando aprendizado de máquina, utilizando o *multi layer perceptron*, que classificará dados de entrada em danos potenciais ou não, a partir disso a estrutura pode ser classificada como saudável ou danificada. Para melhor compreender essas duas tecnologias emergentes e melhor relacioná-las será apresentado uma revisão bibliográfica acerca do escopo de manutenção clássica, vertentes de MSE e MSE, bem como critérios de validação de dados.

O estudo desse trabalho utiliza os testes realizados pela universidade holandesa TuDelft, para o projeto ReMAP, no qual foi obtido diversas medições antes e depois da ocorrência de impacto e fadiga para prever o surgimento de trincas e outras falhas. Esse estudo gerou dados em diversas condições, para embasar a criação de ferramentas de predição de falhas críticas a partir do surgimento de trincas. O aprendizado de máquina será aplicado para identificar o surgimento de trincas, partindo desses dados e classificá-las em dano potencial ou não.

1.1 OBJETIVOS

Para resolver a problemática de predição de surgimento de trincas em asas de avião, propõem-se os seguintes objetivos.

1.1.1 Objetivo Geral

Aplicação de aprendizado de máquina para estudo exploratório de MSE para classificar o surgimento de trincas na estrutura de acordo com seu potencial de falha futura.

1.1.2 Objetivos Específicos

- a) Revisar métodos de manutenção aeronáutica;
- b) Apresentar os métodos de falha mais comuns a aeronaves;
- c) Examinar os critérios necessários para a aplicação de monitoramento da saúde estrutural (MSE) em aeronaves comerciais;
- d) Realizar análise exploratória de dados dos testes;
- e) Demonstrar a sensibilidade do código ao surgimento de trincas na estrutura da asa;

2 FUNDAMENTAÇÃO TEÓRICA

Monitoramento da saúde estrutural (MSE) é uma ferramenta multidisciplinar, que engloba: análise vibracional, mecânica da fratura e computação. Nesse trabalho o MSE é aplicado para otimizar o plano de manutenção de aeronaves, antevendo eventos críticos. Por essa razão, a fundamentação pode ser dividida em duas partes: o entendimento da finalidade e o embasamento teórico de cada área.

Para compreender a finalidade do projeto, é necessário entender a estrutura de um plano de manutenção, os parâmetros e terminologias usadas por esse campo de estudo, sua relação com o ciclo de vida da aeronave e seu custo de operação. Também é necessário, entender os eventos críticos que impedem a operação segura e como eles são usualmente detectados.

O MSE baseia-se no padrão de vibração da asa, portanto é indispensável compreender qual é o padrão para uma asa sem trincas. Deve buscar compreender quais as principais falhas que acontecem nessa região, sua localidade usual e como o aparecimento reflete na rigidez elástica e no padrão de vibração da asa.

Esse trabalho realizará a predição do padrão de vibração pela utilização de aprendizado de máquina (AP), o último tópico abordado nessa seção, explorará os tipos de AP, suas finalidades e sequência de séries temporais. Será apresentado as principais formas de medir a acurácia do código e avaliação da qualidade do dado previsto, para compreensão da previsão realizada.

2.1 ESCOPO DA MANUTENÇÃO CLÁSSICA

Aeronaves, ao entrarem em operação, passam por vários ciclos de pousos e decolagem, entre os quais, são submetidas a pressurização e despressurização da cabine, forças de rajada e manobras aerodinâmicas (XU; KROO, 2014). O ambiente de operação é propício ao surgimento de trincas, creeping e outros tipos de falhas (MEGSON, 2017), por essa razão, é necessário realizar manutenções periódicas a assegurar a segurança dos passageiros.

O momento em que uma aeronave entra em operação é chamado de data inicial de manutenção (T-0), utilizada para prever paradas periódicas, para checar os componentes, pois eles além da variação de pressão, são submetidos a força de impacto e grande oscilação térmica a cada ciclo de operação da aeronave, aumentando a chance de fadiga dos componentes.

A United Airlines realizou um estudo para compreender os modelos de falha de componentes aeronáuticos, a figura 1, mostra os padrões identificados de taxa de falha observados ao longo de sua vida útil, a figura mostra a porcentagem que cada padrão representa em todo o espaço amostral observado.

Figura 1 – Padrões de falha observados ao longo do ciclo de vida da aeronave

	1. Com mortalidade infantil; Taxa de falha constante, com fim de vida bem delimitado (4%)
	2. Sem mortalidade infantil; Taxa de falha crescente, com fim de vida bem definido (2%)
	3. Sem mortalidade infantil; Taxa de falha crescente, sem fim de vida bem definido (5%)
	4. Taxa de falha crescente após o início do ciclo de vida, ela se mantém constante até o fim do ciclo, sem fim de vida definido (7%)
	5. Sem mortalidade infantil, taxa de falha constante, sem fim de vida definido (14%)
	6. Com mortalidade infantil, taxa de falha constante, sem fim de vida bem definido (68%)

Fonte: Kinnison e Siddiqui (2012)

Os casos 1,2 e 3 possuem padrão que possibilita a criação de esquemas de manutenção fixa, uma vez que é possível prever o final da sua vida útil, a representatividade desse grupo é de cerca de 11% (KINNISON; SIDDIQUI, 2012). Segundo o mesmo autor, o grupo 3,4 e 5, representam 87% dos componentes e não é possível estimar o final do seu ciclo de vida. De acordo com Kinnison e Siddiqui (2012), criam-se dois grupos de manutenção:

- Programada: focada no grupo 1,2 e 3;
- Fora de fase: para os grupos 4,5 e 6.

As manutenções podem ser monitoradas por meio do T-0 da aeronave ou do componente, por marcar o início de sua operação, ou por horas de voo, dependendo do perfil operacional da aeronave,

De acordo com Kinnison e Siddiqui (2012), para a viação comercial, nas manutenções programadas e não programadas há checagens do tipo A, B, C e D, descritas abaixo:

- **Check A:** Ocorrem entre 120 e 150 horas de voo, valida a condição dos principais componentes, leva cerca de 8 horas;
- **Check B:** Ocorrem a cada 750 horas de voo, abrange o grupo A e abrange a checagem de sistemas de lubrificação e inspeção aberta de painéis;
- **Check C:** Ocorre a cada 3000 horas de voo, abrange os grupos A e B, chamada de manutenção de hangar. Checa a integridade de sistemas de lubrificação pesado, sistemas de proteção de corrosão, motores e acessórios;

- **Check D:** A cada 20000 horas de voo, inclui os grupos anteriores. O interior da aeronave é desmontado para realizar checagens estruturais detalhadas.

A tabela 1 mostra uma tabela que compara com quantos ciclos de voo, cada um dos grupos de manutenção são performados para diversas aeronaves comerciais.

Tabela 1 – Tipos de checagem de acordo com a quantidade de ciclos de voo para aeronaves comerciais

Tipo de checagem	747-400	747/200/300	A300B4
Trânsito	A cada ciclo de pouso	A cada ciclo de pouso	A cada ciclo de pouso
Diário	Avião parado por mais de 4 horas	Avião parado por mais de 4 horas	Avião parado por mais de 4 horas
A	A cada 600 horas de voo	A cada 500 horas de voo ou 7 semanas	A cada 385 horas de voo ou 11 semanas, dividida em 4 etapas (A1, A2, A3, A4)
B	A cada 1200 horas de voo, é dividida em duas partes (B1, B2)	A cada 1000 horas de voo, dividida em duas partes (B1, B2)	Não há
C	A cada 5000 horas de voo ou 18 meses, dividida em duas etapas (C1, C2)	A cada 4650 horas de voo ou 24 meses	A cada 3000 horas de voo ou a cada 18 meses, dividida em duas partes (C1, C2)
D ou HMM	A cada 25000 horas de voo, ou 6 anos	A cada 25000 horas de voo, ou 5 anos	A cada 12000 horas de voo, ou 4 anos

Fonte: Kinnison e Siddiqui (2012)

Toda vez que uma parada de manutenção tem mais de um critério, por exemplo, horas de voo ou quantidade de semanas, ela acontecerá assim que uma das duas condições for concretizada. Paradas de manutenção são periódicas, ou seja, a aeronave 747-400 fará a checagem A, a cada 600 horas de voo, portanto, mesmo que inicialmente as checagem A e C não se sobreponham, elas podem ocorrer concomitantemente em diversos momentos da vida de operação da aeronave.

Indiferente do tipo, a manutenção é feita com a aeronave fora de operação em uma oficina especializada, com profissionais treinados, portanto, é uma atividade com um custo elevado, que representa cerca de 12 a 15% do custo de operação da aeronave (VIEIRA; LOURES, 2016). O acesso de muitas peças são manuais, portanto é necessário desmontar parte da estrutura para verificar a integridade. Uma estratégia para reduzir o tempo fora de operação é a divisão das manutenções em dois ou mais intervalos.

2.1.1 Modos de falha comuns às aeronaves

Há duas formas de carregamento em estruturas: estático e dinâmico, a diferença entre ambos se dá pela taxa de variação ao longo do tempo. Para carregamentos estáticos a variação não é constante, esse tipo de força pode ser a reação de um apoio, o peso e outros, já para carregamentos dinâmicos a aplicação de força varia com o tempo e pode ser entendida dentro de ciclos de operação de um equipamento (NORTON, 2013).

A maioria das falhas ocorre por fadiga, falha caracterizada pela aplicação de cargas cíclicas a estrutura, uma falha por fadiga pode ocorrer mesmo quando a tensão transversal de uma região é inferior à tensão de escoamento do material (NORTON, 2013). Segundo Norton (2013) um relatório divulgado pelo governo norte-americano, o custo anual do país com reparos devido à fadiga, em 1982 chegaram a 3% do produto interno bruto (PIB) do país.

Um dos maiores exemplos de falha por fadiga aconteceu na aviação com o modelo Comet em 1954, pois, devido ao ciclo de pressurização e despressurização da aeronave e ao formato das bordas da janela, que eram cantos vivos, ocorreu o surgimento de trincas que levaram a falha catastrófica da estrutura (NORTON, 2013). Para mitigar a falha por fadiga é necessário evitar concentradores de tensão (NORTON, 2013).

2.1.2 Fadiga de metais

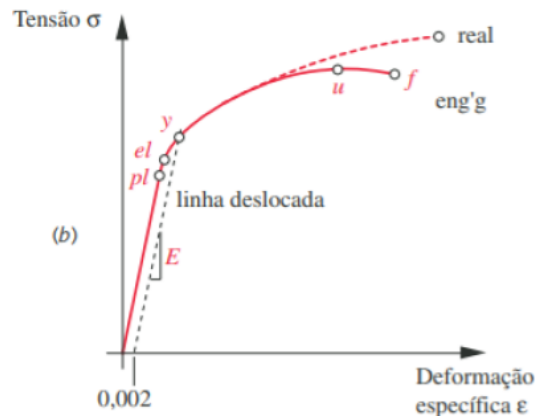
Para entender o conceito de fadiga, faz-se necessário compreender o conceito de tensão e deformação para metais. A estrutura de metais é formada por diversos grãos, com a possibilidade de se movimentar ao nível microscópico (CALLISTER, 2008). Essa movimentação gera mudanças macroscópicas no material.

Uma maneira de entender as mudanças na estrutura, é pelo carregamento de tensão axial de um corpo de prova, sua estrutura de seção transversal pequena, após alguns ciclos ela diminuir até a ocorrência de seu rompimento. O gráfico deformação da figura 2 mostra a variação da tensão, nessa seção transversal. O ponto máximo dele é chamado de limite de escoamento do material.

No gráfico descrito na figura 2, E é o módulo de elasticidade, σ_u o limite de escoamento do material, a partir do qual a deformação não é mais reversível, e σ_f é o ponto de ruptura. Há duas curvas, pois a tensão vista na prática, chamada de tensão de engenharia (*eng'g*) é inferior à teórica (*real*).

Mesmo que não visível ao olho nu, o processo de fabricação gera trincas ao nível microscópico em todos os materiais (NORTON, 2013), ao ser submetida a cargas cíclicas, as trincas passam a se comportar como pequenos concentradores de tensão, ocasionando seu crescimento e conseqüente aumento do valor de concentração de

Figura 2 – Gráfico de tensão por deformação



Fonte: Norton (2013)

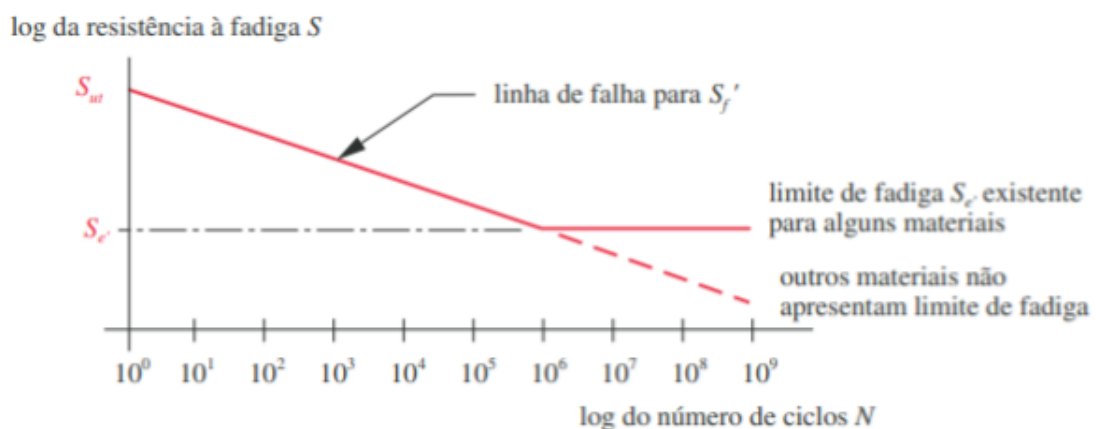
tensão, esse ciclo ocorre até a falha repentina da estrutura (NORTON, 2013).

O desenvolvimento de uma trinca pode ser entendido em três partes de acordo com Norton (2013):

- **Início da trinca:** Etapa de curta duração, a trinca pode surgir no processo de fabricação, ou pelo deslocamento dos grãos que compõe o metal;
- **Propagação:** Corresponde a maior parte do ciclo de vida, para o aumentar é necessário projetar estruturas com a menor quantidade de cantos vivos e formas concentradoras de tensão;
- **Ruptura:** falha repentina da estrutura, ao analisar a região da falha é possível identificar "marcas de praia", decorrentes da evolução da trinca.

Devido à evolução dessas etapas, a cada ciclo de operação da peça ocorre uma redução do limite de escoamento do material, o diagrama S-N ou Curva de Wöhler mostra um exemplo genérico dessa redução, como mostrado na figura 3.

Figura 3 – Curva de Wöhler



Fonte: Norton (2013)

Na figura 3, S_{ut} é a tensão útil do material, valor observado assim que a estrutura entra em operação. Ao longo das etapas de evolução da trinca, a tensão de ruptura passa a ser S'_f , alguns materiais tem um limite mínimo para essa tensão representado por S_e .

2.2 Monitoramento da saúde estrutural

Entre 1990 e 2007 ocorreram 1502 acidentes aéreos, com aeronaves de passageiros nos Estados Unidos, dos quais 26% foram fatais e resultaram na perda de 1104 pessoas, boa parte desses acidentes tiveram origem em falhas estruturais geradas pelo desgaste da estrutura (SU; YE, 2009). O MSE aparece como opção para identificar falhas precoces antes que haja sua evolução para eventos catastróficos, por substituir a manutenção programada para o chamado condition-based maintenance (CBM) (GIURGIUTIU, 2014).

No CBM a manutenção ocorre de forma sistêmica, quando se identifica a falha de algum componente, com ele é mais fácil detectar quando uma falha surge e o seu local, com isso há uma diminuição da carga de trabalho para manutenções de fase e fora delas, chamado de *menhour* (MH), (GIURGIUTIU, 2014). Ao aplicar o MSE o custo de manutenção pode ser reduzido em 30% Dong e Kim (2018).

De acordo com Su e Ye (2009), o MSE pode ser dividido em dois grupos:

- Passivo: estima uma porcentagem de utilização da vida útil da aeronave, pela medição de parâmetros comuns a operação como: velocidade, fator de aceleração g e pressão;
- Ativo: requer a instalação de novos dispositivos na aeronave para realizar a medição pela aplicação de diversos métodos como: ondas Lamb, emissão acústica e outros.

O método passivo tem um custo de implementação menor, porém não tem acurácia na predição de dados: ele estima a redução de vida útil da aeronave, ao passar por eventos adversos, mas sem identificar o surgimento e localidade de trincas na estrutura. O método ativo apresenta uma robustez maior, pode endereçar equipes de manutenção à região, na qual ocorreu o surgimento de uma trinca.

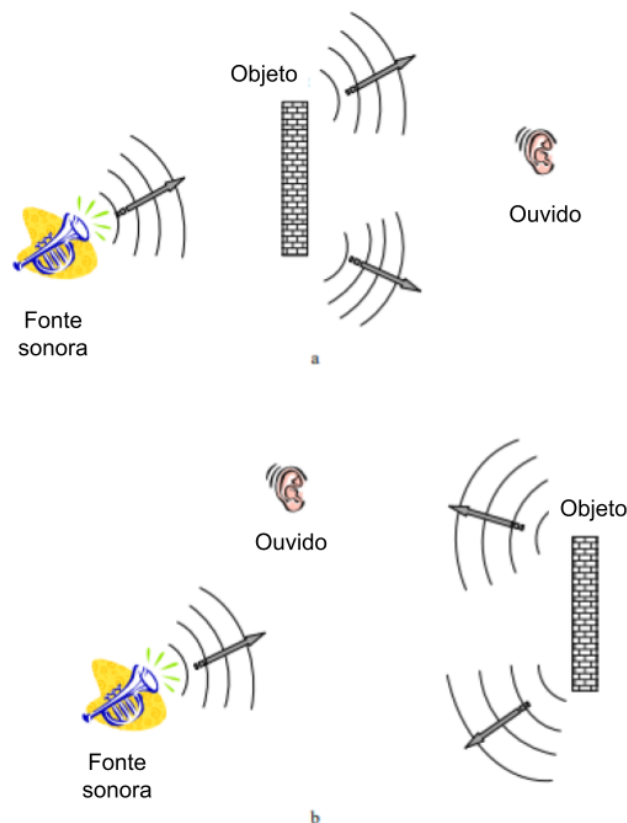
O método ativo apresenta subdivisões nos sistemas empregados para localizar a trinca, como mostrado por (NORTON, 2013), uma trinca gera mudanças locais e globais na estrutura, como mostrado por Su e Ye (2009):

- Métodos globais: podem ser empregados para identificar a trinca, pela aplicação de: frequência eigen, forma modal e amortecimento. Esses métodos só mostram variações quando o dano atinge mais que 10% da razão da dimensão característica pela área da estrutura;
- Métodos locais: possuem acurácia maior ao identificar os danos, desde o

momento do seu surgimento, pela aplicação de ondas elásticas, que utilizam a estrutura para se propagar. Embasam-se em duas maneiras de usar as ondas elásticas para identificar a trinca: pitch-catch e Pulse-echo.

O pitch-catch consiste em lançar uma onda por meio de um atuador e comparar suas propriedades, como amplitude e comprimento de onda, pelas características processadas por sensores receptores. Pulse-echo é outro método local, no qual o objetivo dos sensores é identificar a reflexão da onda gerada pelo atuador com receptores próximos a ele. A figura 4 mostra a diferença entre os dois métodos, utilizando a audição humana como exemplo, nesse caso a fonte sonora representa o atuador, o objeto a trinca e o ouvido os sensores receptores.

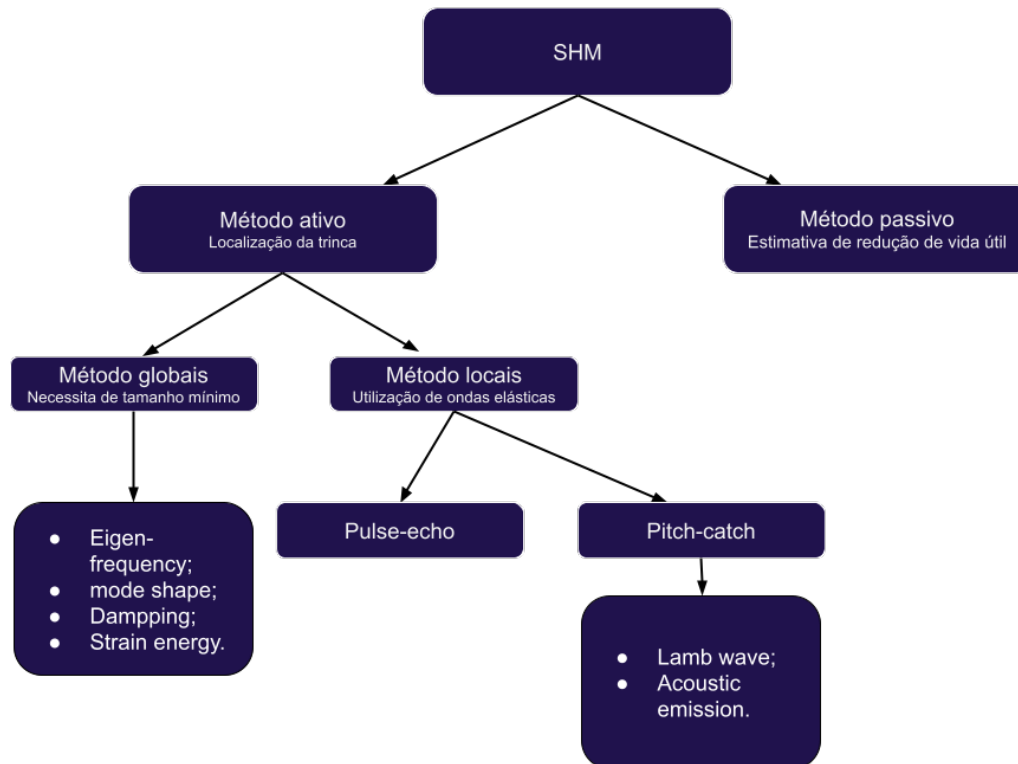
Figura 4 – Exemplificação do método: a. pitch-cath. b. pulse-echo.



Fonte: adaptado Su e Ye (2009)

A figura 5 mostra uma taxonomia dos métodos ativos com as divisões de MSE discutidas nesse capítulo.

Figura 5 – Taxonomia de métodos ativos de abordagem MSE



Fonte: Autora (2022)

2.2.1 Ondas Lamb

Ondas Lamb, ou ondas planas, são ondas elásticas descobertas e apresentadas, em 1917, por Horace Lamb, passaram a ser estudadas para aplicação de detecção de trincas em 1940. A tabela 2 mostra a comparação entre as características das ondas Lamb com outras ondas mecânicas.

Tabela 2 – Tipos de ondas elásticas

Tipo de onda	Definição e características	Tipo de onda	Definição e características
Ondas longitudinais	Viaja em plano pela compressão e rarefação alternada. Por isso, vibra em vai e vem no sentido de propagação da onda	Ondas Lamb	Chamadas de ondas de plano, se propagam em superfícies finas. Formato de propagação sensível a diferenças geométricas e estruturais.
Ondas cisalhantes	Chamada de onda transversal, partículas vibram no sentido perpendicular a propagação da onda	Ondas Stonely	Ondas existentes na junção de superfícies
Ondas de Rayleigh	Chamadas de ondas de superfície. Existem para superfícies muito finas (semi-infinitas). Amplitude decai exponencialmente ao se propagar para longe da superfície.	Ondas creep	Geradas pela refração de ondas longitudinais em bordas do corpo. Comportamento similar a ondas longitudinais.

Fonte: Su e Ye (2009)

As ondas Lamb percorrem distâncias longas, mesmo em materiais de paredes finas e alta taxa de atenuação, são sensíveis à falhas estruturais, percorrem longas distâncias de propagação. Após percorrer uma região danificada, suas características são modificadas, portanto, ao inserir diversos sensores na estrutura é possível triangular a localização do dano (SU; YE, 2009).

Para que a medição tenha boa precisão é importante que a dimensão da região plana seja muito maior que sua espessura, o comprimento de onda gerada pelos atuadores, deve ser na mesma ordem de grandeza da espessura (SU; YE, 2009).

2.3 Aprendizado de máquina

Aprendizado de máquina (AP) é uma ferramenta computacional, capaz de resolver problemas complexos com custo computacional relativamente baixo, sua principal característica é identificar padrões para encontrar valores numéricos, agrupar dados, classificar cenários etc (HARRISSON, 2020). O AP tem as seguintes macro segmentações, conforme a forma de predição dos dados de saída:

- Supervisionado: O código é treinado de acordo com um padrão de dados de entrada e saída, ao entender a relação entre os dois conjuntos, o AP faz predições

para dados novos. Tipos comuns de aplicação: regressão para predição de valores numéricos e classificação, para agrupamentos de dados em categorias pré-definidas;

- Não supervisionado: O código não é treinado com um conjunto de dados, mas realiza a junção de dados de acordo com características definidas pelo próprio código. Exemplo: segmentação de clientes;
- Misto: Junção do aprendizado supervisionado e não supervisionado, pode ser ocasionado por dados de entrada incompletos.

2.3.1 Redes neurais

Redes neurais são um tipo de aprendizado de máquina supervisionado, os dados disponíveis para são divididos em dois grupos: treinamento, os dados de entrada e saída serão fornecidos ao código para identificação dos padrões existente entre os dados de entrada e saída. O conjunto restante será usado para validar os resultados, pois só os dados de entrada são fornecidos e comparas-se o erro entre os resultados previstos e os esperados (MULLER; GUIDO, 2016).

Os dados de entrada são estruturados no formato de matrizes, as colunas representam as características do problema, chamadas de características, já as linhas são os valores numéricos, chamados de padrão. Os resultados são previstos por uma soma ponderada entre sub-variáveis (x), representando cada característica, e pesos (w), calculados pelo código na fase de treinamento, como mostra a equação 1.

$$y = w[0]z[0] + w[1]z[1] + \dots + w[n]x[n] + b \quad (1)$$

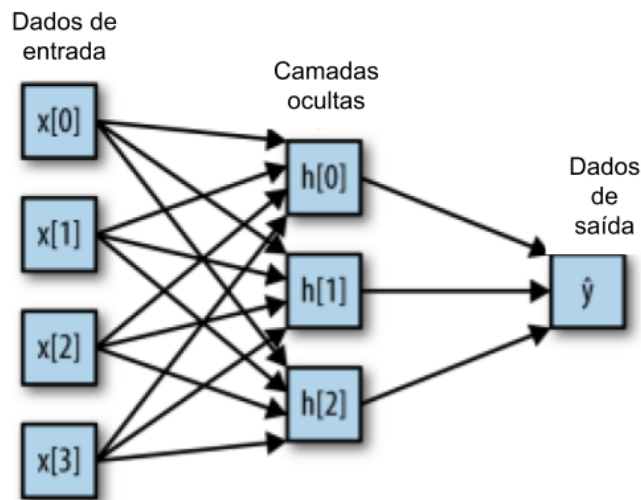
A figura 6, mostra a estruturação da rede neural, cada nível corresponde as sub-variáveis e cada elemento de ligação ao peso.

Embora sejam muito versáteis, é necessário ter muito cuidado ao analisar a acurácia do resultado, pois de acordo com Muller e Guido (2016) os seguintes problemas podem ocorrer durante a fase de treinamento:

- Overfitting: o algoritmo consegue prever com excelência os dados do conjunto de treino, mas ao passar pelo conjunto validado a acurácia é muito baixa, pois embora o código não consegue replicar a adaptação dos dados de treino o conjunto de validação e por conseguinte aos próximos dados de entrada;
- Underfitting: o algoritmo não consegue calcular pesos adequados a etapa de treino e portando, não obtém boa acurácia.

Além dos problemas acima, a acurácia pode invalidar a aplicação do código, sendo requerido um novo treinamento, o valor determinado como aceitável varia para cada situação.

Figura 6 – Representação esquemática rede neural



Fonte: Muller e Guido (2016)

2.3.2 Séries de dados temporais

Séries temporais mostram uma sequência de dados que acontecem dentro de uma sucessão, alterados pelo tempo e resultante de um processo estocástico (AUFFART, 2021). O processo pode ser descrito matematicamente como mostra a equação 2.

$$\begin{aligned}
 x : T &\longrightarrow \mathbb{R}^K \\
 T &\subseteq \mathbb{R} \\
 K &\in \mathbb{R}
 \end{aligned}
 \tag{2}$$

Análise de série temporal são similares a análise exploratória de dados (EDA - do inglês: exploratory data analysis), com a diferença que a análise de série temporal pode englobar elementos descritivos e exploratórios e a EDA abrange exclusivamente o segundo grupo (AUFFART, 2021).

Como uma série temporal é formada por dados sequenciais que capturam o comportamento do sistema no passado, elas podem ser usada para realizar previsões sobre configurações possíveis do sistema.

2.3.3 Análise de anomalias em séries temporais

Variações são inerentes aos processos de engenharia, necessário as incorporar a projetos e análises críticas. Em processos estatísticos e análises de qualidade, a variação é delimitada por valores aceitos e tudo o que for produzido deve ficar dentro deles. A média é uma forma de entender a maior concentração de pontos do sistema, entretanto para correta visualização ela deve ser observada junto a variância, que

indica o qual distante da média o valor observado está (BARBETTA; REIS; BONIA, 2010).

Valores fora do intervalo solicitado podem ser chamados de outliers, eles apresentam uma variação operacional ou uma condição específica. A presença de padrão para o aprendizado de máquina é importante para garantir que o código consegue mimetizar corretamente os dados recebidos, por essa razão, uma etapa anterior ao processamento, mas tão importante quanto é a checagem de anomalias (BARBETTA; REIS; BONIA, 2017).

Segundo Barbetta, Reis e Bonia (2017) ao buscar por outliers em dados, há três opções possíveis: o valor encontrado ser de fato uma anomalia, ele ser um falso positivo, ou seja, não representar uma variação real do esperado ou um falso negativo, quando um valor que é um outlier deixa de ser classificado como tal.

Barbetta, Reis e Bonia (2017) considera três importantes métricas para analisar os pontos anteriores: precisão mostrada (eq. 3), recall (eq. 4) e ranqueamento (eq. 5). Nelas m_t representa a quantidade total de anomalias do conjunto de dados, m_p a quantidade de dados classificados como anomalias, m_g os dados que foram classificados como anomalias e de fato o são e R_i é o conjunto de dados classificado como anomalia.

$$P = \frac{m_p}{m_t} \quad (3)$$

$$R = \frac{m_p}{m_g} \quad (4)$$

$$Ra = \frac{m_t(m_t + 1)}{2 \sum_{i=1}^{m_t} R_i} \quad (5)$$

O ranqueamento serve para indicar qual a posição de importância um dado recebeu dentro do conjunto de anomalias, é fundamental que outliers (m_g) liderem o conjunto (R_i). Imaginar esse conceito como dados em uma régua, na qual apenas os 5 primeiros valores são aceitos, o torna mais tangível, pois caso os valores encontrados sejam: 11,8,9 é importante que eles estejam nessa ordem, para indicar qual o desvio mais crítico.

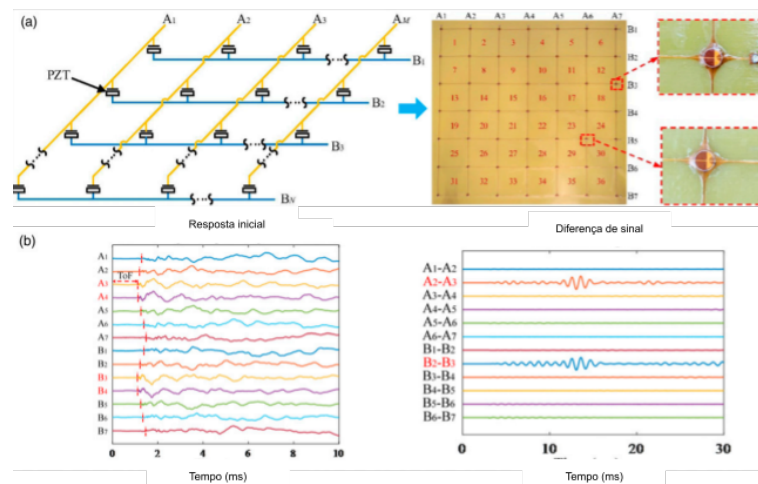
2.4 Trabalhos Relacionados

Por ser uma área multidisciplinar o MSE possibilita diversas combinações e aplicações para CBM, a evolução dos estudos combina com a evolução de materiais

aeroespaciais. Wang et al. (2022) realiza estudo sobre nova composição estrutural de aeronaves, chamada de smart skin, ela faz a junção de vários sistemas difusos na estrutura da aeronave. Um dos sistemas abordado pelo autor são sensores que podem fazer o acionamento de asas amorfas e a aquisição de dados para MSE.

Wang et al. (2022) prevê a criação de uma rede de sensores para realizar a triangulação de resultados, a estrutura proposta por ele, prevê microcontroladores difusos, para que o processamento de dados locais seja possível. O estudo é em estado da arte, e aplica a diferença dos sinais medidos por sensores adjacentes para fazer a predição de falha. A figura 7 mostra a malha utilizada e os resultados obtidos.

Figura 7 – a. Malha de sensores usada. b. Resultado das medições obtidas



Fonte: Wang et al. (2022)

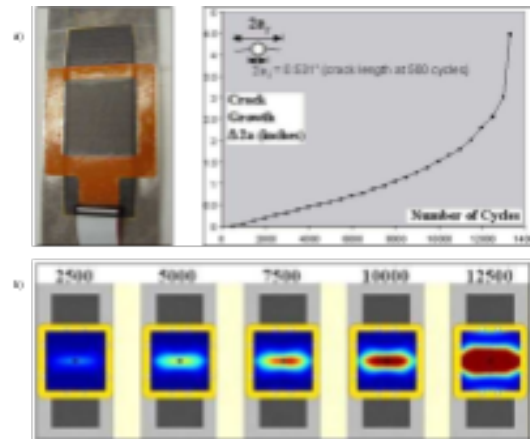
O trabalho de Gorinevsky (2005) acompanha a aplicação de uma camada de sensores posicionadas na estrutura padrão da aeronave, o autor acompanha um estudo de caso, para o modelo Boeing 777, realizado pela empresa de aviônicos Honeywell para associar o MSE ao Computador de Manutenção Central, já comercializado pela empresa. O produto associado ao MSE reduziu os custos de manutenção em 50 a 80% para diversos operadores.

O estudo de (GORINEVSKY, 2005) foi realizado em diversas configurações usando acelerômetros, strain gauges, sensores de fibra ótica, os quais apresentaram os melhores resultados. A figura 8 a) mostra o corpo de prova e evolução dimensional medida da trinca e b) mostra a evolução de uma trinca na estrutura ao longo de vários ciclos de pressurização.

Já o estudo de Qing et al. (2019) usa sensores piezoelétricos aplicados a estrutura existente para prever e acompanhar o surgimento de falhas. A classe de sensores utilizada apresentou um desempenho superior aos demais, devido à capacidade de prever danos em distâncias maiores em relação a acelerômetros, sensores de vácuo ou impedância e outros.

A tabela 3 mostra a comparação feita pelo autor entre MSE e outros ensaios

Figura 8 – a. Imagem do corpo de prova e evolução dimensional da trinca
b. Evolução de trinca por mapa de calor



Fonte: Gorinevsky (2005)

estruturais não destrutivos.

Tabela 3 – Comparação entre ensaios estruturais não destrutivos e monitoramento da saúde estrutural

Característica analisada	Ensaio não destrutivo tradicional	Monitoramento da saúde estrutural
Transdutores	Teste precisa ser montado toda vez que for utilizado	Estrutura de teste pode ser montada a cada teste ou pode ser fixa
Modo de detecção/ Tempo fora de operação	Assíncrono Não pode ser inspecionado em serviço, aumentando o tempo fora de operação	Assíncrono e síncrono Pode ser inspecionado dentro ou fora de serviço Diminuição do tempo fora de operação
Tempo de detecção	Demorado	Automatizado, com redundância
Capacidade de detecção	Só apresenta informação de falha	Além da falha, pode indicar temperatura, e outros dados

Fonte: Qing et al. (2019)

2.5 Considerações Finais

Monitoramento da saúde estrutural é uma área de estudo emergente, com grande potencial econômico e de redução de impacto ambiental. Utilizar ferramentas computacionais robustas para trazer velocidade e confiabilidade a esses sistemas é algo

importante, por essa razão o aprendizado de máquina é uma opção para desempenhar essa tarefa.

Da mesma maneira que o micro-ondas foi criado para missões aeroespaciais, o monitoramento da saúde estrutural apresenta potencial de aplicação em outras áreas como monitoramento de máquinas industriais e outras estruturas.

3 MÉTODO

Os métodos desse trabalho foram baseados nas teorias e discussões apresentadas no capítulo anterior, realiza-se um estudo exploratório com aprendizado de máquina supervisionado e séries temporais para compreender e analisar a evolução da saúde estrutural de uma porção da asa.

Foi aplicado uma rede neural para classificar os dados em: dano potencial e não potencial, isso possibilita a classificação da saúde estrutural em saudável ou danificada. Cada sensor conta com uma rede, dessa forma é possível triangular a predição cujo objetivo é determinar a localização aproximada e eliminar falsos positivos.

Os dados são pré-processados, para agrupar diversos cenários em uma mesma sequência, para que vários conjuntos de testes possam ser analisados mantendo o sentido físico. O processamento se dá pela predição do aprendizado de máquina, passando pelo seu treinamento e o pós-processamento é feito uma triangulação entre os resultados obtidos para localizar a falha e mitigar falsos positivos.

A linguagem de programação adotada foi o Python, em razão da vastidão de bibliotecas para processamento de dados e demais processos inerentes a esse trabalho. A biblioteca de aprendizado de máquina é o Scikit Learn, também foi usado a biblioteca numpy para agrupar e manipular os dados e matplotlib para gerar os gráficos.

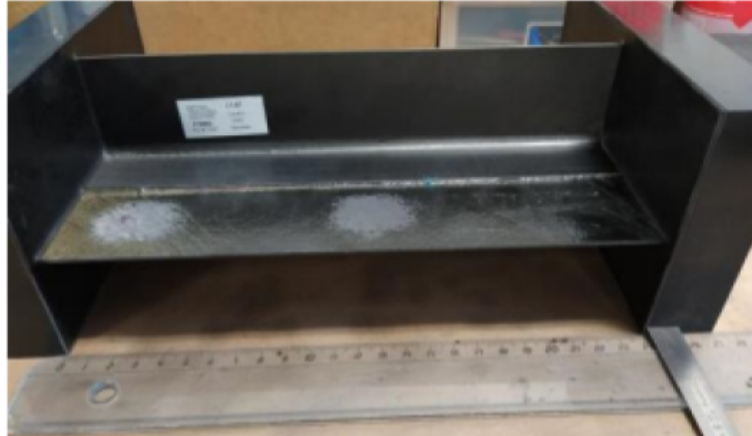
3.1 Conjunto de dados

Os dados são provenientes de testes da universidade holandesa TuDelft, no âmbito do projeto ReMap, cujo objetivo é mudar como a manutenção é feita nos aviões, para a condição de CBM (*condition based maintenance*). A instituição focou em realizar testes de MSE programado e ativo, portanto os dados são obtidos com a estrutura fora de operação por meio da excitação.

A estrutura usada para aquisição de dados pode ser visto na figura 9, ela mimetiza a asa de uma aeronave em contato com uma longarina. A figura 10 mostra a representação esquemática tridimensional do corpo de prova e localização dos oito sensores usados ao longo do teste. A estrutura passou por impactos gerados por pesos acelerados contra a ela, cada formato e força de impacto, resultou em uma configuração de teste.

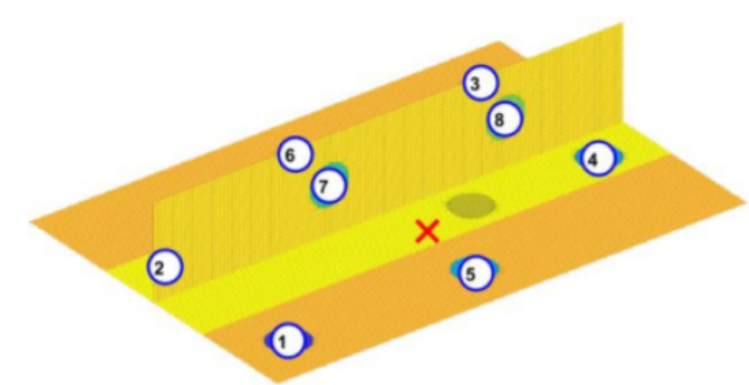
Foram realizadas, ao todo, 6 configurações de testes, com diferença na

Figura 9 – Fotografia do corpo de prova



Fonte: Remap (2019)

Figura 10 – Desenho tridimensional do corpo de prova



Fonte: Remap (2019)

carga máxima, mínima e energia do impacto, como pode ser visto na tabela 4. Cada configuração conta com uma localização de impacto específica mostrada pela figura 11.

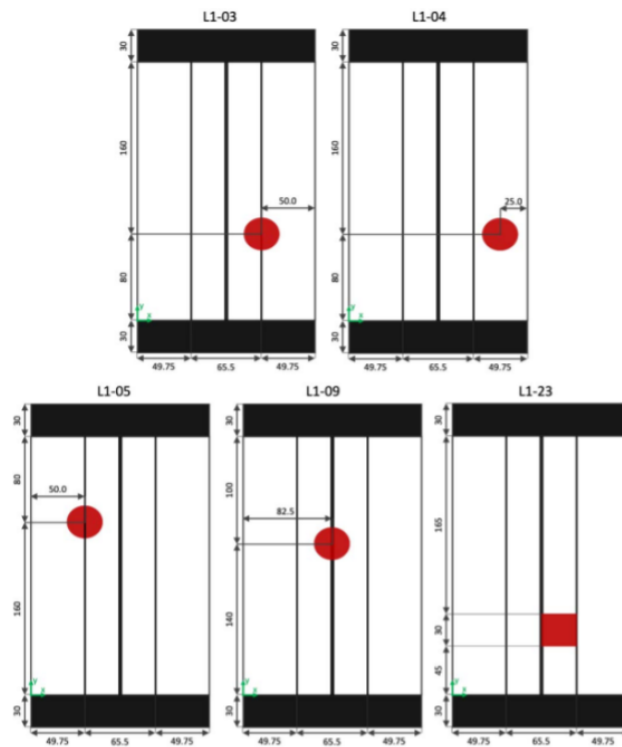
Tabela 4 – Configuração de teste e força de impacto

Configuração	L1-03	L1-04	L1-05	L1-09	L1 - 23 <100000	L1 - 23 >100000
Carga máxima (kN)	-6.5	-6.5	-6.5	-6.5	-5	-6
Carga mínima (kN)	-65	-65	-65	-65	-50	-60
Dano inicial e força de impacto	10 J	10 J	10 J	10 J	30 mm	30 mm

Fonte: Remap (2019)

Os oito sensores piezoelétricos, mostrados na figura 12 são usados para gerar a excitação via ondas Lamb e realizar a aquisição de dados. Em cada medição apenas um sensor funciona como atuador, os demais realizam a leitura. O sinal de entrada é

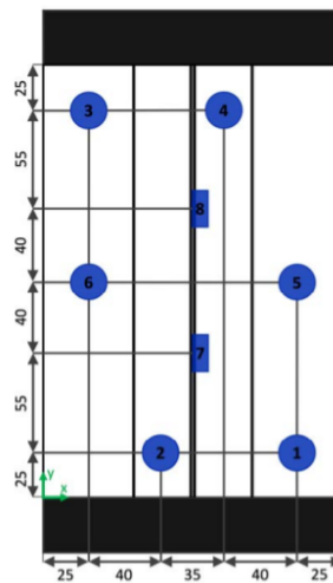
Figura 11 – Localização da região de impacto



Fonte: Remap (2019)

dados pela equação 6, seu gráfico pode ser visto na imagem 13.

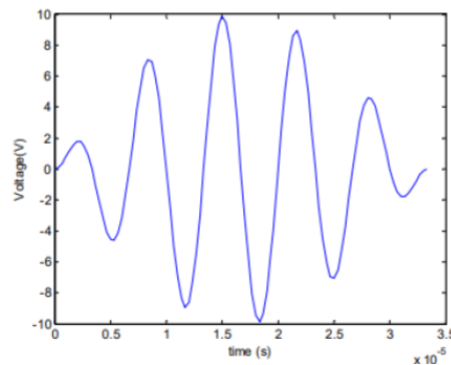
Figura 12 – Localização dos sensores piezoelétricos



Fonte: Remap (2019)

$$x(t) = A \sin(2\pi f_o t) \sin\left(\frac{\pi f_o t}{N_c}\right) \quad (6)$$

Figura 13 – Função de excitação



Fonte: Remap (2019)

O protótipo é submetido em intervalos de 5000 ciclos, com a incidência das cargas dinâmicas mostradas na tabela 4. Entre esses intervalos, ocorre o processo de aquisição dos dados. Esse procedimento é repetido 10 vezes e todos os sensores excitam a estrutura, portanto, a cada medição são adquiridos 80 dados.

No total, o experimento conta com 30 ciclos, sendo os 28 primeiros antes do impacto, um ciclo após ele para avaliar a diferença no perfil de medição e um último após a fadiga. Os dados de cada uma das 10 medições feitas após todos os ciclos, são armazenadas em pastas, em arquivo de extensão “.mat”.

As medições registram a evolução de uma estrutura saudável para danificada. Capturando, desde sua entrada em operação, até o desgaste de propriedade mecânicas pelos carregamentos dinâmicos, etapa referente aos 28 primeiros ciclos. Os dados capturam a mudança gerada pelo impacto, no ciclo 29, e pela ocorrência de fadiga no ciclo 30. Portanto, os 28 primeiros ciclos registram uma estrutura saudável e os dois últimos ciclos uma estrutura danificada. Portanto, os dados coletados representam inicialmente uma estrutura saudável, segue para o período inicial do dano, e, por fim, representa a estrutura danificada.

3.2 Pré-processamento

Devido à organização original para armazenar os dados, foi desenvolvido um programa para extrair os dados e os armazenar de forma adequada ao trabalho, assim erros de manipulação são mitigados e o trabalho otimizado.

Para cada sensor, foi criada uma matriz de dimensão (n,m) , na qual n é a quantidade de medições que ocorreram ao longo do tempo e m representa quantas vezes elas foram repetidas por ciclo de excitação. Portanto, cada coluna é uma série temporal que nos permite analisar mudanças no comportamento da estrutura, a variação dos dados dentro de cada linha, mostra a confiabilidade do sensor.

A figura 14 mostra a organização dos dados de entrada na matriz. Cada coluna

é uma série temporal correspondente a cada uma das 10 repetições do processo de aquisição (a,b,c ... j). A cada etapa são realizadas 2000 medições, logo, cada série temporal conta com 60000 dados. Os 55999 dados, mostrados na figura 14 são referentes as 28 medições antes do impacto, os dados com índice de 5600 a 57999 foram obtidos depois do impacto e a quantidade final, após a fadiga.

Figura 14 – Estruturação dos dados

a1	b1	...	j1	1 medição	Antes do impacto
a2	b2	...	j2		
...		
a2000	b2000	...	j2000		
...		
a55999	b55999	...	j55999		
a56000	b56000	...	j56000	Depois do impacto	Depois da fadiga
a56001	b56001	...	j56001		
...		
a57999	b57999	...	j57999		
a58000	b58000	...	j58000		
a58001	b58001	...	j58001		
...		
a60000	b60000	...	j60000		

Fonte: Autora (2022)

Os dados são agregados sem realizar médias, devido à quantidade consideravelmente menor após o impacto do que antes, dessa forma ocorre o aumento do espaço amostral. Na seção de resultados será mostrado o desvio padrão e média das medidas para cada instante.

O principal objetivo do presente trabalho é explorar as mudanças demonstradas pelas séries temporais, por essa razão é feita uma análise da sequência de dados das colunas. Alguns gráficos foram criados a fim de acompanhar a mudanças visuais no comportamento das séries temporais e realizar uma análise exploratória de dados prévia a construção do aprendizado de máquina.

3.3 Processamento

Um modelo de aprendizado de máquina para cada um dos sensores foi criado, portanto, cada sensor tem uma previsão individual de falha. Nesse formato ajuda a

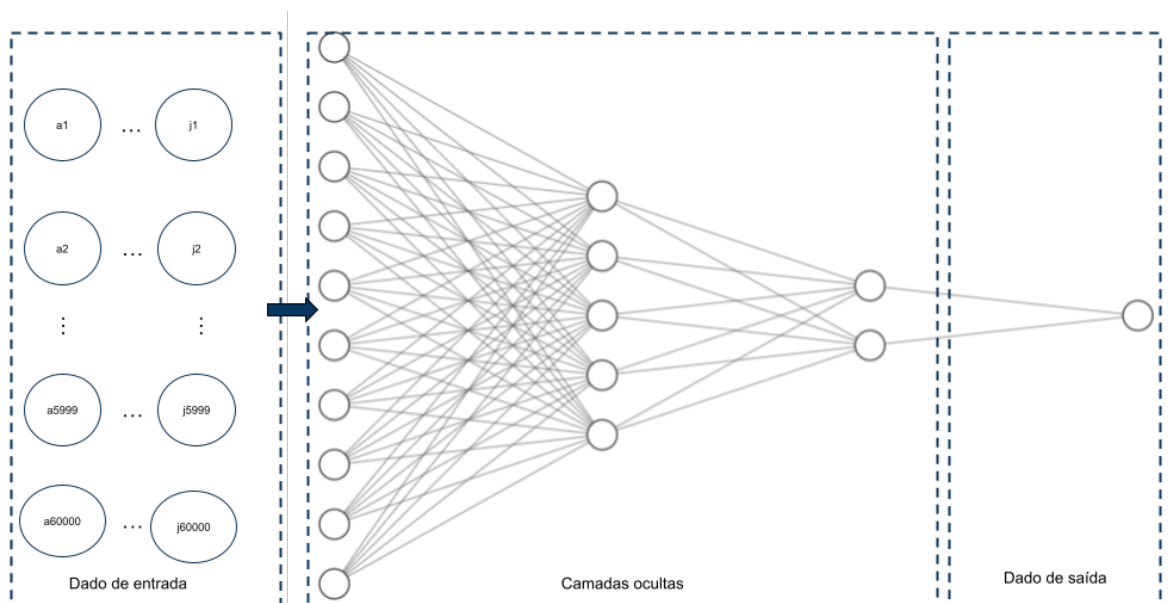
analisar o resultado global e triangular a região da falha, pois devido ao comportamento das ondas Lamb, todos os sensores adjacentes as trincas devem indicar a sua presença. Caso isso não ocorra, o resultado pode ser descartado, diminuindo a ocorrência de falsos positivos.

De todo o conjunto de dados disponível, 80% é usado para fazer o treino da rede e 20% sua validação. Para que o treinamento da rede seja considerado suficientemente bom, a acurácia deve ser superior a 90%, ou seja, dos dados de teste, o nível de acerto deve ter esse valor. O dado de entrada é uma medição de tensão, feita pelo sensor e o dado de saída é a classificação em 0 ou 1, dano não potencial e potencial; respectivamente.

Para esse modelo foi criada uma rede neural de classificação - Multi-layer perceptron (MLP) - foram usadas três camadas, a primeira com 10 perceptrons, a segunda com 5 e a terceira com 2. O formato de rede é mostrado na figura 16. O MLP recebe como dado de entrada a matriz com as 10 séries temporais, mostrada a figura 14, devido à grande quantidade de conexões, a passagem dos dados para o MLP foi mostrado pela seta azul, que liga o bloco de dados de entrada com o de camadas ocultas.

O resultado é uma classificação binária, entre dano potencial ou não, atrelado a uma estrutura danificada ou saudável, respectivamente. Essa classificação vem do resultado da MLP, na qual fornece um vetor de probabilidade. Nesse vetor cada posição representa uma classe, com uma probabilidade de adequação a ela. A depender desse resultado, o dado é entendido como dano potencial ou estrutura saudável. O código utilizado para o processamento pode ser visto no apêndice A.

Figura 15 – Formato rede neural



Fonte: Autora (2022)

Para cada uma das etapas foram aplicadas as bibliotecas descritas abaixo:

- **sklearn.preprocessing:** Realiza a normalização dos dados para o intervalo $[-1, 1]$, pois a NLP é sensível a outliers, pois podem ocasionar overfitting e dessa forma a predição não tem uma boa acurácia;
- **sklearn.model_selection:** Faz a divisão no conjunto de dados, para criar o conjunto de treino e o de testes;
- **sklearn.neural_network:** Exporta a rede neural de classificação, além dos dados tratados ela tem como dados de entrada os parâmetros usados pela MLP.

Os hiperparâmetros adotados, e seus significados, são mostrados na tabela 5.

Tabela 5 – Tabela com hiperparâmetros da rede neural

Hiperparâmetro	Valor adotado	Significado
Tamanho das camadas escondidas	3 camadas com: 10, 5 e 2 perceptrons; respectivamente	Quantidade de camadas e suas dimensões. Ajuste importante para garantir a acurácia, mas sem ter overfitting.
Função de ativação	relu	Função de ativação das camadas ocultas a função relu usa o valor máximo predito em uma regressão linear.
Solver	adam	Gradiente estocástico.
Alpha	10^{-5}	Auxilia a evitar overfitting por usar uma penalização para valores muito discrepantes.
Taxa de aprendizado	Constante	A taxa de aprendizado pelas camadas é constante.
Número máximo de iterações	200	Número máximo de iterações para convergência dos valores dos perceptrons. Está relacionada ao solver adotado.
Random state	1	Semente para geração dos valores iniciais. Usado para congelar a rede e gerar sempre o mesmo resultado.

Fonte: Autora (2022)

3.4 Pós-processamento

Após a classificação dos dados, para cada sensor, o código realiza uma triangulação dos resultados, pois tendo como base a figura 12, caso o sensor 7 indique dano potencial, os sensores 2,5 ou 6 também o indicar o mesmo resultado. Se isso não ocorrer, o resultado é considerado um falso positivo e a predição é descartada. O código é mostrado no apêndice B.

A tabela 6 mostra quais sensores devem indicar falha potencial para o resultado não ser descartado.

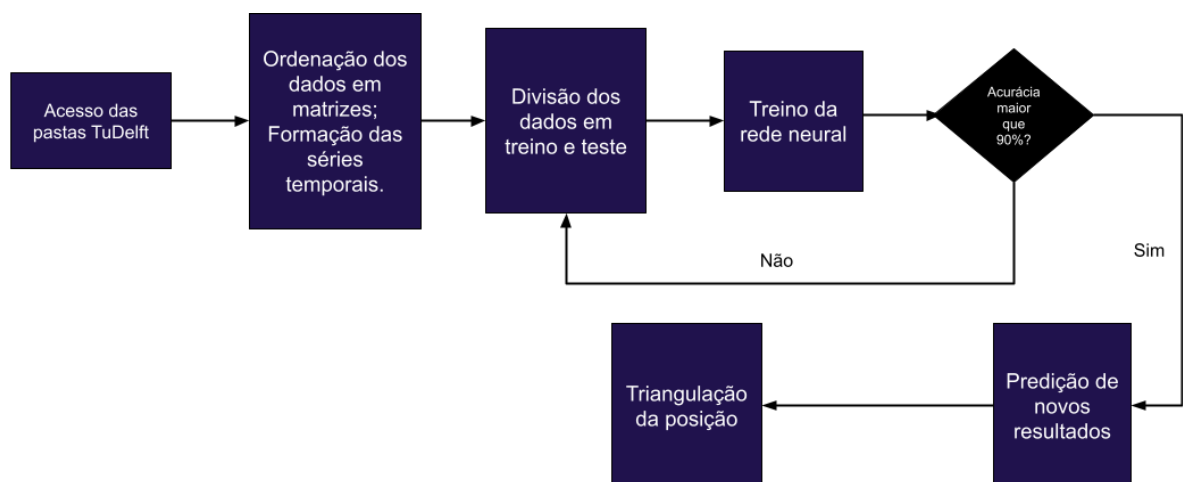
Tabela 6 – Critérios para triangulação de resultado

Sensor analisado	Sensor de validação da falha
2	7
3	4,6 ou 8
4	3,5 ou 8
5	4, 7 ou 8
6	2,3,7 ou 8
7	2,5,6 ou 8
8	3,4,5,6 ou 7

Fonte: Autora (2022)

A figura mostra o fluxograma dos métodos utilizados nesse trabalho.

Figura 16 – Fluxograma do código



Fonte: Autora (2022)

4 RESULTADOS

Neste capítulo serão discutidas as soluções obtidas aplicando os métodos do capítulo 3. Os seguintes tópicos serão discutidos:

- Abordagem da análise exploratória de dados, devido ao grande volume de dados tratados, provenientes da TuDelft;
- Discussão da acurácia das 7 redes neurais criadas, para a configuração L1-09, mostrada na tabela 4;
- Triangulação dos dados para validação do conceito;
- Análise de erro ao aplicar a predição com outra configuração de teste, mas mantendo o treinamento com a L1-09; e
- Expansão da triangulação com outros dados.

4.1 Análise exploratória de dados

Devido ao comportamento das ondas Lamb, espera-se que toda a estrutura tenha um comportamento similar, portando sete sensores, que processam o sinal, devem ter um comportamento similar, porém com aquisições em escalas diferentes, relacionado a distância da fonte de excitação. A presença de dano ou fadiga pode alterar esse padrão.

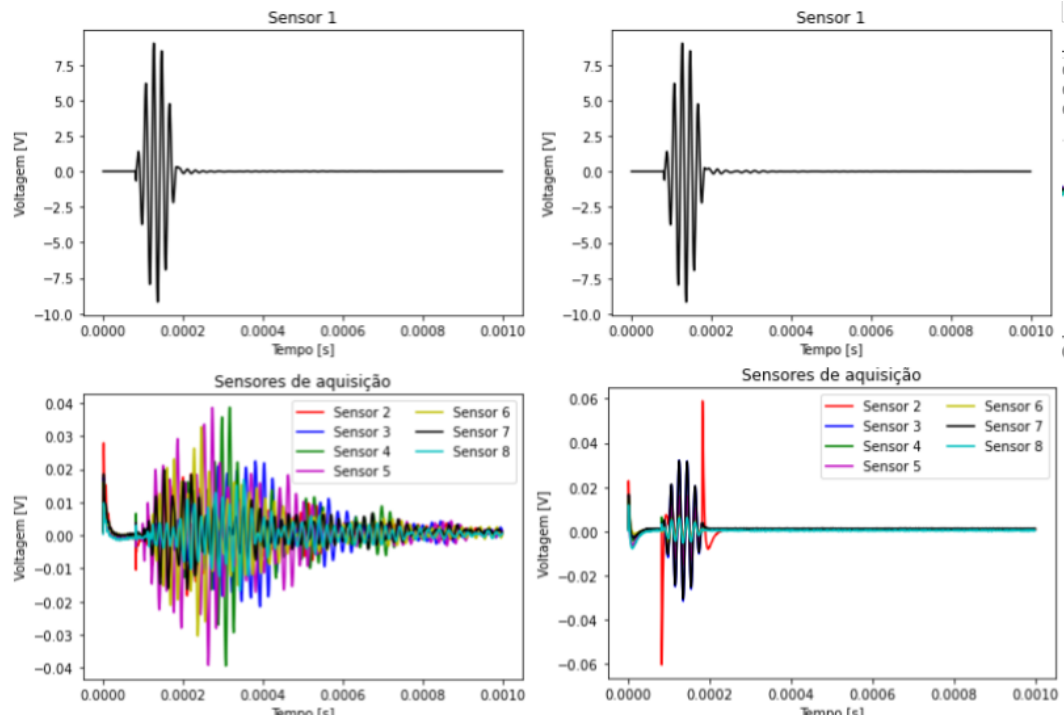
A figura 17 mostra a diferença, entre a primeira e a medição pós-impacto, entre o sensor de excitação 1 e a leitura dos demais sensores. Não há alteração no sensor 1 devido à constância da função de excitação. Para esses gráficos foi realizado a média entre as 10 medições repetidas sequenciais. O primeiro momento inicial pode ser entendido como uma estrutura saudável e o segundo como é o padrão vibracional de uma estrutura com danos.

Ao longo do experimento observaram-se as seguintes características:

- **Na mesma medição:** Há um atraso na leitura realizada pelos sensores em relação ao dado de entrada, ela é proporcional a distância do sensor de leitura ao sensor de excitação;
- **Mesmo sensor entre medições:** Ao comparar o comportamento gráfico de um sensor ao longo do tempo, observou-se que não houve diferença significativa em seu tempo de resposta, mas sim na amplitude do sinal recebido.

A fim de concatenar alguns dados e possibilitar a visualização, foi realizado uma média entre os valores obtidos entre cada ciclo, as figuras 18 e 19 mostram a

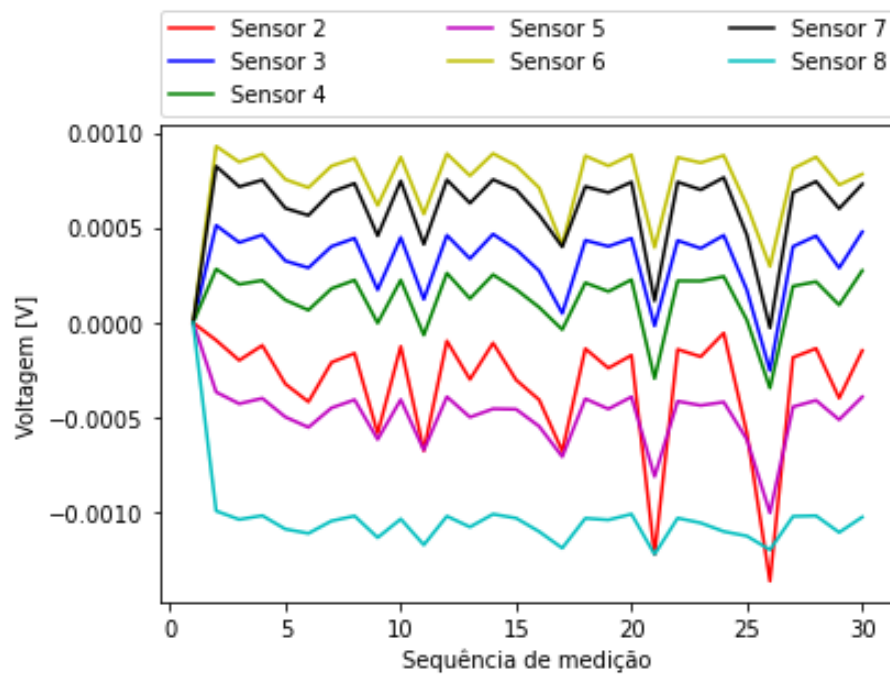
Figura 17 – Comparação de dados, estrutura saudável e com danos



Fonte: Autora (2022)

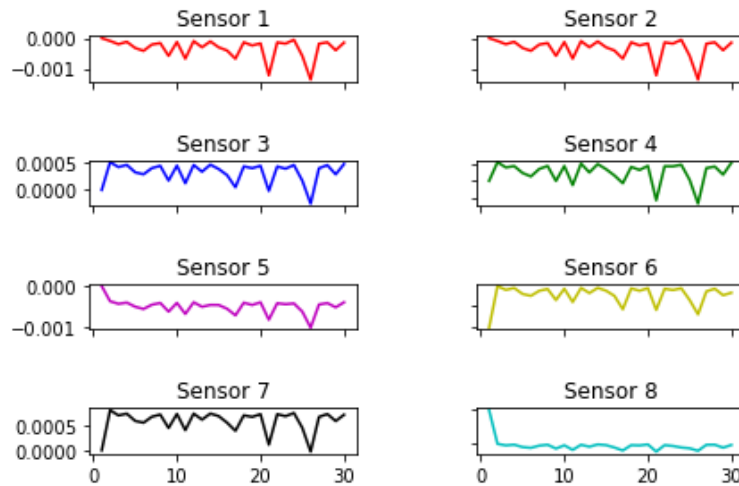
variação dos valores medidos obtidos ao longo do experimento.

Figura 18 – Evolução média global ao longo do experimento



Fonte: Autora (2022)

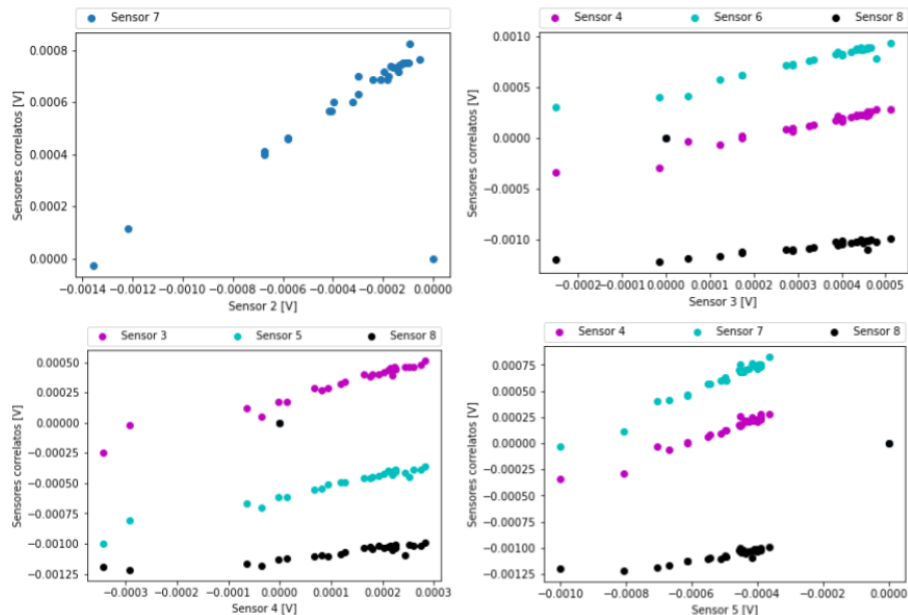
Figura 19 – Evolução média individual ao longo do experimento



Fonte: Autora (2022)

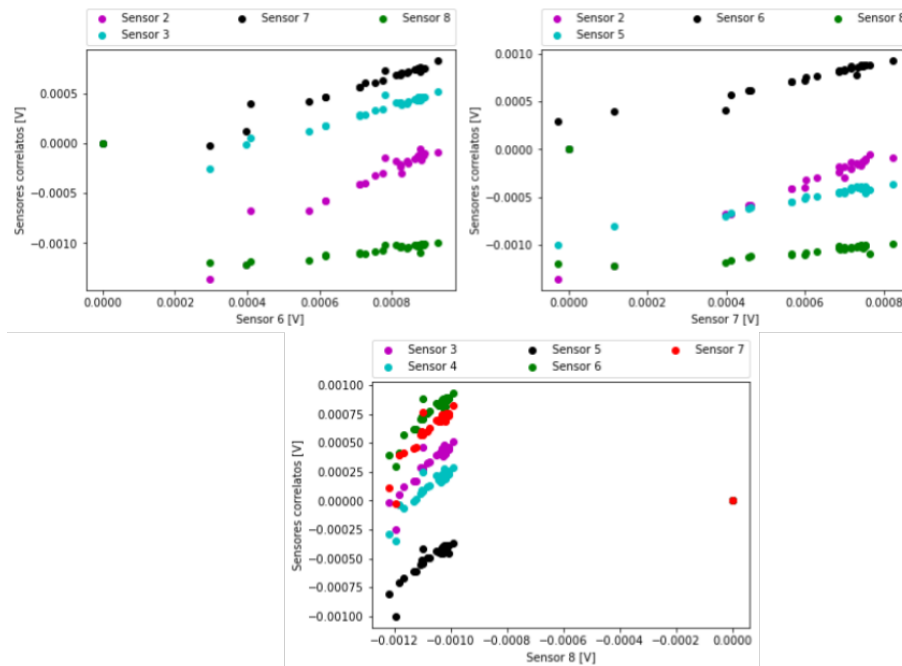
Tomando como partida a tabela 6, as figuras 20 e 21 foram geradas para explorar a relação entre os valores obtidos por seus pares validadores (eixo y) e a medição do sensor de referência (eixo x). Há a presença de alguns outliers que geram uma leve distorção na área do gráfico gerado, o fato é melhor exemplificado pela figura 21 para o sensor 8.

Figura 20 – Evolução média individual ao longo do experimento



Fonte: Autora (2022)

Figura 21 – Evolução média individual ao longo do experimento



Fonte: Autora (2022)

4.2 Treinamento da rede

A aplicação dos hiperparâmetros da tabela 5 mostrada nos métodos, para cada uma das redes neurais resultou nas acurácias e quantidade de falsos positivos, mostradas na tabela 7.

Tabela 7 – Parâmetros de qualidade

Rede neural sensor correspondente	Acurácia	Quantidade de falso positivos
2	93,35	0
3	93,35	0
4	93,35	0
5	93,35	0
6	93,35	0
7	93,35	0
8	93,35	0

Fonte: Autora (2022)

O fato dos dados passarem pelo mesmo processo de normalização e por terem a mesma ordem de grandeza, justifica as acurácias similares para todas as redes. A não aparição de falso positivos é um bom indicativo de que os resultados são condizentes entre si, ao indicar falhas.

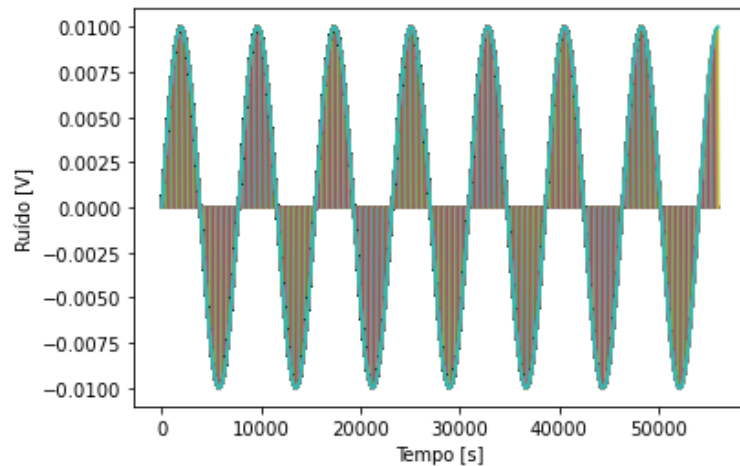
4.3 Predição de novos resultados

Uma nova tentativa de classificação foi feita, usando os dados de treinamento, mas com a introdução de ruído. Nas 5 primeiras medições de cada sensor, foi adicionado a função 7, cujo gráfico é dado pela imagem 22. Já nas 5 últimas medições a função de ruído é dada pela equação 8, cujo gráfico pode ser visto na figura 23.

$$y(t) = 0.01\text{sen}\left(\frac{\pi}{4}t\right) \quad (7)$$

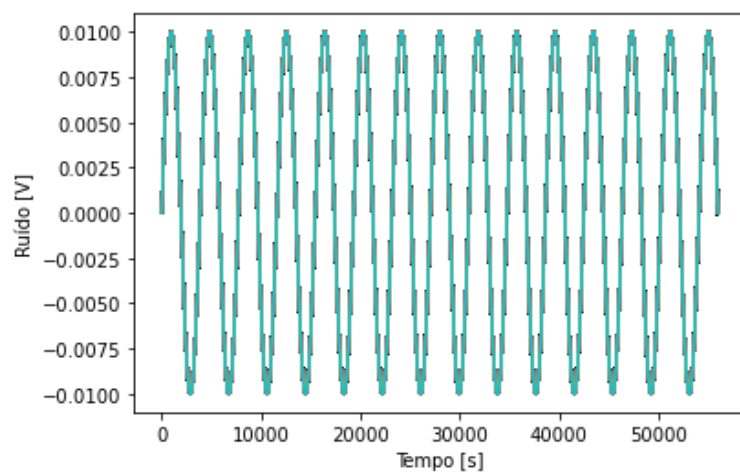
$$y(t) = 0.01\text{sen}\left(\frac{\pi}{2}t\right) \quad (8)$$

Figura 22 – Função de ruído adicionado aos dados da eq. 7



Fonte: Autora (2022)

Figura 23 – Função de ruído adicionado aos dados da eq. 8



Fonte: Autora (2022)

A tabela 8 mostra os resultados para a acurácia e triangulação de dados de cada rede.

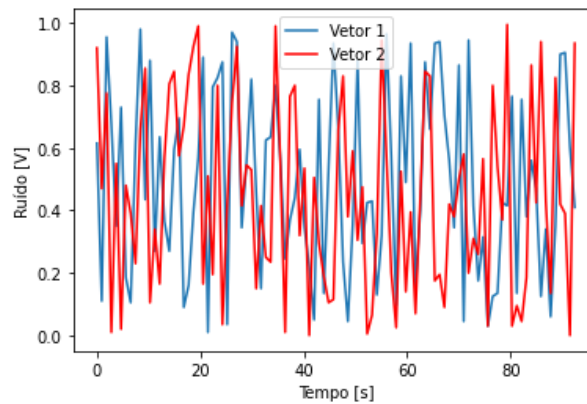
Tabela 8 – Parâmetros de qualidade - após adição de ruído nos dados

Rede neural sensor correspondente	Acurácia	Quantidade de falso positivos
2	93,30	0
3	93,32	0
4	93,30	0
5	93,30	0
6	93,30	0
7	93,30	0
8	93,30	0

Fonte: Autora (2022)

Para investigar se a rede é sensível a variações não harmônicas, foi criado dois vetores com números aleatórios para gerar o ruído dos dados, seguindo a mesma lógica do primeiro vetor ser adicionado as 5 primeiras leituras e o segundo para as 5 últimas. Os valores são inteiros e entre 0 e 200, eles foram normalizados para a ordem de grandeza dos dados dos sensores. A figura 24, mostra os 100 primeiros valores mostrados no gráfico para os dois vetores de ruído. A tabela 9 mostra a acurácia da predição e quantidade de falso positivos.

Figura 24 – Função de ruído adicionado aos dados



Fonte: Autora (2022)

Tabela 9 – Parâmetros de qualidade - após adição de ruído aleatório nos dados

Rede neural sensor correspondente	Acurácia	Quantidade de falso positivos
2	93,33	0
3	93,33	0
4	93,33	0
5	93,33	0
6	93,33	0
7	93,33	0
8	93,33	0

Fonte: Autora (2022)

4.4 Discussões

A rede neural desenvolvida é uma rede neural de recursão (RNN), com camadas ocultas e parâmetro ajustado para mitigar a propagação de erros. Os dados de entrada foram separados em 80% para o treinamento e 20% para o teste, a acurácia é medida pela taxa de acerto na classificação binária de: dano potencial ou não. A rede, para treinamento, recebe a entrada das 10 leituras redundantes e as classifica, a quantidade de dados novos para classificação pode diferir de 10.

A rede mostra bons valores de acurácia, o código desenvolvido para a localização dos falsos positivos não indicou a ocorrência desse problema. Como seu critério de sucesso é pelo menos um dos sensores adjacentes apresentarem erro, isso não impede que o sensor validador também tenha uma predição incorreta. Assim, esse fato limita a confiabilidade do código, sendo a taxa de erro proporcional a quantidade de sensores validadores possíveis.

Embora os resultados de acurácia e quantidade de falsos positivos sejam promissores. A redução que a acurácia teve após a introdução do ruído é muito pequena para ser factível, o fato de nenhum falso positivo ter sido encontrado também chama atenção, pois mesmo com a introdução de ruído harmônico e aleatório, não houve a ocorrência desse erro. Sendo assim, necessário uma investigação desses parâmetros de qualidade, de forma mais detalhada em futuros trabalhos.

O presente trabalho, é um passo inicial para a aplicação de MSE como ferramenta auxiliar a manutenção de aeronaves. Ele mostra o potencial de aplicação das ondas Lamb, pela análise exploratória de dados, é possível observar a correlação de variáveis e a mudança no padrão vibracional detectado por sensores piezoelétricos e ondas Lamb.

Para a sua aplicação cotidiana, deve-se seguir a recomendação de investigar a eficiência do código de validação em encontrar falsos positivos, e sua sensibilidade a configurações adversas, a fim de analisar se os hiper parâmetros de rede adotados sinalizam corretamente a predição de novas falhas na estrutura.

5 CONCLUSÕES

A manutenção de aeronaves é essencial para garantir a segurança e a disponibilidade de aeronaves, soluções para otimizar e atualizar os planos de manutenção são importantes para reduzir o tempo que aeronaves ficam em solo, para as inspeções. Ensaios não destrutivos são eficientes para determinar falhas e propor ações de contenção antes que haja a evolução para eventos críticos, o condition based maintenance (CBM) aplica esse método para alterar as paradas de manutenção pela alteração de propriedades estruturais.

O monitoramento da saúde estrutural (MSE) é uma vertente do CBM, com especificidades benéficas à operação de aeronaves comerciais, os sensores são instalados apenas uma vez, e dessa forma, sua confiabilidade é aumentada por reduzir as variações do processo de fixação. O MSE possibilita a associação de diversos tipos de sensores, possibilitando a redundância das medições, usadas por ferramentas robustas, como o aprendizado de máquina para fazer a predição de danos.

O projeto ReMap da TuDelft é importante para o desenvolvimento e certificação do MSE, pois traz diversos dados, obtidos em ambiente controlado em uma estrutura que mimetiza o revestimento da asa e sua junção com a longarina. O ambiente controlado, torna possível estudar o comportamento desse corpo de prova complexo, mitigando variáveis ambientais, que poderiam influenciar os resultados.

O projeto ReMap possibilita analisar a confiabilidade dos processos de fixação, qual o melhor tipo de sensor para cada aplicação e detecção de danos, além de montar uma base de dados para o treino de soluções de aprendizado de máquina e inteligência artificial. As ondas Lamb têm grande potencial de aplicação pelo MSE, pela sua sensibilidade a mudanças estruturais, sua associação com sensores piezoelétricos é muito vantajosa, pelos custos associados a esse sensor e facilidade de aplicação na estrutura e manipulação para aquisição de dados.

A rede neural apresentada nesse trabalho, teve um bom desempenho com relação à acurácia de predição e eliminação de falsos positivos. A rede apresentou boa adaptação à introdução de ruídos harmônicos e aleatórios. A forma de ordenação dos dados de entrada possibilitou a interpretação do padrão vibracional, possibilitando a classificação dos dados. De modo que pode-se evoluir para uma classificação estrutural entre saudável ou danificada, com suporte de dados coletados dos sensores na aeronave.

Como trabalhos futuros, sugere-se:

- a) mudança no método de detecção de anomalia, possibilidade de usar rede híbrida, para encontrar os outliers por aprendizado não supervisionado;
- b) usar métodos numéricos para aumentar a quantidade de dados de entrada, após o impacto;
- c) junção de configurações de testes diferentes para realizar o treinamento da rede;
- d) explorar o uso do filtro digital com resposta ao impulso de duração infinita(IIR);
- e) elaboração de método de checagem para falsos negativos, por meio do retreino da rede; e
- f) comparação com outros tipos de sensores.

REFERÊNCIAS

- AUFFART, B. **Machine Learning for Time-Series with Python: Forecast, predict, and detect anomalies with state-of-the-art machine learning methods.** [S.l.]: Packt>, 2021. v. 1.
- BARBETTA, P.; REIS, M.; BONIA, A. **Estatística para Curso de Engenharia e Informática.** [S.l.]: Atlas, 2010. v. 3.
- BARBETTA, P.; REIS, M.; BONIA, A. **Anomaly Detection Principles and Algorithms.** [S.l.]: Springer, 2017. v. 1.
- BOEING. **Commercial Market Outlook.** [S.l.], 2022.
- CALLISTER, W. **Ciência e engenharia dos materiais: uma introdução.** [S.l.]: LTC, 2008. v. 7.
- DONG, T.; KIM, N. Cost-effectiveness of structural health monitoring in fuselage maintenance of the civil aviation industry. **MDPI**, v. 5, n. 87, 2018.
- EASA. **EASA releases its Concept Paper ‘First usable guidance for Level 1 machine learning applications.** [S.l.], 2021.
- FAB, F. A. B. **FAB já transportou mais de 200 órgãos para transplante em 2021.** 2021. <<https://www.fab.mil.br/noticias/mostra/37945/>>. Acessado em 03 de Novembro de 2022.
- GAMA, A. G. dos produtores de aviões. **GAMA Annual Report 2019 and 50th Anniversary Edition.** [S.l.], 2019.
- GIURGIUTIU, V. **Structural Health Monitoring with Piezoelectric Wafer Active Sensors.** [S.l.]: Elsevier, 2014. v. 2.
- GORINEVSKY, D. Design of integrated shm system for commercial airera applications. **International Workshop on Structural Health Monitoring**, v. 5, n. 1, p. 1 – 8, 2005.
- HARISSON, M. **Machine Learning – Guia de Referência Rápida: Trabalhando com dados estruturados em Python.** [S.l.]: O’Reilly, 2020. v. 1.
- KINNISON, H.; SIDDIQUI, T. **Aviation Maintenance Management.** [S.l.]: MC Graw Hill, 2012. v. 2.
- MEGSON, T. **Aircraft structures** : for engineering students. [S.l.]: Elsevier, 2017. v. 6.
- MULLER, C. A.; GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists.** [S.l.]: s, 2016. v. 1.
- NORTON, R. **Projeto de máquinas: uma abordagem integrada.** [S.l.]: Bookman, 2013. v. 4.

QING, X. et al. Piezoelectric transducer-based structural health monitoring for aircraft applications. **MPDI**, v. 19, n. 595, p. 2 – 27, 2019.

REMAP. **Deliverable 4.4 SHM data repository**. [S.l.], 2019. Acessado em 01 de Outubro de 2022. Disponível em: <https://h2020-remap.eu/wp-content/uploads/2021/03/ReMAP_D4.4_SHM-data-repository.pdf>.

ADAPTIVE AIRCRAFT MAINTENANCE PLANNING REMAP, R. time Condition-based Maintenance for (Ed.). **Strategic action plan for future CBM adoption**. [S.l.], 2020.

SARANGA, H.; KUMAR. Optimization of aircraft maintenance/support infrastructure using genetic algorithms—level of repair analysis. **Springer Science**, v. 143, n. 91, p. 32–43, 2006.

SU, Z.; YE, L. **Identification of Damage Using Lamb Waves: From Fundamentals to Applications**. [S.l.]: Springer, 2009. v. 42.

TOH, G.; PARK, J. Review of vibration-based structural health monitoring using deep learning. **MDPI applied sciences**, v. 10, n. 5, 2020.

VIEIRA, D.; LOURES, P. Maintenance, repair and overhaul (mro) fundamentals and strategies: An aeronautical industry overview. **International Journal of Computer Applications**, v. 135, n. 12, p. 21–29, 2016.

WANG, Y. et al. Recent progress in aircraft smart skin for structural health monitoring. **Structural Health Monitoring 2022**, v. 21, n. 5, p. 2453–2480, 2022.

WASSINK, J. **Smarter aircraft maintenance could save millions**. 2018. <<https://www.delta.tudelft.nl/article/smarter-aircraft-maintenance-could-save-millions#>>. Acessado em 15 de Novembro de 2022.

XU, J.; KROO, I. Aircraft design with active load alleviation and natural laminar flow. **International Journal of Computer Applications**, v. 51, n. 5, p. 1532–1545, 2014.

APÊNDICE A - CÓDIGO APRENDIZADO DE MÁQUINA

```
#Machine Learning

#Normalizacao de dados
scaler = StandardScaler()

#plt.plot(time_step[:,0],dataset2[:,1], 'b')

#sensor 2
X2_train, X2_test, y2_train, y2_test =
train_test_split(dataset2,
dataset2_clas, test_size=0.8, random_state=0)
clf2 = MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=(10, 5, 2), random_state=1)
clf2.fit(X2_train, y2_train.ravel())
y2_pred = clf2.predict(X2_test)

erro2 = 0
for i in range(0,len(y2_test)):
    if y2_pred[i] != y2_test[i,0]:
        erro2 = erro2 + 1
acracia2 = 100-100*erro2/len(y2_test)

#sensor 3
X3_train, X3_test, y3_train, y3_test =
train_test_split(dataset3, dataset3_clas,
test_size=0.8, random_state=0)
clf3 = MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=(10, 5, 2), random_state=1)
clf3.fit(X3_train, y3_train.ravel())
y3_pred = clf3.predict(X3_test)
```

```

erro3 = 0
for i in range(0,len(y2_test)):
    if y3_pred[i] != y3_test[i,0]:
        erro3 = erro3 + 1
acracia3 = 100-100*erro3/len(y3_test)

#sensor 4
X4_train, X4_test, y4_train, y4_test =
train_test_split(dataset4, dataset4_clas,
test_size=0.8, random_state=0)
clf4 = MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=(10, 5, 2), random_state=1)
clf4.fit(X4_train, y4_train.ravel())
y4_pred = clf4.predict(X4_test)

erro4 = 0
for i in range(0,len(y2_test)):
    if y4_pred[i] != y4_test[i,0]:
        erro4 = erro4 + 1
acracia4 = 100-100*erro4/len(y4_test)

#sensor 5
X5_train, X5_test, y5_train, y5_test =
train_test_split(dataset5, dataset5_clas,
test_size=0.8, random_state=0)
clf5 = MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=(10, 5, 2), random_state=1)
clf5.fit(X5_train, y5_train.ravel())
y5_pred = clf5.predict(X5_test)

erro5 = 0
for i in range(0,len(y2_test)):
    if y5_pred[i] != y5_test[i,0]:
        erro5 = erro5 + 1
acracia5 = 100-100*erro5/len(y5_test)

#sensor 6
X6_train, X6_test, y6_train, y6_test =
train_test_split(dataset6, dataset6_clas,

```

```

test_size=0.8, random_state=0)
clf6 = MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=(10, 5, 2), random_state=1)
clf6.fit(X6_train, y6_train.ravel())
y6_pred = clf6.predict(X6_test)

```

```

erro6 = 0
for i in range(0,len(y2_test)):
    if y6_pred[i] != y6_test[i,0]:
        erro6 = erro6 + 1
acracia6 = 100-100*erro6/len(y6_test)

```

```

#sensor 7
X7_train, X7_test, y7_train, y7_test =
train_test_split(dataset7, dataset7_clas,
test_size=0.8, random_state=0)
clf7 = MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=(10, 5, 2), random_state=1)
clf7.fit(X7_train, y7_train.ravel())
y7_pred = clf7.predict(X7_test)

```

```

erro7 = 0
for i in range(0,len(y2_test)):
    if y7_pred[i] != y7_test[i,0]:
        erro7 = erro7 + 1
acracia7 = 100-100*erro7/len(y7_test)

```

```

#sensor 8
X8_train, X8_test, y8_train, y8_test =
train_test_split(dataset8, dataset8_clas,
test_size=0.8, random_state=0)
clf8 = MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=(10, 5, 2), random_state=1)
clf8.fit(X8_train, y8_train.ravel())
y8_pred = clf8.predict(X8_test)

```

```

erro8 = 0
for i in range(0,len(y2_test)):
    if y8_pred[i] != y8_test[i,0]:

```

```
erro8 = erro8 + 1  
acracia8 = 100-100*erro8/len(y8_test)
```

APÊNDICE B - CÓDIGO TRIANGULAÇÃO DE RESULTADO

```

# Triangulacao de resultado

n2_t    = len(y2_pred)
c2      = np.zeros([n2_t,1])
erro_c2 = 0
for i in range(0,n2_t):
    if y2_pred[i] == y7_pred[i]:
        c2[i,0] = 1;
    else:
        c2[i,0] = 0;
        erro_c2 = erro_c2 + 1

n3_t    = len(y3_pred)
c3      = np.zeros([n3_t,1])
erro_c3 = 0
for i in range(0,n3_t):
    if y3_pred[i] == y4_pred[i]
    or y3_pred[i] == y6_pred[i]
    or y3_pred[i] == y8_pred[i]:
        c3[i,0] = 1;
    else:
        c3[i,0] = 0;
        erro_c3 = erro_c3 + 1

n4_t    = len(y4_pred)
c4      = np.zeros([n4_t,1])
erro_c4 = 0
for i in range(0,n4_t):
    if y4_pred[i] == y3_pred[i]
    or y4_pred[i] == y5_pred[i]
    or y4_pred[i] == y8_pred[i]:

```

```

        c4[i,0] = 1;
    else:
        c4[i,0] = 0;
        erro_c4 = erro_c4 + 1

n5_t    = len(y5_pred)
c5      = np.zeros([n5_t,1])
erro_c5 = 0
for i in range(0,n5_t):
    if y5_pred[i] == y4_pred[i]
    or y5_pred[i] == y7_pred[i]
    or y5_pred[i] == y8_pred[i]:
        c5[i,0] = 1;
    else:
        c5[i,0] = 0;
        erro_c5 = erro_c5 + 1

n6_t    = len(y6_pred)
c6      = np.zeros([n6_t,1])
erro_c6 = 0
for i in range(0,n6_t):
    if y6_pred[i] == y2_pred[i]
    or y6_pred[i] == y3_pred[i]
    or y6_pred[i] == y7_pred[i]
    or y6_pred[i] == y8_pred[i]:
        c6[i,0] = 1;
    else:
        c6[i,0] = 0;
        erro_c6 = erro_c6 + 1

n7_t    = len(y7_pred)
c7      = np.zeros([n7_t,1])
erro_c7 = 0
for i in range(0,n7_t):
    if y7_pred[i] == y2_pred[i]
    or y7_pred[i] == y5_pred[i]
    or y7_pred[i] == y6_pred[i]
    or y7_pred[i] == y8_pred[i]:
        c7[i,0] = 1;

```

```

else:
    c7[i,0] = 0;
    erro_c7 = erro_c7 + 1

n8_t = len(y8_pred)
c8 = np.zeros([n8_t,1])
erro_c8 = 0
for i in range(0,n8_t):
    if y8_pred[i] == y3_pred[i]
    or y8_pred[i] == y4_pred[i]
    or y8_pred[i] == y5_pred[i]
    or y8_pred[i] == y6_pred[i]
    or y8_pred[i] == y7_pred[i]:
        c8[i,0] = 1;
    else:
        c8[i,0] = 0;
        erro_c8 = erro_c8 + 1

# Criacao de ruido
A = 1e-2
t = np.linspace(0,time_step[59998,0],n*30)
y1 = np.zeros([60000,1])
y2 = np.zeros([60000,1])
"""
for i in range(0,60000):
    y1[i,0] = A*np.sin((np.pi/4)*t[i])
    y2[i,0] = A*np.cos((np.pi/2)*t[i])
    y1[1,0] = np.random.randint(200, size=(1))/200
    y2[1,0] = np.random.randint(200, size=(1))/200
"""

y1 = np.random.randint(200, size=(60000))/200
y2 = np.random.randint(200, size=(60000))/200
dataset2b = np.zeros([60000,10])
dataset3b = np.zeros([60000,10])
dataset4b = np.zeros([60000,10])
dataset5b = np.zeros([60000,10])
dataset6b = np.zeros([60000,10])
dataset7b = np.zeros([60000,10])

```

```

dataset8b = np.zeros([60000,10])

for i in range(0,60000):
    for j in range(0,5):
        dataset2b[i,j] = dataset2[i,j]+y1[i]
        dataset3b[i,j] = dataset3[i,j]+y1[i]
        dataset4b[i,j] = dataset4[i,j]+y1[i]
        dataset5b[i,j] = dataset5[i,j]+y1[i]
        dataset6b[i,j] = dataset6[i,j]+y1[i]
        dataset7b[i,j] = dataset7[i,j]+y1[i]
        dataset8b[i,j] = dataset8[i,j]+y1[i]
    for j in range(5,10):
        dataset2b[i,j] = dataset2[i,j]+y2[i]
        dataset3b[i,j] = dataset3[i,j]+y2[i]
        dataset4b[i,j] = dataset4[i,j]+y2[i]
        dataset5b[i,j] = dataset5[i,j]+y2[i]
        dataset6b[i,j] = dataset6[i,j]+y2[i]
        dataset7b[i,j] = dataset7[i,j]+y2[i]
        dataset8b[i,j] = dataset8[i,j]+y2[i]

y2_pred_b = clf2.predict(dataset2b)
y3_pred_b = clf3.predict(dataset3b)
y4_pred_b = clf4.predict(dataset4b)
y5_pred_b = clf5.predict(dataset5b)
y6_pred_b = clf6.predict(dataset6b)
y7_pred_b = clf7.predict(dataset7b)
y8_pred_b = clf8.predict(dataset8b)

erro2b = 0
for i in range(0,len(y2_pred_b)):
    if y2_pred_b[i] != dataset2_clas[i,0]:
        erro2b = erro2b + 1
acracia2b = 100-100*erro2b/len(y2_pred_b)

erro3b = 0
for i in range(0,len(y3_pred_b)):
    if y3_pred_b[i] != dataset3_clas[i,0]:
        erro3b = erro3b + 1
acracia3b = 100-100*erro3b/len(y3_pred_b)

```



```

erro4b = 0
for i in range(0,len(y4_pred_b)):
    if y4_pred_b[i] != dataset4_clas[i,0]:
        erro4b = erro4b + 1
acracia4b = 100-100*erro4b/len(y4_pred_b)

```

```

erro5b = 0
for i in range(0,len(y5_pred_b)):
    if y5_pred_b[i] != dataset5_clas[i,0]:
        erro5b = erro5b + 1
acracia5b = 100-100*erro5b/len(y5_pred_b)

```

```

erro6b = 0
for i in range(0,len(y6_pred_b)):
    if y6_pred_b[i] != dataset6_clas[i,0]:
        erro6b = erro6b + 1
acracia6b = 100-100*erro6b/len(y6_pred_b)

```

```

erro7b = 0
for i in range(0,len(y7_pred_b)):
    if y7_pred_b[i] != dataset7_clas[i,0]:
        erro7b = erro7b + 1
acracia7b = 100-100*erro7b/len(y7_pred_b)

```

```

erro8b = 0
for i in range(0,len(y8_pred_b)):
    if y8_pred_b[i] != dataset8_clas[i,0]:
        erro8b = erro8b + 1
acracia8b = 100-100*erro8b/len(y8_pred_b)

```

Triangulacao de resultado

```

n2_t_b    = len(y2_pred_b)
c2_b      = np.zeros([n2_t,1])
erro_c2_b = 0
for i in range(0,n2_t):
    if y2_pred_b[i] == y7_pred_b[i]:
        c2_b[i,0] = 1;

```

```

else:
    c2_b[i,0] = 0;
    erro_c2_b = erro_c2_b + 1

n3_t_b    = len(y3_pred_b)
c3_b      = np.zeros([n3_t,1])
erro_c3_b = 0
for i in range(0,n3_t):
    if y3_pred_b[i] == y7_pred_b[i]:
        c3_b[i,0] = 1;
    else:
        c3_b[i,0] = 0;
        erro_c3_b = erro_c3_b + 1

n4_t_b    = len(y4_pred_b)
c4_b      = np.zeros([n4_t,1])
erro_c4_b = 0
for i in range(0,n4_t):
    if y4_pred_b[i] == y7_pred_b[i]:
        c4_b[i,0] = 1;
    else:
        c4_b[i,0] = 0;
        erro_c4_b = erro_c4_b + 1

n5_t_b    = len(y5_pred_b)
c5_b      = np.zeros([n5_t,1])
erro_c5_b = 0
for i in range(0,n5_t):
    if y5_pred_b[i] == y7_pred_b[i]:
        c5_b[i,0] = 1;
    else:
        c5_b[i,0] = 0;
        erro_c5_b = erro_c5_b + 1

n6_t_b    = len(y6_pred_b)
c6_b      = np.zeros([n6_t,1])
erro_c6_b = 0
for i in range(0,n6_t):
    if y6_pred_b[i] == y7_pred_b[i]:

```

```
        c6_b[i,0] = 1;
    else:
        c6_b[i,0] = 0;
        erro_c6_b = erro_c6_b + 1

n7_t_b    = len(y7_pred_b)
c7_b      = np.zeros([n7_t,1])
erro_c7_b = 0
for i in range(0,n7_t):
    if y7_pred_b[i] == y7_pred_b[i]:
        c7_b[i,0] = 1;
    else:
        c7_b[i,0] = 0;
        erro_c7_b = erro_c7_b + 1

n8_t_b    = len(y8_pred_b)
c8_b      = np.zeros([n8_t,1])
erro_c8_b = 0
for i in range(0,n8_t):
    if y8_pred_b[i] == y8_pred_b[i]:
        c8_b[i,0] = 1;
    else:
        c8_b[i,0] = 0;
        erro_c8_b = erro_c8_b + 1
```