

Detecção de anomalias em dados de sensores de aeronaves

Rafael da Silva Medeiros
Orientadora: Profa. Dra. Lilian Berton

¹Instituto de Ciência e Tecnologia - ICT
Universidade Federal de São Paulo - UNIFESP
Campus São José dos Campos

rafael.medeiros@unifesp.br, lberton@unifesp.br

Abstract. *A garantia de voos seguros é uma preocupação primordial no setor aeronáutico, exigindo a avaliação cuidadosa de todos os elementos críticos das aeronaves, que são fundamentais para a segurança dos passageiros e tripulantes. A falha inesperada de um componente pode resultar em atrasos, cancelamentos de voos e, em casos extremos, em acidentes aéreos. Nesse contexto, os algoritmos de remaining useful life (RUL), que estimam a vida útil remanescente de um componente antes de uma falha ocorrer, têm se mostrado ferramentas promissoras para otimizar a manutenção e reduzir os custos operacionais das aeronaves. Diante disso, a pesquisa proposta tem como objetivo desenvolver um algoritmo RUL, utilizando modelos de redes neurais, especificamente uma rede neural do tipo LSTM (Long Short-Term Memory), em conjunto com uma camada de atenção, aplicado em dados dos motores Turbofan, simulados no programa C-MAPSS (Commercial Modular Aero-Propulsion System Simulation), disponibilizado pela NASA. Para o futuro, pretende-se avançar ainda mais na pesquisa, explorando outras técnicas de aprendizado de máquina, e realizar análises comparativas com outros algoritmos RUL existentes para validar a eficácia do modelo proposto.*

palavras chave. *aprendizado de máquina, redes neurais, indústria 4.0, sensores, anomalia, manutenção*

1. Introdução

Um dos elementos-chave da indústria 4.0 é a coleta de dados de sensores industriais. Sensores são dispositivos que monitoram e medem diferentes variáveis, como temperatura, pressão, vibração, umidade, entre outros. Esses sensores estão interconectados por meio de redes de comunicação, como a IoT, e são capazes de coletar e transmitir dados em tempo real. O crescimento de fábricas inteligentes oferece uma grande oportunidade para o setor de manufatura se enquadrar nessa revolução industrial. A análise da grande quantidade de *big data* coletada nos sensores no chão de fábrica garante o monitoramento dos ativos de manufatura e fornece meios para realizar a manutenção preditiva a fim de minimizar o tempo de inatividade dos equipamentos [IBM 2023].

Na indústria aeronáutica, esse contexto é de grande relevância, como por exemplo, os motores de aeronaves são equipados com diversos sensores, que fornecem informações importantes sobre o desempenho e condições dos motores. Com o crescimento das fábricas inteligentes, a indústria aeronáutica também se beneficia. A coleta de uma grande quantidade de dados proveniente dos sensores garante o monitoramento dos equipamentos ativos de uma aeronave, como também oferece ferramentas necessárias para implementação da manutenção preventiva.

A indústria aeronáutica se beneficia dessa abordagem, logo que, uma vez que a análise de dados de sensores se torna um aliado indispensável para a manutenção preventiva, que permite antecipar e prevenir falhas em componentes críticos, reduzindo o tempo de inatividade, aumentando o desempenho das aeronaves e garantido a segurança dos voos, o que é prioridade absoluta no setor. Neste contexto, a aplicação de técnicas de análise de dados e aprendizagem de máquina é de extrema importância para aprimorar a eficiência e segurança das operações na indústria.

Nesse trabalho, pretende-se empregar diversos modelos de redes neurais profundas em dados de séries temporais de equipamentos, e comparar a sua eficácia na predição de falhas. Esses modelos conseguem analisar uma grande quantidade de dados em tempo real a fim de descobrir padrões, detectar anomalias e identificar o grau de deterioração do ativo.

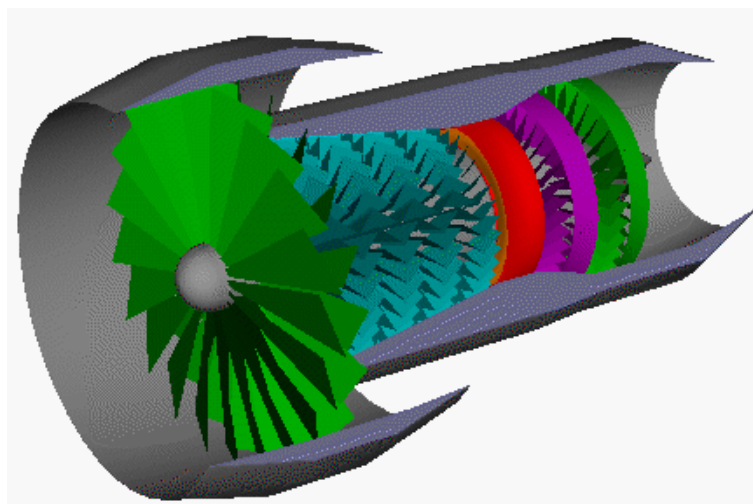


Figure 1. *Turbofan engine* [NASA]

2. Objetivos

Objetiva-se desenvolver modelos de análise de séries temporais de sensores de equipamentos para detectar anomalias. Os objetivos específicos são apresentados a seguir:

- Desenvolver um algoritmo RUL (Remaining Useful Life) baseado em redes neurais LSTM com uma camada de atenção.
- Aplicar o algoritmo em dados dos motores Turbofan para estimar o tempo restante de vida útil dos componentes.
- Realizar experimentos de previsão de séries temporais e avaliação dos resultados.
- Identificar anomalias nos dados estudados.

3. Fundamentação teórica

3.1. Aprendizado de máquina

O aprendizado de máquina é um campo da Inteligência Artificial que busca ensinar máquinas a adquirirem conhecimento e melhorarem seu desempenho com base em experiências e dados de problemas anteriores. Em vez de serem programadas para tarefas específicas, essas máquinas utilizam modelos e algoritmos para identificar padrões, analisar dados e tomar decisões. As abordagens de aprendizado de máquina abrangem uma variedade de sistemas com características particulares e compartilhadas que permitem sua classificação com base na linguagem de descrição, no modo de funcionamento, no paradigma e na forma de aprendizado adotada. Nesta seção, será introduzida alguns dessas características, qual será importante para o entendimento do trabalho.

Neste contexto, existem três tipos de aprendizagem de máquinas, cada uma com características diferentes. O Aprendizado Indutivo que tem a forma de inferência lógica que obtém conclusões a partir de um conjunto específico de exemplos, como o próprio cérebro humano. A indução é caracterizada como raciocínio que parte de um conceito específico e é generalizado. Nesse processo o conceito é aprendido ao realizar inferências indutivas a partir dos exemplos. Entretanto, é importante destacar que as hipóteses geradas por inferência podem ou não ser verdadeiras.

O aprendizado indutivo pode ser separado em duas categorias, o supervisionado e o não supervisionado. No aprendizado supervisionado o algoritmo de aprendizado (ou indutor) recebe um conjunto de exemplos formado por um vetor de características e seus rótulos da classe associada, em geral um conjunto com entradas e saídas desejadas. O objetivo do algoritmo é construir um classificador capaz de determinar corretamente a saída de novos exemplos que não possuem o rótulo da classe.

Por outro lado, no aprendizado não supervisionado, o algoritmo de aprendizado analisa apenas os exemplos de entrada sem seus rótulos de classe. Desta forma o algoritmo procura identificar padrões no conjunto de dados, buscando formar agrupamentos ou *clusters* [Cheeseman et al. 1990]. Após a identificação dos agrupamentos, é comum realizar uma análise para compreender o significado de cada agrupamento dentro do contexto do problema.

Há diversos modelos no campo de Aprendizado de Máquina, como o Simbólico, o Estatístico, o Baseado em Exemplos, o Conexionista e o Genético. Nesta pesquisa, foi empregado o modelo baseado em exemplos.

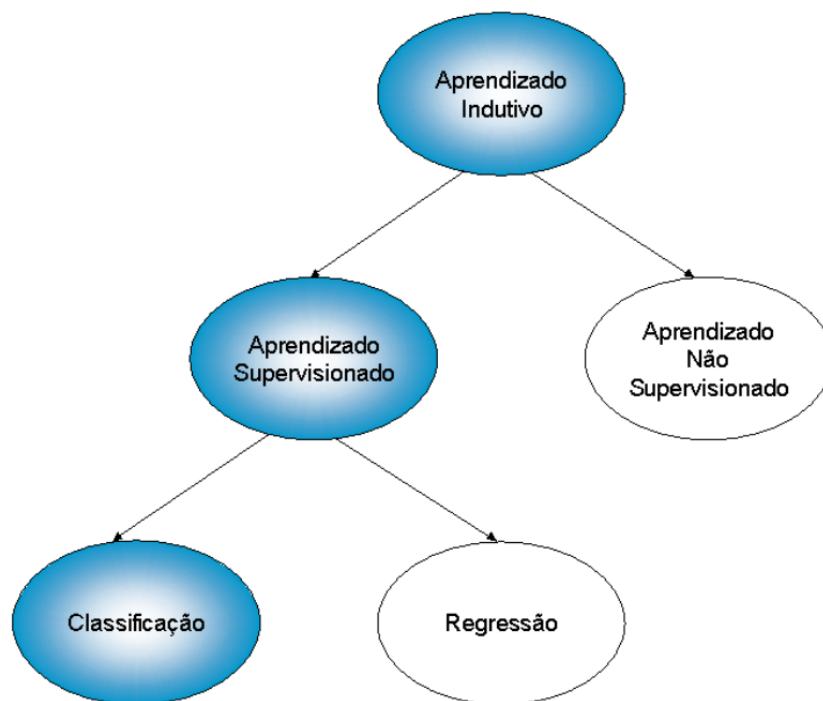


Figure 2. A Hierarquia de aprendizado[Monard and Baranauskas 2003].

O Aprendizado de máquina baseado em exemplos, busca uma maneira de classificar um exemplo utilizando outro exemplo similar de classe conhecida anteriormente, como uma memória, desta forma podendo assumir que o novo exemplo terá a mesma classe. Esse método permite que o sistema classifique novos exemplos nunca vistos com base em exemplos similares conhecidos anteriormente.

Este tipo de aprendizado é chamado de *lazy*, que precisa manter os exemplos na memória para classificar os novos exemplos. Portanto, é essencial saber quais são os exemplos de treinamento que devem ser memorizados por um indutor *lazy*. Logo, é importante reter apenas aqueles exemplos mais representativos do problema.

3.2. Séries temporais

Uma série temporal é composta por um conjunto de observações realizadas em ordem cronológica, podendo ser ou não igualmente espaçadas no tempo. Desse modo, representamos uma Série Temporal Z de tamanho m como uma sequência ordenada de observações, ou seja, $Z = (z_1, z_2, \dots, z_m)$ em que $z_t \in R$ e representa uma observação z no instante de tempo t [Chatfield 2013]. Convencionalmente a análise de séries temporais, é comum realizar a decomposição de suas componentes de tendência, ciclo e sazonalidade [MORETTIN and Toloi 1987].

A tendência de uma série pode ser definida como seu comportamento ao longo do tempo. Isto é, a tendência abrange um comportamento de longa duração, podendo ser tanto constante, linear ou exponencial

Uma tendência constante apresenta uma taxa de crescimento ou declínio estável ao longo do tempo, sem apresentar oscilações consideráveis. Por outro lado, a tendência linear apresenta uma taxa de crescimento e declínio constantes ao longo do tempo. Já a

tendência exponencial indica um crescimento que respeite uma função exponencial, ou seja, os valores da série aumentam (ou diminuem) de uma maneira acelerada. Há uma importância em identificar a tendência da série, pois torna-se possível removê-la com precisão, assim podendo enfatizar os demais componentes da série, o que permite fazer uma análise mais profunda nos dados.

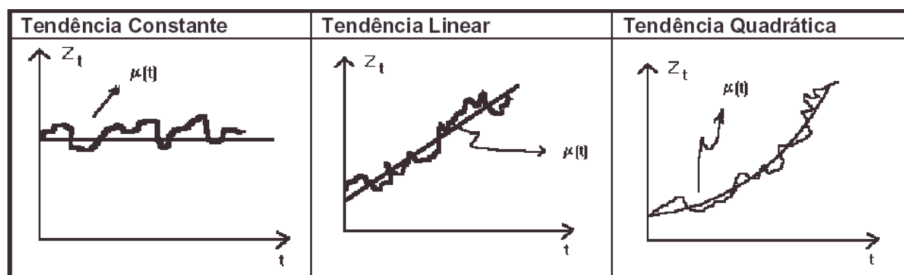


Figure 3. A Hierarquia de aprendizado[Monard and Baranauskas 2003].

A componente do ciclo refere-se como a parte da série temporal que é caracterizada por oscilações de subida e descida, aparentando ser um movimento periódico em torno da tendência ao longo do tempo. A decomposição de uma série de componentes cíclicas é chamada de análise de Fourier, ou análise espectral. [de Figueiredo Soeiro et al. 2012]

Em séries temporais, a sazonalidade indica padrões previsíveis que ocorrem em um determinado período de tempo. Normalmente os padrões estão associados a eventos sazonais, como estações do ano e feriados. Logo o componente sazonal possui movimentos previsíveis, o que se difere do componente cíclico.

3.3. Redes neurais profundas

Redes neurais profundas possuem diversas camadas empilhadas de uma estrutura de Rede Neural Artificial (RNA), que é um modelo computacional baseado no cérebro (sistema nervoso) dos seres vivos. Estas estruturas são compostas por unidades de processamento, os neurônios, que são organizados de forma distribuída e paralela. As unidades são conectadas densamente entre si, o que permite que as informações sejam processadas em diferentes partes da rede.

As Redes Neurais profundas trabalham puramente com a aprendizagem supervisionada, que usa apenas dados de classe rotulada. Como este método de aprendizado é um processo de otimização de parâmetros, este modelo é utilizado quando há um grande volume de dados.

A maioria algoritmos de aprendizado de máquina possuem hiperparâmetros, que são configurações que tem como propósito controlar o comportamento do algoritmo [Goodfellow et al. 2016]). Em geral, os hiperparâmetros são variáveis da estrutura do modelo, como por exemplo, o número de Neurônios (unidades), Função de ativação, taxa de dropout, etc. É importante salientar que os parâmetros são estimados a partir dos dados e os hiperparâmetros são ajustados para ajudar a estimar os parâmetros do modelo durante o processo de aprendizado.

Hiperparâmetros	Características
Número de Camadas Ocultas	Modelos compostos por múltiplas camadas de processamento aprendem a representação do dado em múltiplos níveis de abstração.
Número de Neurônios	Muitas unidades ocultas com técnicas de regularização podem aumentar a precisão. Um número menor de unidades pode causar falta de adequação.
Taxa de Aprendizagem	Relacionado com a técnica de Otimização, uma taxa inadequada resulta em um modelo com baixa capacidade efetiva devido à falha de otimização.
Taxa Dropout	É provável obter um melhor desempenho quando o Dropout é usado, dando ao modelo mais uma oportunidade de aprender representações independentes.
Função de ativação	As funções de ativação são usadas para introduzir a não linearidade em modelos, permitindo que aprendam limites de previsão não lineares.
Momentum	Ajuda a saber a direção do próximo passo com o conhecimento dos passos anteriores, evitando oscilações.
Número de épocas	Número de vezes que todos os dados de treinamento são mostrados à rede durante o treinamento.
Tamanho do batch	É o número de subamostras dadas à rede após o qual a atualização do parâmetro acontece.

Table 1. Hiperparâmetros relacionados à estrutura [Reis 2021]

3.4. LSTM

As Redes LSTM (*Long Short-Term Memory*) um tipo especial de RNN (*Recurrent Neural Networks*), que são um classe de modelos que se destacam com dados sequenciais, essa classe de rede neural possuem conexões entre unidades que formam um ciclo direcionado [Pereira et al. 2017].

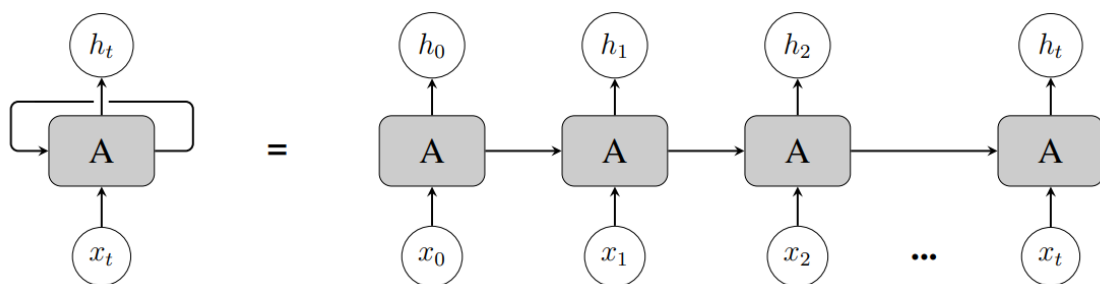


Figure 4. Rede Neural Recorrente[Matsumoto et al. 2019].

No entanto, as RNNs possuem uma limitação conhecida como "dependências de longo prazo". Essa limitação desfavorece a rede quando é necessário buscar uma informação recente para resolver o problema, visto que caso a informação não seja tão recente, há uma grande possibilidade da RNN aprender a informação incorreta.

Para superar estas limitações, as redes LSTM foram introduzidas inicialmente por Hochreiter e Schmidhuber em 1997. Uma rede composta por unidades de memórias, as

células, capazes de aprender de dependências de longo prazo. O'Que se dá através da utilização de portas que interligam as células, sendo elas:

1. **Mecanismo de Esquecimento (*Forget Gate*):** Ele controla quais informações da célula de memória anterior devem ser mantidas ou descartadas. Isso é alcançado usando uma "porta de esquecimento" (forget gate) que calcula um valor entre 0 e 1 para cada unidade da memória da célula.
2. **Mecanismo de Salvamento (*Input Gate*):** O mecanismo de salvamento é responsável por determinar quais informações da entrada atual devem ser adicionadas à memória de longo prazo da célula. Isso é feito usando uma "porta de entrada" (input gate) que decide quais partes dos dados de entrada são relevantes e merecem ser armazenadas na memória de longo prazo.
3. **Transformação da Memória de Longo Prazo em Memória Útil (*Output Gate*):** A memória de longo prazo contém informações relevantes a longo prazo, mas nem todas são necessárias ou úteis em todos os momentos. O mecanismo de transformação da memória de longo prazo em memória útil é controlado pela "porta de saída" (output gate). Essa porta decide quais partes da memória de longo prazo devem ser consideradas ao calcular a saída da célula.

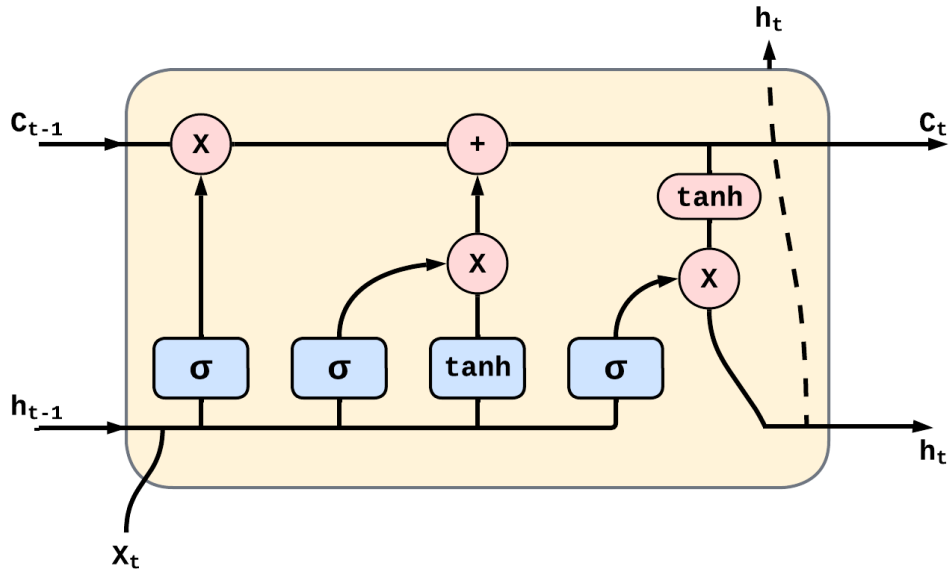


Figure 5. Estruturas de um neurônio de uma rede neural recorrente LSTM.

Onde C_t indica a memória de longo prazo do modelo, x_t representa os dados de entrada, e h_t retrata a saída no instante t . No modelo LSTM, a função sigmóide (σ), atua como um filtro, determinando se um sinal será permitido ou não. Enquanto a função tangente hiperbólica (\tanh) gera novos elementos para a memória de longo prazo. Essas interpretações são expressas nas equações 1 e 2:

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

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

Definadas por [Castelão et al. 2018], as funções de cada porta de uma rede neural LSTM são descritas por:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (4)$$

$$C_t = f_t C_{t-1} + i_t (\tanh(W_C[h_{t-1}, x_t] + b_C)) \quad (5)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \tanh(C_t) \quad (7)$$

. Nas funções apresentadas, f_t indica a porta de esquecimento (*forget gate*), i_t atua como a porta de salvamento (*input gate*), e o_t é a porta da Memória de Longo Prazo em Memória Útil (*output gate*). Os parâmetros de ajuste W e b correspondem aos pesos e ajustes por viés que ocorrem dentro da célula da rede neural. Logo, replicando esta estrutura para todas unidades da rede neural, temos a arquitetura de uma rede neural recorrente LSTM. [Souza 2019]

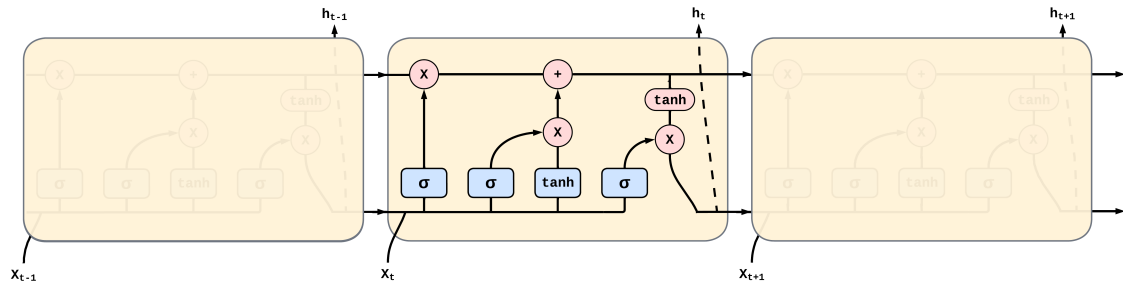


Figure 6. Arquitetura de uma rede neural recorrente LSTM.

3.5. Camada de Atenção

A idealização de atenção foi introduzida por Bahdanau et al. em 2015 para resolver problemas de tradução automática. Embora o conceito tenha sido desenvolvido para este problema específico, ele pode ser utilizado em outros campos do aprendizado de máquina.

Quando lidamos com sequências longas de entrada, modelos do tipo LSTM tem uma queda de desempenho à medida que a sequência aumenta. Isso ocorre pois quanto mais longa a sequência mais difícil é para o modelo prestar atenção em todos elementos. Logo, com apenas a arquitetura LSTM, não é possível destacar partes específicas de uma

sequência. E juntamente com uma camada de atenção, o modelo consegue dar atenção para estas partes.

Na abordagem global a cada instante de tempo, utilizamos os estados ocultos como entrada. Em seguida, determinamos um vetor de contexto c_t que concentra informações importantes relacionadas à próxima saída, y_{t+1} . O vetor de atenção será da forma a_t . [Da Costa et al. 2019]

$$a_t = \tanh(W_e[c_t; h_t]) \quad (8)$$

Em que o vetor c_t também é definido como:

$$c_t = \sum_{j=1}^t \alpha_{t,j} h_j \quad (9)$$

$\alpha_{t,j}$ são os pesos que são multiplicados por cada pedaço da sequência, e é representado pela equação:

$$\alpha_{t,j} = \frac{\exp(e(h_t, h_j))}{\sum_{j=1}^t \exp(e(h_t, h_j))} \quad (10)$$

e $e(h_t, h_j)$ é dado pela equação:

$$e(h_t, h_j) = h_t^T W h_j \quad (11)$$

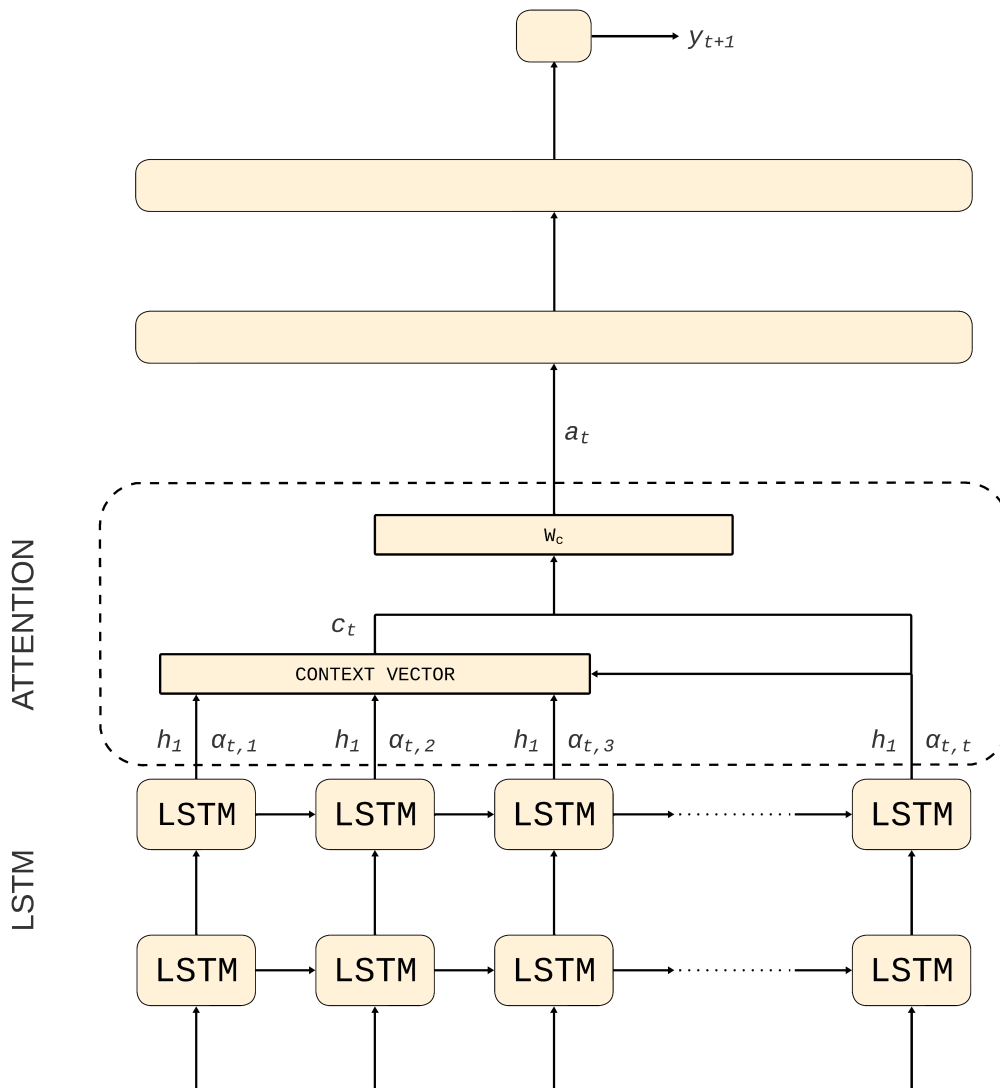


Figure 7. Arquitetura LSTM com uma camada de atenção.

4. Materiais e Métodos

4.1. Datasets

Alguns dos datasets que visamos explorar são descritos a seguir, porém, não iremos nos limitar apenas a eles:

4.1.1. NASA Turbofan Jet Engine Data Set

Este conjunto de dados é o conjunto principal para nossa pesquisa, visto a relevância do tema crítico para a indústria aeronáutica, que se baseia em prognósticos e gerenciamento de integridade para prever o estados de elementos ativos, evitando a falha e interrupção. O conjunto de dados utilizado é uma versão do Kaggle do conjunto de dados públicos de degradação de ativos da NASA, que tem como foco a modelagem da degradação de motores a jato turbofan.

Os dados possuem simulações do tipo *run-to-failure*, de motores a jato turbofan, gerados utilizando o simulador C-MAPPS (*Commercial Modular Aero-Propulsion System Simulation*). Onde foram simulados quatro conjuntos de dados distintos, logo, cada conjunto possui diferentes condições e modos de falha. Cada conjunto de dados inclui 26 variáveis de sensores que capturam a evolução das falhas.

O conjunto de dados é fornecido como um arquivo de texto, contendo 26 colunas de números separados por espaços. Cada linha representa uma instantânea de dados coletados durante um único ciclo operacional, e cada coluna representa uma variável diferente.

	FD001	FD002	FD003	FD004
Trajетórias de treinamento	100	260	100	248
Trajетória de testes	100	259	100	249
Condições	1 (Nível do mar)	6	1 (Nível do mar)	6
Modos de falha	1 (Degradação do HPC)	2 (Degradação do HPC e Fan)	1 (Degradação do HPC)	2 (Degradação do HPC e Fan)

Table 2. Organização do conjunto de dados.

O objetivo principal do conjunto de dados é estimar o RUL (*Remaining Useful Life*) de cada motor no conjunto de testes, ou seja, o número de voos restam para um motor após o último ponto de dados disponível no conjunto de testes.

4.1.2. New NASA Turbofan Jet Engine Data Set

Este novo conjunto de dados é uma versão atualizada do conjunto de dados do *Turbofan Data Set* (N-CMAPSS), fornecidos pela NASA em 2021. Ele consiste em várias séries temporais de medições de sensores realizadas em cada ciclo de operação. À medida que o tempo avança, os motores sofrem degradação. Cada conjunto de dados é subdividido em subconjuntos, que podem apresentar diferentes modos de falha e condições de operação. Tendo como propósito de estimar o RUL utilizando quatro tipos de dados Descrições de cenário - w (Tabela 3), Medidas - x_s (Tabela 4), Sensores Virtuais - x_v (Tabela 5) e Dados auxiliares - α (Tabela 6) [Leal et al. 2022].

Name	# Units	Failure Modes #	Fan E F	LPC E F	HPC E F	HPT E F	LPT E F
DS01	10	1				✓	
DS03	15	2				✓	✓ ✓
DS04	10	3	✓ ✓				
DS05	10	4			✓ ✓		
DS06	10	5		✓ ✓	✓ ✓		
DS07	10	6					✓ ✓
DS08a	15	7	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓
DS08c	10	7	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓
DS08d	10	7	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓

Figure 8. Modos de falha para cada um dos conjuntos de dados.

	Símbolo	Descrição	Unidade
1	alt	Altitude	ft
2	Mach	Número Mach	-
3	TRA	Ângulo do acelerador-resolver	2
4	T2	Temperatura total na entrada do fan	°R

Table 3. Descrições de cenário - w

	Símbolo	Descrição	Unidade
5	Wf	Fluxo de combustível	pps
6	Nf	Velocidade física do fan	rpm
7	Nc	Velocidade física do núcleo	rpm
8	T24	Temperatura total na saída LPC	°R
9	T30	Temperatura total na saída HPC	°R
10	T48	Temperatura total na saída HPT	°R
11	T50	Temperatura total na saída LPT	°R
12	P15	Pressão total no <i>bypass-duct</i>	psia
13	P21	Pressão total na saída do fan	psia
14	P24	Pressão total na saída LPC	psia
15	Ps30	Pressão total na saída HPC	psia
16	P40	Pressão total na saída do queimador	psia
17	P50	Pressão total na saída LPT	psia

Table 4. Medidas - x_s

	Símbolo	Descrição	Unidade
18	T40	Temperatura total na saída do queimador	°R
19	P30	Pressão total na saída HPC	psia
20	P45	Pressão total na saída HPT	psia
21	W21	Fluxo do fan	pps
22	W22	Fluxo para fora do LPC	lbm/s
23	W25	Fluxo para HPC	lbm/s
24	W31	Desvio de refrigerador do HPT	lbm/s
25	W32	Desvio de refrigerador do HPT	lbm/s
26	W48	Fluxo para fora do HPT	lbm/s
27	W50	Fluxo para fora do LPT	lbm/s
28	epr	Relação de pressão do motor (P50/P2)	-
29	SmFan	Margem de estol do fan	-
30	SmLPC	Margem de estol do LPC	-
31	SmHPC	Margem de estol do HPC	-
32	NRf	Velocidade do fan corrigida	rpm
33	NRc	Velocidade do núcleo corrigida	rpm
34	PCNfR	Velocidade do núcleo corrigida	pct
35	phi	Relação entre o fluxo de combustível e Ps30	pps/psi

Table 5. Sensores Virtuais - x_v

	Símbolo	Descrição
36	Fc	Constante de voo
37	hs	Estado de saúde do equipamento

Table 6. Dados auxiliares - α

4.2. Bibliotecas

O projeto será desenvolvido em Python com as seguintes bibliotecas:

- TensorFlow: é uma biblioteca de código aberto para aprendizado de máquina.
- Keras: é uma API alto nível para construir e treinar modelos de redes neurais.
- NumPy: é o pacote básico para a computação científica com o Python, que permite trabalhar com arranjos, vetores e matrizes de N dimensões.
- scikit-learn: é uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python.

- Matplotlib: é uma biblioteca para criar visualizações estáticas, animadas e interativas em Python.

4.3. Avaliação

Para avaliar o desempenho de algoritmos de classificação podem ser utilizadas as seguintes medidas.

4.3.1. Erro Quadrático Médio (MSE)

O MSE (*Mean Squared Error*) ou Erro Quadrático Médio é uma medida comumente usada para avaliar a qualidade de um modelo de regressão. Ela calcula a média dos erros quadráticos entre as previsões do modelo e os valores reais dos dados. Um valor de MSE mais baixo indica que o modelo tem uma melhor precisão, ou seja, suas previsões estão mais próximas dos valores reais.

A fórmula para calcular o MSE é a seguinte:

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_{pred_i} - y_{true_i})^2 \quad (12)$$

- MSE é o valor do Erro Quadrático Médio.
- m é o número de exemplos ou amostras nos dados.
- y_{pred} é o valor previsto pelo modelo para a i -ésima amostra.
- y_{true} é o valor real ou verdadeiro correspondente para a i -ésima amostra.

O MSE é uma métrica comumente usada porque penaliza de forma mais acentuada os erros maiores de previsão, devido ao fato de elevarmos esses erros ao quadrado. Por este mesmo fato, ele pode ser de difícil interpretação se comparado aos dados reais, sendo interessante o uso do RMSE para essa interpretação.

4.4. Raiz do Erro Quadrático Médio (RMSE)

O RMSE (*Root Mean-Square Error*), ou Raiz do Erro Quadrado Médio, nos entrega uma noção da quantidade de erros gerados pelo sistema em suas previsões para problemas de regressão, dando um maior peso para grandes erros [Géron 2019].

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_{pred_i} - y_{true_i})^2} \quad (13)$$

A equação 13 calcula a diferença entre cada previsão feita pelo modelo (y_{pred}) e o valor real correspondente (y_{true}), elevando essa diferença ao quadrado. Em seguida, todas as diferenças ao quadrado são somadas e divididas pelo número de instâncias no conjunto de dados (m). Finalmente, é aplicada a raiz quadrada a essa média dos erros quadráticos, resultando no RMSE.

O RMSE fornece uma medida do desvio médio entre as previsões do modelo e os valores reais. Quanto menor for o valor do RMSE, mais precisas são as previsões do modelo em relação aos dados reais. Essa métrica é amplamente utilizada em problemas de regressão para avaliar a qualidade do ajuste do modelo e comparar diferentes modelos para escolher o mais adequado.

4.5. Erro Médio Absoluto (MAE)

Embora o RMSE seja a principal medida de desempenho para tarefas de regressão, em casos contendo muitos *Outliers* podemos preferir o MAE (*Mean Absolute Error*) ou Erro Médio Absoluto.

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_{\text{pred}_i} - y_{\text{true}_i}| \quad (14)$$

O MAE (Equação 14) é uma medida de erro comumente usada para avaliar a precisão de um modelo de regressão. Ele calcula a média das diferenças absolutas entre as previsões do modelo e os valores reais dos dados.

4.6. Coeficiente de Determinação (R^2)

O coeficiente de determinação, também conhecido como R^2 (*R-squared*), é uma medida estatística utilizada para avaliar a qualidade de um modelo de regressão. O R^2 varia de 0 a 1 e é expresso em termos percentuais da variabilidade da variável dependente que é explicada pelas variáveis independentes do modelo. Quanto mais próximo de 1, maior é a proporção de variabilidade explicada pelo modelo e melhor é o ajuste à amostra.

A fórmula para calcular o coeficiente de determinação é a seguinte:

$$R^2 = 1 - \frac{\sum (y_{\text{true}} - y_{\text{pred}})^2}{\sum (y_{\text{true}} - \bar{y})^2} \quad (15)$$

O numerador da Equação 15 representa a soma dos quadrados dos resíduos, que é uma medida da variabilidade não explicada pelo modelo. O denominador representa a soma dos quadrados totais, que é uma medida da variabilidade total dos valores da variável dependente.

Ao calcular o coeficiente de determinação, subtraímos a proporção da variabilidade não explicada (numerador) da proporção total de variabilidade (denominador). O resultado é uma medida que indica a proporção de variabilidade explicada pelo modelo.

É importante destacar que o coeficiente de determinação não indica a causalidade ou a qualidade preditiva do modelo, mas sim a quantidade de variabilidade explicada em relação à variabilidade total dos dados. Portanto, ele deve ser interpretado com cuidado e em conjunto com outras métricas e análises para uma avaliação completa do modelo.

5. Resultados

5.1. C-MAPPS

Para determinar o melhor modelo LSTM para o conjunto de dados C-MAPPS, foram realizados uma série de testes variando os hiperparâmetros do modelo, incluindo a variação de Dropout e funções de ativação. Que pode dependendo do conjunto explorado o resultado pode variar significamente. Portanto, essas configurações foram ajustadas e otimizadas para alcançar os melhores resultados possíveis.

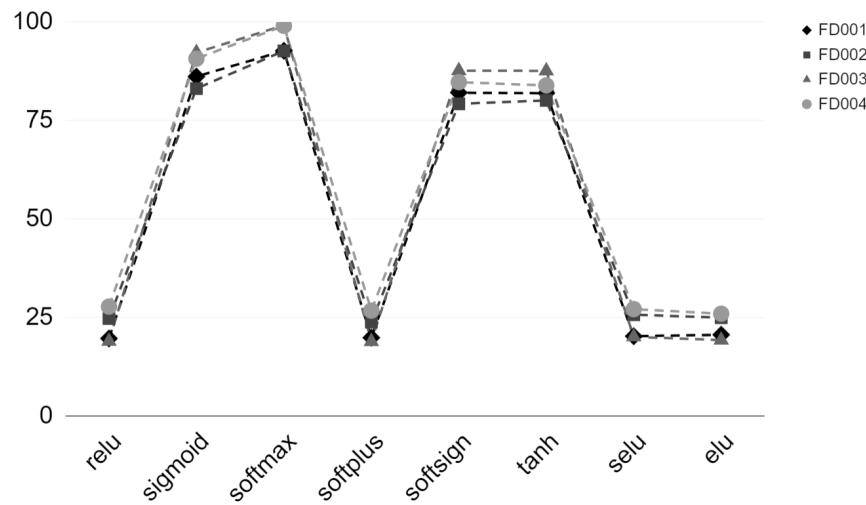


Figure 9. Testes de RMSE para diferentes funções de ativação.

Como à variação de desempenho na camada LSTM em resposta a diferentes funções de ativação, observamos que utilizando a função ReLU obtemos resultados favoráveis em todos os quatro conjuntos de dados. Portanto, selecionamos como a função de ativação mais adequada.

Além da variação da função de ativação, também testamos a influência da variação de dropout na camada LSTM, conforme a Figura 9.

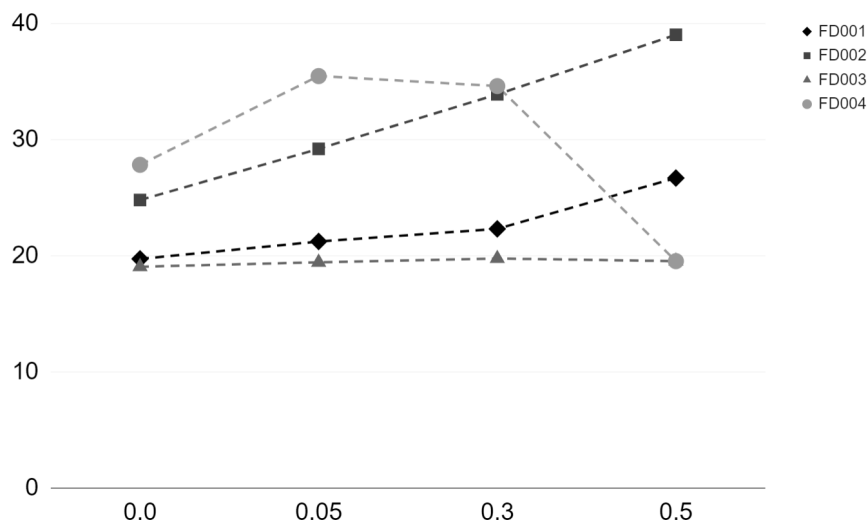


Figure 10. Testes de RMSE para diferentes porcentagens de dropout.

Embora tenha observado melhorias notáveis ao aumentar a taxa de dropout em 50% no conjunto de dados FD004, optamos por selecionar a taxa de dropout de 0%, uma vez que se mostrou a mais adequada para todos os quatro conjuntos de dados.

Foi implementado um modelo de rede LSTM juntamente com uma camada de atenção. Como medida de avaliação para o modelo proposto, utilizamos o RMSE e o

comparamos com outros estudos da literatura. Nesta análise comparativa, apresentamos o desempenho do nosso modelo em relação a outros modelos nos conjuntos de dados C-MAPPS, destacando sua consistência com os outros estudos existentes.

Modelo	RMSE			
	FD001	FD002	FD003	FD004
LSTM + FFNN [Zheng et al. 2017]	16.14	24.49	16.18	28.17
MODBNE [Zheng et al. 2017]	15.04	25.05	12.51	28.66
CNN + FFNN [Li et al. 2018]	12.61	22.36	12.64	23.31
GA + LSTM [Ellefsen et al. 2019]	12.56	22.73	12.10	22.66
LSTM + FFNN [Da Costa et al. 2019]	13.64	17.76	12.49	21.30
LSTM + A [Da Costa et al. 2019]	13.95	17.65	12.72	20.21
LSTM + A proposto	19,27	23,32	18,92	26,43

Table 7. Comparação entre o modelo implementado com outros modelos existentes.

Ao comparar o RMSE do nosso modelo com o modelo GA + LSTM proposto por [Ellefsen et al. 2019], que alcançou os melhores resultados para os conjuntos de dados FD001 e FD003, observamos que nosso modelo proposto apresentou resultados próximos. Sendo mais específico, obtemos um aumento de cerca de 38% no conjunto FD001, um pequeno aumento de 2% no conjunto FD002, aumento de aproximadamente 56% no conjunto FD003 e um aumento de aproximadamente 17% no conjunto FD004.

Agora, comparando com um modelo semelhante ao nosso modelo proposto, LSTM + A proposto por [Da Costa et al. 2019], que se destacou principalmente nos conjuntos de dados FD002 e FD004, registramos um aumento de cerca de 38% no conjunto FD001, um aumento de aproximadamente 32% no conjunto FD002, cerca de 48% de aumento no conjunto de dados FD003 e 30% de aumento no conjunto FD004.

5.2. N-CMAPPS

Ao mudar o conjunto de dados, agora utilizando o N-CMAPPS, foi realizada uma alteração na ativação, substituindo a ReLU pela tahn, visando maximizar os resultados do RMSE.

Conjunto de dados	RMSE
DS01	9.21
DS02	6.94
DS03	9.97
DS04	12.40
DS05	11.23
DS06	12.84
DS07	16.32
DS08a	14.21
DS08c	14.54
DS08d	12.43
Média	12.00

Table 8. Resultados RMSE em todos conjuntos de dados.

Como observado, os valores de RMSE variam entre 6.94 a 16.32, destacando a influência dos diferentes modos de falha de cada conjunto de dados. Os resultados mais baixos, como DS01, DS02 e DS03 indicam que o modelo se adapta melhor nestes cenários e foi capaz de fazer previsões mais precisas nestes conjuntos. Diferentemente, nos demais conjuntos o modelo teve uma maior dificuldade em se adaptar aos padrões dos dados, resultando em valores RMSE razoavelmente maiores.

6. Considerações finais

O modelo LSTM proposto para a previsão de RUL de dados ainda não superou os resultados existentes na literatura, mas demonstrou consistência e abre caminho para futuras melhorias no modelo, mostrando que há uma base sólida para pesquisas futuras.

O aluno participou e apresentou o trabalho no Congresso Acadêmico/Iniciação Científica da UNIFESP 2023. Objetivamos também publicar pelo menos um artigo em um evento brasileiro da área de IA como ENIAC / BRACIS.

References

- Castelão, R., Técnico-IC-PFG, R., and de Graduação, P. F. (2018). Utilização de redes neurais para previsões no mercado de ações. *Projeto final de graduação. Instituto de Computação, Universidade Estadual de Campinas*, pages 13081–970.
- Chatfield, C. (2013). *The analysis of time series: theory and practice*. Springer.
- Cheeseman, P., Stutz, J., Hanson, R., and Taylor, W. (1990). Autoclass iii. *Research Institute for Advanced Computer Science, NASA Ames Research Centre, USA*.
- Da Costa, P. R. D. O., Akcay, A., Zhang, Y., and Kaymak, U. (2019). Attention and long short-term memory network for remaining useful lifetime predictions of turbo-

- fan engine degradation. *International journal of prognostics and health management*, 10(4).
- de Figueiredo Soeiro, J., da Silva, J. L., de Miranda, V. d. F. L., and de Sousa Silva, M. I. (2012). Modelos de séries temporais aplicados ao consumo de energia elétrica do estado do amapá (pp. 127). *Revista da Estatística da Universidade Federal de Ouro Preto*, 2.
- Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., and Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, 183:240–251.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn TensorFlow*. Alta Books.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- IBM (2023). O que é a indústria 4.0? <https://www.ibm.com/br-pt/topics/industry-4-0/>.
- Leal, A. S., Berton, L., and de Castro Santos, L. C. (2022). Learning methods for remaining useful life prediction in a turbofan engine. In *Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional*, pages 556–566. SBC.
- Li, X., Ding, Q., and Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11.
- Matsumoto, D. K. F. et al. (2019). Estudo em séries temporais financeiras utilizando redes neurais recorrentes.
- Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32.
- MORETTIN, P. A. and Toloi, C. M. (1987). Séries temporais, métodos quantitativos. *Atual, São Paulo, SP*.
- NASA. Turbofan engine. <https://www.grc.nasa.gov/www/k-12/airplane/aturbf.html>.
- Pereira, M. d. M. et al. (2017). Aprendizado profundo: redes lstm.
- Reis, C. H. (2021). Otimização de hiperparâmetros em redes neurais profundas. *Minas Gerais*.
- Souza, G. d. O. (2019). Modelos estocásticos e redes neurais: estudo comparativo para previsão de séries temporais.
- Zheng, S., Ristovski, K., Farahat, A., and Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In *2017 IEEE international conference on prognostics and health management (ICPHM)*, pages 88–95. IEEE.