

ESTIMATIVA DO TEMPO DE VIDA ÚTIL PARA MOTORES AERONÁUTICOS COMERCIAIS: UMA ABORDAGEM RECORRENTE E CONVOLUCIONAL

REMAINING USEFUL LIFE ESTIMATION FOR COMMERCIAL AIRCRAFT ENGINES: A RECURRENT AND CONVOLUTIONAL APPROACH

¹William Ludovico Homem

²Vitorino Biazi Neto

¹Depto. de Engenharia Mecânica, Universidade Federal do Espírito Santo
Programa de Educação Tutorial Engenharia Mecânica, Universidade Federal do Espírito Santo.

E-mail: williamlhomem@gmail.com

²Depto. de Engenharia Mecânica, Universidade Federal do Espírito Santo

E-mail: vitorino_biazi@hotmail.com

Artigo submetido em 14/07/2020, aceito em 23/10/2020 e publicado em 28/12/2020.

Resumo: O desenvolvimento de modelos prognósticos possui uma grande importância no campo industrial, uma vez que a utilização da manutenção preditiva, em detrimento da corretiva ou preventiva, permite a minimização de custos de manutenção. Todavia, a criação de modelos baseados na coleta de dados históricos esbarra em dificuldades devido à baixa disponibilização de conjuntos que atendam à característica citada, o que favorece a utilização do banco de dados oriundo do *Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)* para a proposição dessas tecnologias. O texto corrente propõe um modelo recorrente convolucional que estime o tempo de vida útil dos motores aeronáuticos contidos no conjunto C-MAPSS, em que o uso de camadas recorrentes e convolucionais possui o objetivo de interpretar os dados de forma temporal e identificar as características importantes para a estimativa do tempo operacional restante dos equipamentos. Além disso, o artigo aborda os dados de forma conjunta, ao invés da metodologia em frotas, comum aos estudos anteriores a esse, como forma de obter um poder de generalização maior no modelo em questão. Os resultados obtidos são confrontados com os de outros modelos e com soluções de outros autores, sugerindo que o modelo proposto seja uma boa alternativa para a resolução do problema em questão, apresentando uma Raiz do Erro Quadrático Médio (RMSE) de 19,7948 ciclos para os dados conjuntos.

Palavras-chave: rede neural recorrente convolucional; modelos prognósticos; manutenção preditiva; conjunto C-MAPSS.

Abstract: The development of Prognostic and Health Management models has a great importance in industrial field, since the predictive maintenance allows the minimization of maintenance costs, when compared to corrective or preventive maintenance. However, the development of data-driven based

models proves to be difficult due to low availability of datasets that meet the mentioned characteristic, which favors the use of *Commercial Modular Aero-Propulsion System Simulation* (C-MAPSS) dataset for proposing these technologies. The current text proposes a recurrent convolutional model that estimates the Remaining Useful Life of aeronautical engines contained in C-MAPSS dataset, in which the use of recurrent and convolutional layers has the objective of interpreting the data as a time series data and identifying the important characteristics for estimating the remaining operational life of the equipment. It is also chosen to approach the data together, instead of the methodology in fleets, common to previous studies, as a way to obtain a greater generalization capacity in the proposed model. The obtained results are compared with those of other models and with other authors' solutions, suggesting that the proposed model is a good alternative for solving the problem in question, with a RMSE equal to 19.7948 cycles for the joint data.

Keywords: recurrent convolutional neural network; prognostic and health management models; predictive maintenance; C-MAPSS dataset.

1 INTRODUÇÃO

A aplicação de manutenção preditiva e o desenvolvimento de modelos prognósticos é de suma importância para inúmeros segmentos industriais. Em detrimento das convencionais manutenções corretivas e preventivas, essa área busca maximizar o tempo de operação dos equipamentos atuantes em um processo industrial, a partir do monitoramento das suas condições operacionais. Dessa forma, tal estratégia tende a minimizar o custo de manutenção, ao mesmo tempo que aumenta a confiabilidade desses equipamentos (LI et al., 2017).

Em termos gerais, pode-se criar um modelo prognóstico baseado em modelagem física, por meio de dados históricos ou de forma híbrida, mesclando características de ambas abordagens. O uso de dados históricos se mostra atrativo devido à possibilidade de estimar características como o tempo de vida útil (TVU) de um equipamento de forma mais generalizada, sem a necessidade de conhecimento aprofundado do problema. No entanto, essa técnica necessita que um banco de dados robusto seja usado para a identificação das características necessárias à predição. Devido a isso, muitos sistemas utilizam a abordagem física, buscando uma confiabilidade melhor, sem necessidade de realizar um registro prévio (LI et al., 2017; SAXENA et al., 2008).

Foi buscando sanar essa problemática que Saxena et al. (2008) desenvolveram o conjunto de dados C-MAPSS. Registrando o desenvolvimento dos danos que acometem motores aeronáuticos comerciais tipo Turbofan, o autor buscou consolidar a área ao estimular pesquisadores a desenvolverem seus respectivos modelos prognósticos, baseados em dados históricos. Publicado inicialmente pela *2008 International Conference on Prognostics and Health Management*, esse banco de dados é disponibilizado gratuitamente pelo Centro de Excelência em Prognóstico da Nasa (SAXENA et al., 2008; SAXENA & GOEBEL, 2008).

Dentre os muitos modelos propostos, pode-se destacar Heimes (2008) e Peel (2008), que utilizaram modelos baseados em aprendizagem de máquina para a resolução do problema. A capacidade de identificar padrões não lineares em dados expositivos faz com que essa abordagem seja uma poderosa ferramenta para o problema posto. Muitos outros pesquisadores, então, abordaram o tema sob a mesma ótica, utilizando modelos de aprendizado profundo (BABU et al., 2016), máquinas de vetores de suporte (JIN et al., 2019), modelos baseados em aprendizagem semi-supervisionada (ELLEFSEN et al., 2019; VERSTRAETE, DROGUETT e MODARRES, 2020), entre muitas outras abordagens.

Por sua vez, este artigo propõe a utilização de redes neurais artificiais, com camadas recorrentes e convolucionais, para a estimativa do TVU dos motores contidos no conjunto C-MAPSS, por meio do registro dos sensores embarcados nos equipamentos em questão. Além disso, uma abordagem unificada das frotas ali contidas é utilizada, buscado um maior poder de generalização por parte do modelo prognóstico desenvolvido.

A metodologia desenvolvida tem como finalidade ser um estimador confiável, utilizando técnicas de aprendizado de máquinas, do TVU de motores reais funcionando em uma faixa ampla de condições de operação. Sabendo que bancos de dados que mapeiam o desenvolvimento de falhas em tais equipamentos são raros, essa tecnologia tem um papel fundamental na melhoria da eficiência da manutenção preditiva industrial.

2 REFERENCIAL TEÓRICO

O conjunto de dados C-MAPSS, objeto de estudo deste artigo, foi proposto por Saxena et al. (2008) com o intuito de estimular o desenvolvimento de modelos prognósticos baseados na análise do desenvolvimento de falha de motores aeronáuticos e em suas estimativas de vida útil.

Dentre os muitos artigos publicados nessa área, dois autores se destacaram por apresentarem soluções pautadas no uso de redes neurais: Peel (2008) e Heimes (2008).

Peel (2008), que alcançou o terceiro lugar da competição PHM08 (*Prognostic and Health Management 2008*), proposta na *2008 International Conference on Prognostics and Health Management*, sugeriu a utilização de redes neurais do tipo Perceptron Multicamadas e Função de Base Radial para a predição do TVU dos motores contidos no banco de dados. As redes que apresentaram melhores resultados tinham suas saídas mescladas pela utilização de

Filtros de Kalman, reduzindo o ruído da saída. Sob essa perspectiva, o autor obteve um resultado de Raiz do Erro Médio Quadrático (RMSE) igual a 31,3688 ciclos.

Heimes (2008), por outro lado, optou por resolver o problema por meio de Redes Neurais Recorrentes, sanando a demanda de filtragem de ruído não só na saída do modelo, assim como Peel (2008), mas também dentro da própria rede, devido às camadas recorrentes ali existentes. Essa abordagem levou a um modelo final com uma pontuação “s”, proposta por Saxena et al. (2008), de 519,8. Embora o autor tenha optado por não apresentar o RMSE de seu modelo, tal abordagem foi responsável por garantir o segundo lugar na PHM08.

Após esses autores, muitos outros buscaram melhorias para o problema posto acima (LIM et al., 2016; LI et al., 2018; ELLEFSEN et al., 2019; HSU e JIANG, 2018). Por meio da proposição de novos modelos, aplicados em todas as frotas, diferentemente do que acontecia até então, resultados interessantes surgiram.

Redes Neurais Convolucionais foram utilizadas por Babu et al. (2016) nas quatro frotas apresentadas na Tabela 1, obtendo bons resultados e, conseqüentemente, demonstrando a capacidade dos filtros convolucionais de identificar padrões relevantes nos dados de entrada do modelo. Posteriormente, Li et al. (2017) obtiveram resultados ainda mais satisfatórios, utilizando o mesmo tipo de rede em conjunto com uma preparação prévia dos dados de forma temporal.

Ademais, outras abordagens mostraram bons desempenhos, tais como a arquitetura de aprendizagem semi-supervisionada de Ellefsen et al. (2018) e a *Multiobjective Deep Belief Networks* de Zhang et al. (2016).

3 PROCESSOS METODOLÓGICOS

3.1 CONJUNTO DE DADOS

Como bem discutido por Saxena et al. (2008), o desenvolvimento de modelos prognósticos para a estimativa do TVU de equipamentos industriais é deveras dificultado pela falta de dados que descrevam o comportamento dessas ferramentas até o momento de sua falha.

Para que esse registro seja possível, é necessário que haja um grande investimento de tempo e recursos para que frotas de equipamentos sejam instrumentadas, possibilitando, assim, a coleta desses dados. Dessa forma, os bancos de dados que possuem essas informações acabam fora do alcance público, retidas nas mãos das empresas que possuem capital e interesse suficientes para investir nessas tecnologias (SAXENA et al., 2008).

Foi com essa prerrogativa que Saxena et al. (2008) desenvolveram o conjunto de dados C-MAPSS, gerados por um *software* de simulação de mesmo nome, desenvolvido e programado em MATLAB® e Simulink®. Por meio de modelagem física, esse *software* simulou o funcionamento de frotas de motores aeronáuticos tipo Turbofan em diferentes situações de voo e modos de falha, coletando informações relativas ao funcionamento do equipamento, por meio de 21 sensores, tais como temperaturas e pressões das turbinas e compressores.

O conjunto de dados foi dividido em quatro frotas destinadas ao treinamento do modelo prognóstico e quatro frotas para testá-lo, cada uma possuindo um número específico de motores, de condições de voo (altitude de cruzeiro, número de Mach e temperatura do ambiente) e de modos de falha (degradação do compressor de alta pressão e degradação da *fan*, componentes do motor aeronáutico). Cada unidade de motor representa um motor único, mas do mesmo modelo que os demais (SAXENA et al., 2008). A Tabela 1 apresenta a caracterização das frotas que foram separadas como dados de treino e de teste.

No total, foram colhidas informações de 21 sensores para cada ciclo

em que cada motor esteve em funcionamento.

Tabela 1: Descrição das frotas

Dados de Treino			
Frota	N. de motores	Condições de Voo	Modos de Falha
FD001	100	1	1
FD002	260	6	1
FD003	100	1	2
FD004	248	6	2
Dados de Teste			
FD001	100	1	1
FD002	259	6	1
FD003	100	1	2
FD004	249	6	2

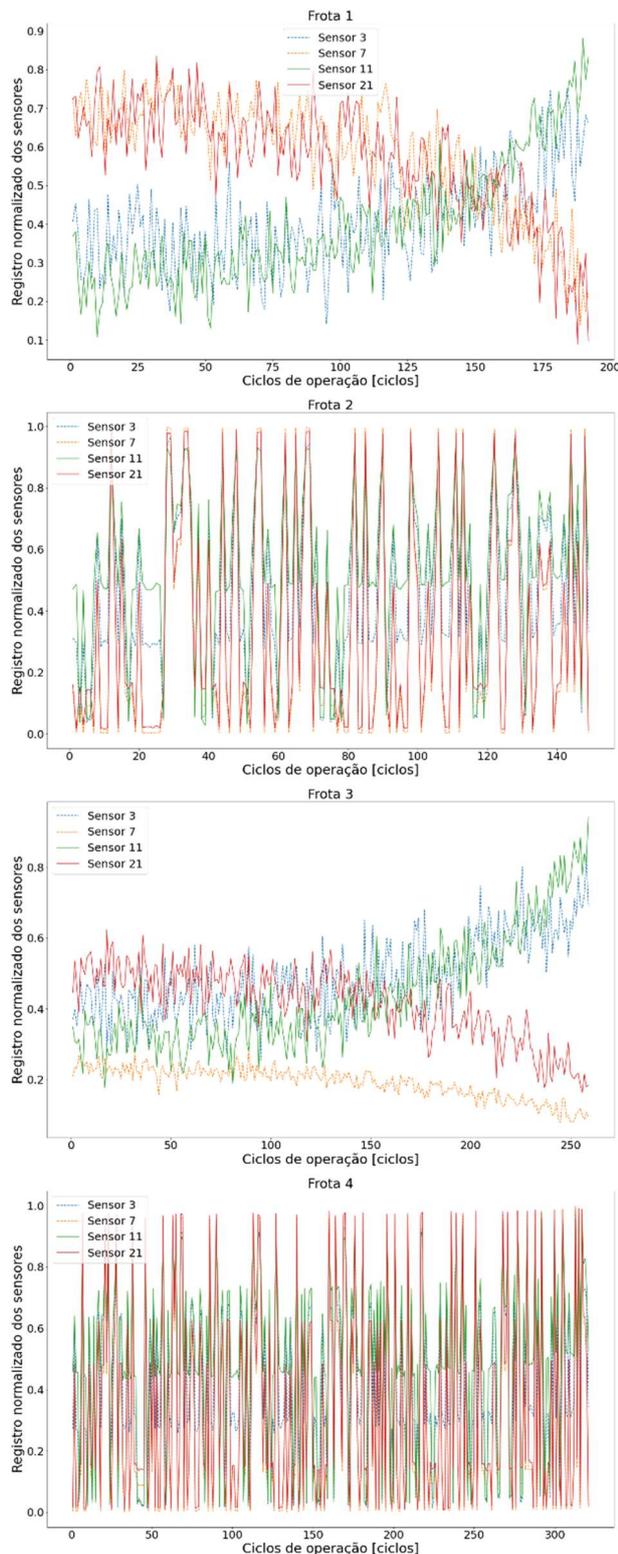
Fonte: Saxena et al. (2008)

A leitura dos sensores foi submetida, ainda, a um ruído, calculado por um conjunto de distribuições normais, com o intuito de simular a complexidade encontrada em dados reais. Para os conjuntos de treino, o registro dos dados cessa no momento da falha da unidade. Por outro lado, para os de teste, o registro é interrompido em um momento anterior, embora a vida útil de cada unidade seja informada. A Figura 1 ilustra os sinais temporais normalizados (abscissas) dos sensores 3, 7, 11 e 21 (ordenadas) para o primeiro motor de cada subconjunto de frotas.

A variável “ciclos”, ilustrada na Figura 1, representa a quantidade de ciclos operacionais (decolagem, cruzeiro e pouso) da unidade de motor em questão e é uma variável de importância, uma vez que se pode utilizá-la para determinar o TVU do motor. Embora o trabalho de Saxena et al. (2008) indique os sensores utilizados durante a simulação, o conjunto de dados

não os identifica. Dessa forma, a análise fica restrita apenas aos seus valores numéricos.

Figura 1: Sensores normalizados da primeira unidade de cada frota



Fonte: elaborado pelo autor

Pode-se verificar que os dados das frotas 1 e 3 são influenciados pelo aumento no número de ciclos de operação dos motores. Os conjuntos 2 e 4, por outro lado, possuem comportamento aleatório em torno de um valor médio bem definido independentemente do número de ciclos, indicando a existência de certa dificuldade de previsão ao submetê-los a um modelo de aprendizagem, quando comparados às frotas 1 e 3, como apontam os estudos de Li et al. (2018), Zhang et al. (2016), Babu et al. (2016), entre muitos outros. Essa característica observada para os 4 sensores selecionados na Figura 1 se estende para os demais sensores.

É fácil observar, também, por meio de uma análise visual qualitativa, que os dados dos sensores não possuem linearidade com a variável temporal à qual estão atrelados (ciclo ou tempo de vida útil), o que pode ser verificado, quantitativamente, ao calcular os Coeficientes de Correlação de Pearson, definidos pela Equação 1.

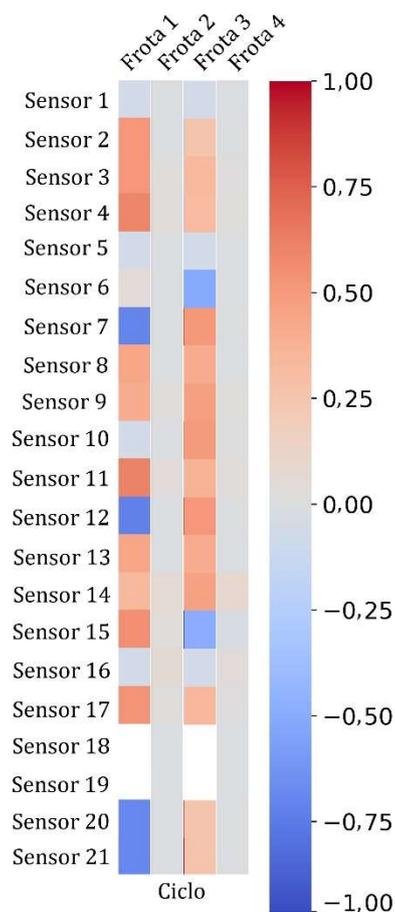
$$r = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

onde x e y e \bar{x} e \bar{y} são, respectivamente, as variáveis de interesse e suas médias (SWINSCOW, 2002).

Esse coeficiente mede o grau de correlação linear existente entre duas variáveis. Se $r = 1$ ou $r = -1$, há uma correlação positiva ou negativa, respectivamente, perfeita entre as variáveis e, caso $r = 0$, não há nenhuma correlação linear.

A Figura 2 ilustra o resultado da análise de correlação entre a variável “ciclos” e os sensores. Pode-se observar, por meio do mapa de cores, que as frotas 2 e 4 possuem uma correlação pouco linear, quando comparada às 1 e 3, corroborando, assim, com a análise gráfica realizada.

Figura 2: Mapa de correlação de Person



Fonte: elaborado pelo autor

3.2 PRÉ-PROCESSAMENTO

Como apontado por Géron (2019), é de extrema importância que os dados sofram uma preparação prévia à aplicação de um modelo de aprendizagem, para que ele possa abstrair de forma eficiente as características intrínsecas ao seu objetivo.

Ao observar os dados C-MAPSS, é possível perceber que não há nenhuma informação explícita sobre o TVU dos motores ali representados. Portanto, se mostra necessária a criação de uma “função alvo” que descreva de forma satisfatória o tempo de operação restante desses equipamentos com base nos dados conhecidos, até que, eventualmente, ocorra uma falha catastrófica.

Para tanto, duas abordagens distintas são comumente utilizadas nas literaturas que abordam o tema deste texto.

A primeira, proposta por Peel (2008), interpreta que o TVU de uma unidade qualquer, pertencente ao conjunto de dados C-MAPSS, decresce linearmente com o início de sua operação. Dessa forma, neste instante, um dado motor possui uma vida útil estimada igual à quantidade de ciclos que ele levaria para apresentar falhas, momento em que sua vida útil se iguala a zero:

$$TVU = C_{falha} - C_{atual},$$

onde “ C_{falha} ” e “ C_{atual} ” são, respectivamente, o ciclo em que aconteceria a falha do motor e o ciclo atual de operação.

Por outro lado, Heimes (2008) sugere que o decrescimento da vida útil de um equipamento só passa a ser observável quando as falhas ali desenvolvidas comprometam o seu funcionamento, como pode ser observado na Figura 3. Sob essa perspectiva, o autor defende que a função que descreve o comportamento do TVU para os dados em questão deve ser linear por partes, mantendo-se constante durante os ciclos iniciais, como ilustra a Equação 2.

$$TVU = \begin{cases} R_{inicial}, & \text{se } R \geq R_{inicial}, \\ R, & \text{se } R < R_{inicial} \end{cases}; \quad (2)$$

$$\forall R = C_{falha} - C_{atual},$$

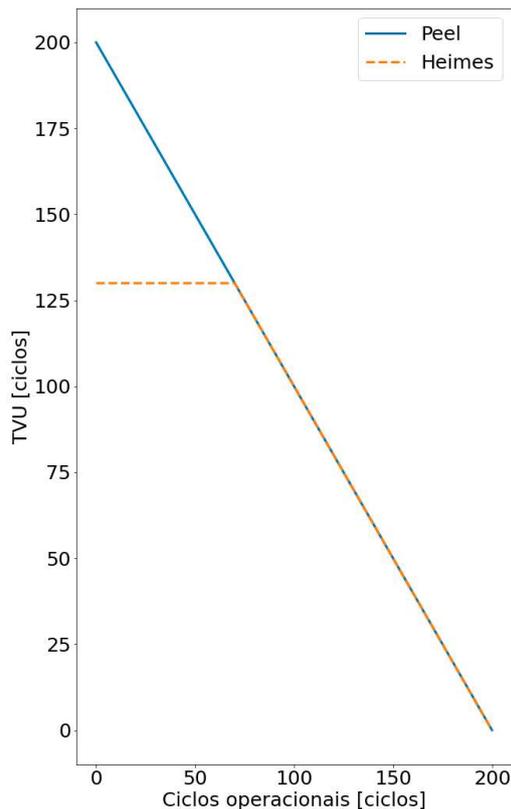
onde $R_{inicial}$ é a faixa constante, anterior ao comprometimento operacional.

À luz das possibilidades, opta-se pela abordagem de Heimes (2008) com um $R_{inicial}$ de 130, como indicado por ele em seu trabalho, por apresentar melhores resultados no modelo proposto no presente artigo, quando comparada com a de Peel (2008).

Como já comentado, o modelo deve prever o TVU de um dado motor a partir das leituras de seus sensores. Portanto, informações como a identificação do motor e o ciclo de operação não devem ser mantidas. As configurações operacionais, embora tenham uma grande influência nos sinais dos sensores, também são retiradas do conjunto de dados para que o modelo não

seja tendencioso em suas escolhas com base na interpretação numérica dessas informações.

Figura 3: Funções alvo para a descrição do TVU



Fonte: elaborado pelo autor

Ao observar o comportamento dos sensores, pode-se identificar, ainda, que, para as frotas 1 e 3, os sensores 1, 5, 6, 10, 16, 18 e 19 se mantêm constantes durante todo o registro. Como bem apontado por Li et al. (2018), esses dados pouco contribuiriam para a aprendizagem do modelo e, portanto, são retirados dos dados de treinamento para que não atrapalhem o processo de aprendizagem.

A maior parte dos artigos que estudaram os dados C-MAPSS utilizou a abordagem sugerida por Saxena et al. (2008), na qual o treinamento e o teste do

modelo de aprendizagem são realizados nos conjuntos separados previamente pelos autores, objetos da competição proposta na 2008 *International Conference on Prognostics and Health Management*.

Todavia, os dados de treino e teste possuem, de forma proposital, comportamentos distintos. O que, *a priori*, era uma estratégia para testar o poder de generalização dos modelos apresentados no desafio, se torna um problema do ponto de visto prático, uma vez que o conjunto de treino não representa, de forma satisfatória, toda a população de unidades de motor.

Por conseguinte, este texto propõe que uma nova amostra de teste seja retirada. Para tanto, unifica-se os dois subconjuntos e, posteriormente, separa-se, de forma aleatória, 30% dos dados para testar o modelo prognóstico, evitando, assim, um provável problema de *overfitting* (super ajuste) que existiria.

Ao adotar tal metodologia, algumas consequências vêm à tona. O número de dados que passam a participar do treinamento se torna muito maior, uma vez que o modelo contacta todas as frotas durante essa etapa, o que é benéfico à aprendizagem. Além disso, o modelo proposto passa a ser capaz de identificar o TVU de um motor que funcione em qualquer configuração operacional, algo que não é possível quando se interpreta os conjuntos de forma separada, tendo em vista que, para este caso, é necessário o treinamento de um modelo individual para cada frota. Como um ônus, o novo conjunto se torna mais complexo e, portanto, espera-se um erro maior em relação ao encontrado em outras literaturas para as frotas 1 e 3, mais simples que as 2 e 4.

Os dados dos sensores, ainda, possuem escalas muito distintas, por se tratar de unidades físicas diferentes, tais como temperatura, pressão, entalpia etc. Para evitar uma possível dificuldade no treinamento, é comum a normalização dos dados de entrada. Para tanto, normaliza-se

os referidos dados por *z-score*, onde o valor normalizado é determinado pela Equação 3.

$$X_{norm} = \frac{X - \bar{X}}{\sigma}, \quad (3)$$

onde X , \bar{X} e σ são, respectivamente, o valor a ser normalizado, a média dos valores e o desvio padrão (MORETTIN, 2017).

São separadas, então, quatro matrizes para utilização durante o treinamento e a validação do modelo. Durante a fase de aprendizagem, o modelo prognóstico entra em contato com uma matriz de 185679 linhas, que representa os registros dos sensores que caracterizam as amostras de treinamento do modelo, e 14 colunas, que representam os sensores selecionados. Além disso, o modelo tem acesso aos rótulos dessas amostras de treinamento, contendo todos os 185679 valores de TVU. Para o teste, o modelo utiliza 79577 amostras e rótulos para estimar o poder de predição do modelo de aprendizagem.

3.3 MODELAGEM E TREINAMENTO

3.3.1 Arquitetura do modelo

Para o problema posto, este texto propõe o uso de uma rede neural com filtros convolucionais e camadas recorrentes do tipo *Long Short Term Memory* (LSTM).

A escolha de utilizar filtros convolucionais se dá devido sua capacidade de identificar padrões relevantes à predição dentro do conjunto de dados de entrada, resultando em um melhor desempenho da rede. Embora essa abordagem seja extremamente comum em visão computacional e processamento de linguagem natural, o uso de Redes Neurais Convolucionais possui bons resultados em problemas de regressão, como mostrado por Li et al. (2018) e Babu et al. (2016) em seus respectivos trabalhos com o mesmo banco de dados.

Devido ao formato do vetor de entrada, já discutido na seção anterior, é

necessário utilizar filtros unidimensionais. Como abordado por Li et al. (2018), para uma entrada temporal $X = [X_1, X_2, \dots, X_N]$, em que “ N ” é igual a 14 no caso exposto por esse texto, a operação dentro das camadas convolucionais é definida pela Equação 4.

$$z_c = \varphi(w_c^T X_{i:i+F_L-1} + b_c), \quad (4)$$

onde φ é uma função de ativação não linear, w_c é o núcleo do filtro convolucional, pertencente ao \mathbb{R}^{F_L} , $X_{i:i+F_L-1}$ é um vetor de tamanho F_L que concatena cada dado em uma sequência maior, começando da posição i e indo até a posição $i + F_L - 1$, e b_c é o termo de *bias* (desvio).

Como discutido por Li et al. (2018), a utilização de camadas do tipo *Pooling* é extremamente comum em Redes Neurais Convolucionais, com o intuito de promover uma extração das características mais significantes para o treinamento do modelo. Entretanto, para problemas com um número pequeno de dimensões, como no caso abordado por este texto, o uso dessas camadas não é recomendado pelo autor supracitado e, portanto, não será adotado.

Após as camadas convolucionais, aplica-se uma camada recorrente para que os mapas de característica sejam tratados de forma temporal, possibilitando, assim, que o modelo leve em consideração dados contactados em iterações passadas. Além disso, Heimes (2008) defende que o uso de camadas recorrentes ajuda na filtragem da saída do modelo, extremamente necessária, uma vez que os sinais de entrada embarcam ruídos oriundos dos sensores.

Para tanto, opta-se pela utilização de células LSTM, proposição de Hochreiter e Schmidhuber (1997), com o intuito de evitar os problemas de “desaparecimento do gradiente” ou “explosão do gradiente” durante a etapa de retro propagação.

A operação recorrente é definida, de forma resumida, por 3 portões: um portão de entrada, um de esquecimento e um de saída. Primeiramente, a célula LSTM decide, levando em consideração a saída

passada e a entrada dessa época, quais informações devem ser descartadas, definidas por meio do portão de esquecimento (f_t) obtido por meio da Equação 5.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (5)$$

onde f_t é a função que define o portão de esquecimento, h_{t-1} é a saída recorrente da época passada, x_t é a entrada presente, em que $[h_{t-1}, x_t]$ representa a concatenação dos dois vetores, σ é a função logística sigmoide e W e b são, respectivamente, os pesos e *bias*, em que o subscrito “f” se refere à camada de esquecimento (OLAH, 2015).

Após isso, a camada recorrente decide quais valores devem ser atualizados e quais são os candidatos a essa atualização. Esse procedimento é realizado pelo portão de entrada (i_t) e por um vetor de novos candidatos (\tilde{C}_t), calculados pelas Equações 6 e 7, respectivamente.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (6)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (7)$$

onde os subscritos i e C se referem ao portão de entrada e ao novo estado da célula, respectivamente, e \tanh é a função tangente hiperbólica (OLAH, 2015).

Por último, o estado da célula (C_t) é atualizado e a camada decide quais valores retornarão na saída (h_t) por meio do portão de saída (o_t), por intermédio das Equações 8, 9 e 10, respectivamente.

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \quad (8)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (9)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (10)$$

onde C_{t-1} é o estado passado, o subscrito “o” se refere ao portão de saída e o símbolo \otimes representa uma multiplicação elemento a elemento (OLAH, 2015).

A escolha dos hiperparâmetros da rede, como os números de filtros convolucionais e de camadas recorrentes, é feita de forma a promover um aprofundamento na rede, permitindo, assim,

que ela possa identificar padrões mais complexos durante a passagem de informação. Para tanto, testa-se combinações distintas de arquitetura, de forma iterativa, selecionando-se a que apresenta melhor resultado.

Com base no exposto, opta-se por 6 camadas convolucionais, sendo as 3 primeiras compostas por 32 filtros cada e as 3 últimas por 64 filtros cada; todos os filtros possuem um tamanho “ F_L ” igual a 10 e função de ativação tangente hiperbólica, como sugerido por Li et al. (2018).

Em relação às camadas recorrentes, opta-se pela utilização de 1024 células LSTM. Após a camada recorrente, utiliza-se uma camada completamente conectada com 512 células e uma camada de saída com 1 célula, ambas com ativação Unidade Linear Retificada (ReLU), ilustrada na Equação 11.

$$ReLU = \max(0, w \cdot x + b), \quad (11)$$

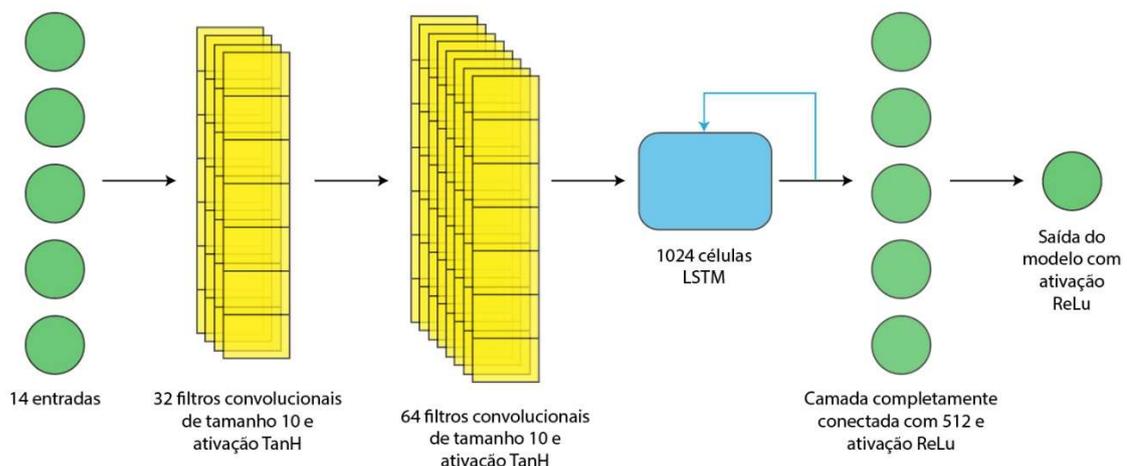
onde w , x e b são, respectivamente, os pesos, entradas e *bias* das camadas às quais estão submetidas a essa função (NIELSEN, 2015).

O uso da camada completamente conectada é importante para agrupar todos os estados de saída das camadas LSTM e, dessa forma, aproximar as saídas da última camada ao formato desejado na camada de saída.

Além disso, a camada completamente conectada é submetida ao efeito de *dropout*, que desativa aleatoriamente 50% das células a cada iteração, evitando, assim, um eventual problema de *overfitting* (NIELSEN, 2015).

A Figura 4 apresenta a arquitetura do modelo proposto de forma gráfica. As setas representam a passagem de informação pela rede, enquanto as camadas completamente conectadas, filtros convolucionais e camadas recorrentes são representadas, respectivamente, pelos objetos de cor verde, amarela e azul.

Figura 4: Arquitetura do modelo proposto



Fonte: elaborado pelo autor

3.3.2 Métricas

Para o processo de treinamento, é necessário definir métricas que serão utilizadas no processo de otimização do algoritmo e na análise dos resultados. Para tanto, opta-se pela utilização do Erro Médio Quadrático (MSE) como função de custo, ilustrada na Equação 12.

$$MSE = \frac{1}{n} \sum_x d^2; \forall d = \hat{y} - y, \quad (12)$$

onde n , y e \hat{y} são, respectivamente, o número de entradas, o valor real esperado e a saída do modelo, em que a somatória é realizada em todas as entradas x (NIELSEN, 2015).

O algoritmo de otimização que opera sobre essa função é o Adadelta, proposto por Zeiler (2012), com uma taxa de decaimento igual a 0,95 e uma constante ϵ igual a $1 \cdot 10^{-7}$. Opta-se pela utilização desse algoritmo em detrimento de outros, como a Descida de Gradiente Estocástica ou Adagrad, em razão da desnecessidade de seleção manual da taxa de aprendizagem, garantindo uma convergência do modelo independente

desse hiperparâmetro (ZEILER, 2012). Como forma de avaliação do modelo, elege-se, todavia, a utilização da RMSE, uma vez que essa função é mais favorável à comparação por apresentar o erro na mesma unidade de suas variáveis dependentes. A RMSE é calculada por meio da Equação 13.

$$RMSE = \sqrt{MSE} \quad (13)$$

Saxena et al. (2008) discutem, ainda, a importância de priorizar modelos que realizam previsões anteriores ao momento de falha dos motores, uma vez que estimativas tardias podem ser catastróficas em um cenário real. A métrica adotada, ilustrada na Figura 5, então, foi utilizada por vários autores, como Li et al. (2017), Babu et al. (2016), Peel (2008) e Zhang et al. (2016), e penaliza a situação mais crítica:

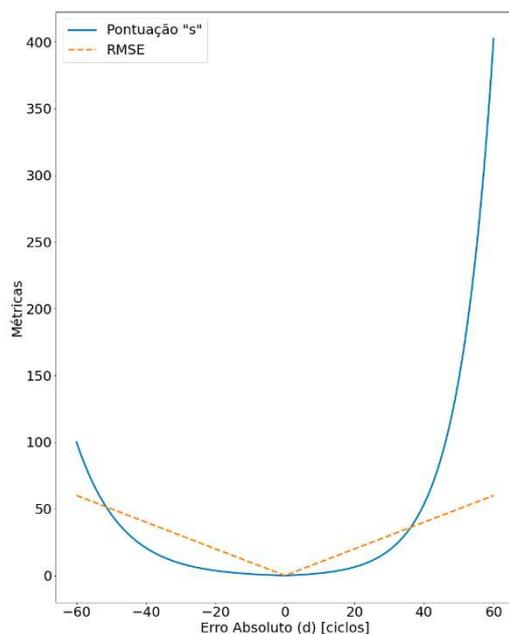
$$s = \begin{cases} \sum_{i=1}^n e^{-\left(\frac{d}{13}\right)} - 1, & \text{se } d < 0 \\ \sum_{i=1}^n e^{\frac{d}{10}} - 1, & \text{se } d > 0 \end{cases} \quad (14)$$

Sendo d o desvio entre o valor real esperado e saída do modelo.

Entretanto, esse texto aplica uma normalização à equação descrita acima, buscando a possibilidade de comparar subconjuntos de tamanhos distintos.

$$s_{norm} = \frac{1}{n} s \quad (15)$$

Figura 5: Métricas em função dos desvios



Fonte: elaborado pelo autor

3.4 PROCEDIMENTO EXPERIMENTAL

Com o objetivo de comparar os resultados obtidos pelo modelo proposto, opta-se pela utilização de outros modelos já abordados na literatura corrente. São eles: rede neural profunda, rede neural puramente convolucional e rede neural puramente recorrente.

Para tanto, a construção desses modelos não é baseada em uma abordagem específica, mas na tentativa de replicar cada camada da Rede Neural Recorrente Convolucional, proposta por esse texto, com o intuito de observar o desempenho dessas estruturas quando operadas individualmente. Após a adoção de uma arquitetura preliminar para cada modelo, os respectivos hiperparâmetros são otimizados

iterativamente, buscando a convergência desses modelos durante o treinamento.

Sendo assim, a Rede Neural Convolucional possui 6 filtros e uma camada completamente conectada, ambos com as mesmas configurações do modelo proposto. A Rede Neural Recorrente, por outro lado, possui uma camada recorrente contendo 256 células LSTM e uma camada completamente conectada com 512 células e ativação ReLu. Por fim, a Rede Neural Profunda simples contém uma camada com 512 células e ativação ReLu.

Assim como realizado no modelo Recorrente Convolucional proposto, camadas *dropout* são acrescentadas, após as camadas completamente conectadas, para evitar o problema de *overfitting*.

Posteriormente, todos os modelos discutidos por esse texto, inclusive o modelo proposto, foram treinados separadamente durante 100 épocas e com um tamanho de lote igual a 1857, com as mesmas amostras de treinamento e rótulos, já discutidos na seção “3.2 Pré-processamento”. O número de épocas e o tamanho de lote, também escolhidos iterativamente, foram os mesmos para todos os modelos, de forma a possibilitar a comparação desses sob as mesmas condições de treinamento.

Os modelos aqui expostos são programados em linguagem Python, utilizando as bibliotecas NumPy (OLIPHANT, 2006), Matplotlib (HUNTER, 2007), Tensorflow (ABADI, 2016) e Keras (CHOLLET, 2015). Os testes e treinamentos são processados em um computador com um processador AMD Ryzen 5 3500X, uma placa de vídeo Nvidia GTX 1050ti e 8GB de memória RAM operando a 3000Mhz.

4 RESULTADOS E DISCUSSÃO

Para o problema em questão, uma análise em termos de RSME e de pontuação “s”, ilustrada na Tabela 2, foi feita entre o

modelo proposto e outros modelos já consolidados na literatura, também implementados pelos autores.

Em termos de RSME, pode-se observar resultados muito parecidos quando se compara o modelo proposto com os modelos convolucional (CNN) e recorrente (RNN). Em contrapartida, a Rede Neural Profunda (DNN) apresenta uma dificuldade relativamente maior em se adequar ao banco de dados, devido a sua arquitetura mais simples. Além da RMSE maior, ela encontra dificuldade em vencer um mínimo local próximo da MSE igual 1646,6634.

Já com relação à pontuação “s”, a Rede Neural Recorrente Convolucional (RCNN) apresenta um valor ligeiramente menor do que as demais, demonstrando, assim, uma maior capacidade de se ajustar a predições prévias, em detrimento às tardias.

A eficácia da abordagem conjunta do banco de dados, por outro lado, é

confrontada com os resultados, em termos de RMSE, encontrados por outros pesquisadores que optaram por uma interpretação das frotas de forma separada, ilustrados na Tabela 3. Pode-se ver que, embora os valores obtidos sejam maiores que os apresentados para as frotas 1 e 3, o modelo proposto (RCNN) possui uma RMSE menor quando comparada com as frotas 2 e 4.

Tabela 2: Resultado dos Modelos

Modelo	RMSE	Pontuações normalizadas
DNN	20,3489	23,1085
CNN	19,6467	26,0468
RNN	19,8147	23,2305
RCNN	19,7948	19,0239

Fonte: elaborado pelo autor

Tabela 3: Abordagens em frotas distintas de outros autores

Frotas	RMSE (Ciclos)			
	CNN (Babu et al., 2016)	MODBNE (Zhang et al., 2016)	DCNN (Li et al., 2017)	DNN Semi-Supervisionado (Ellefsen et al., 2019)
FD001	18,45	15,04	12,61	12,56
FD002	30,29	25,05	22,36	22,73
FD003	19,82	12,51	12,64	12,10
FD004	29,16	28,66	23,31	22,66

Fonte: elaborado pelo autor

Como já discutido outrora, as frotas ímpares possuem uma menor complexidade quando comparadas às frotas pares. Por apresentar sinais com tendências bem definidas, o treinamento de um modelo de aprendizagem utilizando apenas as frotas 1 ou 3 apresenta resultados muito mais satisfatórios do que os das frotas 2 e 4.

Em relação à abordagem conjunta proposta pelos autores, a complexidade do novo conjunto de dados é maior devido à grande diferença entre os comportamentos dos sinais ali contidos. Devido a essa característica, observa-se um maior RMSE, em relação às frotas 1 e 3, aos de autores como Zhang et al. (2016), Li et al. (2017) e Ellefsen et al. (2019).

No que tange os resultados das frotas 2 e 4, expostos na Tabela 3 e de maior complexidade, sem tendência bem definida, o RMSE do modelo RCNN é menor. Esse resultado ilustra uma grande robustez do modelo proposto, denotando uma capacidade de lidar com os sinais de diferentes comportamentos.

Em termos gerais, os resultados expostos acima demonstram que a abordagem proposta por este texto é deveras eficiente no que tange à estimativa do TVU dos motores aeronáuticos contidos no banco de dados C-MAPSS.

Isto porque a referida abordagem unificada, em conjunto com a robusta arquitetura RCNN, promove um alto poder de generalização do modelo, em razão da sua capacidade de lidar com sinais de diferentes comportamentos.

Ademais, a utilização do modelo RCNN permite determinar o TVU por intermédio de apenas um conjunto de dados unificado, em detrimento dos 4 modelos distintos, que necessitam ser treinados e utilizados separadamente. Desta forma, a predição do TVU pode ser realizada sem que haja uma classificação dos sensores com base na condição operacional do motor.

5 CONCLUSÕES

Neste artigo, uma nova abordagem para a estimativa do TVU das unidades contidas no banco de dados C-MAPSS foi proposta. Camadas recorrentes e convolucionais foram aplicadas em conjunto para a identificação de padrões relevantes à predição e para o tratamento dos dados de forma temporal. Ademais, a técnica de *dropout* foi usada para evitar os eventuais problemas de *overfitting*.

Um estudo prévio dos dados foi realizado, identificando algumas características nos conjuntos para que eles pudessem ser preparados. Sensores constantes foram identificados e retirados,

uma vez que a ausência de variação ocasiona uma dificuldade na aprendizagem.

Os dados que alimentam o modelo citado foram pré-processados para que ele tivesse um melhor desempenho. Diferentemente de outros autores, as frotas foram unidas, buscando uma maior generalização do modelo, e um novo conjunto de teste foi retirado de forma aleatória, com o intuito de que esse representasse melhor os dados. Ademais, uma “função alvo” baseada em uma interpretação linear por partes foi criada para a estimativa do TVU das unidades e os dados dos sensores foram normalizados por *z-score*.

Métricas foram definidas para pautar a etapa de aprendizagem do modelo prognóstico e sua posterior análise. MSE foi utilizada como função de custo e otimizada por meio do algoritmo Adadelta. Em contrapartida, a verificação do modelo foi realizada por meio de RMSE e de uma pontuação “s”, que punia mais severamente as predições tardias.

O modelo proposto foi confrontado, então, com outros modelos e seus resultados foram comparados com os de outros autores, demonstrando um bom desempenho para o problema, com valores ligeiramente melhores e com um poder de generalização maior.

Dados que mapeiem o desenvolvimento de falhas de equipamentos, tais como motores aeronáuticos, são raros e, portanto, a metodologia aplicada se mostrou uma alternativa eficiente para a estimativa do TVU de unidade de motores reais, por meio da aplicação posterior de técnicas de Aprendizagem por Transferência ou outras que cumpram o mesmo objetivo.

REFERÊNCIAS

ABADI, Martín, et al. Tensorflow: A system for large-scale machine learning. **12th Symposium on operating**

systems design and implementation, 2016. p. 265-283.

BABU, Giduthuri Sateesh; ZHAO, Peilin; LI, Xiao-Li. Deep convolutional neural network based regression approach for estimation of remaining useful life.

International conference on database systems for advanced applications. Springer, Cham, 2016. p. 214-228.

CHOLLET, F. Keras. **GitHub**. 2015.

ELLEFSSEN, André Listou et al. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. **Reliability Engineering & System Safety**, v. 183, p. 240-251, 2019.

GÉRON, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. **O'Reilly Media**, 2019.

HEIMES, F. O. Recurrent neural networks for remaining useful life estimation. **IEEE International Conference on Prognostics and Health Management**, Denver, CO, USA, 2008. pp. 1-6.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, v. 9, n. 8, p. 1735-1780, 1997.

HSU, C.; JIANG, J. Remaining useful life estimation using long short-term memory deep learning. **2018 IEEE International Conference on Applied System Invention (ICASI)**, Chiba, 2018, pp. 58-61.

HUNTER, John D. Matplotlib: A 2D graphics environment. **Computing in science & engineering**, v. 9, n. 3, p. 90-95, 2007.

JIN, Baihong, et al. A one-class support vector machine calibration method for time

series change point detection. **2019 IEEE International Conference on Prognostics and Health Management (ICPHM)**. IEEE, 2019. p. 1-5.

LI, X.; DING, Q.; SUN, J. Remaining useful life estimation in prognostics using deep convolution neural networks. **Reliability Engineering & System Safety**, v. 172, p. 1-11, 2018.

LIM, Pim et al. A time window neural network based framework for Remaining Useful Life estimation. **2016 International Joint Conference on Neural Networks (IJCNN)**, Vancouver, BC, 2016.

MORETTIN, P. A.; BUSSAB, W. O. Estatística básica. 8a edição. **Saraiva Educação SA**, São Paulo, 2017.

NIELSEN, M. A. Neural networks and deep learning. San Francisco, CA, USA: **Determination press**, 2015.

OLAH, C. **Understanding lstm networks**. 2015. Disponível em: <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Acesso: 09 de junho de 2020.

OLIPHANT, T. E. A guide to NumPy. USA: **Trelgol Publishing**, 2006.

PEEL, L. Data driven prognostics using a Kalman filter ensemble of neural network models. **2008 International Conference on Prognostics and Health Management**. IEEE, 2008. p. 1-6.

SAXENA, Abhinav et al. Damage propagation modeling for aircraft engine run-to-failure simulation. **2008 international conference on prognostics and health management**. IEEE, 2008. p. 1-9.

SAXENA, Abhinav; GOEBEL, Kai. Turbofan engine degradation simulation data set. **NASA Ames Prognostics Data**

Repository, 2008. Disponível em:
<<https://ti.arc.nasa.gov/tech/dash/groups/pecoe/prognostic-data-repository>>. Acesso em: 21 de setembro de 2020.

SWINSCOW, Thomas Douglas Victor et al. **Statistics at square one**. London: Bmj, 2002.

VERSTRAETE, D.; DROGUETT, E.; MODARRES, M. A. Deep Adversarial Approach Based on Multi-Sensor Fusion for Semi-Supervised Remaining Useful Life Prognostics. **Sensors**, 2020, 20, 176.

ZEILER, M. D. Adadelta: an adaptive learning rate method. **arXiv preprint arXiv:1212.5701**, 2012.

ZHANG, C. et al. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. **IEEE transactions on neural networks and learning systems**, v. 28, n. 10, p. 2306-2318, 2016.