Remaining Useful Life Prediction for Experimental Filtration System: A Data Challenge

Kürşat İnce[1,3], Engin Sirkeci[2,3], and Yakup Genç[3]

[1] *Naval Combat Management Technologies Center, HAVELSAN Inc., Pendik/İstanbul, Türkiye*
kince@havelsan.com.tr

[2] *Defense Systems Technologies Vice Presidency, ASELSAN Inc., Yenimahalle/Ankara, Türkiye*
esirkeci@aselsan.com.tr

[3] *Computer Engineering Department, Gebze Technical University, Gebze/Kocaeli, Türkiye*
{*kince,esirkeci,yakup.genc*}*@gtu.edu.tr*

ABSTRACT

Maintenance costs of industrial systems often exceed the initial investment cost. Predictive maintenance, which analyzes the health of the system and suggests maintenance planning, is one of the strategies implemented to reduce maintenance costs. Health status and life estimation of the machinery are the most researched topics in this context. In this paper, we present our analysis for Fifth European Conference of the Prognostics and Health Management Society 2020 Data Challenge, which introduces an experimental filtration system for different experiment setups, and asks for remaining useful life predictions. We compared random forest, gradient boosting, and Gaussian process regression algorithms to predict the useful life of the experimental system. With the help of a new fault-based piecewise linear RUL assignment strategy, our gradient boosting based solution has been ranked 3rd in the data challenge.

1. INTRODUCTION

Maintenance costs of industrial systems often exceed the initial investment cost. Technical advances in Industry 4.0 enables us to predict future state/behavior of the industrial system by collecting and analyzing operational data, and to decide accordingly. With that respect, predictive maintenance, which analyzes the health of the system and suggests maintenance planning, is one of the strategies implemented to reduce maintenance costs. Since access to data is easier than before, health status and life estimation of the machinery are the most researched topics in this context (Tsui, Chen, Zhou, Hai, & Wang, 2015). In view of the high impact and extreme costs usually associated with maintenance tasks, studies have been carried out to predict failures and reduce overall effects. However, it is necessary to determine maintenance decisions at the right time with a prognostics information. The vast majority of these studies focuses on the estimating the remaining useful life (Nguyen & Medjaher, 2019).

By enabling widespread integration of diagnostics and prognostics into modern production systems, uncertainties associated with life cycle of system have reduced. Prognostics and health management (PHM) is performed with varying degrees of success for a number of different reasons. There are currently no standards to demonstrate best practices comparatively because each problem can be solved in a variety of ways. The PHM Data Challenge, an open data competition specialized in PHM, is an opportunity to competitively determine leading solutions for industrial problems. PHM Data Challenge, an open data competition specialized in PHM, includes diverse issues in industrial data analytics and thus provides abundant resource for study and appropriate approach development. The data in the competitions cover a wide spectrum of real-world industrial problems. The proposed issues and winning algorithms each year serve as a diverse library of case studies from which we can learn about the current challenges in practice, the thinking flow of addressing these challenges, and the advantages and disadvantages of different methods (Huang, Di, Jin, & Lee, 2017).

In this paper, we present our analysis for Fifth European Conference of the Prognostics and Health Management Society

2020 Data Challenge (PHME20 Challenge) (Giordano & Gagar, 2020). The challenge proposes degradation of an experimental filtration system for different experiment setups, and asks for useful life predictions. Through a data pipeline, we compared random forest (RF), gradient boosting (GB), and Gaussian process (GP) regression algorithms to estimate the useful life of the experimental system.

This analysis paper is organized as follows: Section 2 describes remaining useful life of a system, and gives information about the challenge dataset and other public datasets which we considered relevant to this challenge. Section 3 describes how we attacked the Data Challenge, and the workflow we used for analysis. Section 4 describes our implementation and results that we had for the Data Challenge.

2. REMAINING USEFUL LIFE PREDICTION AND PHME20 DATA CHALLENGE

The remaining useful life (RUL) of a system is defined as the time left from the current time to the end of the systems's useful life. The major task of RUL prediction is to forecast the time left before the system losses its operation ability, based on the condition monitoring information. There are two major issues/research areas related to the remaining useful life prediction: predicting the remaining useful life based on the condition monitoring information and measuring the prediction accuracy of different approaches (Lei et al., 2018). PHME20 data challenge competitors are challenged to showcase their abilities on an experimental filtration system's condition monitoring data.

Filtration systems are used in several engineering processes including automotive, chemical, nuclear reactor, and process engineering applications. Besides, several industrial applications such as food, petroleum, pharmaceuticals, metal production, and minerals embrace filtration process
(Sparks, 2012). The aim of the filtration systems is to keep the rest of the system running smoothly, thus they play a vital role in maintaining the process operating (Skaf, Eker, & Jennions, 2015).

Filtration systems are subject to clogging because of the contamination in the liquids. PHME20 dataset explores this fact through an experimental rig that has been constructed to simulate filter clogging failure with different contamination degrees. The rig contains liquid tanks, stirrer, pump, pulsation dampener, filter, pressure and flow rate sensors, and data acquisition system, as shown if Figure 1. The suspension contains polyetheretherketone particles and water in different concentrations. Particles have three possible sizes: small $(45-53\mu m)$, medium $(53-63\mu m)$, and large $(63-75\mu m)$. The dataset contains sensor readings of the experiments for different particle sizes and concentrations, acquired at 10 Hz. The filtration system is said to be clogged when the pressure drop ($Upstream\_Pressure - Downstream\_Pressure$) is higher than $20psi$.

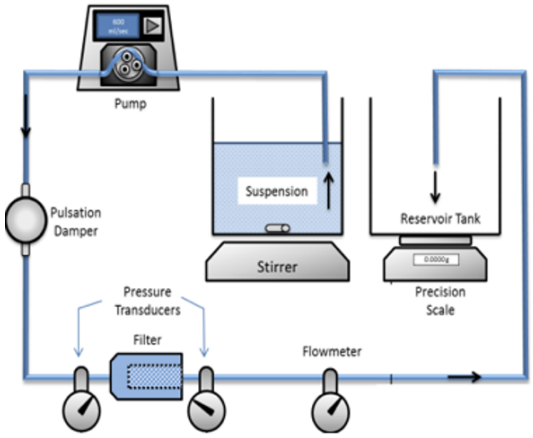

Figure 1. The experimental rig for the PHME20 Data Challenge

Data Challenge's public dataset contains experiments divided into training and validation subsets. A private test subset is allocated for evaluations of the challenge submissions. Concentration is given by $Solid\ Ratio$, and particle size is given by $Particle\ Size$ in experiment metadata. A $Profile$ is assigned to each combination of concentration and particle size in the dataset. We consider these metadata as operating conditions of the experiments. Training and validation subsets contain 32 experiments in total. The aim of the PHME20 Data Challenge is to predict RUL for training, validation and test datasets for each 10 seconds intervals.

Similar to the current data challenge, PHM08 Data Challenge (Saxena & Goebel, 2008) also aims for RUL prediction. It's sibling, NASA Turbofan Engine Degradation Simulation Dataset (CMAPSS dataset) (Saxena & Simon, 2008) is one of the most analyzed dataset in predictive maintenance
researchers. Both PHM08 and CMAPSS dataset are composed of multiple run-to-failure data of turbofan engines simulated using CMAPSS simulation software. We have analyzed CMAPSS dataset using various algorithm, including deep learning architectures, gradient boosting and Gaussian process, and found out that Gaussian process with piecewise linear rule assignment produced better results. We find various similarities between CMAPSS dataset and
PHME20 Data Challenge, and that is what we want to explore for more.

Another well-known dataset for RUL estimation is a battery dataset, (Saha & Goebel, 2007). Battery capacity estimation, and accurate prognostics is an important component of a battery management system. Capacity of lithium-ion batteries are affected not only by the cycle count but also by the environmental conditions the battery is operating. Richardson et al. in (Richardson, Osborne, & Howey, 2017) use Gaus-

sian process regression to predict cycle capacities, and remaining useful life for lithium-ion batteries. Richardson et al. shows that Gaussian process proves useful when there is limited amount of data samples that are coming from dynamic environmental conditions, such as a battery pack.

3. METHODS AND TECHNIQUES

We have built a data processing workflow using Jupyter Notebook (Kluyver et al., 2016) to attack the Data Challenge. The workflow contains data preprocessing, model training, and evaluation phases as shown in Figure 2.

3.1. Data Preprocessing

We have executed preprocessing steps such as initial RUL assignment, operating conditions (profile) assignment, scaling etc. before the training phase.

Data Merging: The dataset contains individual data files for each experiment. We merged individual data files to have two separate datasets for training and validation. We have added experiment setups for each sample of reading for further processing. The merged training and validation subsets have $ExperimentID$, $ReadingID$, $Time$, three experiment setups, readings from three sensors, and $Pressure\_Drop$.

RUL Assignment: RUL assignment is the process of assigning RUL labels to the samples in the dataset. The datasets does not contain any ground truth RULs for the experiments. Since we knew that the experiments are run-to-failure, we could assign RULs for each sample.

The most common RUL assignment strategies are Linear RUL Assigment, and Piecewise RUL Assignment. Linear RUL Assignment suggests that the maximum of the RUL values is the length of the sequence of the sensor readings. RUL value drops by one with every reading, and finally reaches to zero at the end. The sensor readings does not end with clogging failure, so we have assigned negative RULs also. Piecewise Linear RUL Assignment (PwL) suggests that RUL is constant until a degradation in the experiment, and that the RUL starts to decrease with the fault (Heimes, 2008). We experimented with initial RUL values of 100, 125, and 150.

For the challenge, we have also developed a new RUL assignment strategy. In this strategy we sought for the faulty timestamp using the following heuristic: we have observed that $Flow\_Rate$ is constant with some added noise, so we tried to capture the timestamp where this behavior changes. Using the $Flow\_Rate$ values between timestamps 400 and 1400, we estimated a line for this nominal behavior, and calculated the intersection of the $Flow\_Rate$ and the line at the furthest timestamp, which is assumed to be the start of the degradation. This is demonstrated in Figure 3. We used linear RUL model before (initial RUL to fault) and after (fault to experiment end) that intersection point. We named this assignment strategy as RUL_Fault. A visualization RUL assignment strategies is shown in Figure 4.

Operational Conditions Assignment: The training and validation subsets has $Profile$ to describe the experiment setups. Since these subsets have disjoint profiles, we needed to build our schema for operation condition assignment. We used K-Means clustering algorithm with $Solid\ Ratio$ and $Particle\ Size$ features to assign operational conditions ($Kmeans\_Profile$) for each sample.

Scaling: Data normalization is performed on each $KMeans\_Profile$ using StandardScaler from Scikit-Learn Library (Pedregosa et al., 2011). Initially training data is scaled, and then the scaling parameters are applied to validation data.

Feature Selection: We used the following features throughout out experiments: $Flow\_Rate$, $Upstream\_Pressure$, $Downstream\_Pressure$, $Pressure\_Drop$, $Particle\ Size$, $Solid\ Ratio$, and $Kmeans\_Profile$.

Resampling: The runtime performance of GP is $O(n^3)$, where n is the number of samples in the dataset. In order to achieve reasonable training times, we resampled the 10 Hz input data to d Hz, where $d \in \{1, 0.5, 0.33, 0.25, 0.2\}$, using $5/d$ strides. Initial results showed that $d = 0.2$ Hz gives reasonable training times. Resampling is used for GP only.

Windowing: We have used window sizes of 5, 10, 15, 20, 25, 30, 40, and 50 to create a context on the time series data. Windowed dataset have passed to model training phase.

3.2. Data Modeling

We have used RF, GB, and GP regression algorithms to estimate the useful life of the experimental filtration system.

Random Forest: RF is an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time. RF is a nonlinear modeling tool and overcomes low accuracy of single decision-tree and overfitting (Breiman, 2001). Tuning the hyperparameters can often increase generalization performance. Depending on the implementation tree size can be controlled using hyperparameters such as maximum depth, maximum number of nodes, and minimum number of points per leaf node. RF method is very suitable for solving failure problems when priori knowledge is unclear, there is incomplete data. We have used RF implementation in Scikit-Learn library (Pedregosa et al., 2011).

Gradient Boosting: GB, one of the most powerful techniques for performing classification and regression tasks, builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function (Friedman, 2001). GB is an ensemble learner: a final model based on a collection of indi-

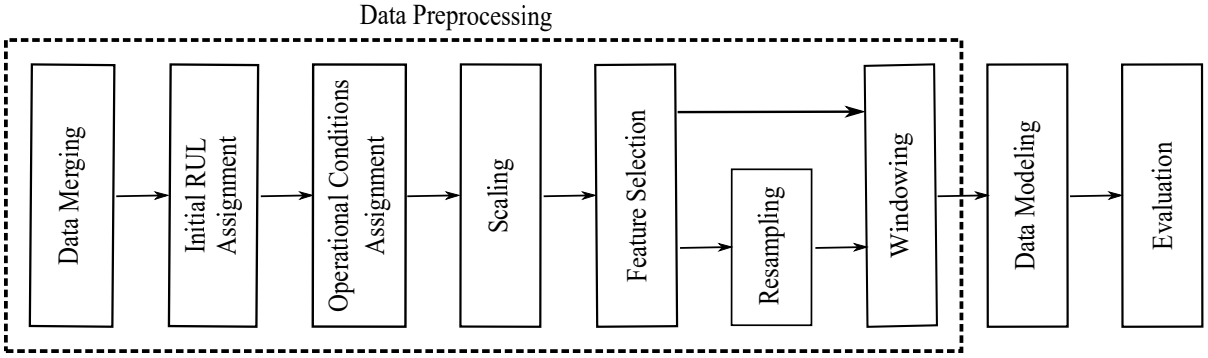

Figure 2. Data Processing Workflow

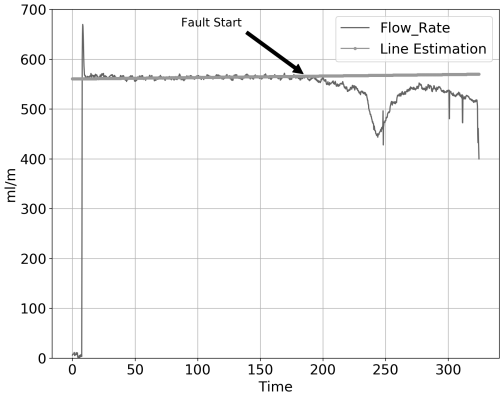

Figure 3. Fault Estimation for PwL_Fault RUL Assignment

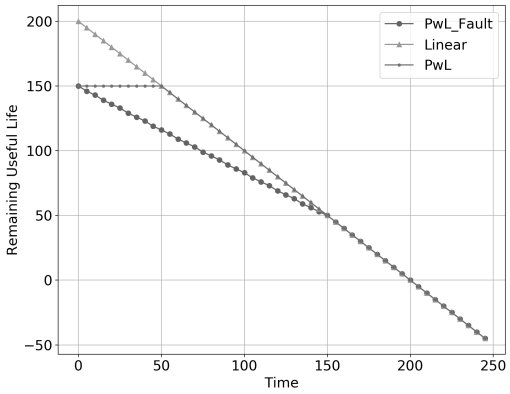

Figure 4. Linear, PwL, and PwL_Fault RUL Assignment Strategies

vidual models. The predictive power of these individual models is weak and prone to over-fitting but combining many such weak models in an ensemble will lead to an overall much improved result. We have used CatBoost (CB) (Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2017) implementation in our research.

Gaussian Process: A GP is a probability distribution over possible functions (Rasmussen & Williams, 2005). GPs are a generic supervised learning method designed to solve regression and probabilistic classification problems. Their greatest practical advantage is that they can give a reliable estimate of their own uncertainty. GP extend multivariate Gaussian distribution to infinite dimensionality. The key idea of GP is to model the underlying distribution training data as a multivariate normal distribution. Learning a distribution enables the model to output a prediction and an uncertainty associated with the prediction. We have used GP implementation in Scikit-Learn library (Pedregosa et al., 2011).

3.3. Evaluation

We have used and reported mean absolute error (MAE) in our experiments. The submissions to the data challenge have been evaluated using the challenge specific penalty scores:

$$\begin{aligned} Penalty(TV) &= \sum MAE(M_i(TV)) \\ Penalty(TE) &= \sum MAE(M_i(TE)) \\ PenaltyScore &= 1.5 \times Penalty(TE) + Penalty(TV) \end{aligned} \tag{1}$$

where

- M_i = Model generated with $i\%$ of the training data, $i \in \{25, 50, 75, 100\}$
- TV = Training + Validation datasets
- TE = Test dataset

Table 1. Model Parameters

Algorithm	Model Parameters
RF	max_features: 'auto', n_estimators: 100, bootstrap: True, max_depth: 10, min_samples_leaf: 2, min_samples_split:2
CB	learning_rate: 0.03, iterations: 1000
GP	Kernel: Rational Quadratic length_scale: 1.0, alpha: 0.1

Table 2. MAE Scores Using 25% of the Training Data (Best result is given in bold.)

		Window Size				
		10	15	20	30	50
RF	Linear	8.65	7.79	8.43	8.39	8.72
	PwL	5.98	6.46	6.07	6.06	6.48
	PwL_Fault	4.89	4.99	5.08	4.71	4.76
CB	Linear	8.27	7.82	7.17	7.01	8.14
	PwL	5.12	4.92	4.92	4.80	5.08
	PwL_Fault	4.63	4.05	3.96	3.76	3.79
GP	Linear	12.42	12.20	14.18	13.73	7.11
	PwL	11.01	9.69	9.38	5.88	**2.19**
	PwL_Fault	7.14	6.56	7.25	6.98	3.75

4. EXPERIMENTS AND RESULTS

4.1. Model Training

After data preprocessing we trained our data models with the transformed data. We used model parameters that are given in Table 1. These values for RF were obtained through a grid search procedure while the rest were default values provided by the implementations.

4.2. Results and Discussion

PHME20 Data Challenge asks four models depending on the percentage of the data that is used for training purposes. We have used training and validations subsets separately, so for our analysis we have used the required percentage of the training experiments only. We have reported MAE scores for each case. During the experimentation we have seen that MAE score drops when initial RUL for PwL drops. So we make a decision and only reported PwL for initial RUL 150. Results are shown in Tables 2-5.

Interestingly, when there was more training data, CatBoost learned at smaller window sizes, i.e. best scores were achieved for window size 15 rather than 30 as in previous experiments. On the other hand GP achieved best results for window size 50 in all cases. Although GP achieve better, we were unable to submit GP models for the challenge. The data challenge strictly forbids submissions that are greater than 6 MBs. Size of the RF model increases with number of trees in the ensemble, the depth of these trees. Our experiments showed

Table 3. MAE Scores Using 50% of the Training Data

		Window Size				
		10	15	20	30	50
RF	Linear	7.81	7.07	7.97	8.10	8.12
	PwL	4.94	5.41	4.66	5.21	5.23
	PwL_Fault	4.21	4.37	4.36	4.30	4.30
CB	Linear	9.04	8.06	8.02	7.74	7.68
	PwL	4.64	4.63	4.61	4.47	4.54
	PwL_Fault	4.53	4.31	3.87	3.99	3.95
GP	Linear	10.75	9.71	9.28	8.94	6.08
	PwL	7.87	6.95	5.91	3.64	**1.66**
	PwL_Fault	5.77	5.54	5.14	4.29	2.96

Table 4. MAE Scores Using 75% of the Training Data

		Window Size				
		10	15	20	30	50
RF	Linear	6.97	6.94	8.59	7.81	6.95
	PwL	4.56	4.63	6.16	5.20	4.86
	PwL_Fault	4.25	4.19	5.53	4.64	3.98
CB	Linear	8.72	7.61	9.31	8.63	8.79
	PwL	4.87	4.62	6.12	4.95	5.66
	PwL_Fault	4.41	4.45	5.48	4.74	4.56
GP	Linear	9.39	8.58	8.38	7.58	5.17
	PwL	6.61	5.35	4.68	2.76	**1.02**
	PwL_Fault	4.35	4.13	3.89	3.23	2.18

that four RF models are as big as 28 MBs. MAE scores for smaller models were upto 3x worse than that is reported here. Scikit-Learn implementation of GP stores training samples and covariance matrix in the model. So the size of the GP models increases with number of samples in the dataset. Although we used resampling of the dataset, four GP models are 14.3 MBs in total. CB models are about 1 MBs each, so we were able to submit our CB models.

Submissions to the data challenge are evaluated using public training and validation datasets, and a private test dataset. Our CB models, which ranked 3rd in the data challenge, scored 86.74 using the challenge's penalty score.

5. CONCLUSION

We presented our analysis for PHME20 Data Challenge, which asks for prediction of RUL of an experimental filtration system for different experiment setups. We built a data pipeline, and compared random forest (RF), gradient boosting (CatBoost), and Gaussian process regression (GP) algorithms.

Gaussian process regressor predicted better than other algorithms in all experiments. A new heuristic-based RUL assignment strategy (PwL_Fault) is introduced. PwL_Fault achieves better than Linear RUL assignment (Linear) in all cases.

Table 5. MAE Scores Using all the Training Data

		Window Size				
		10	15	20	30	50
RF	Linear	7.45	7.12	8.07	7.52	7.38
	PwL	4.57	4.72	5.43	5.29	4.85
	PwL_Fault	4.52	4.22	4.88	4.39	4.20
CB	Linear	9.18	8.37	9.01	8.51	8.90
	PwL	4.79	4.62	5.55	4.69	5.53
	PwL_Fault	4.61	4.36	5.26	4.62	4.34
GP	Linear	9.32	9.07	8.56	8.00	5.77
	PwL	6.05	5.62	4.70	2.79	**1.00**
	PwL_Fault	4.30	3.86	3.72	3.12	2.10

PwL_Fault performs better then piecewise linear version (PwL) for RF and CatBoost, but not for GP. CatBoost results shows that while training data size increases the optimal windows size decreases. While performing the best, GP does not yield the smallest model which prevented us from submitting our best model but the model using CatBoost which carried us to the 3rd place in the challenge.

REFERENCES

Breiman, L. (2001, October). Random forests. *Machine Learning*, *45*(1), 5–32. doi: 10.1023/A:1010933404324

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Giordano, D., & Gagar, D. (2020). *Fifth european conference of the prognostics and health management society 2020 data challenge.* (http://phmeurope.org/2020/data-challenge-2020)

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management, phm 2008.* doi: 10.1109/PHM.2008.4711422

Huang, B., Di, Y., Jin, C., & Lee, J. (2017, 05). Review of data-driven prognostics and health management techniques: Lessions learned from phm data challenge competitions..

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., ... Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (p. 87 - 90). doi: 10.3233/978-1-61499-649-1-87

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834. doi: 10.1016/j.ymssp.2017.11.016

Nguyen, K. T., & Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering and System Safety*, *188*, 251 - 262. doi: 10.1016/j.ress.2019.03.018

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2017). *Catboost: unbiased boosting with categorical features.*

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning (adaptive computation and machine learning)*. The MIT Press.

Richardson, R. R., Osborne, M. A., & Howey, D. A. (2017). Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, *357*, 209–219. doi: 10.1016/j.jpowsour.2017.05.004

Saha, B., & Goebel, K. (2007). *Battery Data Set.* (https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/)

Saxena, A., & Goebel, K. (2008). *Phm08 challenge data set.* (http://ti.arc.nasa.gov/project/prognostic-data-repository)

Saxena, A., & Simon, D. (2008). *Turbofan engine degradation simulation data set.* (http://ti.arc.nasa.gov/project/prognostic-data-repository)

Skaf, Z., Eker, O. F., & Jennions, I. K. (2015). A simple state-based prognostic model for filter clogging. *Procedia CIRP*, *38*, 177 - 182. (Proceedings of the 4th International Conference on Through-life Engineering Services) doi: 10.1016/j.procir.2015.08.094

Sparks, T. (2012). *Solid-liquid filtration: A user's guide to minimizing cost and environmental impact, maximizing quality and productivity*. Elsevier Science.

Tsui, K.-L., Chen, N., Zhou, Q., Hai, Y., & Wang, W. (2015, 05). Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, *2015*, 1-17. doi: 10.1155/2015/793161