Proceedings of the International Conference on Industrial Engineering and Operations Management
Toronto, Canada, October 23-25, 2019

Aircraft Engine Remaining Useful Life Prediction Framework for Industry 4.0

Hussein A. Taha
Department of Mathematics and Industrial Engineering
Polytechnique Montréal
Montréal, H3T 1J4, Canada
Hussein.Taha@polymtl.ca

Ahmed H. Sakr
Department of Mathematics and Industrial Engineering
Polytechnique Montréal
Montréal, H3T 1J4, Canada
Ahmed-1.Sakr@polymtl.ca

Soumaya Yacout
Department of Mathematics and Industrial Engineering
Polytechnique Montréal
Montréal, H3T 1J4, Canada
Soumaya.Yacout@polymtl.ca

Abstract

This article proposes a Condition-Based Maintenance (CBM) approach for aircraft engines and Remaining Useful Life (RUL) monitoring, and failure prevention. Due to the unavailability of run-to-failure data, Turbofan Engine Simulation data, obtained from NASA repository, is used to train and test our model. Data Acquisition and Management system framework and planning are proposed for online monitoring and RUL prediction. In practice, sensor measurements usually suffer from noise contamination, hence the prediction models are challenged by noise contaminated data for both training and testing tasks. This is done to assess their prediction ability in a similar condition of having noisy data. Linear and nonlinear prediction models are developed, with performance comparison addressing both regression and classification problems. Models performance indices consider both prediction accuracy and percentage of predictions before the actual failure (PBAF). The proposed model considers continuous learning and improvement to account for any further operational changes that affect the model prediction ability. This is reached by ingesting the model with the actual RUL during the maintenance of the engine unit, and by comparing it to the predicted one.

Keywords Condition-based maintenance, Failure prediction, Engine Degradation, IoT, Industry 4.0.

1. Introduction

Aircraft engine is a critical component. Its failure causes loss of lives. The traditional maintenance strategies, that are proposed by the designers, usually involve Reliability Centered Maintenance (RCM). These strategies propose preventive maintenance tasks that are based on reliability analysis of the operating systems. These strategies improve effectively the reliability of the engine. However, the costs are high due to unnecessary maintenance or replacement actions. Condition-Based Maintenance (CBM) is used for cost minimization while achieving reliability improvement. Online monitoring and data analysis lead to better maintenance planning and maintenance duration reduction. In addition to performing effective maintenance plans, airlines can achieve better consistency of flight scheduling.

CBM is a condition monitoring concept which is used to decide when the operating asset requires maintenance (Jardine et al. 2006) (Koenig et al. 2006). This provides a proactive scheduling for the maintenance process. The CBM strategy begins with data acquisition from sensors' readings, which are analyzed to extract useful information about the system's state (Jardine et al. 2006). The performance of CBM is challenged by data cleanness and prediction models' accuracy (Byington et al. 2002) (James et al. 2013). Normally, an engine condition should trigger maintenance actions within enough time before failure. Consequently, efficient models that accurately predict the RUL are required while overcoming the noise contamination problems (Saxena et al. 2008). Researchers proposed supervised learning prediction models for aircraft engine degradation (Byington et al. 2002) (Liu et al. 2015) (Lu et al. 2017, 2019) (Ragab et al. 2016) (Yan 2006) (Yuan et al. 2016) (Zhao et al. 2017). However, their models do not consider continuous learning, hence there is no possibility for accuracy improvement or considering any new events that the model was not trained for. In practice, industrial operations usually have operational modifications that require continuous monitoring to avoid inaccurate predictions (Zhang et al. 2017). The online monitoring of operating assets has become possible through Internet of Things (IoT) technologies adopted by the Industry 4.0 paradigm. These give a chance for sensors to transmit the captured engine data to a cloud database during operation (Zhang et al. 2017). The cloud storage of the data facilitates the engine monitoring even if the aircraft is in the air. Hence, maintenance scheduling is achieved, and flight rescheduling is planned to avoid conflicts. Our proposed framework consists of:

- Data transfer and cloud storage platform
- RUL prediction model

In this paper, the data, that is used for training and testing of the prediction model, is obtained from NASA Prognostics Data - Turbofan Engine Degradation Simulation results (Saxena et al. 2008). Simulation is used due to the difficulty of having run-to-failure real data for these engines.

This article is organized as follows: System planning, and framework layout are presented in section 2. Data prepossessing and overview of the prediction models are given in section3. Section 4 discusses the obtained results. Finally, section 5 presents our conclusion and future works.

2. System planning and framework layout

The data represents simulation results for 100 engine units. It is provided by a text file of 26 columns and indexed into units, cycle time, three types of operational settings, and 21 sensors' measurements. Each row is a snapshot of the data that is taken during a single operational cycle. Table 1 shows detailed description of sensors' measurements. The actual RUL for an operational cycle is the difference between the unit's total life until failure, and the current cycle's number. It is calculated for the training and the testing data as the failure cycle for each unit is given by its simulation results.

The main objective for a condition-based maintenance strategy is to predict the number of remaining operational cycles before failure, i.e the number of operational cycles after the current cycle, during which the engine will continue to operate. However, this prediction task is challenged by data contamination due to sensor noise. The measurements types are summarized as follows:

- Temperature measurement
- Pressure measurement
- RPM measurement
- Air Mass flow measurement

Table 1. Descriptions of sensor signals (Liu et al. 2015)

Index	Predictor name	Unit
1	Total temperature at fan inlet	K°
2	Total temperature at LPC outlet	K°
3	Total temperature at HPC outlet	K°
4	Total temperature at LPT outlet	K°
5	Pressure at fan inlet	psia
6	Total pressure in bypass-duct	psia
7	Total pressure HPC outlet	psia
8	Physical fan speed	rpm
9	Physical core speed	rpm
10	Engine pressure ratio	-
11	Static pressure at HPC outlet	psia
12	Ratio of fuel flow to "16"	pps/psi
13	Corrected fan speed	rpm
14	Corrected core speed	rpm
15	Bypass ratio	–
16	Burner fuel-air ratio	–
17	Bleed Bleed enthalpy	–
18	Demanded fan speed	rpm
19	Demanded core fan speed	rpm
20	HPT coolant bleed	lbm/s
21	LPT coolant bleed	lbm/s

A data acquisition system is needed for transfer and storage of the sensors' measurements. Aircrafts have data acquisition system with aviation Arinc429 standard (Balmus 2016). It is used to transfer data such as air data, radar altimeter data, and GPS data. The measurements are used for engine operational control (Imani and Montazeri-Gh 2019). Our proposed system layout includes sensors' measurements data transfer to an onboard server as shown in Figure 1. The server is selected with internet/cloud connecting feature; thus, it facilitates the engine remote monitoring and RUL prediction, even when the aircraft is in operation.

Figure 1. Proposed system layout

3. Methodology

The methodology that is applied for model training and testing is performed using Scikit-learn library for machine learning on Python 3.7. Python is an open-source general-purpose programming language. The Scikit-learn is a free machine learning library that features various classification and regression algorithms. The Python code loads the input data from CSV file. The CSV file is developed from the raw text file using MS Excel.

3.1 Data prepossessing

The preprocessing of the data is an important step before training machine learning models. Some problems within the data, such as correlated predictors, presence of outliers, missing data instances, cannot be handled well by some machine learning techniques and may affect their prediction capabilities. Hence, it is advisable to preprocess the data to improve the performance of the models. The preprocessing applied here includes the following:

- Outliers detection and removal
- Removing highly correlated predictors

The outliers are detected by Box plot. The data instances that have a Z-score higher than 3 are considered outliers and are removed. Figure 2 depicts sensor 7 data as an example for outlier removal. The data instance that is red colored has a Z-score greater than 3. This instance is removed from the input data. The same procedure is applied for the other predictors.

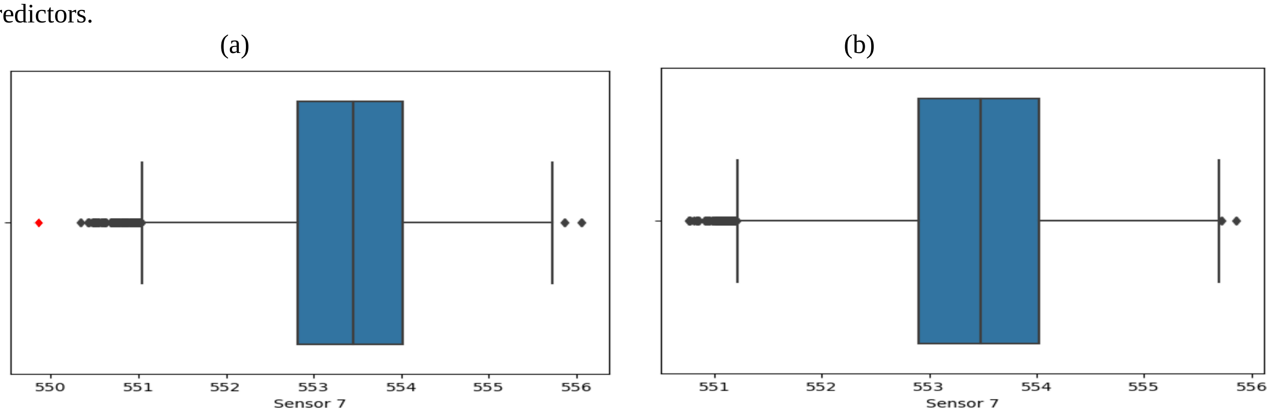

Figure 2. Box Plot for sensor 7 before (a) and after (b) removing the outliers

Figure 3 shows the correlation matrix for the predictors. The matrix represents the coefficient of correlation between each of the predictors and the others. This coefficient ranges from -1 to 1. The sign defines the type of proportionality between the predictors. The relationship is directly proportional for a positive coefficient of correlation, while is inversely proportional for a negative one. Large absolute value of the coefficient of correlation, greater than 0.95, shows high correlation. The values are color coded to aid visualization. Highly correlated predictors, Setting 3, Sensor 1, Sensor 5, Sensor 10, Sensor 16, Sensor 18, and Sensor 19, are removed from the input data before models' training as shown in Figure 4.

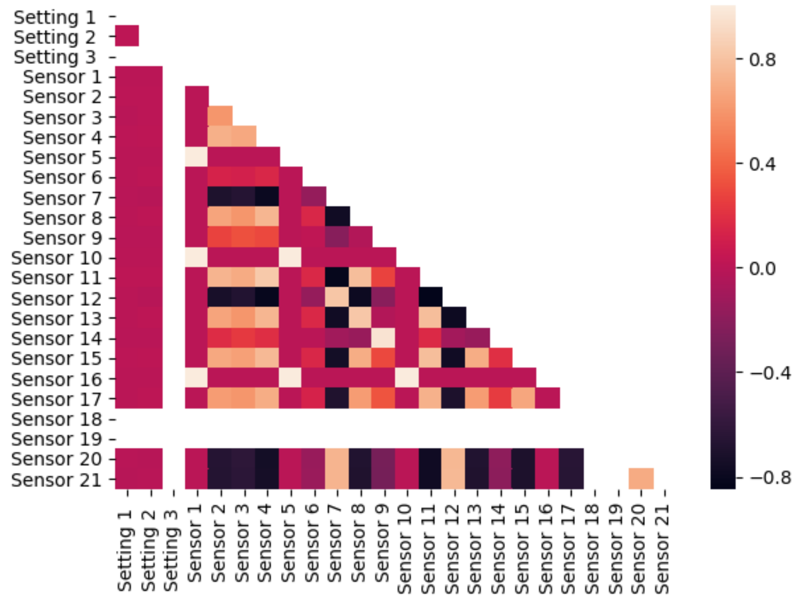

Figure 3. Correlation matrix for predictors

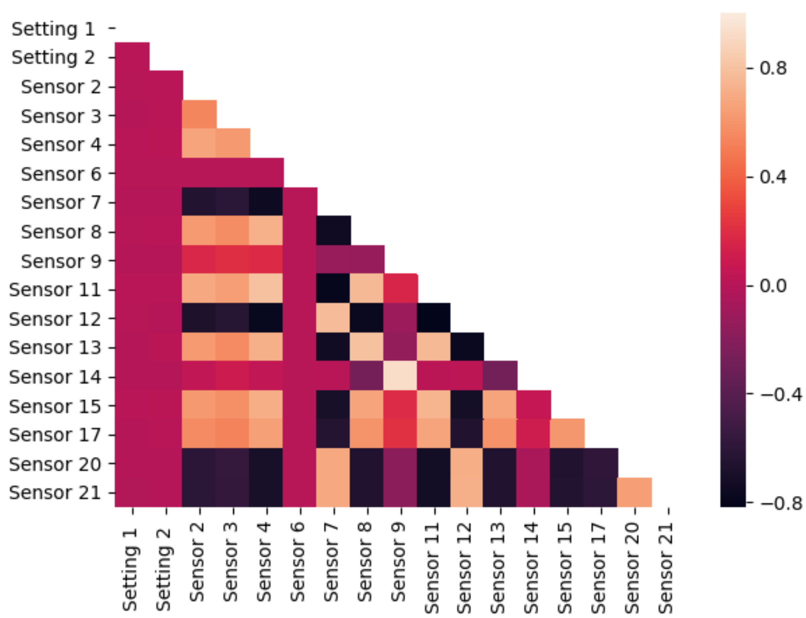

Figure 4. Correlation matrix for predictors after highly correlated predictors removal

3.2 Prediction Models

To predict the RUL, both linear and non-linear models are explored including parametric and non-parametric types. Different transformations for the output are tested in order to select the best form for RUL prediction. The best form is selected based on the prediction performance of the models. The performance is measured by the root mean square error (RMSE) for predictions using the testing data. For this data, the best form for the output is the inverse form, 1/RUL, for all the tested models. The input data is standardized to eliminate the effect of the predictors data units on the prediction models. The explored models include the following:

- Linear parametric:
  - Multiple linear regression
  - Ridge regression
  - Partial least square regression (PLS)
- Non-linear parametric:
  - Polynomial regression
- Non-linear Non-parametric:
  - K-nearest neighbors (KNN)
  - Random Forest (RF)
  - Neural Network (NN)

Equation (1) presents the multiple linear model where y_p is the predicted RUL value according to the transformation that is applied for the RUL of training data,X_j is the j^{th} predictor, P is the number of predictors which is 17 for the input data after removing highly correlated ones, and β_0, β_j are model parameters. The Ridge regression model is shown by Equation (2) where λ is the Ridge parameter. A value of 0.2 is selected for this parameter based on the best performance for prediction. The polynomial model, degree 2, is given by Equation (3). This degree is selected to avoid the overfitting problem that the polynomial model suffers from when the degree is high. The overfitting results in low training error, but high test error and poor prediction ability. This problem is named as the bias-variance trade-off in literature (James et al. 2013). The Ridge and the PLS models are explored for their ability to control and reduce the regression coefficients variance, hence improving the prediction performance. The Ridge model involves shrinking the coefficients towards zero, while the PLS considers dimensions reduction for the predictors (James et al. 2013)

The KNN regression model is given by Equation (4) where K is number of neighbors, and F_i is inverse of the distance between two neighbors. y_i is the RUL value, according to the applied transformation, for i^{th} nearest data point to the given X. The number of neighbors K is selected to be 5 according to the best prediction accuracy found. This avoids the overfitting problem as small K values are avoided. For the Random Forest model, the best parameters for prediction performance using the testing data are 100 trees with a depth of 20. The square root of predictor number is considered when looking for the best split. The Neural Network model consists of three hidden layers with sizes of 10, 8, 4, and the activation is rectified linear unit (ReLU). The size of hidden layers is selected to be between the size of the input layer and the output layer as recommended in (Karsoliya S. 2012). Model performance is assessed by *RMSE* which is shown by Equation (5) (James et al. 2013).

$$y_p = \beta_0 + \sum_{j=1}^{p} (\beta_j X_j) \tag{1}$$

$$y_p = \beta_0 + \sum_{j=1}^{p} (\beta_j X_j) + \lambda \sum_{j=1}^{p} \beta_j^{\,2} \tag{2}$$

$$y_p = \beta_0 + \sum_{j=1}^{p} (\beta_j X_j) + \sum_{j=1}^{p} (\beta_j X_j^{\,2}) \tag{3}$$

$$y_p = \frac{F_i}{\sum_{i=1}^{K} (F_i)} \sum_{i=1}^{K} (y_i) \tag{4}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_i (y_i - y_{pi})^2} \tag{5}$$

4. Results

The prediction of the RUL may have an error which results in a prediction of failure before the actual failure, PBAF, or after the actual failure, PAAF, as shown in Figure 5. Both cases are considered error from the point of view of RUL prediction. However, having a predicted life which is beyond the actual life is worse than having a prediction which is shorter than the actual life. For this, the performance of the models is measured not only based on the value of error, but also based on the PBAF%. The best-case scenario, in this case, is having the least possible value of error, along with the highest PBAF%.

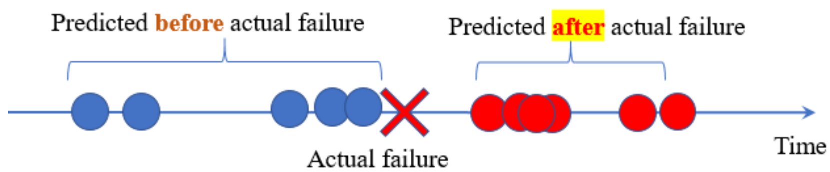

Figure 5. Positions of possible error in RUL within time axis

4.1 Regression Method

The regression is performed in this context to predict the value of the RUL. Figure 6 shows the performance measurement of the selected models based on RMSE and PBAF%. As shown in Figure 6 the Random Forest regressor is the most suitable over studied models with the lowest RMSE and high PBAF%. The Neural Network regressor gives the highest RMSE value among all models. The Neural Network yields the highest PBAF%, nearly 70%, while the Random Forest yields fewer PBAF%, up to 58%.

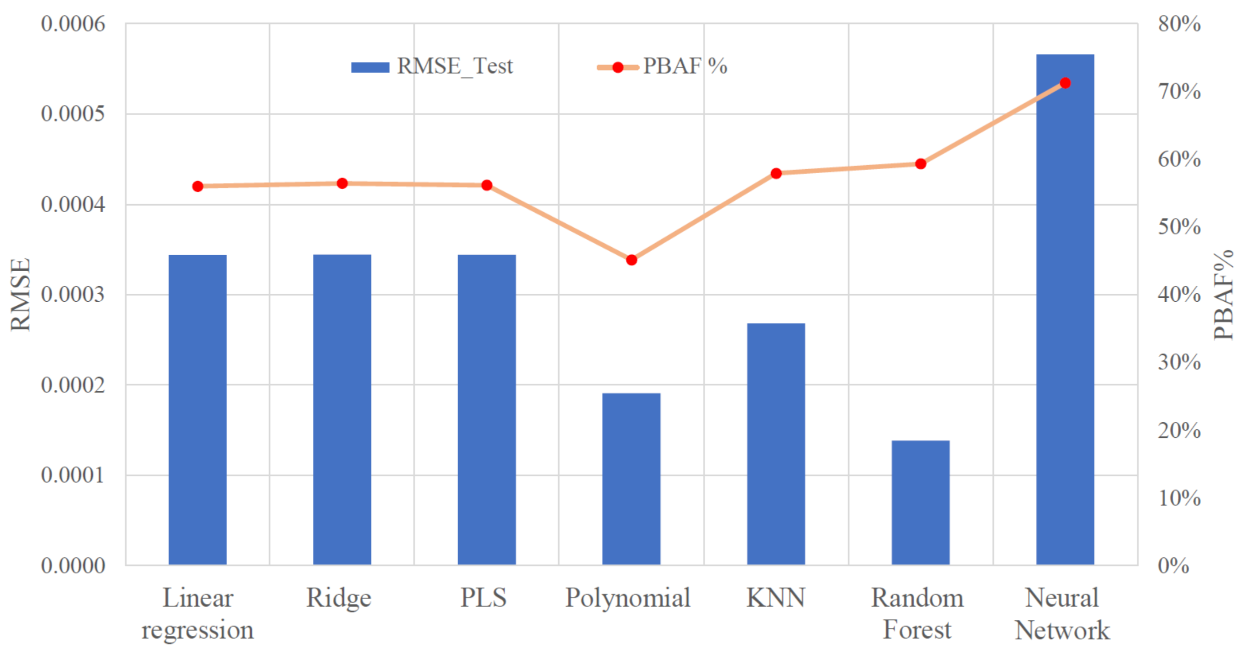

Figure 6. RMSE and PBAF% for the proposed prediction models

Figure 7 shows the relative importance of predictors based on the Random Forest model as the best model in this case. In Random Forest, the decrease of the Residual Sum of Squares (RSS) at each split is recorded. The predictor that has the highest value of RSS total reduction in all splits is the most important. The predictors importance gives better understanding for the most important engine readings that are affected by the RUL of the engine. The figure shows that Static pressure at HPC outlet, Sensor 11, is the most important predictor for RUL prediction. The Total pressure in bypass-duct, Sensor 6, has no importance as shown in the figure. The physical sensor measurement is no longer required; hence the amount of measurement data size are reduced accordingly.

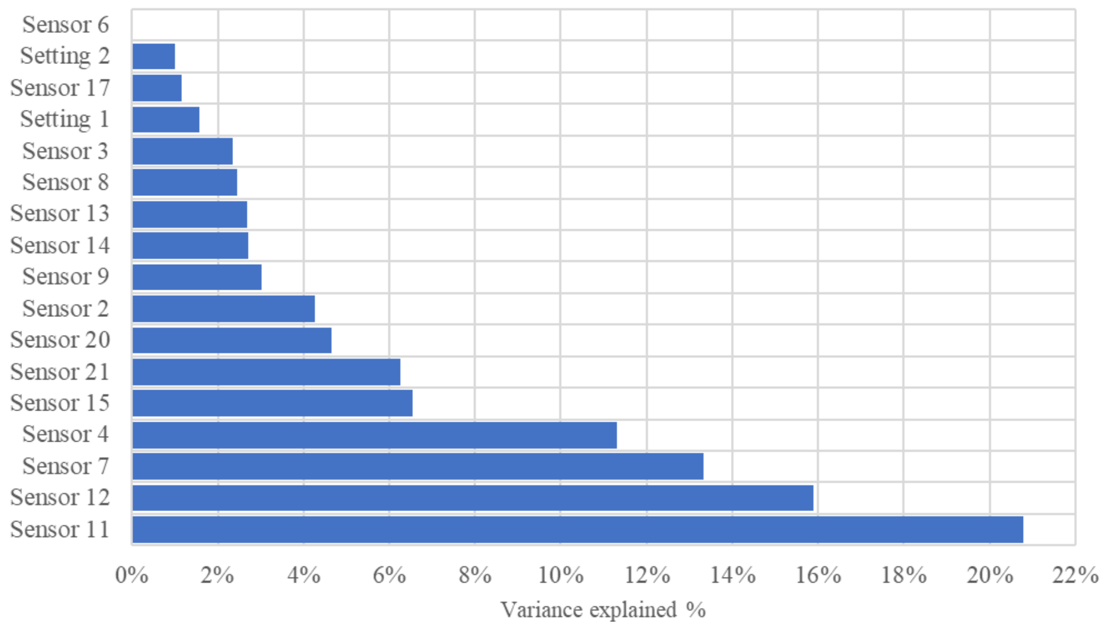

Figure 7. Predictors relative importance based on Random Forest model

4.2 Classification Method

Due to the less satisfactory results of the explored regression models, an alternative methodology is proposed which involves classification of 2 RUL classes instead of directly predicting its exact value. The RUL values are transformed into percentages for each engine unit, then a class is assigned for each data instance based on the RUL% value. This value is assigned according to the desired maintenance strategy. For demonstration, the RUL% value that differentiates the classes is arbitrarily selected to be 20%. The classes are as follows:

- Class 1: RUL is more than 20 %
- Class 2: RUL is less than 20 %

Three different classifiers are tested at different classification thresholds:

- Logistic regression
- KNN classifier (5 neighbors)
- Random Forest classifier (Depth=8, 50 trees, Max features → "Sqrt")

The number of neighbors in the KNN and both the depth and number of trees in the Random Forest are selected according to the best class prediction performance found.

Figure 8 shows both the error rate and the PAAF%. The error rate represents the proportion of the false classifications obtained for the test data. The Random Forest classifier gives the minimum error rate and PAAF%. Classification methods calculate the probability of selection for each class and perform the selection according to its classification threshold. This threshold affects the classification error rate and the percentage of false classification in each class. The false classification in a certain class is changed when the threshold is modified (James et al. 2013). The PAAF% is not acceptable error type, and it is reduced by decreasing the classification threshold as shown in Figure 9. Although the error rate has increased for all classifiers, the Random Forest shows a promising result as the PAAF% is successfully reduced to only 1.24% at 7.43% general error rate.

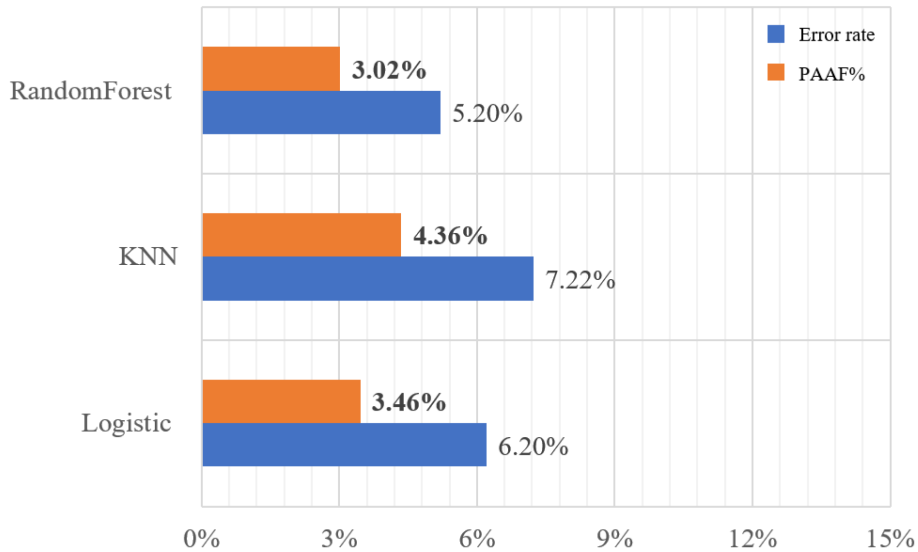

Figure 8. Error rate and PAAF% at default, 0.5, classification threshold

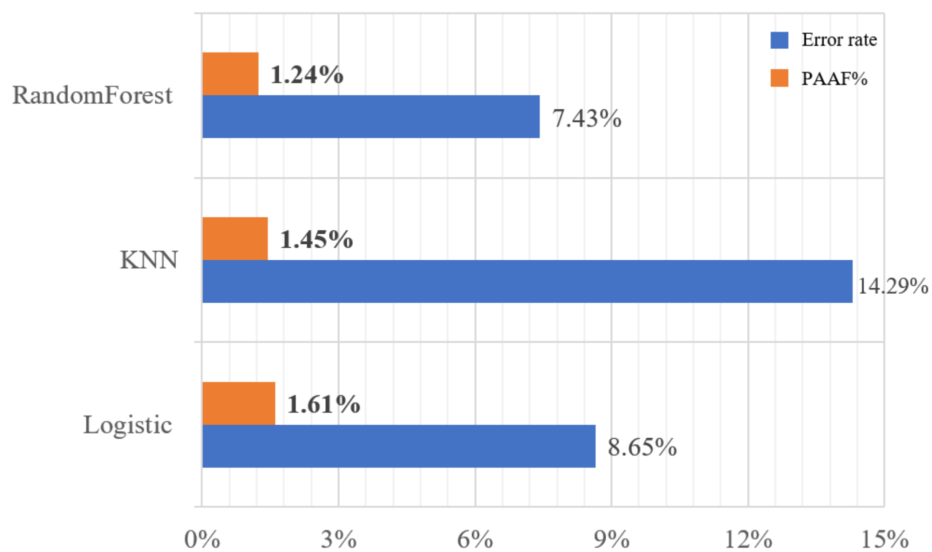

Figure 9. Error rate and PAAF% at 0.2 classification threshold

Figure 10 shows the execution time in seconds for different parts of the Python code that is used for the application of our methodology. These times are based on 2.5 GHz Core-i5 CPU with 8 Gb of RAM. The data loading and preprocessing is shown in green bar. The regression models are shown in blue bars. The classification models are shown in orange bars. The Random Forest classifier takes less execution time for training as compared to the Random Forest regressor. This is due to that the classifier has less depth and number of trees than the regressor.

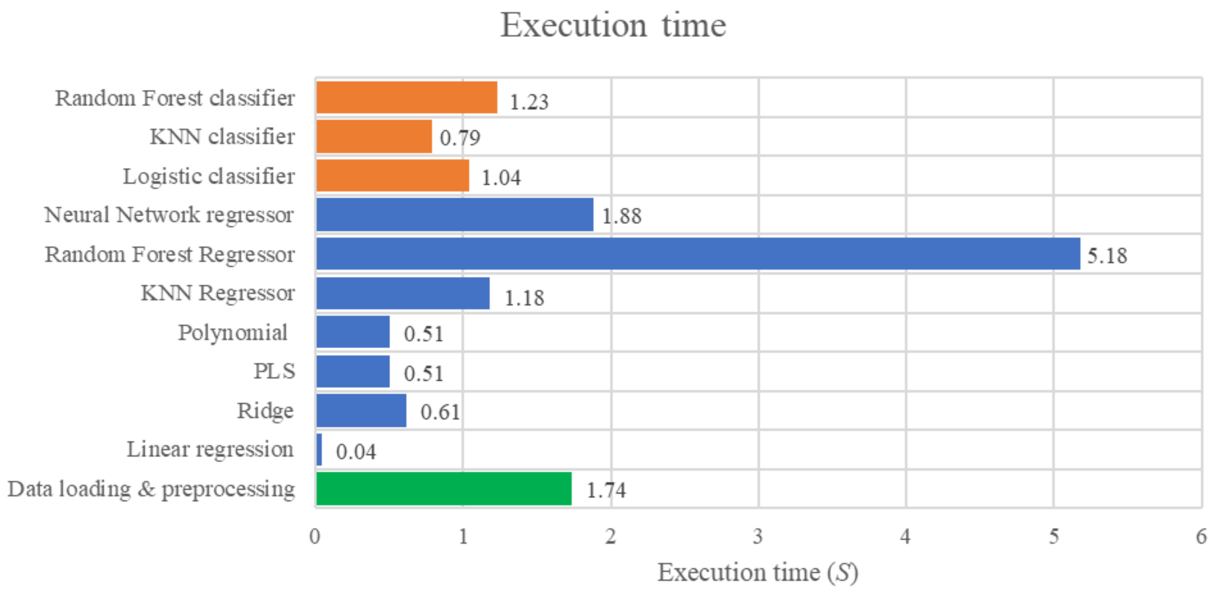

Figure 10. Execution times for different models using Python

4. Conclusion and Future work

This research proposed a framework for aircraft engine's RUL prediction. This framework included On-line remote monitoring and continuous learning with cloud connection facility. The RUL prediction model parameters are meant to be updated every maintenance operation, which helps improving the accuracy and the predicting capabilities of the model. The sensor noise problem was overcome by our model which affirms its robustness. This promotes its ability to provide reliable predictions with real data that is normally contaminated with noise. The input data were preprocessed before exploring the prediction possibility. The preprocessing included outliers and highly correlated variables removal for reaching better modelling performance. We studied both regression and classification methodologies for performing RUL prediction. The Random Forest classifier showed promising results. It offers safe and conservative condition-based maintenance. It could provide RUL classes prediction, above/below a certain level. This was demonstrated at 20% level of RUL. The classes prediction was achieved at only 1.24% PAAF% and 7.43% general error rate.

For the future work, the framework will be validated with real system and embedded sensors. Moreover, spare part stocks, arriving time and downtime to be taken into consideration for a complete maintenance planning system.

5. References

Balmus E. (2016). Aircraft Data Acquisition. INCAS Bulletin. 8(1), 141-151.

Byington, C. S., Roemer, M. J., & Galie, T. (2002, March). Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft]. Paper presented at the Proceedings, *IEEE Aerospace Conference*.

Imani, A., & Montazeri-Gh, M. (2019). A Min–Max multiregulator system with stability analysis for aeroengine propulsion control. *ISA transactions*, 85, 84-96.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112): Springer.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7), 1483 - 1510.

Karsoliya S. (2012). Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture. International Journal of Engineering Trends and Technology. 31, 714 – 717.

Koenig, F., Found, P. A., & Kumar, M. (2019). Innovative airport 4.0 condition-based maintenance system for baggage handling DCV systems. *International Journal of Productivity and Performance Management*, 68(3), 561-577.

Liu, L., Wang, S., Liu, D., Zhang, Y., & Peng, Y. (2015). Entropy-based sensor selection for condition monitoring and prognostics of aircraft engine. *Microelectronics Reliability*, 55(9), 2092 - 2096.

Lu, F., Wu, J., Huang, J., & Qiu, X. (2019). Aircraft engine degradation prognostics based on logistic regression and novel OS-ELM algorithm. *Aerospace Science and Technology*, 84, 661 - 671.

Ragab, A., Yacout, S., & Ouali, M. (2016, Jan). Remaining useful life prognostics using pattern-based machine learning. Paper presented at the 2016 *Annual Reliability and Maintainability Symposium* (RAMS).

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008, Oct). Damage propagation modeling for aircraft engine run-to-failure simulation. Paper presented at the 2008 *International Conference on Prognostics and Health Management*.

Yan, W. (2006, Oct). Application of Random Forest to Aircraft Engine Fault Diagnosis. Paper presented at the *The Proceedings of the Multiconference on "Computational Engineering in Systems Applications"*.

Yuan, M., Wu, Y., & Lin, L. (2016, Oct). Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. Paper presented at the 2016 *IEEE International Conference on Aircraft Utility Systems (AUS)*.

Zhang, Y., Ren, S., Liu, Y., & Si, S. (2017). A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products. *Journal of Cleaner Production*, 142, 626 - 641.

Zhao, Z., Liang, B., Wang, X., & Lu, W. (2017). Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering & System Safety*, 164, 74 - 83.

Biography

Hussein A. Taha received his BSc and MSc form Cairo University 2011, 2014 respectively, Electrical Power and machine engineering. He worked as electrical engineer in ASIC Industrial Automation and SAMSUNG Electronics companies respectively. He worked as a teaching and research assistant since 2015. Currently, he is formally a Ph.D. student at Polytechnique Montréal University. His research interests are artificial intelligent in industrial control systems, conditional monitoring and fault diagnosis, and energy management.

Ahmed Sakr is a PhD student in Mathematics and Industrial Engineering at École Polytechnique de Montréal in Canada. He holds a MSc. and a B.Sc. in Mechanical design and production Engineering from Cairo University in Egypt. His research interests and specialties include: Machine learning and artificial intelligence application in operating systems, Manufacturing and service centers modeling, simulation, and analysis, Decision Making for Industrial Systems for optimal performance and waste reduction, Analysis of variance and Design of experiments, Mechanical vibrations and dynamics of mechanical systems. He has developed passion for interdisciplinary tactics to improve operating systems performance. His doctoral research involves simulation and improvement of complex product manufacturing systems performance through the realization of the Big data analytics and IoT technologies. He completed his MSc. and BSc. at Cairo university in renewable energy and vehicle dynamics respectively.

Soumaya Yacout is a Professor in the Department of Mathematics and Industrial Engineering at École Polytechnique de Montréal in Canada. She holds a D.Sc. in operations research from Georges Washington University in U.S.A.; and a B.Sc., and a M.Sc. in Industrial Engineering from Cairo University in Egypt. She designed and taught courses on quality engineering, reliability, and maintenance for undergraduate, graduate, and professional engineers. Her research interests include Condition Based Maintenance, and distributed decision-making for product quality. She has publications in journals including Quality Engineering, International Journal of Production Research, Computers and Industrial Engineering, The European Journal of Operational Research, and JESA. She is a senior member of the American Society for Quality; and a member of the Institute of Industrial Engineering, and the Canadian Operational Research Society.