A Framework for Rapid Prototyping of PHM Analytics for Complex Systems using a Supervised Data-Driven Approach

Katarina Vuckovic[1], Ben Burke[2], and Shashvat Prakash[3]

[1,3] *Collins Aerospace, PHM Enterprise Engineering Windsor Locks, CT, 06096, USA*
katarina.vuckovic@collins.com
shashvat.prakash@collins.com

[2] *Insight Enterprises Inc., 2701 E. Insight Way, Chandler, AZ 85286*
ben.burke@insight.com

Abstract

Prognostic and Health Management (PHM) solutions are becoming increasingly popular in industries that rely on large systems such as aircraft, spacecraft, and power plants. PHM analytic solutions are designed to monitor the health of each subsystem and component and apply predictive analytic to improve system reliability and safety, reduce the cost and decrease time spent on unscheduled maintenance. However, identifying correlations between different components and associated monitors in these large systems can be challenging. To address this issue and achieve maximum utilization of available monitoring signals, a methodology is required that can identify correlations between degraded or failed components and the features engineered from the monitors and sensors. This paper introduces a framework that enables rapid prototyping of analytics, allowing users to seamlessly move from designing and discovering features to developing models for a specific event or component of interest. The framework has three main components: feature exploration, data preparation, and model development. Feature exploration focuses on feature engineering using raw monitor data from all available monitors. Data preparation purges the data, and down-selects relevant features based on correlation defined in the feature exploration part. The data preparation step also creates a training dataset. Model development enables quick testing and comparison of multiple supervised Machine Learning (ML) models. To demonstrate the framework, this paper presents an example of a remaining useful life model for an aircraft component. While the examples and simulations are aircraft-focused, the principles behind the framework can be applied to other large systems.

1. Introduction

Prognostic and Health Management (PHM) has become an essential function for safe system operation and scheduling economic maintenance (Kim, Choi, & Kim, 2021). Thus far, many prognostic algorithms have been proposed for component level prognostics. Examples include bearings (Hamadache, Jung, Park, & Youn, 2019), gears (Hsu, Chang, Hsu, Chen, & Hwang, 2022), batteries (Meng & Li, 2019), and filters (Vuckovic & Prakash, 2022). Single components are easier to model and test by themselves because the component degradation data can be captured during accelerated life testing in a controlled environment. This data, in combination with the domain or physical knowledge, enables model based solutions. However, as we move towards more complex systems with many integrated components, the correlations and interdependencies become intricate and models rarely exist to support them. In recent years, Digital Twins (DTs) that create virtual representation of the complex systems have gained interest (Tao, Zhang, Liu, & Nee, 2018). Over time, the DT may be used to gather data and develop models for PHM applications. Either way, the behavior of the complex system and its components is best modeled using data-driven approaches.

PHM solutions have gained much interest in industries that require high levels of reliability and safety. Examples of these systems such as aircraft (Fei, Bin, Jun, & Shunhua, 2020), nuclear power plants (Zhao et al., 2021), and military defense systems (Lang et al., 2021). Currently, PHM development often occurs much after the system is designed. Adding new sensors to such systems typically requires substantial amount of work to implement and certify. Therefore, adding new sensors may not be feasible nor desirable solution in many cases. As a result, PHM engineers do not always have the luxury to instrument sensors in all the desired locations. Instead, they have to rely on what is already available on the system.

Creating a comprehensive PHM solution for a complex system is a significant endeavor. Typically, developers are given access to historical data from all available sensors or monitors within the system, as well as a list of failures (system or component failures), which are referred to as ”events” in this context. An event can represent a degradation in performance or complete failure. However, the relationship between the events and sensor data (signals) is not always clear, and symptoms of a degraded component may be reflected in the performance of a different component. Therefore, it is necessary to systematically identify the correlation between events and sensor data using data-driven methods.

This paper introduces a framework to first, isolate the relevant data, second, to engineer features into training data sets, and third, identify the correlation between these features and the events of interest (EOI). Once this processing framework is established, a further Machine Learning (ML) framework will allow developers to rapidly train and test multiple supervised ML models using the dataset generated in the previous step.

1.1. Contributions

The contributions of this paper are summarized as follows:

- The work proposes a novel data-driven framework that enables PHM development for complex systems consisting of many components and signals. The framework guides the developer from concept generation all the way to performance validation.
- The framework provides a methodology for identifying relevant features in the ”data preparation” component. Identifying relevant features is key to developing successful analytic.
- The ”Model Training” component enables rapid protyping of a wide range of supervised ML models which significantly reduced the time that the developer needs to allocate to a given EOI.
- A numerical example for developing component Remaining Useful Life (RUL) estimation using a regression model is presented. The example goes through the framework and demonstrates the analytic development process and the performance of the resulting analytic.

The rest of the paper is organized a follows. Section 2 provide a high level overview of the architecture and defines all the components within it. The next four section discuss each component of the framework in detail starting with the approach for identifying opportunities in Section 3. This is followed by the data preparation and feature engineering component in Section 4 which leads to the automated-ML (auto-ML) model training component in Section 5. In this component several models are evaluated to identify the best ML model for this given use case. The last part of the architecture is the performance validation component described in Section 6. Section 7 provides a simulation example for a regression model to demonstrate the framework process. Finally, Section 8 summarizes the work and highlights the main points.

2. ARCHITECTURE OVERVIEW

This section presents a high level overview of the different components within the framework shown in Figure 1. The process flows from left to right, separated into four phases, starting with ”Opportunity Identification” as the first phase. Here, the types of events targeted in the ’events of interest’ set must be clearly identified, especially in presence of multiple components and failure modes. The steps and approaches involved will be discussed in the next section.

The next phase is ”Data Preparation”, where features are engineered from raw sensor data and correlated to events. Since there are many features available, this part also aims to down-select the features to only the essential ones. These features are then used to train the ML models. There are four different model types and each of them requires a different training dataset format (classification, regression, time-series classification, and time-series regression).

The third phase, ”Model Training”, consists of auto-ML libraries designed specifically for PHM applications. The libraries enable the developer to quickly asses different models and to optimize them with different hyper-parameters.

The final phase is ”Performance Validation”. The framework considers two approaches to evaluating the performance based on whether the model type is classification or regression.

3. OPPORTUNITY IDENTIFICATION

In a plethora of signal-EOI combinations, it may be challenging to identify the order in which the analytics should to be tackled in an incremental development process. Therefore, a systematic approach to prioritize the work is needed to establish a successful framework. This section outlines some methods that can be used to identify opportunity and assign priority. It is focused on addressing the question of how to determine the priority of developing analytics for different systems or components.

3.1. Customer Request

Customer requests and requirements are certainly the most direct way of obtaining priority and narrowing the focus to specific events. However, the requirements are not always available. Furthermore, even if the customer desires an analytic for a specific component, it is not always obvious if there are parameters that can support the PHM model. Therefore,

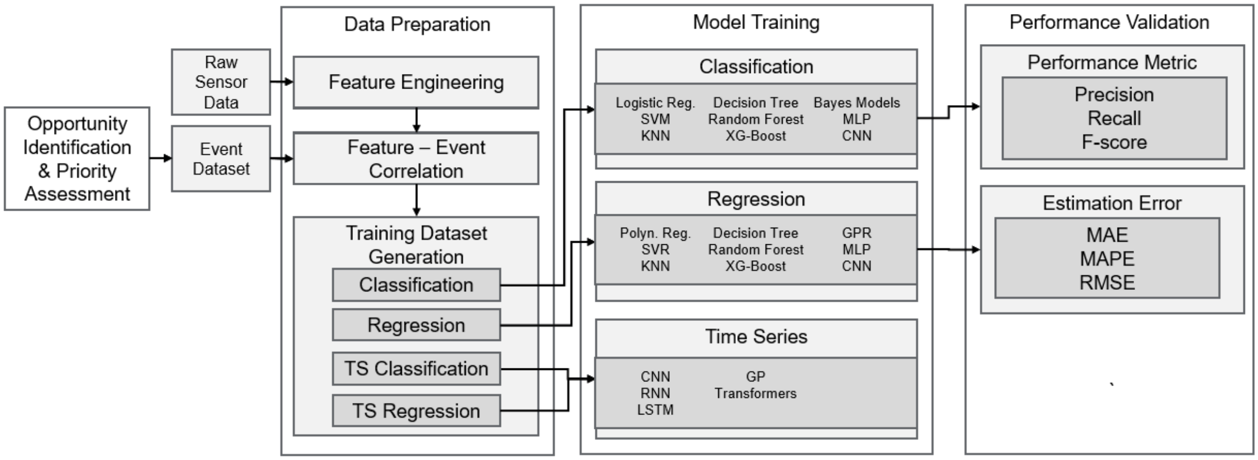


Figure 1. Block diagram illustrating the major component of the framework and the flow of the process.

developers frequently have to perform their own analysis to assess the feasibility and benefits of analytics in order to set priority.

3.2. Cost-Benefit Analysis

As cost is a key driver for the PHM business model, the Cost-Benefit Analysis (CBA) is the best way to determine if implementing the analytic is profitable. Identifying components and failure modes for PHM solutions would involve weighing factors like the cost of operational disruption against potential reduction in serviceable life. For example, components with high removal rates and high cost of unscheduled maintenance are typically good candidates. In the case of commercial airlines, unscheduled maintenance is expensive due to the down time that can result in flight delays, cancellations, loss of potential revenue, or loss of customer goodwill. In addition, labor and equipment costs may be compounded when diagnosis is incomplete or leads, at times repeatedly, to the wrong component. Airlines in particular tend to observe a failure symptom and replace several components until the issue is resolved.

3.3. Safety and High Risk Components

Safety is paramount and will always supersede cost, a reality that is most apparent in military and civilian aviation applications. Although safety would be factored into the initial design and type certification, PHM strategies can enhance existing safety margins. Furthermore, efforts are underway to consider prognostic strategies as an eventual replacement for some airworthiness inspection intervals (*International Maintenance Review Board Policy Board, Aircraft Health Monitoring (AHM) Integration in MSG 3*, 2018). For example, PHM models that predict engine degradation or failure can effectively reduce the need for scheduled inspections (Khan et al., 2021).

3.4. High Correlation

There may be, on occasion, specific event and signal combinations which occur frequently together. If the correlation is pronounced and the analytic development effort low, then pursuing such an analytic may be beneficial even if it does not necessarily lead to a prognostic recommendation or carry any other sort of priority. The benefit instead could be in alerting to a need for further modeling, monitoring, or engineering analysis.

4. DATASET PREPARATION

4.1. Feature Engineering

The engineering task starts by applying a set of transformations to each raw signal collected during a flight in an effort to generate features. The transformations applied are various statistical functions that convert raw signals into a single value per flight. Each feature is tracked over time and saved into a feature-specific dataset.

Correlations are then performed against each unique feature and each unique EOI, automatically. If the feature is statistically different before and after the the event (i.e. significant drop or increase of feature within a set window), then the instance is triggered as a *correlated event* between the feature and event, and saved for future reference and investigation in another dataset.

4.2. Feature Down-selection and Explainability

The correlated events dataset is then filtered to a specific EOI based on the above *opportunity criteria*. This event could be a specific component, set of components, type of maintenance, or a derivation thereof. This single event will become the dependent variable for the ML model.

The correlated events dataset saved earlier, is used to discover the desirable independent variables for the EOI chosen. Because each feature is correlated to a specific event within the EOI, total percentage of explain-ability for an entire EOI can be derived for the feature.

For example, if a maintenance action was performed 100 different times on the entire fleet for a specific EOI (i.e. component removal), and a feature was found to be correlated to 5 of them, then level of feature explain-ability for that EOI is 5%.

This percentage-of-feature-explain-ability is used as the criteria to down-select the features based on a defined threshold (quartile, specific-threshold, relative value). In practice, a threshold limit may be set such that if the percentage is below the limit, then the correlation is not statistically significant even if it is the most correlated feature. If this occurs the analytic cannot move forward with model development given these features.

4.3. Dataset Formatting Based on Model Type

Once the features are identified, the the data is pre-processed to create a training dataset. The framework differentiates between four different training dataset formats: 1) classification, 2) regression, 3) time-series classification, and 4) time-series regression.

The raw dataset generally has more healthy (normal operation) data than the unhealthy data, therefore the dataset needs to be correctly balanced. Additionally, for the train/test split, the framework separates the dataset by aircraft, thus ensuring that the whole sequence prior to an event remains in tack during testing/training.

4.3.1. Classification

One popular type of classification model is a binary classifier that differentiates between the healthy and unhealthy data. Occasionally, it may be challenging to identity the transition from healthy to unhealthy and labeling the data may becomes a burdensome task.

However, the framework relies on the correlated EOI trigger to classify the training dataset. For example, everything within a certain window before the correlated EOI, could be considered unhealthy while all other events could be considered healthy. Care must be taken with the approach however as the dataset will become strongly imbalanced and will require the proper tuning prior to modeling.

4.3.2. Regression

The RUL estimation problem can be tackled using a regression model. The input of the model is the feature at some time instance t prior to the correlated EOI and the corresponding output is the RUL. The RUL of a component can be defined in terms of number of days or cycles remaining until failure. In aircraft PHM modeling, the RUL may be expressed in terms of flights or flight hours, depending upon the dominant failure mechanism.

4.3.3. Time-Series

Time-series models can be either classification or regression models in the same manner as defined in the previous two subsections. The main difference is that time-series analysis involves analyzing a set of data points collected over a period of time to identify patterns, trends, and relationships between variables, while the previously mentioned models treat each instance in time as a single data point. By using time-series analysis, it becomes possible to monitor variations in data over an extended period, and leverage the historical data to enhance predictions by recognizing how the feature deteriorates over time.

4.3.4. Data Challenges

To develop a high quality model, there are certain requirements that we impose on the quality and quantity of the dataset. When considering the dataset, we have to ensure that i) the number of component removals is large enough to develop a training dataset, and ii) that the same component is removed on multiple aircraft to avoid aircraft bias. These requirements are best illustrated in the graph in Figure 2. Each data point on the graph represents a different component. The x-axis represents the total number of removals of a given component, while the y-axis represents the total number of different aircraft on which this same component was removed. For example, if we consider component $C_m = (26, 25)$, this means that the dataset contains 26 removals of component C_m across 25 different aircraft. Since the number of removals can never exceed the number of aircraft, all the data points will be at or below the $y = x$ line. The best case scenario is when the data points are in the upper right corner, where there are a lot of removals across different aircraft. The data points in the lower right corner indicate that there are a lot of component removals but they are all located on a few aircraft, this can often point to a systemic aircraft issues rather than the component. The data points that are in the lower left region indicate very few removals and probably not enough data to develop a model.

Next, the data needs to be labeled for supervised training. We have already mentioned that this can be a challenge especially for classification where we have to determine when the component transitions from healthy to unhealthy. In an RUL case, labeling the data is trivial as we can count backward from the time that the component was removed. We only consider a window of 50 flight prior to removal and assume that during this time the component transitions from healthy to unhealthy. The maximum RUL is subjective to the component

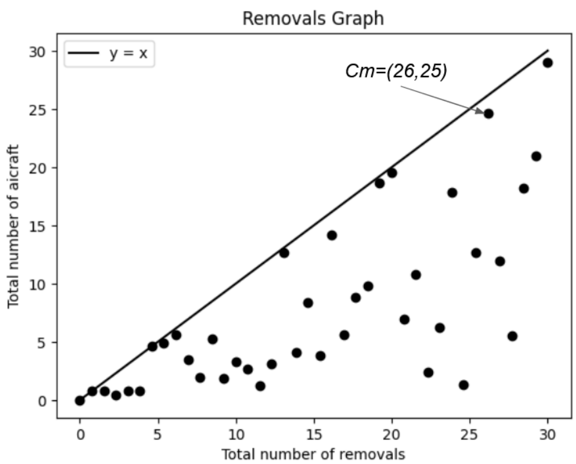


Figure 2. **Component removal analysis graph showing removal information of different aircraft components.**

and could vary between components depending on how frequently the component is replaced. However, since there is generally more healthy that unhealthy data, we run into data imbalance issues. By limiting the dataset to only a certain window prior to the removal, we attempt to address this issue at least to some extent.

5. Model Training

The rapid prototyping framework heavily relies on the model training component, which is a crucial factor. This component comprises three sub-components, namely regression, classification, and time-series. In Section 4.3, the developer determines the model type and formats the training dataset accordingly. Each of the three model types has its auto-ML library, which is specifically designed for PHM applications. The auto-ML libraries provide baseline implementation a collection of common supervised ML models, enabling the developer to swiftly train and evaluate multiple models. Table 1 summarizes the common models present in the libraries. Furthermore, the user also has the ability to develop new models and add them to the existing libraries. The auto-ML environment also offers the flexibility to modify and retrain models. This transparency and flexibility sets the auto-ML environment apart from many other existing auto-ML tools such as auto-sklearn, AutoKeras, and Databricks AutoML. Moreover, the model training component utilizes Machine Learning Operations (MLOps) to streamline and automate the entire ML lifecycle (Aronchick & Boykis, 2020).

6. Performance Validation

In order to fully assess model performance, various factors must be taken into account, including accuracy or error metrics, uncertainty analysis and lead time. It is important to note the evaluation of model performance differs across classification and regression model types. While a comprehensive study on prognostic model evaluation metrics has already been conducted (Prakash, Vuckovic, & Amin, 2023), this section aims to summarize the most essential metrics.

Agnostic to the actual performance metrics of choice is the device used to track them over multiple experiments and modeling scenarios. Utilizing a tool, such as the open sourced MLFlow, can allow for quick and even automated experimentation and model selection. As development of the framework continues, our tracking tool will serve as not just a repository but also a meta-data-driver to facilitate in tracking, retraining, model selection, model serving, model & feature drift and more; all in a automated and dynamic manner. A true MLOps implementation.

6.1. Performance Metrics for Classification Models

The performance of classification models is commonly reported in terms of *precision* and *recall* defined in (1) and (2). Precision is the probability of the event occurring given an alert has been enunciated, while recall is the probability of alerting given the event will occur. Figure 3 shows the confusion matrix where one axis is the predicted class and the other is the true value class. If both true and predicted values agree, the model scores a True Positive (TP). If the model alerts without the event it is a False Positive (FP), while a False Negative (FN) means the model failed to adequately detect the event.

$$\begin{aligned}\text{Precision} &= P(\text{ImpendingFailureEvent}|\text{PrognosticAlert}) \\ &= \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}\end{aligned} \tag{1}$$

$$\begin{aligned}\text{Recall} &= P(\text{PrognosticAlert}|\text{ImpendingFailureEvent}) \\ &= \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}\end{aligned} \tag{2}$$

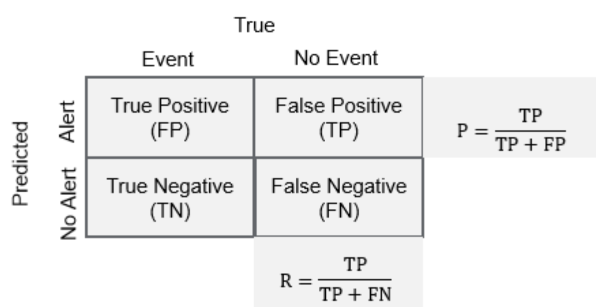


Figure 3. **The binary classifier confusion matrix, showing the precision and recall conditions.**

Occasionally, precision and recall are combined into the F-score metric as in (3). The F-score provides a balance between precision and recall. Combining two parameters into

Table 1. Auto-ML models available in the Model Training Component.

Classification	Regression	Time-Series Classification/Regression
Logistic Regression	Polynomial Regression	CNN
Decision Trees	Decision Trees	Gate Recurent Unit (GRU)
Random Forest	Random Forest	Recurrent Neural Network (RNN)
XGBoost	XGBoost	Long-Term Short-Term Memory (LSTM)
Support Vector Machine (SVM)	Support Vector Regression (SVR)	Transformers
k- Nearest Neighbor (KNN)	KNN	GP for time-series
Bayes Models	Gaussian Process Regression (GPR)	
Multi-Layer Perceptron (MLP)	MLP	
Convolutional Neural Network (CNN)	CNN	

one makes comparing the performance among the different models simpler.

$$\text{F-score} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

6.2. Standard Error Metrics for Regression Models

The accuracy of regression models is typically evaluated by calculating the error. There are a few error metrics commonly used in RUL regression estimation: 1) the Mean Absolute Error (MAE), 2) the Mean Absolute Percent Error (MAPE), and 3) the Root Mean Square Error (RMSE) (Liu & Chen, 2019). The equations are defined in (4) - (6), respectively, where x_i is the true and $\hat{x}_i$ is the predicted RUL, and n is the total number of test samples.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{4}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{x_i - \hat{x}_i}{x_i}| * 100\% \tag{5}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{n}} \tag{6}$$

Assessing the performance of a model using multiple error metrics is valuable. Each metric has its strengths and limitations, and choosing the right one depends on the data characteristics and the goal. MAE is less sensitive to outliers, while RMSE is useful for identifying the impact of outliers. However, both are specific to the data scale. MAPE can compare results across different scales but has drawbacks, such as the need to exclude true value data points that are equal to zero and the fact that it penalizes negative errors more than positive errors. Choosing the right metric leads to accurate analytics, and using multiple metrics provides a comprehensive understanding of the results.

6.3. Additional Validation Considerations

Beyond the numerical validation results, there are also other model validation aspects that may have to be considered. Ethics may play a role when the models have a community impact or when safety is at risk. Models may also have to pass certification depending on how and where the analytic is being deployed. ML based products are still in the early stages of Federal Aviation Administration (FAA) certification development and ML-based certification is still a challenge in the aviation industry. No matter if the requirements are coming from an external certification authorities or from internal internal requirements, the models generally need to be explainable to be approved for deployment. This explainability tends to come from subject matter experts that can provide some justification based on physics models or knowledge domain of the systems, especially when it comes to signal selection and feature engineering. The developers should consider these verification step as well as any additional requirements mandated by the internal or external governing bodies, and the should consult with their verification and validation representatives during the development. However, these consideration are outside the scope of this framework, which mainly focuses on the data-driven model development.

7. SIMULATION NUMERICAL EXAMPLE

i

Table 2. Error metric for evaluated regression models.

Model	MAE	RMSE	MAPE
Linear Regression	12.5	14.5	170.9%
Random Forest	4.6	5.61	55.3%
XGboost	12.1	14.1	154.8%
SVR	11.8	14.1	150.1%
ElasticNet	12.5	14.5	171.0%
MLP	12.5	14.4	171.8%
KNN	10.1	12.2	119.4%
GPR	0.1	0.2	0.5%

The objective of this example is not to discuss the specific components and features but rather show that this data-driven approach works on a real dataset and can be repeated for multiple components. For any new component, the combination of features will vary and the best model type will likely be different but the framework and approach will still be applicable. The objective of the framework is to reduce the time and effort that a developer needs to invest in designing a high quality analytic. Therefore, we do not focus on the explainability of the model in this section.

8. CONCLUSION

This paper presented a framework for development of PHM analytics using a data-driven approach. The framework is intended for complex systems where many components and signal monitors/sensors co-exist. The latent correlation between the components and signals are sometime difficult to deduce using domain knowledge and physics based models. The framework performs both feature engineering, feature selection, and model training. Using this framework, the developer can quickly identify the relevant features given a component of interest and then apply a set of different ML models to find an optimal solution. The ML models are predefined in the framework to default hyper-parameters, however the developer has full control of the models and can optimize and modify them as needed. The performance of the models is evaluated using standard performance metrics. Finally, the paper presented a numerical example to RUL estimation for a given aircraft component using four features.

9. FUTURE WORK

Future work will focus on developing analytics and presenting numerical results for the classification and time-series models. Furthermore, the current architecture is designed to only train supervised models. As part of the future work, we intend to investigate unsupervised methods and expand our framework to include these models. We believe that the unsupervised methods will be particularly helpful in cases where it is difficult to label the data. For example determining when the component transitions from healthy to unhealthy and labeling the data accordingly is a challenge that may be overcome by supervised methods. Another research direction for future work is on improving feature engineering. Thus far, we presented a method for combining features at the ML phase, however, signals from multiple sources can also be combined during the feature engineering phase. We intend to explore all these improvements in order to generate a more robust and more comprehensive framework. (Hrnjica & Softic, 2021)

REFERENCES

Aronchick, D., & Boykis, V. (2020). *Mlops: Continuous delivery and automation pipelines in machine learning*. O'Reilly Media.

Fei, X., Bin, C., Jun, C., & Shunhua, H. (2020). Literature review: Framework of prognostic health management for airline predictive maintenance. In *2020 chinese control and decision conference (ccdc)* (p. 5112-5117). doi: 10.1109/CCDC49329.2020.9164546

Hamadache, M., Jung, J. H., Park, J., & Youn, B. D. (2019). A comprehensive review of artificial intelligence-based approaches for rolling element bearing phm: shallow and deep learning. *JMST Advances*, *1*, 125–151.

Hrnjica, B., & Softic, S. (2021). The survival analysis for a predictive maintenance in manufacturing. In *Advances in production management systems. artificial intelligence for sustainable and resilient production systems: Ifip wg 5.7 international conference, apms 2021, nantes, france, september 5–9, 2021, proceedings, part iii* (pp. 78–85).

Hsu, T.-H., Chang, Y.-J., Hsu, H.-K., Chen, T.-T., & Hwang, P.-W. (2022). Predicting the remaining useful life of landing gear with prognostics and health management (phm). *Aerospace*, *9*(8), 462.

International Maintenance Review Board Policy Board, Aircraft Health Monitoring (AHM) Integration in MSG 3 (Tech. Rep.). (2018). Issue Paper 180,Air Transport Association of America.

Khan, K., Sohaib, M., Rashid, A., Ali, S., Akbar, H., Basit, A., & Ahmad, T. (2021). Recent trends and challenges in predictive maintenance of aircraft's engine and hydraulic system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *43*, 1–17.

Kim, S., Choi, J.-H., & Kim, N. H. (2021). Challenges and opportunities of system-level prognostics. *Sensors*, *21*(22), 7655.

Lang, B., Zhao, H., Mi, C., Huang, Q., Song, L., & Ma, G. (2021). Application and analysis of prognostics and health management technology in weapon equipment. In *2021 3rd international conference on system reliability and safety engineering (srse)* (p. 56-59). doi: 10.1109/SRSE54209.2021.00016

Liu, J., & Chen, Z. (2019). Remaining useful life prediction of lithium-ion batteries based on health indicator and gaussian process regression model. *IEEE Access*, *7*, 39474-39484. doi: 10.1109/ACCESS.2019.2905740

Meng, H., & Li, Y.-F. (2019). A review on prognostics and health management (phm) methods of lithium-ion batteries. *Renewable and Sustainable Energy Reviews*, *116*, 109405.

Prakash, S., Vuckovic, K., & Amin, S. (2023). Prognostic model evaluation metrics. In *2023 ieee aerospace conference* (p. 1-11). doi: 10.1109/AERO55745.2023.10115952

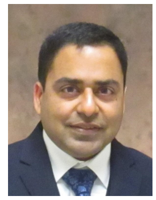
Tao, F., Zhang, M., Liu, Y., & Nee, A. Y. (2018). Digital twin driven prognostics and health management for complex equipment. *Cirp Annals*, *67*(1), 169–172.

Vuckovic, K., & Prakash, S. (2022). Remaining useful life prediction using gaussian process regression model. In *Annual conference of the phm society* (Vol. 14).

Zhao, X., Kim, J., Warns, K., Wang, X., Ramuhalli, P., Cetiner, S., … Golay, M. (2021). Prognostics and health management in nuclear power plants: An updated method-centric review with special focus on data-driven methods. *Frontiers in Energy Research*, *9*, 696785.

BIOGRAPHIES

Katarina Vuckovic received her B.S. in Aerospace Engineering (2017), B.S. in Electrical Engineering (2017), M.S. in Electrical Engineering (2019) from Florida Institute of Technology. She is currently pursuing her Ph.D. in Electrical Engineering at the University of Central Florida. She has been with Collins Aerospace for six years working as a systems engineer on wireless communication systems, aircraft automation applications, and prognostics and health management of aircraft components.



Shashvat Prakash received a B.S. in Mechanical Engineering from University of Illinois, Urbana, an M.S. in Mechanical Engineering from Carnegie Mellon University, and a Ph.D. in Mechanical Engineering from the Georgia Institute of Technology. He has worked on satellite attitude and orbit control at NASA Goddard, combustion and control of aviation turbines at General Electric, and prognostics and health management (PHM) of aircraft components at Raytheon Technologies-Collins Aerospace. Currently a Senior Principal Engineer, he has over 13 years of experience in the commercial aviation industry.

Ben Burke received a B.S. in Communications from Brigham Young University - Idaho and a M.S. in Data Science and Business Analytics from the University of North Carolina at Charlotte. He has worked in the cloud, data and data science space as a consultant for almost a decade, helping clients create unique, native, and distinct solutions harnessing cloud architecture, data engineering and data science skills. Currently he is a Sr. Data Scientist with Insight, who provides consulting services for Collins Aerospace.