Data Mining and Neural Network Applications in Aerospace Industry Health and Usage Monitoring Systems

Ryan Patton
Missouri University of Science and Technology, Department of Engineering Management and Systems Engineering
223 Engineering Management, 600 W. 14th St., Rolla, MO 65409-0370
(673) 341-4572,
rtpwx3@umsystem.edu

Abstract—**Major airlines equip their fleets of aircraft with bountiful sensors with the ultimate goal of uncovering as much information on the aircraft as possible to monitor the aircraft's performance along with its health and usage. Until airlines, commercial contractors, and defense contractors place a broader emphasis on monitoring aircrafts' health and usage before an abnormal performance occurs, the data gathered from various sensors attached to the aircrafts and other measurements will not yield near to their potential. For various reasons, the aerospace industry is playing catch-up with discovering the potential of mining and tracking all the historical and real-time data gathered from each aircraft. Given the recent tragedies with Boeing's 737 Max, applying data mining algorithm techniques to sensor data could not only save airlines, commercial contractors, and defense contractors billions of dollars long-term, but their reputations as worldwide leaders of safety.**

Keywords—Health and Usage Monitoring, Prognostics and Health Management, neural networks, data mining, K-means clustering

I. Introduction

Health and usage monitoring (HUMS) emerged as a vital room for improvement in the aerospace industry within the last decade. Although efforts to improve aircraft monitoring have taken place for nearly half a century, the evolvement of sensors and other technical data measurement techniques make real-time monitoring a more well-rounded activity to capture the availability, reliability, and safety of aircraft [8]. The difficulty in prioritizing HUMS lies within the aerospace industry's nature of business. Commercial and defense contractors will draw up contracts with airlines and/or the military with requirements laid out for the product. Once requirements are stipulated and agreed upon by the stakeholders, all essential activities revolve around fulfilling requirements first and foremost. Utilizing limited resources to put towards an advanced HUMS system with little or no direct payoff is not feasible when plenty of other measures stipulated by the contract are already in place to guarantee the safety of the aircraft. Only recently, with the prevalence of big data, has the aerospace industry placed a broader emphasis on HUMS [10].

The HUMS controls three main airborne tasks: cockpit voice recording; flight data recording; and the ground task of maintenance data analysis. HUMs provides automated power assurance checks, in-flight crew alerts when thresholds are exceeded, package data for evaluation by ground-based applications that feature automated rotor and shaft balance solutions, drivetrain and engine health assessments, and automated logbook entries for maintenance records [8]. By analyzing maintenance data, incipient defects in major components can be detected before they can hazard the safety of flight, the exceedance of a range of limitations can be detected, and many items of equipment can increase their reliability by minimizing airframe vibration [4]. One such company in the United Kingdom, Smiths Aerospace, is integrating its own data mining tool to automatically analyze data instead of relying on real-time alerts and downloaded post-flight analysis by ground-based systems. By doing so, they can apply fleet analysis methodologies to trend the current health and usage of an aircraft and on a larger scale, compare a particular aircraft's health and usage with others in the fleet [5].

UTC Aerospace Systems (UTAS), a pioneer in vehicle health management (VHM) in the late 1990s and early 2000s, received recognition from the Army for a 27 percent higher sortie rate after comparing aircraft fleets equipped with their Vehicle Health Management System (IVHMS) to non-IVHMS equipped aircraft. In addition to the higher sortie rating, the IVHMS-equipped aircraft achieved a 52 percent reduction in unscheduled maintenance and a 17 percent reduction in total maintenance for the UH-60L Blackhawk. Given the data supporting the UTAS IVHMS from the early 2000s, HUMS-equipped aircraft today may be capable of achieving drastically better results because of simultaneous developed technologies affecting the health and usage of the system and concurrent developed technologies from the HUMS itself [14].

Modern data mining practices involving HUMS aim to study condition indicator (CI) patterns so healthy CIs can be distinguished at certain thresholds from bad CIs. Oftentimes, CIs are derived from vibration data from various parts. Because of the scope of secured proprietary knowledge amongst aerospace companies, many components still need further analysis to provide a baseline for condition based maintenance metrics for both the companies and Federal Aviation Administration (FAA) approval [1]. One main challenge with producing a comprehensive HUMS for universal aircraft adoption stems from the various aircraft parts each needing specific tests and monitoring methods. The measures by which a bearing is deemed out of its normal threshold range may be different than the measures by which a turboprop engine is deemed out of its normal threshold range.

Coming up with a Prognostics and Health Management (PHM) system necessitates advancements in big data mining methodologies to analyze the intricacies of helicopters, unmanned aerial systems (UAS), spacecrafts, planes, and other various aircrafts [11].

This paper proposes a method for mining, processing, and analyzing turbofan engine degradation data before testing the data in a neural network formation. After feeding the data through a relevant neural networks algorithm, a strategy for improving data sorting for implementation into a PHM system is proposed. Detailed analysis of the mined data and real graph findings illustrate further extensions this work may have in the field of HUMS.

The paper summarizes as follows. Section 2 gives a brief overview of data mining methodologies used in the aerospace industry with subsections A., B., C., and D. describing four highly applicable methods before a general summary subsection E. Section 3 details problem formulation related to the specific data mining methodologies and related neural network practices to be used in the experiment that follows. Section 4 describes training, testing, validating, and performance analysis of the mining methodologies used. Section 5 answers post-testing analysis, the relevancy of the experiment in HUMS, and proposed future work. Lastly, Section 6 concludes the paper with modern data mining observations in the aerospace industry.

II. Data Mining Methodologies in Aerospace

Health management (HM) applications at different levels of the system result in multi-tiered benefits. At the component-level they support: diagnostics/built-in-tests, usage/life tracking, mechanical diagnostics, gas path debris monitoring, performance trending, prognostics, blade health monitoring, and propeller balancing. At the aircraft-level they support: corrosion monitoring, exceedance monitoring, flight regime recognition, individual aircraft tracking, loads monitoring, operational usage monitoring, mission capability assessment, and maintenance assessment. At the higher fleet-level they support: fleet trending, force life management, performance metrics, knowledge discovery, and data gathering [6].

Throughout the various levels, the HUMS decision-making process uses machine learning techniques to incorporate component history, diagnostics, and parametric data. All the data being fed from the aircraft can be used to reduce learning error, to pattern-search for skewed populations, and to combat issues arising from data matrices formed for machine learning [4].

One such effort undertaken by the United States Army and a couple industry consultants transformed the data collected from an Army helicopters' rotorcraft drive train components to analyze the fleet's health. By using supervised classification learning algorithms to reduce misclassification error through an error function (also known as a penalty function), they could see the distance between the hyperplane classification boundary and a point misclassified. To reduce the data skew pointing towards equal representations of healthy components and faulted components in their manned air vehicles, the scientists proposed four methods for adjusting the learning process to avoid skewing the results towards always determining healthy [7].

A. Normalized Error for Skewed Classes

When the error function is computed, the classes normalize to their size so the error measured on the dense class is given equal weight to the measured sparse class. By doing so, the smaller class is essentially over-sampled. The sum of the normalized error from the healthy data plus the normalized error from the faulted data is computed as follows:

$$Err(g) = \frac{\sum_{i=1}^{N_h} Err(g(N_{h,i}))}{N_{healthy}} + \frac{\sum_{k=1}^{N_f} Err(g(N_{f,k}))}{N_{faulted}} \tag{1}$$

For training iterations, the data separates into healthy and faulted. Each data point is compared to the decision boundary learned by the machine to identify misclassifications. The total number of misclassifications are summed and normalized by the appropriate population size, healthy population for false positives and faulted population for false negatives. The information is then sent back to new iterations of the boundary can be trained and tested using the same hypothesis, iterating until the minimum number of misclassifications is found [9].

B. Receiver Operating Characteristic (ROC) Curve Penalty

Based on two applications for computer error diagnostics, Airworthiness Applications 6-nines of reliability and Advisory Applications on Aeronautical Design Standard 79D (ADS-79) guidelines, the performance criteria percentages can be converted into weight penalties caused by false diagnostic outputs [6]. Penalties can be used to multiply the error measured during training. During training, data is separated between healthy and faulty. Each data point is then compared to the decision boundary learned by the machine to identify misclassifications. After recording the total number of misclassifications and their distances, the total number of healthy misclassifications are summed and multiplied by the penalty. The total number of faulted misclassifications are summed and multiplied by the penalty [6]. The information is then looped back to the machine to create a new evolution of the boundary, iterating until the minimum number of misclassifications is found.

C. Compound Error

The compound error method is creating by combining Methods A. and B. The error is normalized as in Method A. and then penalized based on Method B. criteria. The follow equation applies:

$$Err(g) = R_{FP} * \frac{\sum_{i=1}^{N_h} Err\big(g(N_{h,i})\big)}{N_{healthy}} + R_{FN} * \frac{\sum_{k=1}^{N_f} Err(g(N_{f,k}))}{N_{faulted}} \tag{2}$$

During training evolutions, the data is again separated into healthy and faulted. Each data point is compared to the decision boundary learned by the machine to identify misclassifications, recording the total number of misclassifications and their distances from the boundary. Summarizing and normalizing the total number of misclassifications allows for the multiplication of the total normalized number of healthy misclassifications by the FN penalty and the total normalized number of faulted misclassifications by the FP penalty. Finally, the information is sent back and used to create a new evolution of the boundary, iterating until the minimum number of misclassifications is found [9].

D. Normalized Error and ROC Curve Training

Method D. capitalizes on training the machine towards performance goals after a hypothesis is chosen. By doing so, the learned decision boundary shifts to include a balance of the training data so the FP and FN penalties are balances based on structure. Each data point is compared to the decision boundary learned by the machine to identify misclassifications. The total number of misclassifications and their distances from the boundary are then recorded. The total number of misclassifications are summed and normalized by the appropriate population size. The information sent back to the machine aids in applying a heuristic rule to balance the error and shift the boundary toward the desired FP and FN rates. The reported shifted model will iterate until it achieves the desired rates, generating a new hypothesis each time the desired rates are not achieved [9].

E. Summary

The four methods listed above demonstrate practical approaches to incorporating HUMS data in the application of machine learning as it relates to data mining. The methods to follow address data mining applications as they generally apply to the aviation industry at large.

Applying data mining to aerospace products may be difficult due to temporal, spatial, or spatio-temporal nature of flight data. Prior to selecting and tuning techniques, variables such as noise, the underlying structure for data quality checks, and constructing features based on knowledge of the source of data must be taken into account to focus on time-series data mining [11]. The primary data mining techniques used for observation include clustering, classification, indexing, and segmentation. Cluster focuses on finding natural groupings of the time series in the database under some similarity/dissimilarity measure. Classification assigns an unlabeled time series to one of multiple predefined classes such as phase of flight. Indexing finds the nearest matching time series from a database given a query time series and some similarity/dissimilarity measure. Segmentation constructs a model from piecewise segments such that it closely approximates the time series, given a time series containing n data points. Another technique garnering consideration is time-delay for time-series comparisons because time shifts can yield significant effects as they render similar series completely dissimilar. In contrast, Dynamic Time Warping (DTW), a new distance measure proposed to address the time-delay issue by shifting between two time series' similarities, shows little signs of catching on for broad use in aviation data mining [11, 8]. For time-series, the techniques previously mentioned group by their use of the original data.

For data mining applications as they pertain to flight data, the flight data is first divided up into sub-phases such as takeoff, cruise, and descent so the flight parameters can be analyzed according to flight conditions [11]. The premise behind the division of sub-phases is that flight data parameters are relatively homogeneous within a given phase and will share similar characteristics or features with the same parameters in other flights. Each phase may be represented by a mathematical signature, a vector of attributes providing a statistical representation of the parameter time series. A quadratic regression model may be applied to recursively fit on a time-window of the parameter time series. By doing so, the mean, maximum, minimum, and variance may be estimated while discovering discrete parameters which are estimated with a state transition matrix. Clustering and atypical determination may be applied to analyze a set of flights of interest, given an atypical score to evaluate using the parameter signature arrays of the flights [13]. The atypical score for each flight can then be drawn for comparison by issuing a global atypical score. The atypical score may be calculated by the following equation:

$$A_i = \sum_{j=1}^{n} PCA(j)^2{}_i / \lambda_j \tag{3}$$

The global atypical score derived from the equation above bodes well to a gamma distribution. The p-value of each observation against the gamma distribution of the set of flights is used as a measure of atypicality [4], establishing a comparison across phases of flight. The clustering membership score for flight I in a given phase may be estimated by:

$$cms_i = \frac{n_i}{N} \tag{4}$$

Where n_i is the number of flights in the cluster that flight I is a member of, and N is the total number of flights. Taking the clustering membership score, the global atypicality score for a flight in a given phase can be calculated as:

$$G_i = -\log(p_i) - \log(cms_i) \tag{5}$$

Based on the global atypicality score, flights may be ranked according to the largest score across all phases. Other ranking schemes are in development to prioritize most effectively the different phases of flight specific atypicalities [11].

Inductive health monitoring with clustering and statistical metrics sheds light on the process to build a knowledge set of nominal behavior with the arrangement of training data into an array so that each data point in the set is represented by a vector of values for parameters of interest. The process to build the knowledge set employs agglomerative clustering to establish subsets of nominal conditions on the basis of the underlying similarity of states. Once the database of nominal condition clusters is built health monitoring is implemented by testing whether the current system state subscribes to one of the clusters [1].

Anomaly detection may be analyzed using density-based clustering. Density-based clustering compares multi-parameter signals for a given flight phase across multiple fights to identify anomalous data subsets. The data transformation process anchors all parameters in all flight data records about a common event like landing, samples all parameters with at fixed intervals, and concatenates the sample of all parameters into a single vector v. Since the size of the vectors can consist of thousands of elements, dimensionality reduction is needed to apply principle component analysis to the vectors. This retains the top principal components explaining approximately 90 percent of the variance [11]. Clustering then takes place using DBSCAN where the number of outliers is found to be relatively insensitive to the algorithm's parameter setting the minimum number of points for density neighborhood. The radius of the point neighborhood is found to provide the suitable sensitivity to achieve the desired proportions of clusters. A couple algorithms used for anomaly detection are ClusterAD, MKAD, and traditional FDM, with ClusterAD and MKAD outperforming exceedance detection in the identification of operational anomalies. ClusterAD offers advantages for continuous flight data while MKAD offers great capabilities for discrete sequence data [6].

Anomaly detection in nominal flight data sequences generally uses SequenceMiner. SequenceMiner captures the chronological evolution of aircraft states as described by discrete settings and switches recorded in the flight deck as a set of symbols. For anomalous sequences, they first must be found in a sequence pool and then they can be described as to what makes the sequences anomalous. If taking a parametric or model-based approach, the Hidden Markov Model (HMM) is used. The HMM consists of a set of hidden states and a set of observation. HMM needs to be developed and tuned by training the model artifacts like the transitional matrix where the probability of evolving from on hidden state to another is specified. After development, the HMM can estimate the probability of an observation sequence of interest. To contrast HMM, a discriminative approach to anomaly detection relies on data characterization on the basis of similarity for classification or clustering. The longest common subsequence (LCS) is adopted by SequenceMiner to similarly measure discrete sequential data by Euclidean distance. The main advantage of LCS is its scalability because it frees up the need to train model artifacts like the transitional matrix. The downside of LCS is its lack of interpretability to explain why dissimilar sequences are abnormal. Identifying anomalous sequences using SequenceMiner will apply k-mediod CLARANS clustering algorithm to account for the length of the LCS normalized by the length of the complete sequences. When analyzing symbolic sequences of real flights, SequenceMiner identified 12 flights with potential anomalies in a set of over 2,200, while reported results indicated 5 out of the 13 candidates actually experienced significant operational anomalies [6].

Multiple Kernel Anomaly Detection (MKAD) strives to concurrently handle discrete and continuous data types, improving upon prior work where the focus mostly remained on or the other, to identify anomalies based on a supervised learning approach. MKAD has its origins I the support vector machine (SVM) problems whose goal is to create a classifier function for testing points using its dot product with training points. A kernel function measures similarity between two vectors by providing a scalar quantity. The multiple kernel function simply weights the sum of multiple constituent kernels. Applying to MKAD, a discrete data kernel function and a continuous data kernel function are given varying weights. The discrete data kernel functions has its roots in the normalized longest common sequence used previously in SequenceMiner as the similar measure. To implement MKAD follow the following steps: 1) train data acquisition, 2) transform continuous data using Symbolic Aggregate approximation (SAX), 3) solve for the Lagrangian multipliers, 4) construct the classifier and determine phi, and 5) apply the classifier for testing [6].

The final method for anomaly detection using the nearest neighbor and symbolic dynamic filtering uses the Orca algorithm. The Orgca algorithm implements distance-based anomaly detection based on the concept of the Kth nearest neighbor. For each point, the average distance to its K nearest neighbors provides a measure of its relative proximity to neighboring points. The top t points with the largest value are flagged as outliers. The parameters K and t are tuning variables. The indexed-Orca (iOrca) algorithm provides further improvements with a more aggressive strategy for updating the cutoff threshold whereby data points are indexed according to distance to a reference point. Symbolic Dynamic Filtering (SDF) may be used for feature construction alongside iOrca for anomaly detect. SDF reduces continuous data series into a symbolic sequence, similar to the SAX approach [6]. This overall scheme reduces data and noise while incurring some loss of information.

By presenting numerous data mining applications towards aerospace problems, a more fundamental approach may be taken towards aerospace datasets in the future. Many of the techniques listed aimed to reduce the number of parameters while preserving the variability of the data. Since most commercial and defense fixed-wing aircraft can be expected to perform in about the same way undergoing about the same phases, their data may be pooled together. One challenge opposing this conglomeration of data is the inexactness of rotorcraft phases requiring their own independent analysis, regardless of the similarity between their parameters and datasets [10]. In addition, fixed wing aircraft may drastically vary in the amount of data they capture since there are no limitations on where sensors may be placed on an aircraft as long as safety is not impeded. Complex parameter trends show promise in the unsupervised and supervised algorithms proposed to offer a strategic advantage in identifying inherent nominal and anomalous conditions, applying them as a basis for anomaly detection/classification, and updating them with more incoming data [12].

Although the various methods above all apply equally to different problems within aerospace, the remainder of the paper focuses on one particular method used in determining a turboprop engine's degradation over time.

III. Problem Formulation, Training, Testing, and Validating

With a focus on HUMs, many anomalous aerospace datasets were considered for experimentation such as a lithium-ion battery's degradation, gearbox fault detection, bearing faults, capacitor electrical stress, and small satellite power. The data used in the following tests is turbofan engine degradation provided by National Aeronautics and Space Administration (NASA) Ames Center and was carried out using C-MAPSS [3]. The data consists of four sets simulated under different combinations of operational conditions and known fault modes, recording several sensor channels to characterize fault evolution. Each set of data contains approximately 20,000 lines of data, each representing an operational cycle, with 26 columns. Column 1 represents the unit number, Column 2 the time, in cycles, Columns 3-5 the operational settings, and Columns 7-26 the sensor measurements. In the training sets, the engine develops a fault at some point during the series and the fault grows in magnitude until system failure.

In the test set, the time series ends at some point prior to system failure. Using the available data, the number of operational cycles before failure is predicted. The test data is validated using vectors of True Remaining Useful Life (RUL) values. The data is analyzed in MATLAB's Deep Learning toolbox for simplicity and built-in Graphic User Interfaces (GUIs) [15].

A. Data Processing and Mining Methodologies

The degradation data is split into a training data set and validation data set for later performance evaluation. Group variables are specified to capture variables of interest so the ensemble data can be sampled and visualized. Based on ensemble data observation, the degradation process shows no clear trend for run-to-failure measurement, exemplified in Figure 1.

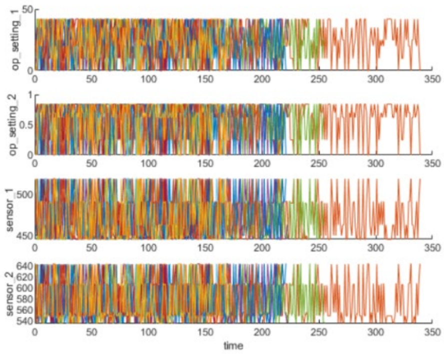

Figure 1: Training Regimes Showing No Trend.

From the sensor signals, clearer degradation trends may be extracted from the operating conditions. Each cell in the training data is concatenated into a single table with three operational settings and shown on a 3-Dimensional (3D) scatter plot. The scatter plot seen in Figure X shows 6 regimes with each point in the regime clustering close in proximity to each regime.

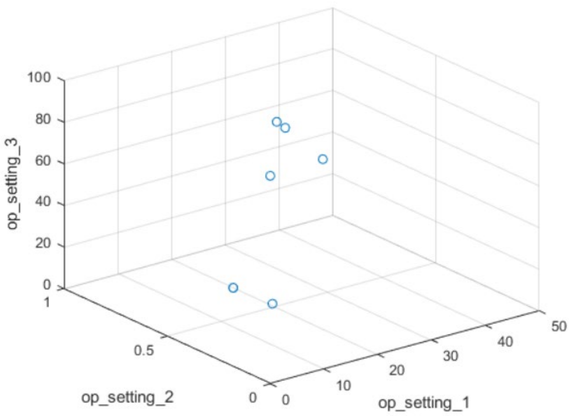

Figure 2: Plot Clusters.

From Figure 1, the clusters may be improved by locating each of the 6 automatically. A K-means algorithm may be used, varying the initial conditions along the way, to reach the results with the lowest cost. In this particular instance, the algorithm ran 10 times where similar results were achieved. Figure 2, seen below, highlights the ability of the algorithm to find the 6 working regimes quickly and at a low cost.

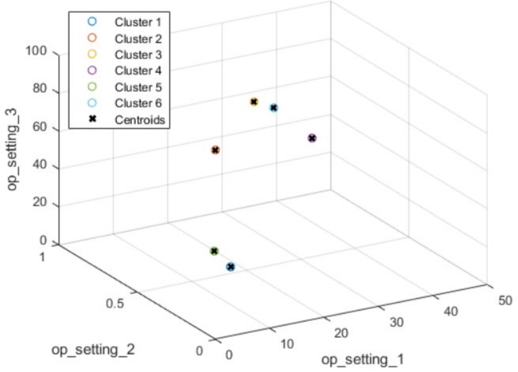

Figure 3: Conditional Plot Clusters.

With plot clusters returning accurate centroidal values, the next logical step is to normalize the different working regimes. The mean and standard deviations are computed for each sensor measurement by the work regimes identified in the plots. Revisiting the data statistics in each regime helps the normalization process. For each ensemble member, the operating points of each row are extracted and the distance is computed to each cluster centers to find the nearest cluster center. Subtracting the mean and dividing it by the standard deviation of that cluster for each sensor measurement develops trends in the data. For standard deviations close to 0, the normalized sensor measurement may be set to 0 to pre-negate their values for RUL estimation. The trends from normalizing the data may be seen in Figure 3.

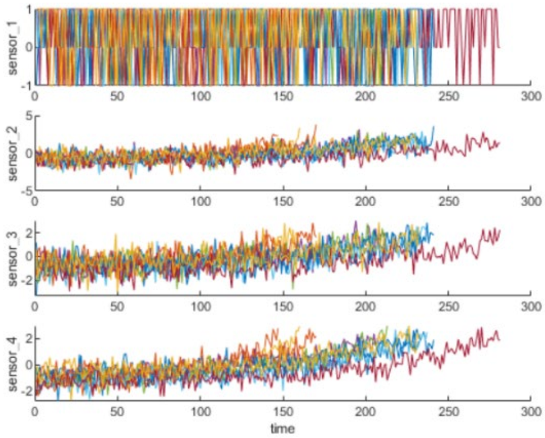

Figure 4: Normalized Ensemble Sensor Plots.

The 26 sensor measurements are compared to select the most trendable to feed into the health indicator for prediction. A linear degradation model is estimated for each sensor measurement and the slopes of the signals are ranked. The 8 sensors with the largest slopes are picked to advance. Some of the top candidates' trends are shown in Figure 5.

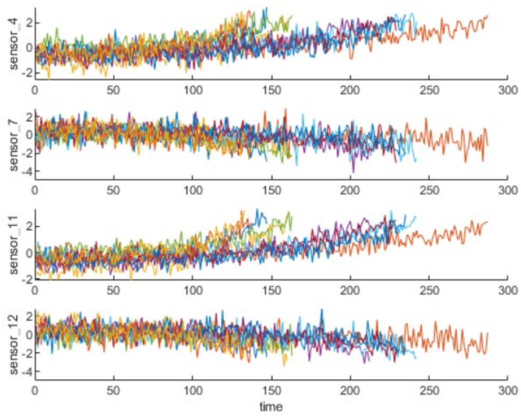

Figure 5: Top Sensor Measurements Trends.

The top trending sensor measurements may now be fed into a health indicator neural network to determine when they may fail and their relative health conditions.

B. Neural Network Implmentation

By assuming a healthy condition for each run-to-failure data and assigning a value of 1 to start with a degradation failure to 0 over time, the data will degrade linearly over time while fusing the sensor values. The health condition is visualized as follows in Figure 6. The results from fusing the sensors and incorporating weights/biases may be seen in Figure 7. Figure 8 uses the same process as Figure 7 with validation data.

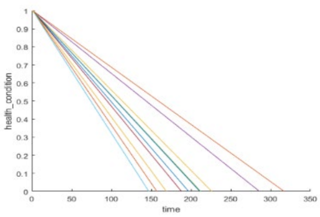

Figure 6: Linear Health Condition Trends.

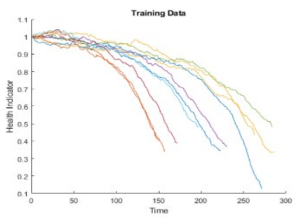

Figure 7: Fused Health Indicators.

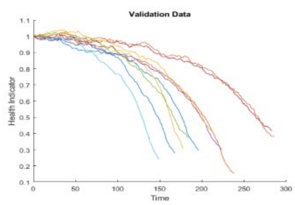

Figure 8: Validation Data Fused Health Indicators.

The RUL model generated from the training data will fit each fused dataset with a 2nd order polynomial. The ideal model will find the nearest 50 ensemble members in the training dataset, fit a probability distribution based on the 50 ensemble members, and use the median of the distribution as an estimate of RUL. The RUL model will be used for evaluation of the sample validation data to predict each of the sensors regimes' respective RUL.

IV. Performance Analysis

The similarity RUL model will use percentage increments of 50%, 70%, and 90% of sample validation data. K nearest neighbor plots are generated with the data truncated at the three percentages with their nearest neighbors. Figures 9, 10, and 11 show the K Nearest Neighbor Plots at each percentage truncation.

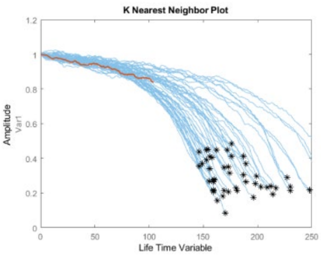

Figure 9: 50% K Nearest Neighbor Plot.

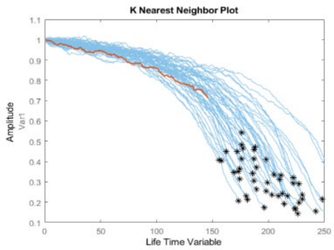

Figure 10: 70% K Nearest Neighbor Plot.

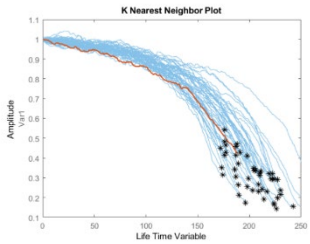

Figure 11: 90% K Nearest Neighbor Plot.

The probability distribution of the estimated RUL at each percentage truncation may be seen as follows in Figures 12, 13, and 14. The distributions show an improvement in the error between the estimated RUL and the true RUL at higher percentages.

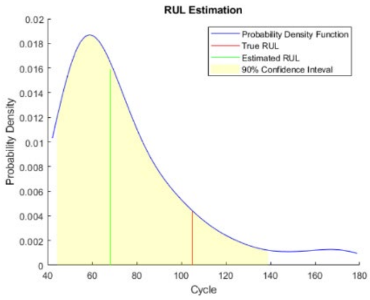

Figure 12: 50% Probability Distribution Plot.

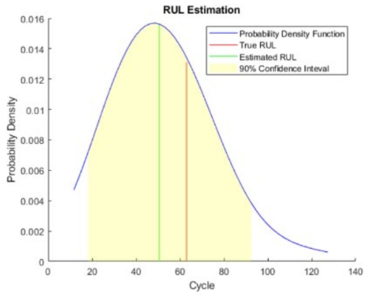

Figure 13: 70% Probability Distribution Plot.

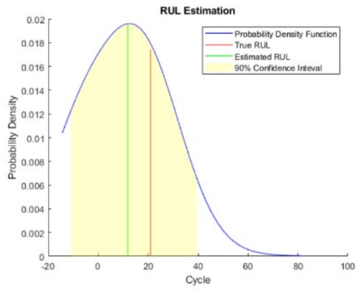

Figure 14: 90% Probability Distribution Plot.

The same evaluation then proceeds to use the whole validation set and compute the error between estimated RUL and true RUL for each breakpoint. In Figure 15, the histogram of the error for each breakpoint with its probability distribution provides a visual for relative error between ensembles. The prediction error in Figure 16 shows box plots to visualize the median and 25-75% quantile outliers. Figure 17 visualizes the concatenation of the predictions and from this, it can be observed that the error concatenates more around 0 with more data so less outliers are present.

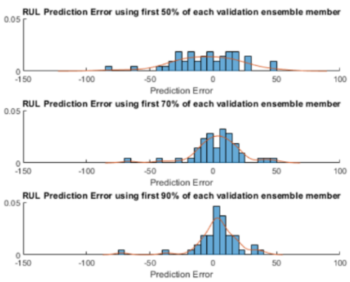

Figure 15: RUL Predictions Errors - % Ensemble Member.

Figure 16: Prediction Error of Each Validation Ensemble Member.

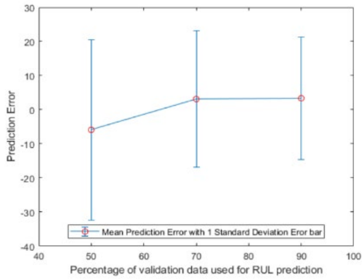

Figure 17: Percentage of Validation Data Used for RUL Prediction.

The preceding work exemplifies academic data mining and neural network principles to determine turbofan degradation time-series failure points. The validity of the results captured the entire datasets' information because of the attention given to pre-processing the data. The RUL estimations succeeded in that the condition indicators change over time is observable and connected with the system degradation process in a reliable, measurable way.

V. Observations and Future Work

The success of the applications to the problem in Sections 3 and 4 show the small-scale potential for HUMS and PHM in aerospace systems. Implemented enterprise-wide, HUMS and associated data relay may happen in real-time across all major aircrafts within another decade or two. Eventually, aircrafts of the same fleet may communicate with each other and compare health data to collectively gather more accurate failure predictions relevant to their particular subsystems. Although the experiment above focused strictly on turbofan engine data, each subsystem and component of the aircraft could execute its own strategy for health monitoring as long as analysis efforts identify the variables contributing to its failure and apply an algorithm capturing its interactions. Building on complete subsystem and component PHM strategies, cross-system functionalities could then identify previously undetected variable connections.

Data mining could tremendously improve the sorting processes in real-time across satellites, drones, helicopters, airplanes, and space vehicles. With large sets of data constantly being gathered from aerospace systems, it is vital that proper data mining techniques are applied to improve the costliness of sorting through the vast amounts in real-time to give conclusive indicators of health. Some of the data mining techniques applied from Section 1 show high promise in this area. While the tech industry has received the majority of the focus for advancements in data mining algorithms, many of their fundamental features can be incorporated into algorithms yielding successful results in the aerospace industry. More research is needed to figure out the nuances in aerospace systems' anomalous behavior to understand various applications of primarily used in the past for more software-heavy systems. Introducing environmental variables into the hardware-focused aircraft complicates the process of interpreting the noise of the data to reach definitive conclusions about the health of systems, their subsystems, and their components. HUMS and PHM will continue to grow in their stressed importance in the aerospace industry as computational abilities continue to scale from their lagging integration status. Data mining methods specialized for the aerospace industry will follow suit along with HUMS and PHM advancements.

The dataset worked in this paper may be further validated and applied to a dataset put together for a PHM08 Challenge by NASA [2]. The new dataset does not contain known estimated RUL values inspiring a continuation of the work presented within this paper to test how the mining methods fare for prediction error when the data may not be trained in the same way.

All of the different algorithms presented in Section 1, specifically for anomaly detection in aerospace systems, reveal the field is playing catch-up to the tech industry and the algorithms may still see improvements over time as the environment may be better defined. The general applications of data mining algorithms will also improve as patterns emerge across environmental functions as newer, more rounded datasets become available. Although clustering is generally agreed upon to be the best method(s) to work with the data as of now, more algorithms may take advantage of the uniqueness of aerospace data to improve costliness, noise reduction, and accuracy. Although any language may be used with the deep learning methodologies presented, industry practice favors current development efforts in Python which could be a jumping off point for further development away from the canned features MATLAB offers in its deep learning toolbox.

In addition to direct applications with HUMS, data mining principles applied to HUMS may show crossover for solving other relevant developmental problems in the aerospace industry. For example, the data sorting algorithms applied to a HUMS problem may help sort aerospace flight data and reveal a better way to sort data for automated simulator aircraft testing. Many of the problems HUMS and PHM encounter slowing progress pertain to how best to model the data within an aerospace cross-functional framework. The complicated nature of flight in different conditions is the limiting factor that when better assumptions are applied and more accurate models are produced may provide a whole assortment of breakthroughs in the industry.

VI. Conclusions

Section 1 gave a general description of current efforts and problems arising in the aerospace industry to better model, mine, and apply data to give an indication as to a system's health. Section 2 described modern data mining methodologies accounting for anomalous behavior in aerospace systems. Sections 3 and 4 overviewed a specific problem highlighting the fundamental applications of data mining and neural networks towards health indicators in an aerospace subsystem, a turbofan engine. Section 5 presented a glimpse of what may be in store for the future work of data mining in the aerospace industry. The research and work presented in this paper aim to give a simplistic survey on the uniqueness of data mining development efforts in HUMS, as pertinent to aerospace vehicles and their support systems. In the future, HUMS and PHM will play a larger role in the aerospace industry as commercial and defense companies take a look at what can be done to improve costs, reliability, and safety of their aircraft.

REFERENCES

[1] A. Gavrilovski, H. Jimenez, D. Mavris, A. Rao, S.-H Shin, I. Hwang, and K. Marais, "Challenges and Opportunities in Flight Data Mining: A …," ResearchGat, AIAA SciTech, 23-Sep-2016, San Diego, CA. [Online]. Available: https://www.researchgate.net/profile/Arjun_Rao7/publication/290194327_Challenges_and_Opportunities_in_Flight_Data_Mining_A_Review_of_the_State_of_the_Art/links/56955f5aeab58a9a4e4f4/Challenges-and-Opportunities-in-Flight-Data-Mining-A-Review-of-the-State-of-the-Art.pdf?origin=publication_detail. [Accessed: 15-March-2020].

[2] A. Saxena and K. Goebel (2008). "PHM08 Challenge Data Set", NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA.

[3] A. Saxena and K. Goebel (2008). "Turbofan Engine Degradation Simulation Data Set", NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA

[4] D. He and E. Bechhoefer, "Development and Validation of Bearing Diagnostic and Prognostic Tools using HUMS Condition Indicators," *2008 IEEE Aerospace Conference*, Big Sky, MT, 2008, pp. 1-8.

[5] D. Learmount, "Smiths to take HUMS to next level," Flight Global, 22-Jun-2004. [Online]. Available: https://www.flightglobal.com/smiths-to-take-hums-to-next-level/55020.article. [Accessed: 25-Apr-2020].

[6] D. Wade, R. Lugos, and M. Szelistowski, "(PDF) Using Machine Learning Algorithms to Improve HUMS Performance" [Online]. Available: https://www.academia.edu/26956620/Using_Machine_Learning_Algorithms_to_Improve_HUMS_Performance. [Accessed: 20-Apr-2020].

[7] D. Wade, T. Vongpaseuth, R. Lugos, J. Ayscue, A. Wilson, L. Antolick, N. Brower, S. Krick, M. Szelistowski, and K. Albarado, "(PDF) Machine Learning Algorithms for HUMS Improvement on …," ResearchGate, 05-May-2015, AHS Forum 71, Virginia Beach, VA. [Online]. Available: https://www.researchgate.net/publication/286926665_Machine_Learning_Algorithms_for_HUMS_Improvement_on_Rotorcraft_Components. [Accessed: 25-Apr-2020].

[8] "Health and Usage Monitoring Systmes," Wikipedia, 07-Apr-2020. [Online]. Available: https://en.wikipedia.org/wiki/Health_and_usage_monitoring_systems. [Accessed: 25-Apr-2020].

[9] Helicopter Health and Usage Monitoring (HUMS)," Meggit [Online]. Available: https://meggittsensing.com/aerospace/product/helicopter-hums-to-prevent-failures-increase-safety-and-reduce-costs/. [Accessed: 18-Apr-2020].

[10] H. Yan, X. Liqing, L. Xingwang, W. Ying, and F. Haiyan, "The Investigation and Demand Analysis of HUMS on the Civil Helicopter," 2017 Prognostics and Health Management Conference (PHM-Harbin), 2017.

[11] J. Kappas and P. Frith, "From HUMS to PHM: Are We There Yet?," Semantic Scholar, 2017. [Online]. Available: https://www.humsconference.com.au/Papers2017/Not_on_USB_Proceedings/191_HUMS2017_Kappas.pdf. [Accessed: 25-Apr-2020].

[12] J.M. Pena, F. Famili, and S. Letourneau, "Data Mining to Detect Abnormal Behavior in Aerospace Data," *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discvoery and data ming – KDD '00*, pp. 290-297, 2000.

[13] K. Hawko, "The Role of HUMS," StackPath, 05-Oct-2017. [Online]. Available: https://www.aviationpros.com/airports/airport-technology/article/12365423/the-role-of-hums. [Accessed: 11-Apr-2020].

[14] M. Seale, A. Hines, G. Nabholz, A. Ruvinsky, O. Eslinger, N. Rigoni, and L. Vega-Maisonet, "Approaches for Using Machine Learning Algorithms with Large Label Sets for Rotorcraft maintenance," 2019 IEE Aerospace Conference, 2019.

[15] "residualSimilarityModel," *MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/predmaint/ug/similarity-based-remaining-useful-life-estimation.html. [Accessed: 02-Apr-2020].