A Hybrid Learning Approach to Prognostics and Health Management for Military Ground Vehicles Using Time-Series and Maintenance Event Data

W Glenn Bond, Haley Dozier, Thomas L Arnold, Antoinette Silas,
Indu Shukla, Quyen T Dong, Brandon E Hansen, Corey Hollis Mize

US Army Engineer Research and Development Center, Information Technology Laboratory, Vicksburg, MS, 39180, USA

William.G.Bond@erdc.dren.mil
Haley.R.Dozier@erdc.dren.mil
Thomas.L.Arnold@erdc.dren.mil
Antoinette.R.Silas@erdc.dren.mil
Indu.Shukla@erdc.dren.mil
Quyen.T.Dong@erdc.dren.mil
Brandon.E.Hansen@erdc.dren.mil
Corey.H.Mize@erdc.dren.mil

ABSTRACT

Attempts to leverage operational time-series data in Condition Based Maintenance (CBM) approaches to optimize the life cycle management and Reliability, Availability, and Maintainability (RAM) of military vehicles have encountered several obstacles over decades of data collection. These obstacles have beset similar approaches on civilian ground vehicles, as well as on aircraft and other complex systems. Analysis of operational data is critical because it represents a continuous recording of the state of the system. Applying rudimentary data analytics to operational data can provide insights like fuel usage patterns or observed reliability of one vehicle or even a fleet. Monitoring trends and analyzing patterns in this data over time, however, can provide insight into the health of a vehicle, a complex system, or a fleet, predicting mean time to failure or compiling logistic or life cycle needs. Such High-Performance Data Analytics (HPDA) on operational time-series datasets has been historically difficult due to the large amount of data gathered from vehicle sensors, the lack of association between clusters observed in the data and failures or unscheduled maintenance events, and the deficiency of unsupervised learning techniques for time-series data. We present a method of discovering patterns in vehicle operational data that determines models for predicting the likelihood of imminent failure, referred to as Parameter-Based Indicators (PBIs). Our method is a data-driven approach that uses both time-series and relational maintenance data. This hybrid approach combines both supervised and unsupervised machine learning and data analytic techniques to correlate labeled maintenance event data with unlabeled operational time-series data utilizing the DoD High Performance Computing (HPC) capabilities at the U.S. Army Engineer Research and Development Center. In leveraging both time-series and relational data, we hope to demonstrate a means of fast, purely data-driven model creation that is more broadly applicable and requires less a priori information than physics informed, data-driven models. By blending these approaches, this system will be able to relate some lifecycle management goals through the workflow to generate specific PBIs that will predict failures or highlight appropriate areas of concern in individual or collective vehicle histories.

1. OVERVIEW OF THE HYBRID LEARNING APPROACH

Data analytics and machine learning are applied to vehicles and complex systems in several disciplines to prognosticate remaining useful life, assess system health, and to inform decision makers of a wide variety of life-cycle relevant facts. Event driven and sensor data has been collected for years if not decades, from some platforms. This data can be used to both provide a repository of knowledge regarding system history and state, and to inform life cycle and operational needs such as availability, maintenance, sustainment, logistics, and acquisition. These ideas have evolved, as the data collected has become broader and deeper in terms of scope, to reflect a more real-time stance by leveraging insights gained.

U.S. Army ground vehicle operational sensor data has been collected for several ground vehicle platforms over a period of roughly eight years. In total, it represents approximately seven Terabytes of searchable data that is growing with time and due to the expansion of the program. A wide array of values is collected every second the vehicle is powered on by a Controller Area Network (CAN bus) system. Beyond basic error code recording that the CAN bus system is typically deployed to monitor, this system collects and stores all of the temperature, pressure, level, RPM, and other types of data gathered by the CAN bus sensors, placing the values, one row per second, into a compressed daily file. Time stamped rows consist of startup, one Hertz, and fault data, representing startup and normal operation modes as well as CAN bus recognized faults. Several levels of sensor, component, and vehicle metadata are also recorded into each daily file. Data collected in the daily files for each vehicle is then downloaded from the vehicle and uploaded to a storage filesystem, and converted to a database ingestion friendly format. The files are ingested into one database that reflects the original data, and another in which the values are cleaned and imputed, along with aggregative augmentation, as part of the ingestion process. In this way, we protect the data provenance of our analytic workflows while harnessing the speed of database ingestion for data cleaning.

Currently, the database represents over one million days of operation across several vehicle types and consumes five Terabytes for raw files and seven Terabytes each for the two databases. The database is stored on a dedicated system, separate from the computation nodes. A virtual machine, running on a third system, runs the server that processes queries to fill dataframes for the model creation workflows running on compute nodes. In this way, we optimize access to HPC machine learning model creation processing for all of the data.

Physics informed linear models have been used to successfully demonstrate prognostication of health-related events for vehicles by means of estimating operational parameters and comparing the expected to the observed values. Such methods are created as a result of an observed relationship, such as that between revolutions per minute and oil pressure in vehicle sensor data, and have demonstrated a high degree of success in predicting the occurrence of certain faults that lead to maintenance events (Vogl, Weiss, and Helu, 2019).

Given the size and scope of vehicle operational sensor data, such relationships are difficult to discern without a priori knowledge of an exploitable relationship like the oil pressure predicted value model mentioned above. Our research asks what will be required to leverage the processing and memory of DoD High Performance Computing (HPC) with data analytics and Machine Learning (ML) in order to find many other, more general Parameter-Based Indicators (PBIs)?

The nature of the time-series operational sensor data and the dimension reduction and cluster analysis techniques used to identify normal and abnormal operational intervals is detailed in section 2.1. We discuss HPDA techniques for process monitoring, manifold creation for comparison of data, and identification of normal and abnormal operating regimes. This portion of the workflow establishes a spectrum of operating regime related parameter ranges that will help identify healthy operation while identifying the intervals of operation where parameters were not all within the healthy spectra. Groups of parameters that vary from their counterparts within healthy identified clusters will be identified for later investigation as candidates for new PBIs, indicating possible maintenance or fault occurrence is likely based on received, streamed operational data.

Sections 2.2, 2.3, and 2.4 discuss the application of more mainstream ML techniques to the time-series data. Because fault data occurs within the time series data, alongside healthy operation data, some ML models target identification of parameters that predict fault occurrence within the data. These scenarios can both identify parameter ranges and regimes that are outside healthy ranges and identify parameter functions that can predict likely fault occurrence, as identified within the streamed operational data. These models can go further, however, to assist in identifying intervals of operation that should be considered healthy or not healthy and may also predict some maintenance events.

Where faults and cluster results can be correlated with identified parameter patterns, the problem becomes how to label them given the text-based nature of maintenance event label entry. Section 2.5 discusses a partial solution to this uncertainty. Text-based labels, entered by maintenance personnel at the time of accomplishment, are inconsistent as represented in the maintenance data and so are transformed to vector space representation for clustering and classification. In this way, a significant subset of maintenance event labels is consistently labeled for time-wise correlation with the identified but unlabeled outside-normal operation parameter subsets.

Despite some of the uncertainties identified in section 2, these techniques are combined into a hybrid workflow. Techniques for dealing with uncertainties in correlating maintenance events with identified faults and vice-versa are discussed. A workflow is established that leverages the established relationships from these sections, allowing creation of novel PBIs based on questions from users of ground vehicle data that translate to queries of the relational maintenance data. In this way, we broaden the scope of PBI creation over the physics-based approach by reducing the need for a priori knowledge to create models, generalizing the technique to a significantly faster, data-driven approach.

2. The Hybrid Model Workflow

The Hybrid Model is a workflow that begins by combining elements of machine learning and process monitoring with manifold learning and cluster analysis to detect anomalous patterns in streaming parameters. These patterns are essentially subsets of the set of sensor inputs plus aggregations of them. Together, these patterns correlate well with future occurrences of fault conditions being reported by each vehicle's Controller Area Network bus (CAN bus) system or with logged maintenance events as reported in the log data. These candidate Parameter-Based Indicators (CPBIs) need labels to be useful as indicators of current and future possible, specific conditions. Labels are derived from time-wise correlation between CPBIs and maintenance logs.

Maintenance descriptions are manually entered text, so the description of the same event, repeated at different times over different vehicles can vary significantly. To standardize their label set, events in the maintenance log database have been vectorized and the vectorized events classified. In this way, a significant portion of the comprehensive set of maintenance events can be labelled consistently. Each event can then be analyzed for time intervals and vehicle identification numbers of occurrence. In this way, we establish a somewhat reliable subset of maintenance events.

Uncertainty in vehicle log data may be erroneous for a few reasons in addition to text entry problems. Maintenance events are sometimes initiated when no fault exists or the fault that does exist does not match the event, resulting in a miss in the correlation calculation below. Simply put, the following may be true and will confound the correlative process:

1) Maintenance events might be ineffective at repairing the fault, such as replacing a battery when the problem is another component.
2) Faults may not match with a maintenance event because no maintenance event was initiated. This can happen because of redundancies or intermittency issues preventing the fault from exhibiting sufficient symptoms to be discovered.

Either situation, in addition to misclassification of log entries by the vectorization process, results in a miss in terms of the correlation calculation and a lowering of the confidence in the candidate PBI.

The set of unlabeled, candidate PBIs are established separately through several means detailed below. Any one from the set may be analyzed based on time intervals and the Vehicle Identification Number (VIN) of each occurrence. Correlating a PBI with a label, then, involves achieving a correlation threshold computed by taking the number of matches over the number of matches plus misses as defined below. Matches are defined on a per element basis. Elements of the set of candidate PBI occurrences fall within a threshold value I_{Match}, defined below. I_{Match} is the difference between the time stamp of the fault (T_{Event}) and the time at which the maintenance event was recorded (T_{CBPI}) for a particular vehicle.

$$I_{Match} = T_{Event} - T_{CPBI} \tag{1}$$

In equation (1), the fault's time stamp must precede the maintenance event, so only positive values of I_{Match} will be considered. The VIN must match as well. Then for each element of the set of CPBIs, this threshold is used to evaluate the pair as a match (*M*) or miss (*m*). Matched candidate PBI/Event pairs, then, are those in which a fault occurred within a specified time interval of a following maintenance event performed on the same vehicle. The correlation of the set of CPBIs and the set of maintenance events, across VINs, can then be evaluated as a ratio of the matches and misses within the sets.

$$C = {M_{total}}/{(M_{total} + m_{total})} \tag{2}$$

Correlation (C) is defined by Eq. (2) as the total number of elements within the two sets which fall within the defined interval, I_{Match}, with matching VIN, divided by the total matches (M_{total}) plus the sum of the unmatched items from both the maintenance event set and the candidate PBI set (m_{total}). Defining thresholds for the correlation and matching interval can then be set by using the existing time-series data to test the candidate PBI. These values can then be manipulated to yield appropriate certainty levels as defined by existing PBI models, such as the oil pressure model. In this way, the loop is closed and we establish dataset level values for these thresholds.

2.1. Anomaly Detection

For Army ground vehicles, faults are considered to be an unallowed deviation of one or more parameters of the vehicle system (or subsystems). These deviations from the vehicle's regular operation can lead to substandard vehicle performance or even vehicle failure. Process monitoring techniques can be used to detect a malfunction in the system, isolate the malfunction's location, and identify the severity of the malfunction. This information can then be used to mitigate the fault before a failure occurs, reducing failures and maintenance down time.

The goal of this section is to develop parameter-based indicators in support of fault detection, isolation, and identification. The development of these indicators relies heavily on the detection of anomalous operational data. Due to the high-dimensional, interconnected parametric (sensor) data collected from Army ground vehicles, manually defining baseline operational data is inefficient and can fail to capture subsystem dependencies. Many traditional process monitoring techniques are additionally ill-suited to the high number of parameters in vehicle systems and subsystems (Schimert, 2008). The solution to this dilemma is to use unsupervised learning techniques designed to programmatically detect a correlation between systems and sensors and find a lower-dimensional structure in the data (manifold learning). By using dimension reduction, clustering, and process monitoring techniques, normal and abnormal operating conditions can be established and evaluated for faults. With this data, the visualization of trends can be made apparent through the use of charts. This work took chart visualization one step further and developed a monitoring graph where the abnormal operations are superimposed upon the normal operation graph. Identification of parameter-based anomalies is possible, and these anomalies can lead to faults and failures. Due to the nature of the sensor dataset, it is inherently multivariate, and work must be done to obtain a reduced set of variables to determine abnormal behavior (Jelali, 2013).

2.1.1. Unsupervised Learning Methods for Determining Normal and Abnormal Operating Conditions

Unsupervised learning is a type of machine learning algorithm that is used to make inferences from datasets without using labeled response variables. These methods are particularly useful when the goal of the algorithm is to detect patterns in a dataset that may have been previously undetected without substantial human supervision. Unsupervised learning methods are necessary for condition-based maintenance on ground vehicles due to the vast amount of unlabeled data as well as disparities in the knowledge of normal and abnormal operating conditions.

Traditionally, normal operation can be considered to be data that is collected on dates where no faults occur. This method is problematic in a variety of ways, most of which is that changes in the data that might predict faults that often occur well before the fault alert is triggered. In addition to this, the predefined fault alerts may not encompass all of the vehicle's possible malfunctions. Therefore, to determine the normal and abnormal operation of the vehicle, first, an unsupervised dimension reduction technique is used to identify a lower-dimensional subspace of the dataset. Then a clustering algorithm can be used to identify clusters in the lower-dimensional subspace (Jelali, 2013). A clustering algorithm can also be used to group similar structures that can be correlated time-wise to the predefined faults recorded over time. Data that is not correlated time-wise with the predefined faults can then be analyzed further and compared to maintenance logs to determine whether each cluster of data can be labeled as a normal operating condition or as data that leads to a fault not previously logged.

In this effort, Independent Component Analysis (ICA) is used to create the lower-dimensional subspace of the sensor data. ICA is a linear dimension reduction method, traditionally used in signal processing, that transforms the dataset into columns of independent components. The goal of ICA is to reduce the dimensions of the dataset by identifying and projecting the data into a new space that maximizes the statistical independence of the dataset while preserving the essential structure of the data. Similar to Principal Component Analysis (PCA), this technique removes correlations in the dataset, but unlike PCA, this method also removes higher-order dependencies. Many ICA algorithms use PCA to whiten (preprocess) the data. For preliminary results, the scikit-learn package FastICA (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot & Duchesnay, 2011) was used.

Once ICA is implemented, clusters can be found using clustering algorithms. The most widely used algorithm for clustering is K-means clustering. This method aims to partition data into a user-defined number of clusters. The allocation of data into clusters is based on the similarity of the data to the center (or mean) of any given cluster. Preliminary results have shown that, while k-means is an effective method for small amounts of data, the method does not scale well to the large amount of data collected for vehicles. Therefore, the Facebook AI similarity search algorithm (FAISS) is used (Johnson, J., Douze, M., and Jegou, H., 2017). The advantages of FAISS are that it

- provides several similarity search methods
- is optimized for memory usage and speed
- offers a GPU implementation (Johnson et al.)

The goal of the FAISS algorithm is that, given the number of clusters from a user, the algorithm uses a quantizer to allocate each vector (row of the dataset obtained from ICA) to a cluster based on the similarity of the vector to that cluster in a computationally efficient manner. Once each row of the dataset is assigned to a cluster, the fault data can be correlated to clusters time-wise. The groups of clusters that contain faults can be labeled as abnormal operating conditions. Clusters that are not correlated time-wise with the predefined faults are then assumed to be a normal operating condition.

2.1.3. Dynamic Process Monitoring

As previously stated, the US Army ground vehicle sensor data represents a myriad of interconnected and interrelated sensors that help maintainers diagnose and troubleshoot faults. US Army engineers and analysts also use this data to help understand trends for the entire ground vehicle fleet. Most of this sensor data is in a form that can be digested easily; however, a significant portion of the data is noisy, imperfect, and highly variable. After preprocessing the data, it is in a form that allows for further exploration of newly detected trends. These trends are readily observable in historical data, and they can be correlated to maintenance events occurring. Often, these trends also can be detected multiple days before the actual corresponding fault, which aids in automated efforts of the detection of Parameter Based

Indicators (PBIs). This section explores abnormal historical operating conditions further using a supervised learning classification model and then further explored the creation of dynamic alerts.

A critical part of process monitoring is obtaining an empirical model of the system's normal operating condition, subsystem, or sensor to be monitored. One sensor of interest on ground vehicles is the engine oil pressure. For example, to create a model for the normal operating condition of this sensor based on the other sensors in the dataset, statistically significant outliers from the normal dataset should first be removed, Then an XGBoost Regressor (Chen & Guestrin, 2016) model can be trained to predict the normal operation of the engine oil pressure using the gradient boosted trees algorithm, a supervised learning method based on function approximation. The algorithm is trained by optimizing the objective functions over several iterations, t,

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i) + \wp(f_t)) \qquad (3)$$

where y is the real values, $\hat{y}$is the predicted value, $\wp$ is a complexity penalizing term, f corresponds to a tree structure, and l is the differentiable loss function (Chen & Guestrin, 2016). XGBoost is used due to its efficiency and the highly scalable nature of the algorithm.

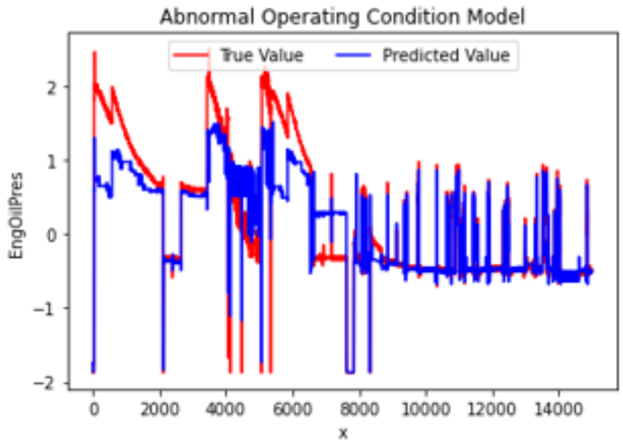

Figure 1. Predicted engine oil pressure from model (blue) and the actual, recorded values of the oil pressure (red). The data is scaled using the scikit learn standard scalar.

From here, data can be projected onto the normal engine oil pressure model to identify anomalies. Points that lie well outside the normal model's prediction can be flagged as alerts to when the oil pressure is outside of the bounds of normal operation based on the readings from the other sensors. This model is a way to programmatically create dynamic Upper Control Limits (UCL) and Lower Control Limits (LCL) for the sensor. Dynamic UCL and LCL for sensors of interest will provide an avenue for future efforts on automating the detection of potential faults. The efforts listed in this section play a part in a combined effort to determine additional CIs, which will be used as a metric for vehicle health and fault prediction.

2.2. Fault Detection and Isolation

Early fault detection is critical to mitigate unnecessary maintenance, facilitate precise logistical maneuvers, reduce vehicle downtime, and increase system health and availability. These predictive models are often physics-based and heavily rely on the expertise of a subject matter expert to understand each component's underlying physics. Physics-based models can be proven beneficial when observing a few components. However, in a multivariate scenario with greater than 50 channels, feature reduction and developing physics-based models for each channel can be time-consuming and susceptible to error. Further, physics informed models require some a priori system information that extends development time and reduces generalization in model creation. In this section, a supervised learning model provides a novel approach to fault detection (Dai & Zhao, 2013). The preliminary results indicate that artificial neural network models, namely Long Short-Term Memory (LSTM) models, provide promising fault detection capabilities for highly dimensional data observed over millions of samples.

The vehicle's operational behaviors, as presented in the 1Hz dataset, coupled with the observed fault data for that particular vehicle over the same timeframe, provided the framework to design a supervised learning model for enhanced fault prediction. We hope to identify a significant number of fault conditions and component failures before they occur, increasing the mission capability and cost savings. Logistical and maintenance decisions will also benefit from insights provided by these models.

2.2.1. Long Short-Term Memory Model

Artificial neural networks (ANNs) are a subset of the supervised machine learning discipline and can process operational data, outputting fault detection and prediction (Helbing & Ritter, 2018). Univariate regression models find functional correlations in the form of coefficients from the input data as an assumed linear relationship (Park, El-Sharkawi, Marks, Atlas, & Damborg, 1991). However, ANNs utilize regression techniques as well as non-linear functionality to model future behaviors of components.

This research effort explores the application of a multivariate Long Short-Term Memory (LSTM) model as a data-driven approach for multivariate topology under an immense sample size of more than a million samples per vehicle. LSTMs employ deep-learning, artificial recurrent neural network techniques (Fu, Huang, Qin, Liang, & Yang, 2018). Deployed on preliminary data, the LSTM model observed multiple channels of sensor data and provided fault detection and diagnosis. The detection levels identify whether the operating conditions could classify as normal operating conditions, represented with a value of 0. Values significantly higher than 0 indicate some level of fault status. The LSTM model further identified the type of numerical fault code that the system would encounter. A fault code is a list of numerical identifiers that range from 0 to 30, which correlate to a particular failure or error for a component within the vehicle system.

The LSTM model, through a series of network layers shown in Figure 1, provided the capability to distinguish normal operational behavior from a fault. It further identified the type of numerical fault code that the system would encounter. A fault code is a list of integer identifiers that range from 0 to 30, which correlate to a particular failure or error for a component within the vehicle system.

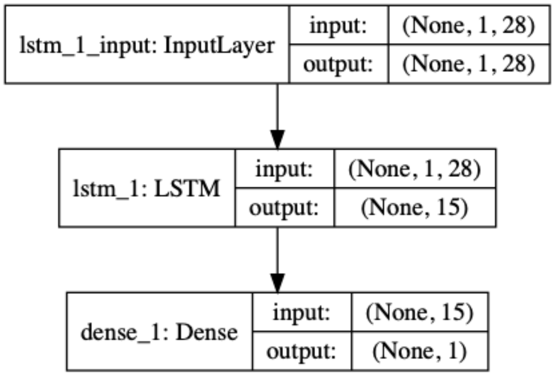

Figure 2. LSTM network layer based on operational data.

The two available datasets were multivariate 1Hz operational data that spanned over one year and the vehicle's corresponding fault data for the same timeframe. The independent datasets, coupled as a supervised learning problem, improved insight for fault detection. Input data for LSTM is filtered 1Hz operational data with reduced channels. Channel reduction decreases dimensionality and mitigates model overfitting. The trained model must observe the behaviors of the channels to recognize faults. However, the raw data, in its entirety, presented heavily saturated normal operational data. The unsupervised anomaly detection described in Section 2.3 allowed for a pre-processing technique in reducing the sample size provided to the LSTM model. Analysis from the unsupervised anomaly detection model identified areas within the data where a fault was likely to occur. The training batch size and epochs were reset, reflecting a more constricted time frame. The LSTM model could then train precisely around the days deemed to present a fault. The separate multivariate fault dataset correlated time-wise to the operational dataset and only contained 13 channels. A single extracted feature reduced the multivariate fault dataset to univariate and consisted of a numerical fault code and its identified timestamp. The fault data's sample size pre-processing techniques included filtering of one fault code presented on any given day.

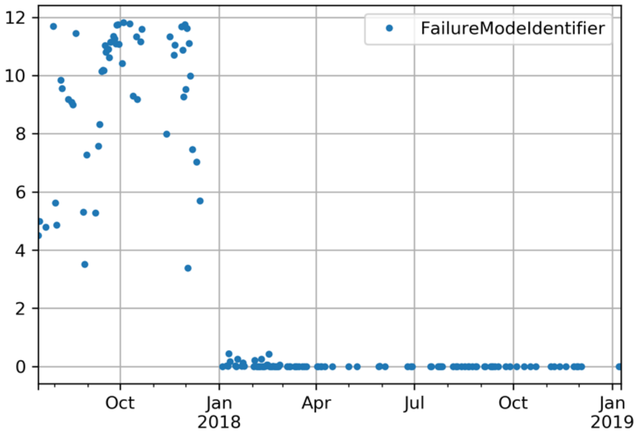

Figure 3. Fault detection model.

The graph shown in Figure 3 depicts the resulting LSTM model output as a preliminary finding. For a single vehicle, the trained LSTM model received only its operational data. The model accurately predicted the days when faults occurred within the actual fault dataset. The model proved successful in learning the correlation between operational data and fault data. Experimentally, LSTM provided an 89% accuracy in fault detection and isolation, detecting and identifying fault code values, including healthy operational behaviors. Future analysis across the more than 4500 vehicle fleet will allow for increased accuracy with the model and ability to identify additional fault codes. In addition to increasing the model sample size across the fleet, future work will further develop thresholds to further classify floating-point fault values toward established fault codes.

2.3. Operational Data Forecasting

The Vector Autoregressive (VAR) model is one of the most successful, flexible, and easy to use models for the analysis of multivariate time series data (Zivot, E. & Wang, J. 2006), so for this work, we use the VAR model for forecasting 1 Hz operational data. To test our method, we are using just one year of data from one vehicle, which contains 17+ million rows and 50+ columns. The entire 1Hz operational data was collected from roughly 4500 vehicles over seven years and will be evaluated later, once the workflow is established. Due to the presence of some sensors sharing stochastic trends, we implemented a multivariate time series VAR to model on our 1 Hz data. This model is then used to generate expected future values of operational data. The patterns and trends of forecasted data our analysis reveals will be used for the condition monitoring and to detect possible anomalies from fault detection model.

2.3.1. Forecasting Operational Data Methodology:

Before implementing the VAR model, we began with structural analysis typically performed using VAR models. These analyses include Pearsons correlation test, Cointegration test, Grangers Causality test and Augmented Dickey-Fuller Test. We employ a Johansen test to validate cointegrating relationships between non-stationary time series data and indicate that our time series data share statistically significant relationships. Granger's Causality Test confirms the relationships of multiple variables in the time series, justifying the VAR modeling approach to forecast 1Hz operational data. Finally, we use KPSS and Augmented Dickey-Fuller Test (ADF Test) to validate the stationary nature of our time series.

Forecast:
We use a Durbin Watson Test to measure the autocorrelation in residuals from regression analysis. Exploratory test results indicate that there is a slight negative correlation autocorrelation problem in the data set. We anticipate that increasing sample size and data cleaning will reduce Durbin-Watson statistics.

Fig. (4) shows the forecast result for 1Hz operational data. For the current sample size, we identified that some sensors favor certain lag orders over the others.

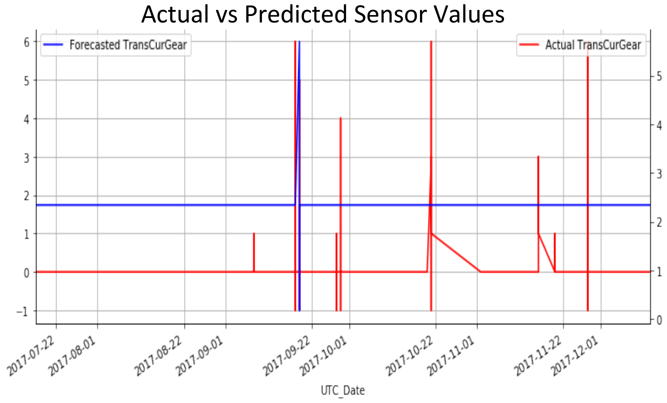

Fig 4. Forecasting Vehicle Operational Data

2.4. Automation

As mentioned previously, the motivation behind a portion of this work is to be able to reduce the time taken to handle unscheduled maintenance events. If a fault can be predicted for a vehicle using noisy datasets for that specific vehicle, the same can be done for other ground vehicles. Fault prediction models and forecasting have the ability to assist in this effort. The time taken to build a fault prediction model or forecast is dependent on the size of the dataset and the ML algorithm used, so this process can be very time consuming. A component to the solution of the overarching goal is to speed up the fault prediction modeling and forecasting process by the automation of the segments from sections 2.2 and 2.3 above. The automation process includes preprocessing the data for manipulation, analyzing the data according to indispensable features, and using the data in machine learning algorithms to produce these fault prediction models and forecasting based on a given dataset. Another aspect to support the automation is the use of HPC. Attainable software exists for high performance computing workflows to interpolate results from one coding script to another. In Fig. (5) below, there is a more detailed version of the automated workflow.

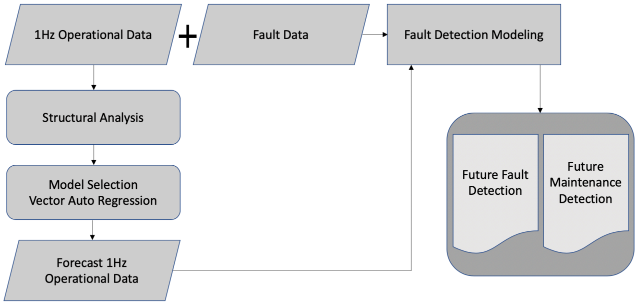

Figure 5. The automated fault detection, isolation, and data forecasting workflow.

The historical operational data fuels both forecasting and fault detection models. The fault detection model, discussed further in Section 2.3, makes use of both operational and fault datasets. Structural analysis and linear regression techniques produce operational forecasts of the vehicle.

An automated HPC workflow can more readily generate models capable of fault detection and maintenance needs across the entire fleet. As shown in Fig. (5), the proposed HPC workflow will align operational forecasting with fault detection models to provide an enhanced model for future fault and maintenance detection. Additional benefits of automation include reduction of human labor, consistency, reproducibility, and increased production of models.

2.5. Vectorization of Maintenance Event Labels

The maintenance text entries used to label fault patterns is free text, entered at the time of the maintenance event and includes details about the fault. The number of issues that can occur within a ground vehicle are many and often very specific. More generally, different fault events fall under larger categories that describe basic features of a fault, such as the component or vehicle subsystem that requires repair. Here we present a method for generating a consistent set of labels for maintenance events in order to pair them with their causal fault patterns. Providing meaningful labels for the text in maintenance logs has the benefit of reducing the number of possible labels for a maintenance event and helps to generalize that label to make it more comprehensible.

We explore different methods of unsupervised text-based pattern recognition. This includes approaches in surveying the text directly to finding frequency patterns within it. After text labels are converted to a numerical format, the set undergoes cluster analysis within the vector space.

The final method of choice for investigation was the use of textual vectorization paired with other unsupervised learning approaches. The motivation for using this method stemmed from the fact that it required no a priori knowledge to create the labels though it does require a priori knowledge to determine the meaning of labels after generation. For each text entry in the set, a process of cleaning and tokenization was undergone in order to prepare it for vectorization. Cleaning the text consisted of removing all irrelevant characters that would not serve in the analysis, including stop words, numerical characters, punctuation, and extra spacing. A process of lemmatization is undergone in order to change given words in the text entries to their basic dictionary form; this minimizes unique yet redundant tokens. After cleaning and lemmatization, the words are split by spacing in order to create a token list for use in the vectorization algorithm. The ordering of the tokens is maintained for use with this algorithm.

The vectorization algorithm used in this labeling method is the Doc2Vec algorithm (Le & Mikolov, 2014), which was developed from the Word2Vec algorithm (Mikolov, Chen, Corrado, & Dean, 2013) for use with vectorizing groups of tokens as one vector in a vector space. We used the entire collection of maintenance fault descriptions under the assumption that each one would be representative of a universal fault within ground vehicles independent of vehicle types in order to create a general vector space where each individual entry in the maintenance dataset maps to a point in the space. Multiple iterations of this process with Doc2Vec were undergone using different vector sizes in assigning vectors to text entries. This ranged between vector sizes of two to three hundred.

After vectorization, the cumulative vector space is analyzed using clustering methods. The method of clustering used in particular was DBSCAN (Ester, Kriegel, Sander, & Xu, 1996), which takes into account the existence of outliers in the dataset. Outliers would be representative of faults that are infrequent and do not have to do with major components of a ground vehicle. The DBSCAN clustering algorithm also determines the number of clusters within the vector space analyzed, which further reduces the amount of a priori knowledge needed to find labels in the maintenance text entries.

Clustering finds groups of spatially related points. For a point in the set, the cluster it belongs to determines its label. If the point is an outlier, its label is classified as an outlier. This structure can also help to determine new points by seeing how they fall within existing clusters. This process undergoes multiple iterations, with different parameters denoting the minimum distance a point needs to be from a core cluster point and how many points are required to form a cluster.

The resulting set of possible clusters can be analyzed for quality based on evaluation heuristics such as the Silhouette Score (Rousseeuw, 1987), along with specific evaluation based on number of clusters and the comprehension of points within each cluster. Determining what each cluster represents is done by looking at the points in each cluster and their associated text entries. This helps to determine labels for each original point in the maintenance text entries using the label for each cluster.

3. Conclusions and Future Work

We began the hybrid learning workflow for PHM on US army ground vehicles in order to generalize and deepen progress made in the area of physics informed linear modelling. These models work extremely well, but they require some a priori knowledge of the system. To expand the scope and speed of creating these models, we are exploring the possibilities of a more data-driven approach that leverages both the accrued sensor data from the vehicle CAN bus systems and the corresponding maintenance data. The approaches outlined in section 2 have combined to produce promising results when applied to relatively small data. When leveraged against the large datasets that will be considered in the next phase of the project, these cursory results should show significant improvement.

We have succeeded in manifold and cluster creation, corresponding to healthy versus unhealthy operation of the vehicle considered, as explained in section 2.1. These clusters represent potential Parameter-Based Indicators. Similarly, the supervised approaches outlined in sections 2.2 and 2.3, and joined into a workflow in section 2.4, find similar candidates for PBI status. What each approach is missing, with the exception of the small subset of Fault data correlation, is consistent labelling. This is addressed in section 2.5, where we vectorize the maintenance data accompanying the operational data, yielding a consistent label set.

Beyond knitting these solutions together, we will expand the data into High Performance Computing workflows on DoD supercomputing resources. A time-sharded database of all operational data for every vehicle, composed of over one million days of operation across approximately 4,500 vehicles, is online and ready for fleet wide querying of operational data. This database will fill dataframes for ML workflows on HPC in negligible time, making the memory-loaded data ready for parallel and multi-processing workflows.

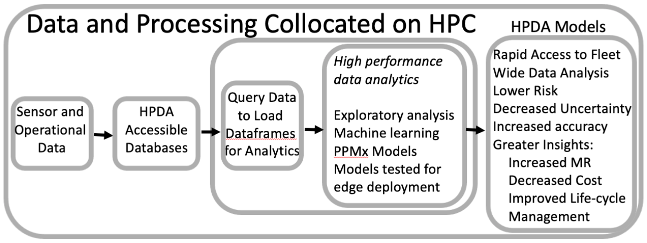
Data and Processing Collocated on HPC
HPDA Models
Sensor and Operational Data
HPDA Accessible Databases
Query Data to Load Dataframes for Analytics
High performance data analytics
Exploratory analysis
Machine learning
PPMx Models
Models tested for edge deployment
Rapid Access to Fleet Wide Data Analysis
Lower Risk
Decreased Uncertainty
Increased accuracy
Greater Insights:
Increased MR
Decreased Cost
Improved Life-cycle Management

Figure 6: Massive ground vehicle data is ingested into a constantly available, time-series data optimized database where it can be queried to load dataframes for a large variety of available HPDA tools for exploratory data analysis, ML work, and workflow management, resulting in a large number of HPDA models to facilitate decision making at all levels of life cycle management.

This system will support the collaboration of the tools from section 2, combined through workflow software, for iterative processes on HPC such as hyperparameter evaluation, where one workflow considers several parameter sets, yielding all results for visualization and refinement. While we received promising results from each of the above efforts, every one of them will benefit from both the ability to greatly expand the data considered for training and testing models, and from collaboration between themselves, producing more robust, more certain models. In particular, entire vehicle types can be explored for fault and candidate PBI intervals using several clustering and machine learning strategies in concert. The CPBIs can then be correlated with labels whose certainty is improved through integration with these methods and continued refinement. CPBIs can then be tested with a large amount of testing data that already exists on the HPC platform, using the successes of their predecessors, the physics-based linear models, as benchmark.

References:

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp 785-794), August 13-17, San Francisco, CA. doi: 10.1145/2939672.2939785

Dai, X. and Gao, Z. From Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis, *IEEE Transactions on Industrial Informatics*, vol. 9, (no. 4), pp. 2226-2238, Nov. 2013, doi: 10.1109/TII.2013.2243743.

Ester, M., Kriegel, H., Sander, J., Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *2nd International Conference on Knowledge Discovery and Data Mining* (226-231), August 2-4, Portland, Oregon. doi: 10.5555/3001460.3001507

Fu, Y., Huang, D., Qin, N., Liang, K. and Yang, Y. (2018). High-Speed Railway Bogie Fault Diagnosis Using LSTM Neural Network, *37th Chinese Control Conference* (pp. 5848-5852), Wuhan, Hubei, China. doi: 10.23919/ChiCC.2018.8483987.

Helbing, G. & Ritter, M. (2018). Deep Learning for fault detection in wind turbines, *Renewable and Sustainable Energy Reviews*, Volume 98, Pages 189-198, ISSN 1364-0321, doi: 10.1016/j.rser.2018.09.012.

Jelali M. (2013) Statistical Process Control. Control Performance Management in Industrial Automation. Part of the series Advances in Industrial Control. Pp. 209-217. Springer, London. doi: 10.1007/978-1-4471-4546-2_8

Johnson, J., Douze, M., and Jegou, H. (2017) Billion-scale similarity search with GPUs. arXiv e-print. arXiv:1702.08734

Le, Q. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. arXiv e-print. arXiv:1405.4053v2

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv e-print. arXiv:1301.3781

Park, D.C., El-Sharkawi, M.A., Marks, R.J., Atlas, L.E. and Damborg, M.J. Electric load forecasting using an artificial neural network, *IEEE Transactions on Power Systems*, vol. 6, (no. 2), pp. 442-449. doi: 10.1109/59.76685.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011) Scikit-learn: Machine Learning in Python., *Journal of Machine Learning Research* vol. 12, pp. 2825-2830. doi: 10.5555/1953048.2078195

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65. doi: 10.1016/0377-0427(87)90125-7

Schimert, J. (2008). Data-Driven Fault Detection Based on Process Monitoring using Dimension Reduction Techniques. *2008 IEEE Aerospace Conference.* (pp. 1-12), March 1-8, Big Sky, Montana. doi: 10.1109/AERO.2008.4526621.

Vogl, G.W., Weiss, B.A. and Helu, M. (2019). A review of diagnostic and prognostic capabilities and best practices for manufacturing. *Journal of Intelligent Manufacturing, vol.* 30, pp. 79–95. doi: 10.1007/s10845-016-1228-8

Zivot, E. and Wang, J. (2006) Vector Autoregressive Models for Multivariate Time Series. *Modeling Financial Time Series with S-PLUS®*. Springer, New York, NY doi: 10.1007/978-0-387-32348-0_11