Case Study: Models for Detecting Low Oil Pressure Anomalies on Commercial Vehicles

Howard E. Bussey[1], Nenad G. Nenadic[2], Paul A. Ardis[3], Michael G. Thurston[2]

[1] *Tixlers Letters, PO BOX 831, Pittsford NY 14534*
howard@tixlers.com
[2] *Golisano Institute of Sustainability, Rochester Institute of Technology, Rochester, NY 14623 USA*
nxnasp@rit.edu mgtasp@rit.edu
[3] *GE Global Research,1 Research Circle Bldg. K1-4A6, Niskayuna, NY 12309*
ardis.p@ge.com

Abstract

We present a case study of anomaly detection using commercial vehicle data (from a single vehicle collected over a six-month interval) and propose a failure-event analysis. Our analysis allows performance comparison of anomaly detection models in the absence of sufficient anomalies to compute the Receiver Operating Characteristic curve.

Several heuristically-guided data-driven models were considered to capture the relationship among three main engine signals (oil pressure, temperature, and speed). These models include regression-based approaches and distance-based approaches; the former use the residual's z-score as the detection metric, while the latter use a Mahalanobis distance or similar measure as the metric. The selected regression-based models (Boosted Regression Trees, Feed-Forward Neural Networks, and Gridded Regression tables) outperformed the selected distance-based approaches (Gaussian Mixtures and Replicator Neural Networks). Both groups of models were superior to existing Diagnostic Trouble Codes. The Gridded Regression tables and Boosted Regression Trees exhibited the best overall metric performance.

We report a surprising behavior of one of the models: locally-optimal Gaussian Mixture Models often had zero detection performance, with such models occurring in at least 25% of the iterations with seven or more Gaussians in the mixture. To overcome the problem, we propose a regularization method that employs a heuristic filter for rejecting Gaussian Mixtures with non-discriminative components.

1. Introduction and background

Equipment health and condition monitoring enables maintenance to minimize the effects of equipment degradation or failure. Building on existing concepts for predictive maintenance, Reliability Centered Maintenance (RCM) (Nowlan & Heap, 1978) provided a formalism for Condition-Based Maintenance (CBM). Being based upon objective evidence of equipment degradation or impending failure, CBM has significant economic and safety benefits: it reduces incidence of unscheduled failures and downtime and reduces occurrence of unnecessary or early scheduled maintenance.

Health or condition monitoring is the process of collecting asset data, extracts the information and provides it to CBM. Affordable sensors, data storage, and networking enable comprehensive monitoring of all types of assets. In order to make this data actionable for CBM, models are needed to identify and characterize anomalies, and then to relate the anomalous patterns to forward looking failure risk for decision making purposes (prognostics). The models are typically classified as expert-system, physics-based, data-driven, and hybrid. This paper takes the data-driven modeling approach.

Health monitoring is generally an incremental (not all-at-once) process, as data is typically not available to develop comprehensive diagnostic and prognostic algorithms from the outset (Sikorska, Hodkiewicz, & Ma, 2011). Most modern vehicles are equipped by the original equipment manufacturer with built-in sensors on a data bus, and diagnostic systems that detect major drive train failures. The diagnostic coverage on these systems can be limited, and they typically detect problems with limited warning horizon before maintenance action is required. Telematics systems, such as General Motors OnStar$^{\text{TM}}$are increasingly being used to monitor private, commercial, and military vehicles. Data provided by these systems, over a large fleet of vehicles, can be used to develop new anomaly detection and failure prediction al-

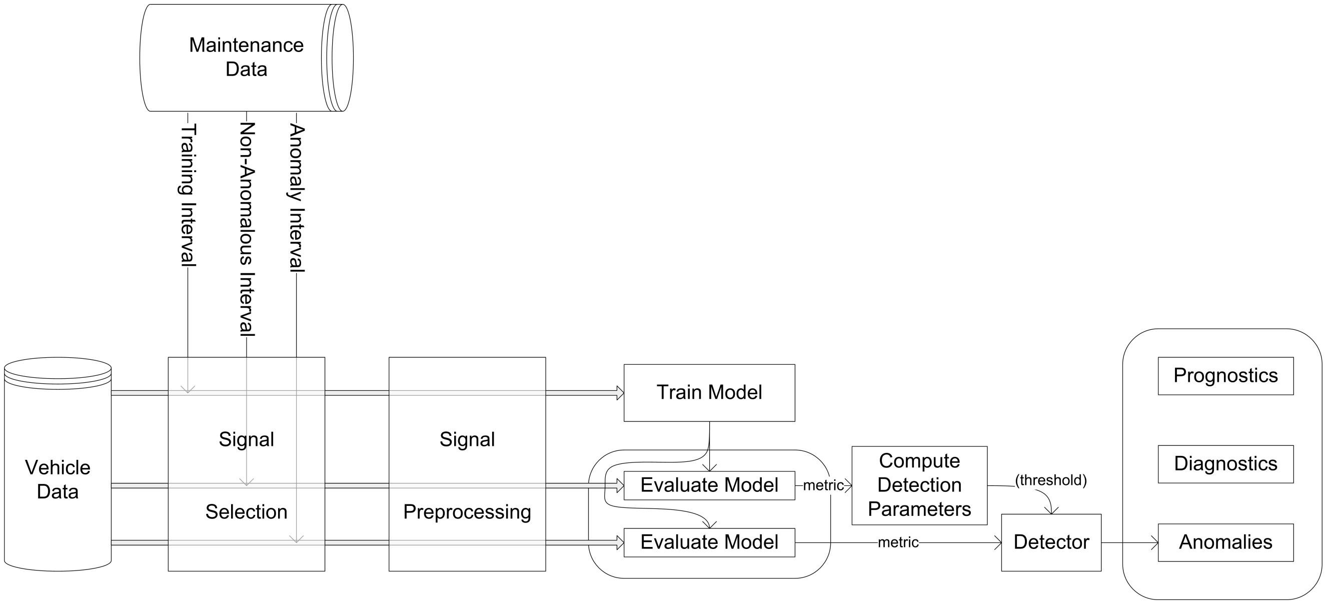


Figure 1. Analysis process, showing steps of building the model, detection anomalies, diagnosing faults, and predicting future failures (prognostics).

gorithms more cost effectively than through traditional engineering testing. On-board computers, coupled to the vehicle data bus, can filter vehicle data and run algorithms locally, or they can relay data to a back-end system for processing. These systems can also support cost effective addition of vehicle sensors to augment existing capabilities. In addition to driver services and logistics support, these systems are used to collect information to support product improvement, and have growing levels of Prognostic Health Management (PHM) capability.

Consolidation of vehicle fleet data in a data warehouse provides an opportunity to develop CBM knowledge and algorithms incrementally. As failures occur within the fleet, the vehicle and maintenance data can be correlated, analyzed, and used to create autonomous health monitoring agents with embedded anomaly detection, diagnostics, and prognostics. With larger fleets, more accurate and extensive algorithm sets can be developed. Our approach is opportunistic, based upon the failures, and data-driven, exchanging data mining and statistical machine learning in place of in-depth expert knowledge.

As shown in Figure 1, anomaly detection is the first layer of information extraction in condition-based maintenance. The ability to reliably detect system performance changes, in the context of different operating and environmental conditions, is the first step towards condition monitoring. The value of anomaly detection is the ability to trigger useful alerts and to pave way to more sophisticated PHM. In the context of truck fleet operations, an anomaly warning can be provided to maintenance or operational supervisors to prompt them to review the condition of the truck or the behavior of the driver.

Observed anomalies and their links to the associated failure modes (established by maintainers) form a labeled data set suitable for supervised machine learning. Automated classification of observed anomalies enables the second level of PHM – diagnostics. Using observations of operational failures for classification training is well suited for environments where failures can be, or have historically been, tolerated; this approach is cost effective and requires no additional risk. In particular, the present case study is concerned with health monitoring of commercial truck fleets, where failures can be very costly, but are tolerated as a part of doing business. The variant of this approach, in which unsupervised anomaly detection identifies candidate events for human expert analysis, may be suitable for systems such as nuclear reactors where system failures are unacceptable. In this case, the data-driven approach would augment the physics- or expert-knowledge-based systems presently in use. This paper focuses on the development of a methodology for anomaly detection in truck engine behavior using data captured from a commercial fleet telematics system. To achieve this capability, we use data-driven models, each with an intrinsic metric. We will describe five such models, motivate their choices, and compare their performance in following sections.

Once anomaly detection is in place, additional observed failures can be used to improve anomaly detection algorithms and parameters, as well as to develop diagnostics and prognostics. The development of data-driven prognostics is enabled (and improved) by more examples of the same failure mode, which allow for the development of models of the progression of failures subject to operational and environmental context (regression, tracking). Alternatively, correctly classified anomalies with accurate physics-based (or other expert knowledge-based) models can be considered without requiring a large number of examples. Data-driven diagnostic development is enabled by examples of a variety of distinct failure modes; from a machine learning viewpoint, diagnostics can be perceived as a discrete classification problem. Since the available data have only one failure, we were unable to address the diagnostic and prognostic areas.

Building a system for anomaly detection includes the following three steps: 1) selecting and pre-processing the relevant signals; 2) selecting, building, and tuning a model equipped with a metric; and 3) selecting and tuning an inference engine that indicates anomalies, based upon the model metric. While design, parameterization and parameter tuning of all three blocks impact the performance of the system, this report focuses on model selection and tuning. In all cases the models operate on the same three signals: engine oil temperature, engine oil pressure, and engine speed. Moreover, all systems discussed in this paper employ a simple inference engine a low-pass filter followed by a comparator. When the filtered metric exceeds the threshold, the signals are considered anomalous.

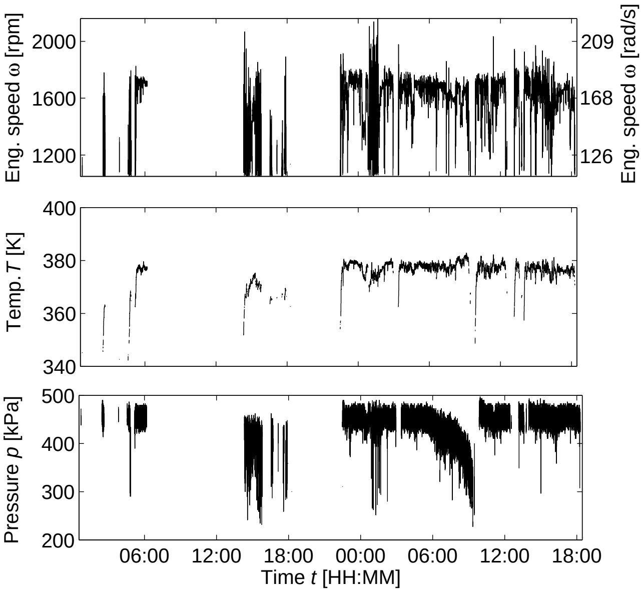


Figure 2. Example engine speed, oil temperature, and pressure

Chandola, Banerjee, and Kumar (2009) survey anomaly detection techniques, touching on methods used here. Our work falls under their industrial damage classification, for which they report on work using parametric and non-parametric statistical modeling, Neural Networks, spectral, and rule-based systems. Bishop (2006) describes these machine learning techniques in further detail, including specifics of training and testing that are used in our work. Vachtsevanos, Lewis, Roemer, Hess, and Wu (2006) present a somewhat different model for data-driven anomaly detection (fault or failure detection in their terminology - see their section 5.2.3). The literature reporting anomaly detection results using standard vehicle data over long periods is sparse. Golosinski, Hu, and Elias (2001) report on 1.2 hours of data from a single vehicle. Kargupta et al. (2004) report on analysis based upon a vehicle simulator. McArthur, Booth, McDonald, and McFadyen (2005) report on a processing system using data from a single engine. Cheifetz, Same, Aknin, and de Verdalle (2011) report on data from 22 consecutive operating cycles of a commercial bus. Our experiments are intended to provide further empirical insight, especially with regard to longer performance periods and the specifics of model construction.

The study data include a period during which the vehicle was driven with an active oil leak. We employed an opportunistic data-driven methodology in our analysis. Because we have only one labeled failure event in the data, we: (a) create several models from the training data; (b) for each model, find the minimum threshold that results in a zero false-alarm rate during the normal period; (c) measure detection performance during the low-oil period using the models and their respective detection threshold values. For this failure, we have approximately 144 hours of training data from a two-week interval, failure data representing about 15 hours of operation during approximately 19 clock hours, and the normal period of five months (1500 hours) following repair.

2. PROBLEM AND PROCESS

Figure 2 shows a segment of the vehicle data: engine speed and oil temperature and pressure, recorded over a two-day period during which the vehicle was operated with an oil leak. The data show the vehicle operating with steadily declining oil pressure starting between 5:30 and 6:00 AM. With this rich contextual information, one can conclude that the pressure is legitimately anomalous. However, if only the pressure information is available, the most one can say is that the pressure exhibits a downward trend. For this fault, anomaly detection based upon only the oil-pressure is insufficient. The manufacture recommends pressures of at least 150 kPa when the engine is idling, and at least 300 kPa when the engine speed is greater than 1100 RPM. If anomaly detection used only the idle condition minimum pressure, the anomaly would be missed in its entirety. Using the higher limit, the anomaly is detected only in the last few minutes, and might cause false alarms if applied when the engine is idling. Some anomaly detection algorithms use a mode-based approach, where the operating modes and associated signal limits are defined a priori and used to identify anomalous operations. Based on the rules presented above, a mode-based oil pressure anomaly detector would identify anomalies sometime after 9 AM on the

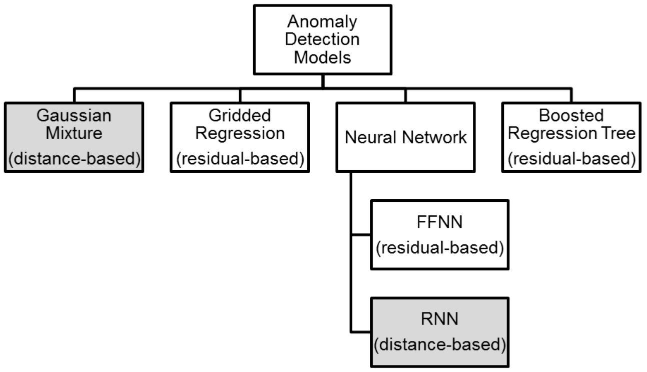

Anomaly Detection Models
Gaussian Mixture (distance-based)
Gridded Regression (residual-based)
Neural Network
Boosted Regression Tree (residual-based)
FFNN (residual-based)
RNN (distance-based)

Figure 3. Anomaly detection approaches in this investigation

second day of second day of operation.

Figure 3 maps the five models this study explored: three residual-based models - Gridded Regression table (GR), Boosted Regression Tree (BRT), and Feed-Forward Neural Network (FFNN) - and two distance-based models - Gaussian Mixture (GMM) and Replicator Neural Network (RNN). In the residual-based systems, the models predict the pressure, based upon temperature and engine speed. The metric is the absolute value of the z-score (*the standard score*) of the residual, where the residual mean and variance are determined from the model and training data. In the distance-based systems, the metric reflects how different all three signals are from the model.

2.1. Data Source and Preparation

As indicated in Figure 1, signal preprocessing is often necessary before the data is used for building models. The preprocessing here includes filtering out irrelevant data (e.g. during idling), removal of short-duration transient data, eliminating non-informative data (e.g., if some data is missing), and excluding data segments so short they cannot be handled in subsequent processing (e.g., a 20 s drive between two 5 minute idle periods).

We use data from a commercial truck (including both maintenance data and operational data from the vehicles' data buses) as provided to RIT by Vnomics Corp. Examination of the maintenance data showed that there was one oil leak event; that single event is used as the fault event for this study. The vehicle data were obtained from J1587 and J1939 packets available on the J1708 and CAN buses on heavy-duty trucks. This data did not include oil level information, even though that signal is defined in the J1939-71 and J1587 specifications. The Vnomics' Vehicle Health Management Software (Vnomics, 2012) collected the asynchronous on-board signals and used lossy data compression to save space in the database. The compression algorithm compares the current signal value to the last stored data value and stores the current signal value if the difference exceeds a fixed threshold. The thresholds are provided in Table 1.

Table 1. Thresholds used in data compression algorithm.

Signal	Threshold
Oil Temperature	0.2 C/K
Engine Speed	10 RPM
Oil Pressure	6.89 kPa

For this investigation, the asynchronous signal values are read from database and time-synchronized to a 1 s periodic stream using sample and hold interpolation. In addition to synchronization, some data are removed. For instance, we remove low-RPM (idle) data so that it isn't over-emphasized during training. There are two irrelevant data removal schema, as show in Table 2. In schema 1, a wide range of physically-feasible engine oil temperatures are accepted. In schema 2, the temperature range is narrower to exclude data collected while the engine is warming up.

Table 2. Data Removal Schema.

Schema	Signal	Minimum (inclusive)	Maximum (exclusive)
1	Temperature	-20	120
	RPM	1050	2500
	Pressure	50	550
2	Temperature	90	120
	RPM	1050	2500
	Pressure	50	550

The training interval was selected after inspection of the operational and maintenance to find the first period with no maintenance events and no obvious data anomalies. For this vehicle, that was immediately following a stuck at high oil pressure sensor fault. The selected training period, with approximately 142 hours of operational data, is the two weeks following replacement of the sensor. After removing irrelevant data, there remain 75 hours of training data. The non-anomalous period follows the repair of the oil leak. The anomalous period is a two-day period starting 2/2/2010.[1]

2.2. Metric Filtering

For all of these models, the metric is filtered with an infinite impulse response low-pass filter with low passband frequency of 0.0017 Hz (1/600 Hz) and reject-band frequency of 0.05 Hz. These values were chosen to provide a filter time-constant of 5 minutes. This filter is appropriate for detecting anomalies related to a slow oil leak.

In addition to the low-pass filtering, the metric filtering must deal with data gaps introduced by the irrelevant data removal step described in 2.1. In addition, short segments (e.g. 60 s) are statistically insignificant when a fault event evolves over a period of an hour or longer; because they cause numerical instability, we removed them. Finally, the filter used above is applied on the remaining segments on a segment-by-segment

[1] To encourage further research in this area, we have made the data available: http://www.rit.edu/gis/research-centers/csm/EOP_Case_Study.php. This has irrelevant data discarded according to schema 1.

basis. This filter exhibits some ringing, so to prevent high amplitude ringing, the filter is initialized with 10,000 s of input points equal to the median of the first 50 samples in the segment.

Because our goal is to study performance of several system models, the same data preparation and detection processing steps are used for all of the models.

2.3. General Modeling Process

For a problem of this type, the inputs consist of n observed signals $S_1, S_2, \ldots, S_n$. Data is divided into training $D_{training}$, event D_{event}, and normal D_{normal} sets, such that the sets are subsets of $\mathbb{R}^n$:

$$D_{training}, D_{event}, D_{normal} \in \mathbb{R}^n \tag{1}$$

and the sets are disjunct

$$\begin{aligned} D_{training} \cap D_{event} &= \varnothing \\ D_{training} \cap D_{normal} &= \varnothing \\ D_{normal} \cap D_{event} &= \varnothing \end{aligned} \tag{2}$$

The modeling is the process of identifying parameters of a model $\mathcal{M}$ and detection threshold Θ, given metric m, that maximizes discriminability between the training and event data:

$$\max |m(\mathcal{M}(D_{training}), \mathcal{M}(D_{event})) > \Theta| \tag{3}$$

subject to zero false alarms

$$|m(\mathcal{M}(D_{training}), \mathcal{M}(D_{normal})) > \Theta| = 0 \tag{4}$$

Overall, our goal is to provide a long and stable detection horizon for known faults, subject to the requirement that there are no anomalies detected during the normal interval (false alarms). As a final note, we prefer low-complexity models that use zero expert system knowledge and have short training times.

All five models, described in the next section, were able to detect anomalies on the first day of the low-oil event, which took place approximately 19 hours before the last mission during the low oil period. Analyzing these anomalies showed transient pressure drops when the engine speed briefly increased to a range between1500 and 2000 RPM. Figure 4a, from the training interval, shows a small pressure variation, approximately 50 kPa, with no clear pattern of increasing or decreasing. Figure 4b shows the data from one of the anomalous intervals. Here the pressure drops approximately 100 kPa as the engine speed increases from 1500 to 2000 RPM. In both cases, the pressure is above 400 kPa when the engine speed is steady around 1500 RPM.

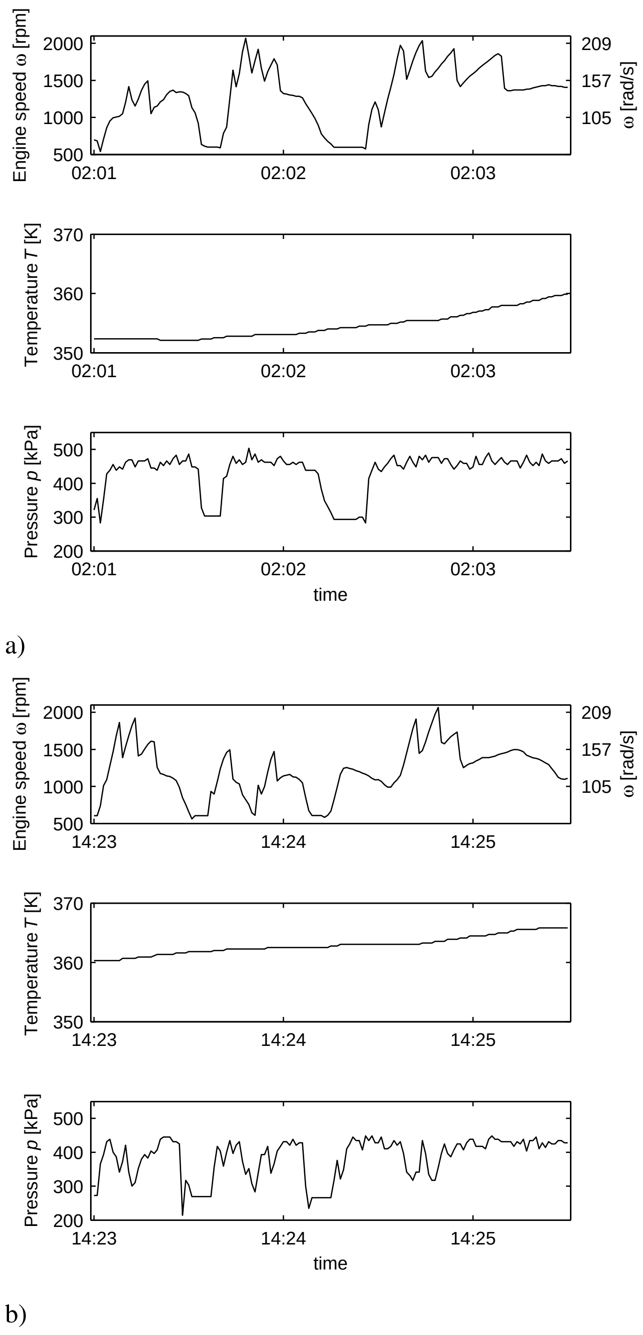


Figure 4. (a) Signals on the first training day (1/14/2010) showing the normal behavior where engine speed spikes make little change in the oil pressure. (b) Anomalous signals at 14:23 on first day of low oil event (2/2/2010), where the pressure drops to approximately 325 kPa when the engine speed increases sharply from 1500 RPM to 2000 RPM - once just after 14:23, and again just before 14:25.

3. MODELS' DESCRIPTIONS AND PERFORMANCES

This section describes the five models in turn, with the application-specific decision processes associated with the models and their performance.

3.1. Model 1 – Gridded Regression

The Gridded Regression (GR) model has a look-up table used to estimate engine oil pressure p as a function of engine speed ω and engine oil temperature T; and the residual mean and variance, used to calculate the z-score metric. Here, the domain, the temperature-speed (ω-T) plane, is subdivided into rectangular subdomains, or bins, as depicted in Figure 5a. The temperature and speed ranges are determined a priori, based on the expected ranges of the signals; consequently, some of the bins are empty during training. The discrete pressure estimates $\hat{p}$ over the domain are given by

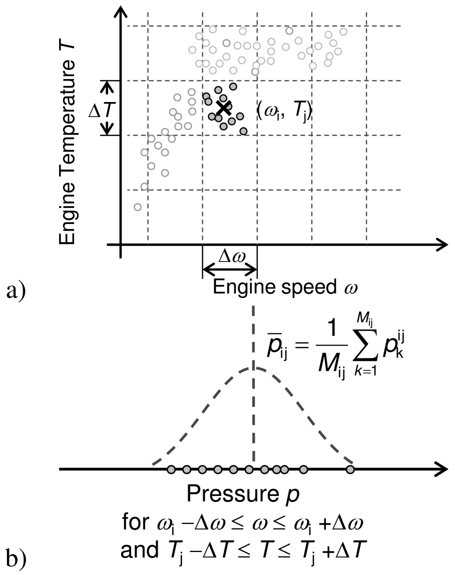

Engine Temperature T
ΔT
(ω_i, T_j)
Δω
Engine speed ω
a)
p̄_ij = 1/M_ij Σ_{k=1}^{M_ij} p_k^ij
Pressure p
for ω_i −Δω ≤ ω ≤ ω_i +Δω
and T_j −ΔT ≤ T ≤ T_j +ΔT
b)

Figure 5. A sketch of GR model. (a) Discretized (ω-T) plane. Data points within (ω_i-Tj) bins are highlighted. (b) Mean pressure of the data.

$$\hat{p} = f(\omega, T) = \overline{p}_{ij} \tag{5}$$

where f is the point sample of a 2-dimensional Gaussian distribution in terms of ω and T. $\overline{p}_{ij}$ is the mean pressure of the training data corresponding to (ω_i-T_j) subdomain bounded by $\omega_i - \Delta\omega/2 \leq \omega < \omega_i + \Delta\omega/2$ and $T_j - \Delta T/2 \leq T < T_j + \Delta T/2$ (see Figure 5b), as in:

$$\overline{p}_{ij} = \frac{1}{M_{ij}} \sum_{k=1}^{M_{ij}} p_k^{ij} \tag{6}$$

Another way to think of this model is a piece-wise constant (in this case two-dimensional) fit function with error bars. In the metric evaluation operations, subtracting estimates from the measurements yields error $\varepsilon_p = p - \hat{p} = p - \overline{p}_{ij}$. The residuals are considered collectively, over all bins. The metric used for detecting anomalies is the absolute value of the z-score of the residuals, computed as:

$$m = |z_p| = \left| \frac{\varepsilon_p - \overline{\varepsilon_p}}{\sigma_p} \right| \tag{7}$$

Figure 6a shows that the Gaussian distribution fits the residual data, $\varepsilon_p \sim N(0, \sigma_p^2)$, reasonably well. Figure 6b quantifies this fit further, showing that 99.8% of the residuals match the expected range from -25 to +25.

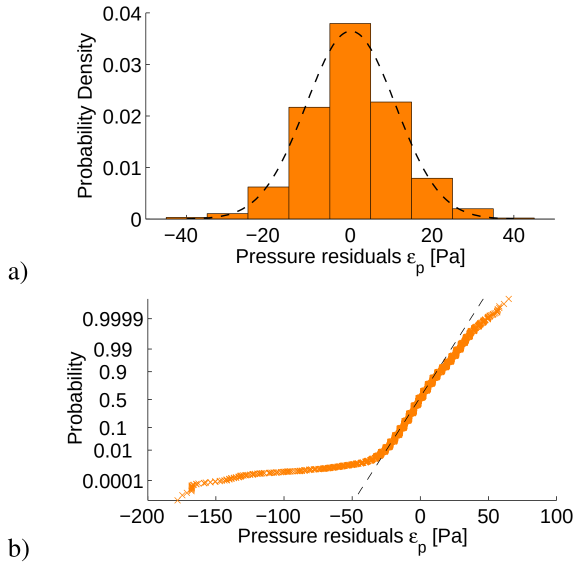


Figure 6. Distribution of the 75 hours of training data. (a) Histogram with a fit. (b) Test of normal data.

3.1.1. GR: Parameters and Performance

The oil temperature and RPM ranges were divided into 10 equal intervals, resulting in a 10x10 grid. The model estimate for each bin in the grid is the mean oil pressure for the data samples in that bin. If the count of data in the bin was too low, the model estimate for that bin was NaN (not-a-number) a flag value causing that bin to be effectively ignored in the rest of the experiment. The residuals were computed over all of the training data, and the histogram of the residuals in Figure 6 shows the distribution is well-modeled by a Gaussian distribution. The variance of the residuals is computed and stored with the model, to be used in subsequent z-score calculations. For each data point in the test and non-anomalous intervals, the GR model is used to predict the oil pressure, based upon the RPM and oil temperature. The metric is the absolute value of the z-score of the residual. The metric is smoothed by the low-pass filter described in Section 2.2. For the non-anomalous interval, the smoothed metric value is used to determine the detection threshold, guaranteeing the no false alarm criterion. That threshold is compared with the smoothed metric for the test interval, and the results are shown in Figure 7. The anomalies between 15:00 and

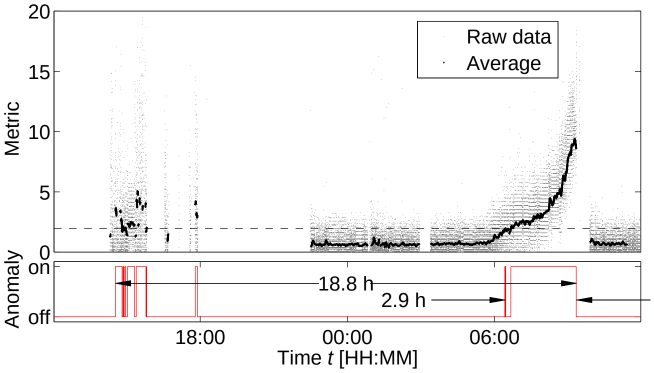


Figure 7. Performance of GR Model. Detection horizon is about 2.9 hours.

18:00 are correlated with vehicle oil level and pressure vehicle Diagnostic Trouble Codes (DTCs) recorded at 14:15 and 15:38; however, they are not included in the detection horizon calculation, which is based upon the period between 22:35 on day 1 and 09:21 on day 2 of these data. This narrower time range is used because the vehicle operators, aware of the oil leak, added oil from time to time in this period. However, the period from 22:35 until 09:21 the next morning, as Figure 2 shows, represents a single event when the oil pressure dropped from normal to abnormally low.

Tuning this model requires selection of the number of bins for temperature and engine speed. The number we used represents a compromise between too few bins, which would increase the prediction error, and too many bins, which would result in too few training points per bin. Given the bin count selection, training is deterministic for a given training data set.

The selection of 10 bins was based on trial and error in this study. Optimal or near-optimal bin counts could be selected through either exhaustive or random exploration of the bin count space for each independent variable.

3.2. Model 2 – Gaussian Mixtures

Model 2 is an automatically trained GMM comprising a set of multivariate normal distributions, $\boldsymbol{N}_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, and their weights π_k where $\sum \pi_k = 1$. The distributions, $\boldsymbol{N}_k$, are trained to maximize the generative likelihood of all points (T_t, ω_t, p_t) in the training data. The metric used in this model is the likeliest Mahalanobis distance (Duda, Hart, & Stork, 2000), which is the Mahalanobis distance to the mean of the Gaussian G_k that maximizes $\mathcal{P}_t = pr_k(T_t, \omega_t, p_t) \cdot \pi_k$.

Two variants of the model were considered: one (schema 1) explored wide temperature range and the other (schema 2) was restricted in a narrower temperature range.

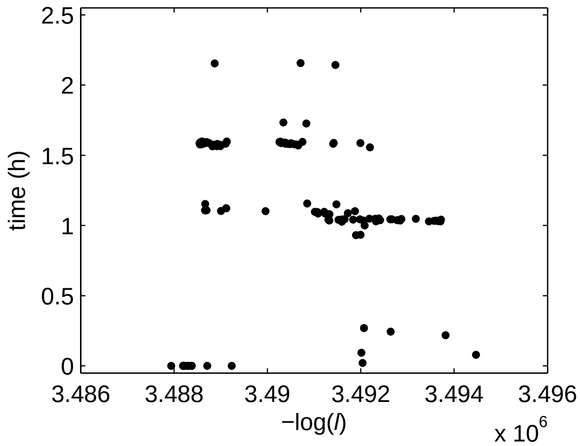


Figure 8. Performance of GMM(7)s. Each dot on the figure represents one trained GMM(7). The models with the better likelihood generally have better detection performance, although the models with the best likelihood have zero detection performance.

3.2.1. GMM: Parameters and Performance

The modeled employed seven fitted Gaussian distribution mixture components. The number of components was determined heuristically by searching the parameter space between one and 15 Gaussian components in the GMM: mixtures with less than seven components exhibited shorter detection horizon, while mixtures with more components showed no consistent advantage in detection horizon, and sometimes resulted in a large proportion of models with zero detection performance. Candidate GMMs were trained with Matlab® using the `gmdistribution.fit()` method. This uses an expectation maximization algorithm to find locally optimal models meeting hard-coded convergence criteria.

Initial experiments showed inconsistent performance with detection horizons ranging from 0 to 2.2 hours (see Figure 8). The cause for this is explained in section 3.2.2. The results shown in this section use models trained with the combined expectation maximization and rejection criterion filter. The metric performances similar to the one in Figure 7, and are not repeated for each of the models for brevity.

Changing the irrelevant data removal to schema 2 and rerunning the same experiment resulted in no performance improvement, showing that the GMM training and rejection filtering process is robust in that the detection horizon is the same for two different temperature ranges. While the horizons are the same (see the GMM(7) schema 1 and schema 2 results in the figure), the schema 2 results, based upon data in a narrower temperature range, show less variation at the onset of detection (07:40) on the second day.

Training required repeated creation of GMMs from different random subsets of the training data, with selection of

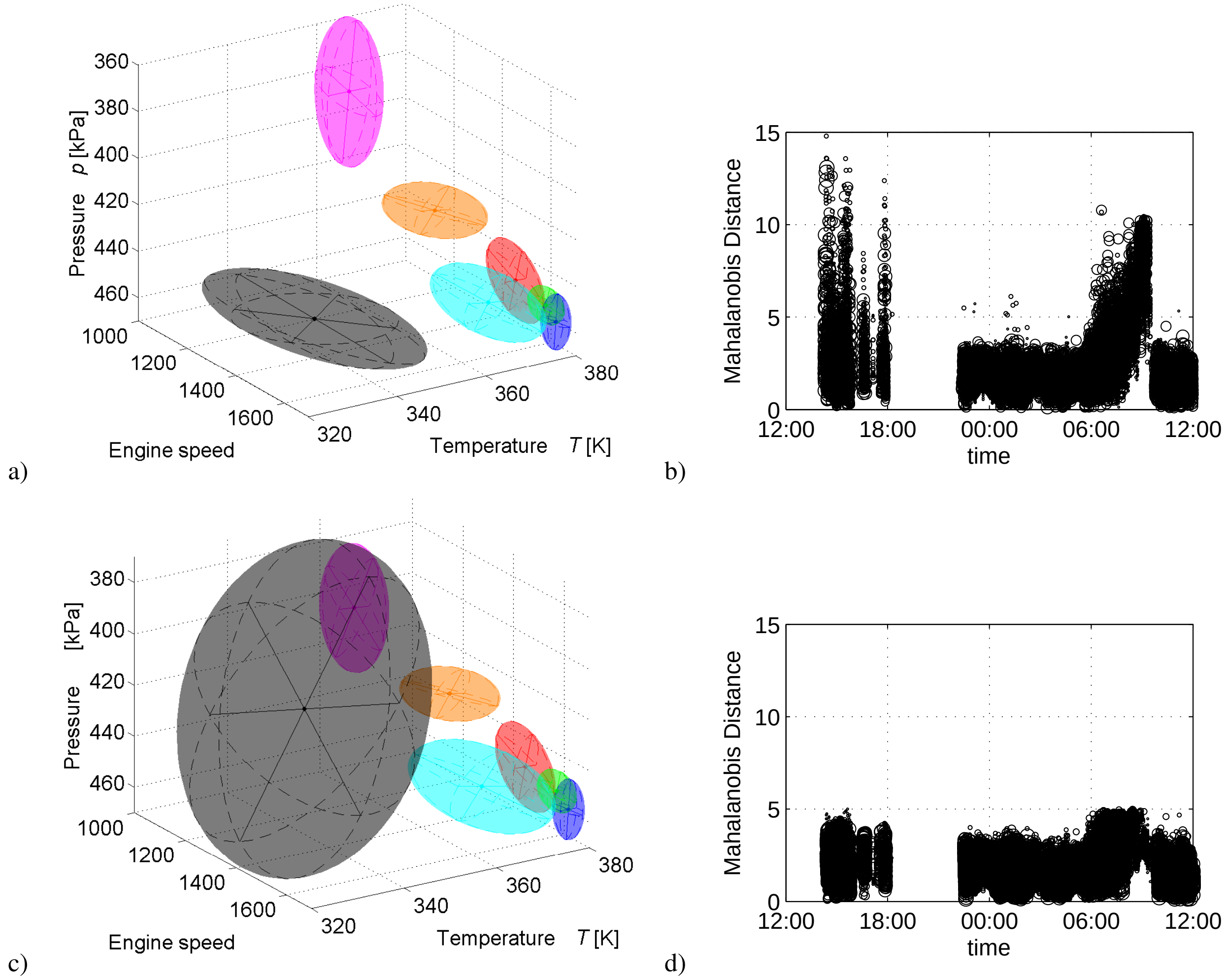


Figure 9. Visualizations of a GMM. (a) with good (1.6 h) prediction horizon; (b) GMM with zero prediction. Most of the Gaussians (except the grey one) have similar positions and sizes as the ones in (a).

the GMM with the smallest average Mahalanobis distance to the most likely Gaussian for all the training data. The number of Gaussians in the GMM was selected by searching for the smallest number of components where the improvement of the average Mahalanobis distance stopped to avoid overfitting.

3.2.2. Gaussian Mixture Rejection Filtering

We investigated observed inconsistency in performance of randomly-initialized GMMs in order to understand why some resulted in zero detection performance. Figure 8 shows the relationship between the model performance and the likelihood, l, of the training data given the trained model for GMMs with seven components each. The figure shows that several of the learned models those with the best training performance have zero detection. The results for the other GMMs show a general correlation between training performance (larger model posterior likelihood, l, or smaller $-\log(l)$) and detection performance. The GMM visualizations in Figure 9 – one with 1.6 hour detection horizon and one with zero performance – show the likely cause of this. (The ellipsoids represent the envelope enclosing the points within the one standard deviation probability, that is where $|z| \leq 1$.) In the GMM with good performance, Figure 9a, the component Gaussians are all fairly compact. The other, Figure 9c, shows that one of the Gaussians encloses a large volume of the $[T_t, \omega_t, p_t]$ space. With this model, the metric values are all less than 5.

The GMMs like the one shown in Figure 9c are non-discriminative. The most likely Mahalanobis distances of any point in the training, anomaly, or post repair data set, is small enough that no anomalies are detected according to the problem statement in section 2.3. Figures 9b and 9d show the likeliest Mahalanobis distance of the low-oil interval data,with respect to the clusters of the two models shown in Figure 9a and Figure 9b, respectively. In a more detailed examination of the results, we found that the maximum Mahalanobis distance of any point in the training data to the large-ellipsoid component of Figure 9 (or any of the ones with zero detection performance) was less than 7. Based on this, the rejection criterion used to reject GMMs with non-discriminative Gaussian components is *for each Gaussian component in the GMM, compute the Mahalanobis distance between that Gaussian and each point in the training data. Reject the GMM if the maximum Mahalanobis distance for any component is than a threshold.* For this study, the rejection threshold value was 10. This value must be selected, based on the performance of the trained GMMs, by comparison of results of several GMMs with reasonable detection horizons with several GMMs with zero or near-zero detection horizons.

We applied this criterion to 20 candidate GMMs; 7 (35%) were rejected. We selected the GMM for modeling from the remaining GMMs by finding the GMM with the highest likelihood of the training data. The GMM with the longest detection horizon (see Figure 8) 2 hours did not have the highest likelihood. That model could not be selected according to the rules presented in the problem statement (section 2.3) because it used data other than the model and the training data.

We tested the need for this rejection filtering by using the algorithm of (Figueiredo & Jain, 2002) for training GMMs. We found a clear threshold for the rejection criterion after training 120 different GMMs. We found that GMMs that were rejected had zero detection performance. Although this alternate means to train GMMs confirmed the need for rejection filtering, and has several advantages over the native Matlab method especially finding the optimal number of components in the GMM we did not use this algorithm for the work reported here because the GMMs trained with this algorithm did not perform as well as the ones trained by Matlab's `gmdistribution.fit()` method.

3.3. Model 3 – Feed-Forward Neural Network

Two Artificial Neural Network (ANN) models were explored. The first one, Feed-Forward Neural Network (FFNN) can be viewed as a neural network analogue of Gridded Regression. An FFNN was trained to estimate the engine oil pressure, given the oil temperature and the engine speed. A new unknown function f_{NN} is trained to express pressure in terms of the other two variables and unknown parameters – weights $\mathbf{w}$

$$\hat{p} = f_{NN}(T, \omega; \mathbf{w}) \tag{8}$$

The metric used was the same as for the GR model: the absolute value of the z-score of the residuals. The hidden neurons employ sigmoid activation functions because linear activation functions reduce the neural network to a simple linear equation

$$\hat{p} = w_0 + w_1 T + w_2 \omega \tag{9}$$

whose performance was considerably worse than that of the GR model.

3.3.1. FFNN: Parameters and Performance

At first, a two-layer neural network was employed for modeling the functions[2], with twenty neurons in the hidden layer, given by

$$\hat{p}(T, \omega; \mathbf{w}) = \sigma\left(\sum_{j=1}^{40} w_{kj}\sigma\left(w_{j1}T + w_{j2}\omega + w_{j0}\right) + w_0\right), \tag{10}$$

where $\sigma()$ is the logistic sigmoid and $\mathbf{w}$ are the weights. This standard neural network topology, known as the universal function approximator, with its expressive power, and its relation to Kolmogorov theorem is discussed in (Duda et al., 2000, Section 6.2.2). However, in our case, significantly better performance was achieved after the two-layer topology 2-20-1, was replaced by a three-layer 2-3-3-1 topology with the same total number of neurons, which is not surprising because deeper network have better expressive power. The final topology was selected comparing various candidate topologies. The number of layers and neuron counts were randomly selected within narrow ranges. A simple program trained FFNNs with the selected topology and evaluated the event horizon. The best model, with the longest event horizon, was used. Figure 10 shows the topology of a six-neuron FFNN. This simple network performed strictly as well, or better than, FFNNs with larger numbers of neurons or additional neuron layers. The selected topology was simplest in terms of neuron counts, and is expected to have better generalization than its more complex counterparts

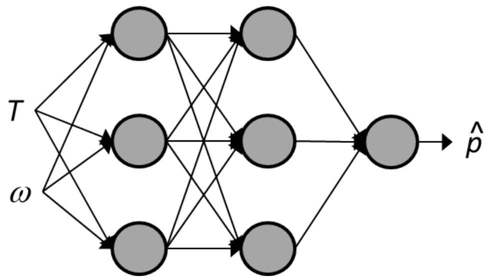


Figure 10. Topology of FFNN – two hidden layers with three neurons each.

After training on good data, the network showed a 3 hour detection horizon with no false alarms.

3.4. Model 4 – Replicator Neural Network

The second ANN model, Replicator Neural Network (RNN), can be considered as neural network analogue of GMM. An RNN (Hawkins, He, Williams, & Baxter, 2002) has 3 hidden layers, with sigmoid activation functions in the first and third

[2] This article employs the notation where the number of layers of a neural network is equal to the number of adaptive weights, as in (Bishop, 2006).

layers. The middle hidden layer has one neuron for each input signal, and the activation function is a differentiable step function that quantizes the input into one of the steps. The output of the network is the vector $[\hat{T}_t,\, \hat{\omega}_t,\, \hat{p}_t]$.

We found that the RNN model did not train well with the original input data; the average length of the residual vector was dominated by prediction error of ω. This necessitated scaling the training, anomaly event, and normal data. For the metric, Hawkins et al. (2002) suggests using the outlier factor, which is defined as the mean of the square of the Euclidean-norm of each residual:

$$OF_t = \frac{1}{3}((T_t - \hat{T}_t)^2 + (\omega_t - \hat{\omega}_t)^2 + (p_t - \hat{p}_t)^2) \qquad (11)$$

We also investigated an alternative metric: the Mahalanobis distance of the residuals from the mean of a single Gaussian modeling the residuals from the training interval. The outlier factor weights all components of the residual equally, whereas the Mahalanobis distance metric adapts to the statistics of the residual signals.

3.4.1. RNN: Parameters and Performance

An RNN was trained to replicate T, ω, and p. The signal values were pre-scaled into the range [0.1 0.9]. Mahalanobis distance metric resulted in a 1.2 hour detection horizon. Hawkins' (Hawkins et al., 2002) outlier factor metric resulted in zero anomaly detection.

Guided by an automated exploration of the parameter space, we selected a RNN with 10 neurons in the first and last hidden layers, and 3 neurons in the middle hidden layer, corresponding to our three signals in this study. The activation function of the middle hidden layer has 32 steps.

Mahalanobis distance metric resulted in a 1.2 hour detection horizon.

3.5. Model 5 – Boosted Regression Tree

The BRT (Elith, Leathwick, & Hastie, 2008) model estimates $\hat{p}_t$ based upon $(\omega_t,\, T_t)$. From the modeling perspective, it is comparable to the GR model because both use speed and temperature to predict the pressure, then calculate the absolute value of the z-score given by Eq. (7) as the metric.

3.5.1. BRT: Parameter and Performance

A BRT, with 200 sub-trees, was trained on data with range filtering according to schema 1. The detection horizon was 2.9 hours, as shown in Figure 11. We trained models with 10, 20, 50, 100, and 200 sub-trees, and found that the performance for the 10 sub-tree BRT was much lower (1.2 hours), while the BRTs we investigated with 20 – 200 sub-trees all produced detection horizons within 0.1 hours of each other.

In another variation on this experiment, we used data using schema 2 for the range filter (restricted oil temperature) and found that performance improved substantially: for the 20, 50, 100 and 200 sub-tree BRTs, the detection horizon was 3.0 hours, and the detection horizon of the 10 sub-tree BRT was only slightly less - 2.8 hours.

4. RESULTS COMPARISON

Figure 11 and Table 3 summarize the results of this investigation. Figure 11 offers two comparisons based on two detection horizons: one measures the time between the first observed anomaly (the day before the final failure) and the final failure, and the other measures the time between the first detection of anomaly during the final mission and the final failure.

The performances of the detectors according to the first, accross-the-mission comparison are nearly indistinguishable, ranging between 18.8 and 19 hours, which amounts to just over one percent (1.05%).

The second, within-the-mission comparison, however, separates the performances of different detectors. According to this comparison, the GR and BRT methods produced the best overall performance. The detection horizon during the last mission, 2.9 hours, is more than twice as long as the RNN, and nearly twice as long as the GMMs. The FFNN performance, 2.7 hours, was nearly as good. Note that all detectors considerably outperformed thefa existing DTCs, which appeared only 0.1 hour before the failure.

Table 3 lists detection horizons within the last mission with the times required to train the associated detectors. There is little correlation between training time and detection performance; the models with the best detection performance take the longest and shortest times to train. FFNN took by far the most time to train, but it also resulted in the most compact model, which is has an efficient execution and is less prone to overfitting.

Table 3. Performance of algorithms.

Method	Details	Detection Horizon (hours)	Training Time (s)
GMM	$-20 < T < 120$ training; no GMM rejection filtering	0	45
RNN	10+3+10 topology, 8 steps	1.1	670
GMM	$-20 < T < 120$ training; GMM rejection filtering	1.6	45
GMM	$90 < T < 120$ training; GMM rejection filtering	1.6	45
FFNN	3+3 topology	2.7	1780
BRT	$-20 < T < 120$	2.9	40
GR	$-20 < T < 120$	2.9	1

5. DISCUSSION AND CONCLUSIONS

This paper proposes an approach for incremental introduction of PHM capabilities by development of anomaly detection,

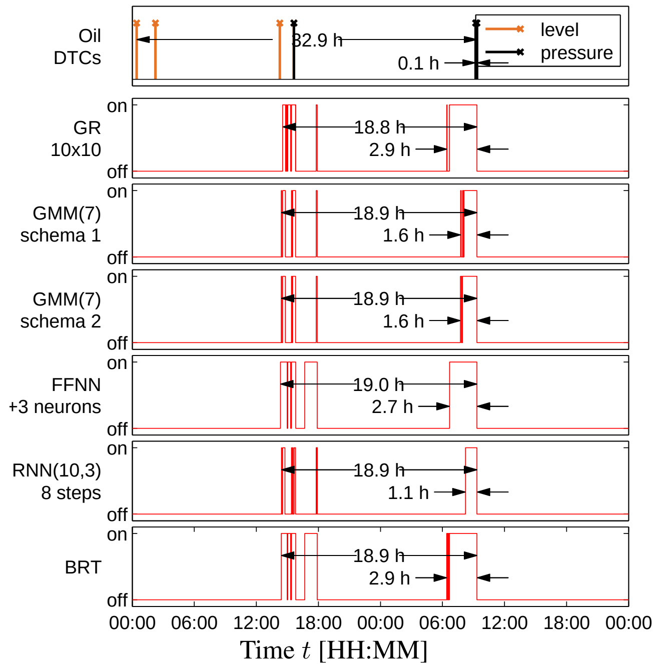


Figure 11. Anomaly detection performance of all models. Each graph shows the on/off state of the anomaly detection using a comparator on the averaged metric. In addition, the top graph shows the diagnostic trouble codes from the vehicle's electronic control unit. For GR see section 3.1.1; for GMM(7)-both schema-see section 3.2.1; for FFNN see section 3.3.1; for RNN see section 3.4.1, and for BRT see section 3.5.1.

even in the presence of a single known failure. We evaluated detectors by disallowing any false alarms during the period of normal operation and measuring detection horizon. The conservative requirement of zero-false-alarm tolerance aimed to compensate for potential overfit problems due to the lack of test and verification data. Rather than waiting to observe a statistically significant set of failures, we propose to start learning from the very first failure instance and carefully consider newly triggered anomalies by verifying the presence of real (incipient) failures. Any new undetected failures would also have to be incorporated in the models. All observed failures and their modes would be documented to allow for future classification and diagnostics, and any observed failure progression, with known failure modes, would be used for future prognostics development. In the context of this vision of PHM, we described its first layer – a tentative anomaly detector that consisted of a pre-filtering, data-driven model, a filter and a threshold comparator. The most space is given to comparison of five candidate data-driven models.

We found that residual based models (GRs, FFNNs, and BRTs) outperformed distance based models (GMMs and RNNs) in this application. The better performance of residual models is probably due to small engineering knowledge that was captured in them by expressing engine oil pressure in terms of engine speed and engine oil temperature. The distance-based models met the nearly zero expert knowledge goal, at least with respect to expert engineering knowledge of the vehicle, but required skills and effort in the machine learning area to select useful models and metrics. In particular, such knowledge and effort was necessary to identify and correct the root cause of the inconsistent GMMs' performance. We reported that locally-optimal GMMs often failed to detect anomalies. To overcome the problem we proposed a heuristic filter that rejects candidate GMMs with non-discriminative components and controls the volume of the largest mixture component.

The old technique of gridded residual, often neglected in favor of more recent methods, not only achieved the best detection horizon, but also trained the fastest. BRT shared the first prize with GR with respect to detection horizon and trained reasonably fast (still not nearly as fast as GR), but its model complexity was much higher. FFNN, by contrast, required by far the most amount of time for training, but achieved a very good result with a the most compact model, which is less likely to overfit. All models performed markedly better than the tradition, vehicle built-in DTCs.

This study employed a very simple anomaly detector – a filter with a threshold comparator. As more failures are observed, more sophisticated inference engine should be considered, especially those that combine multiple learners, such as a model ensemble, which may have a built-in bias against potentially overfitted models.

While this work investigated models for anomaly detection, the results suggest further work to create diagnostic and prognostic algorithms based on these techniques. Implementation of fleet-wide data collection and analysis would allow a statistically significant set of known failures to be created. This in turn would allow estimation of a Receiver Operating Characteristic curve and enable known PHM engineering techniques that are based on such curves to be applied.

ACKNOWLEDGEMENTS

This work was made possible by the Department of the Navy, Office of Naval Research under Award No. N0004-07-1-0823. We gratefully acknowledge our industrial partner, Vnomics Inc. for providing access to the vehicle data.

NOMENCLATURE

Symbol	Definition
ω	Engine speed in radian/s (1 radian/s $\approx$ 9.55 RPM)
ω_i	Sequence of engine speeds in i^{th} speed group
$\Delta\omega$	Width of each speed group
T	Engine oil temperature in °K (°K $\approx$ °C+273)
T_j	Sequence of engine oil temperatures in j^{th} temperature group
ΔT	Width of each temperature group
p	Engine oil pressure in kilo-Pascals (kPa)
p_{ij}	Sequence of pressures in bin of (ω_i, T_j)
p_{ij}^k	k^{th} value of p_{ij}
$\overline{p}_{ij}$	Mean of pressures in bin of (ω_i, T_j)
$\hat{p}$	Estimate of engine oil pressure
S_i	The i^{th} signal
$D_{training}$	Sequence of observed signals used for training
D_{event}	Sequence of observed signals in known event(s)
D_{normal}	Sequence of observed signals during normal operation
$\mathcal{M}$	A model for a set of signals, based on data from a training interval
m	A real-number sequence resulting from evaluating a model over data from a given interval
Θ	Anomaly detection threshold
M_{ij}	The number of values in p_{ij}
ϵ	Residual, or prediction error
z	z-score of prediction error ϵ
σ_p	standard deviation of sequence of prediction errors
$N(\mu, \sigma^2)$	Univariate normal (Gaussian) distribution for mean μ and variance σ^2
$\boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal (Gaussian) distribution for mean and covariance $\boldsymbol{\mu}, \boldsymbol{\Sigma}$
π_k	Weight of distribution $\boldsymbol{N}_k$ in Gaussian Mixture Model
$\mathcal{P}$	Sequence of maximum weighted probabilities signals from Gaussian Mixture Model
pr_k	Probability of (T, ω, p) for k^{th} distribution in Gaussian Mixture Model
l	Posterior likelihood of signal for given model
$\boldsymbol{w}$	Weights in Neural Network
$\sigma(x)$	Logistic sigmoid function of x
OF_t	Outlier Factor for signal at time t

REFERENCES

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 15.

Cheifetz, N., Same, A., Aknin, P., & de Verdalle, E. (2011). A pattern recognition approach for anomaly detection on buses brake system. In *Intelligent transportation systems (itsc), 2011 14th international ieee conference on* (pp. 266–271).

Duda, R., Hart, P., & Stork, D. (2000). *Pattern classification. 2nd edn wiley.*

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, *77*(4).

Figueiredo, M., & Jain, A. (2002, march). Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(3), 381 -396. doi: 10.1109/34.990138

Golosinski, T. S., Hu, H., & Elias, R. (2001). Data mining vims data for information on truck condition. *Computer Applications in the Minerals Industries*, *88*, 397–402.

Hawkins, S., He, H., Williams, G., & Baxter, R. (2002). Outlier detection using replicator neural networks. In Y. Kambayashi, W. Winiwarter, & M. Arikawa (Eds.), *Data warehousing and knowledge discovery* (Vol. 2454, p. 113-123). Heidelberg: Springer.

Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., … others (2004). Vedas: A mobile and distributed data stream mining system for real-time vehicle monitoring. In *Proceedings of siam international conference on data mining* (Vol. 334).

McArthur, S. D. J., Booth, C. D., McDonald, J. R., & McFadyen, I. T. (2005). An agent-based anomaly detection architecture for condition monitoring. *Power Systems, IEEE Transactions on*, *20*(4), 1675-1682.

Nowlan, F., & Heap, H. (1978). *Reliability-centered maintenance* (Tech. Rep. No. AD/A066 579). United Airlines, San Francisco, CA.

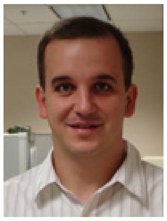
Sikorska, J. Z., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, *25*(5), 1803-1836.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Hoboken, NJ: John Wiley & Sons, Inc.

Vnomics. (2012). *Products - vehicle health management software* (Vol. 2012) (No. 3/16/2012).

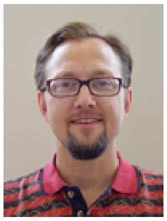
BIOGRAPHIES

Howard Bussey earned Bachelors degrees in Electrical Engineering, Mathematics, and Psychology from the Universities of Colorado and Minnesota (Boulder, CO, USA and Minneapolis, MN, USA), and a Masters degree in electrical engineering from the University of California (Berkeley, CA, USA). He is CTO/co-founder of Tixlers' Letters, providing educational resources. He was a Senior Staff Engineer in the Golisano Institute of Sustainability, and has also worked at Eastman Kodak, Bell Communications Research, Bell Laboratories, and the National Oceanographic and Atmospheric Administration. He has six patents and 13 publications. He is a member of Eta Kappa Nu and Tau Beta Pi.



Nenad Nenadic received his B.S. in Electrical Engineering from University of Novi Sad (Novi Sad, Serbia) in 1996 and his MS and Ph.D. in Electrical and Computer Engineering from University of Rochester (Rochester, NY, USA) in 1998 and 2001, respectively. He joined Kionix Inc. in 2001, where he worked on development of microelectromechanical inertial sensors. Since 2005, he has been with Center for Integrated Manufacturing Studies (CIMS) at Rochester Institute of Technology, where he is currently a Research Associate Professor. His research interest include design, analysis, and monitoring of electromechanical devices and systems. He has two patents in electromechanical design and six publications. He co-authored a textbook ”Electromechanics and MEMS”. He is a member of IEEE.

Paul Ardis received his B.S. in Computer Science from Purdue University (West-Lafayette, IN, USA) in 2005 and his M.S. and Ph.D. in Computer Science from the University of Rochester (Rochester, NY,USA) in 2007 and 2009 respectively. He is a Lead Scientist at GE Global Research, and was formerly a Research Associate Professor at the Center for Integrated Manufacturing Studies (CIMS) at Rochester Institute of Technology and Research Scientist under contract to the Air Force Research Laboratory. He holds two patents in signal processing and has published eleven scholarly articles in machine learning,computer and human vision, biometrics, and machine diagnostics and prognostics. His research interests include data-driven predictive modeling, decision theory, surveillance, natural language processing, and machine sensing.



Michael Thurston received his B.S. and M.S. in Mechanical Engineering from Rochester Institute of Technology (Rochester, NY, USA) in 1988, and his Ph.D. in Mechanical and Aerospace Engineering from the University of Buffalo (Buffalo, NY, USA) in 1998. He is the Technical Director and Research Associated Professor at the Center of Integrated Manufacturing Studies at Rochester Institute of Technology. He formerly held positions in air conditioning system development at General Motor and Delphi, and as a Researcher at the Applied Research Laboratory at Penn State University. He holds 7 patents in the areas of air conditioning and asset health monitoring. His research interests include: sustainable design and production, condition based maintenance and prognostics, and asset health management. He is a member of the Society of Automotive Engineers, and was awarded the Boss Kettering Award for product innovation by Delphi.