

# Unsupervised Anomaly Detection for Liquid-Fueled Rocket Propulsion Health Monitoring

Mark Schwabacher<sup>1</sup> and Nikunj Oza<sup>2</sup>  
*NASA Ames Research Center, Moffett Field, CA, 94035*

*and*

Bryan Matthews<sup>3</sup>  
*Perot Systems Inc., Moffett Field, CA, 94035*

**This paper describes the initial results of applying four machine-learning-based unsupervised anomaly detection algorithms—Orca, GritBot, the Inductive Monitoring System, and one-class Support Vector Machines—to historical data from the Space Shuttle Main Engine. The paper describes five anomalies detected by the four algorithms.**

## I. Introduction

The ability to detect anomalies in sensor data from a complex engineered system such as a spacecraft is important for at least three reasons. First, detecting anomalies in near-real time during flight can be helpful in making crucial decisions such as whether to abort the launch of a spacecraft prior to reaching the intended altitude. Second, for a reusable spacecraft such as the Space Shuttle, detecting anomalies in recorded sensor data after a flight can help to determine what maintenance is or is not needed before the next flight. Third, the detection of recurring anomalies in historical data covering a series of flights can produce engineering knowledge that can lead to design improvements.

The current approach to detecting anomalies in spacecraft sensor data is to use large numbers of human experts. Flight controllers watch the data in near-real time during each flight. Engineers study the data after each flight. These experts are aided by limit checks that signal when a particular variable goes outside of a predetermined range. The current approach is very labor intensive. Also, humans may not be able to recognize faults that involve the relationships among large numbers of variables. Further, some potential faults could happen too quickly for humans to detect them and react before they become catastrophic. On future missions to Mars, there will be a speed-of-light delay of up to 20 minutes in between when the data are sensed and when flight controllers on Earth first see them, furthering the need for automated anomaly detection.

One approach to automating anomaly detection is the model-based approach. This approach encodes human knowledge into a model, which is then used to automatically detect faults. Examples of systems that use the model-based approach include Livingstone<sup>1-3</sup>, Titan<sup>4</sup>, TEAMS-RT<sup>5</sup>, RODON<sup>6</sup>, SHINE<sup>7</sup>, and MEXEC<sup>8</sup>. Building the models is very labor intensive; it therefore may not be feasible to model every part of a highly complex system such as a spacecraft. It also may not be possible to model all possible failure modes. We therefore consider supplementing the model-based approach with the data-driven approach.

The data-driven approach seeks to build a model for detecting anomalies directly from the data, rather than building it based on human expertise. In this paper, we explore a particular data-driven approach, which is based on anomaly detection algorithms from the machine learning and data mining communities.

## II. Anomaly Detection

Anomaly detection algorithms, also known as outlier detection algorithms, seek to find portions of a data set that are somehow different from the rest of the data set. A supervised anomaly detection algorithm requires training data consisting of a set of examples of anomalies, and a set of examples of non-anomalous (or nominal) data. From the

---

<sup>1</sup> Computer Scientist, Intelligent Systems Division, MS 269-3, AIAA member.

<sup>2</sup> Computer Scientist, Intelligent Systems Division, MS 269-2.

<sup>3</sup> Electrical Engineer, Intelligent Systems Division, MS 269-2.

data, the algorithm learns a model that distinguishes between the nominal and the anomalous data points. Supervised anomaly detection algorithms typically require tens or hundreds of labeled examples of anomalies, plus a similar number of labeled examples of nominal data points, in order to obtain adequate performance. Unsupervised anomaly detection algorithms are trained using only nominal data. They learn a model of the nominal data, and signal an anomaly when new data fails to match the model. They typically require tens or hundreds of nominal data points in order to obtain adequate performance.

We have been using historical data from the Space Shuttle Main Engine (SSME)<sup>9</sup> to develop and test algorithms that we hope will be useful for future launch systems such as Ares I<sup>10</sup> and Ares V<sup>11</sup>. For the SSME, the number of examples of anomalies available in historical data is fairly small. We therefore decided to use unsupervised anomaly detection algorithms, since they do not require labeled examples of anomalies. In a previous publication<sup>12</sup>, we described four candidate anomalies detected by two unsupervised anomaly detection algorithms, Orca<sup>13</sup> and GritBot<sup>14</sup>. This paper presents several additional anomalies detected by these two algorithms and by two other algorithms, the Inductive Monitoring System (IMS)<sup>15</sup> and a one-class Support Vector Machine (SVM)<sup>16</sup>. The next four subsections describe these four algorithms.

### A. Orca

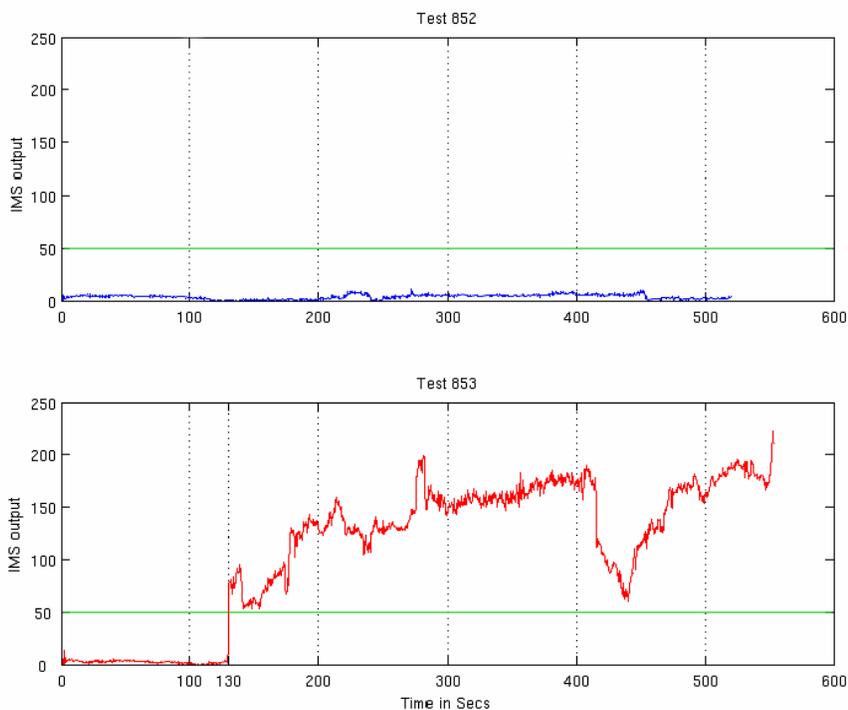
Orca, which was developed by Stephen Bay and Mark Schwabacher, uses a nearest-neighbor approach to anomaly detection. It defines an anomaly to be a point whose nearest neighbors in feature space are far away from it. It uses a novel pruning rule to obtain near-linear-time performance, allowing it to scale to very large datasets<sup>13</sup>.

### B. GritBot

GritBot is a commercial product from RuleQuest Research<sup>14</sup>. Rather than just looking for points that are anomalous with respect to the entire dataset, GritBot searches for subsets of the dataset in which an anomaly is apparent. For each anomalous point, it reports a description of the relevant subset of the dataset, based on values of discrete variables or ranges of continuous variables, in which the target variable usually has a particular value (if it is discrete) or range of values (if it is continuous). The point is considered to be an anomaly because the target variable at that point is significantly different from the value of the target variable at the vast majority of the other points in the subset.

### C. Inductive Monitoring System

The Inductive Monitoring System (IMS)<sup>15</sup>, developed by David Iverson at NASA Ames Research Center, is similar to Orca in that it is distance-based. The major difference is that during the training step, it clusters the nominal training data into clusters representing different modes of the system. At run time, it uses the distance to the nearest cluster as an anomaly measure. Before applying IMS to the SSME data, we pre-processed the data by first normalizing the data to have zero mean and unit variance, then clustering the variables into sets of variables with similar values, and then averaging the variables in each cluster to produce a single representative variable that was input to IMS. The clustering



**Figure 1. IMS detection of HPFTP failure.** IMS was trained on test 851 (which was nominal) and then tested on test 852 (top, also nominal) and test 853 (bottom, with an HPFTP failure at 130 seconds). The green line represents a candidate threshold above which values are considered to be anomalous.

and normalization were both based on the training data.

#### D. One-class Support Vector Machine

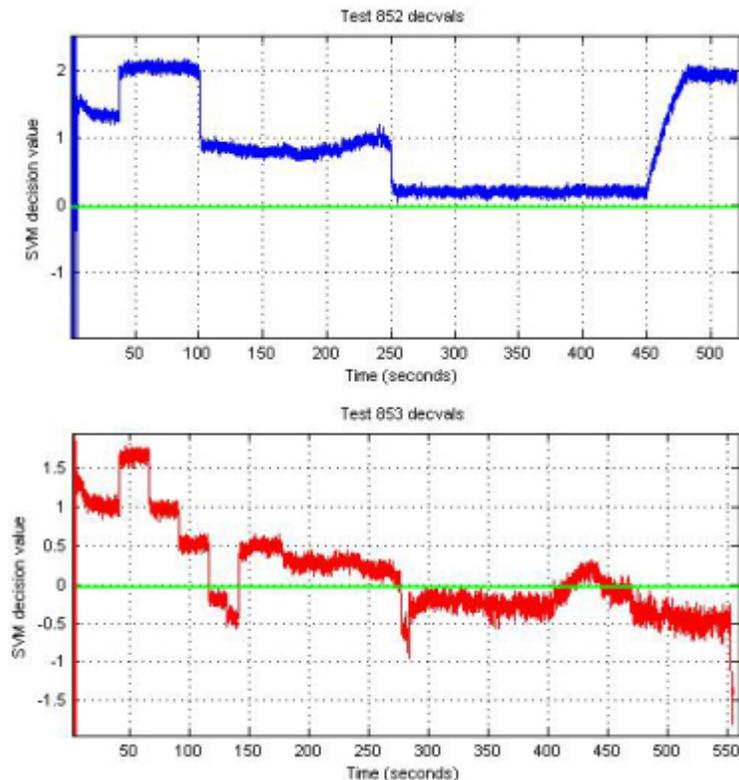
One-class Support Vector Machines (SVMs) seek to describe the range of normal training data in such a way as to enable it to distinguish normal data from abnormal data in the future. The name “one-class SVM” is due to the possibility that only one class of data (normal data) may be available during training (if abnormal training data is available, it can be used). One-class SVMs first map the training data from the original data space into a much higher-dimensional feature space and then find a linear model in that feature space that allows normal data to be on one side (and to be separate from abnormal training data if available). The idea is that linear models in a higher dimensional space correspond to more complicated nonlinear models in the original data space. The use of linear models allows SVMs to retain the benefit that the algorithm finds the globally optimal solution given the training set, while still effectively using nonlinear models. For each test point, the one-class SVM returns a measure of how strongly normal or anomalous the data point is.

### III. Space Shuttle Main Engine

The Space Shuttle propulsion system consists of two solid rocket boosters, and three Space Shuttle Main Engines (SSMEs)<sup>9</sup>. The SSMEs are located on the Orbiter. During the initial ascent, all five engines are fired together. Approximately two minutes after launch, the solid rocket boosters run out of fuel and are jettisoned, and the Shuttle continues its ascent using only the three SSMEs. The SSMEs are liquid-fueled rocket engines employing cryogenic liquid hydrogen and liquid oxygen as propellants. These propellants are stored in the external tank, which is jettisoned when the Shuttle reaches its intended orbit, about nine minutes after launch.

The SSMEs are reusable. After each flight, they are removed from the Orbiter, inspected, serviced as necessary, and then installed onto an Orbiter (which is not necessarily the same orbiter). Each engine is periodically acceptance tested by firing it on the ground using test stands located at NASA Stennis Space Center.

Each SSME has approximately 90 sensors measuring quantities such as temperature, pressure, fuel flow rate, rotational velocity, and vibration. Many of the sensors are redundant for reliability reasons. For example, four identical pressure sensors may be placed right next to each other with the expectation that they will all provide approximately the same value. When one of the sensors provides a substantially different value from the other three, it can be inferred that the sensor has failed. When the SSMEs are used on the Shuttle, additional relevant information is available from sensors located in the Shuttle’s fuel feed system. When the SSMEs are fired on a test stand, the test stand acts as a fuel feed system, and provides additional relevant information from sensors located on the test stand. A small



**Figure 2. One-class SVM detection of HPFTP failure.** *The one-class SVM was trained on test 851 and then tested on tests 852 (top) and 853 (bottom). Negative values indicate abnormal system behavior. On test 853, the SVM first indicates abnormal system behavior at 115 seconds, then indicates more anomalous behavior at 130 seconds when the actual failure occurred, and then indicates normal behavior before returning to indicating abnormal behavior at 276 seconds.*

number of additional sensors are placed directly on the SSME during testing. The initial results reported in this paper use only the data from the sensors that are located on the SSME and available both in flight and on the test stand.

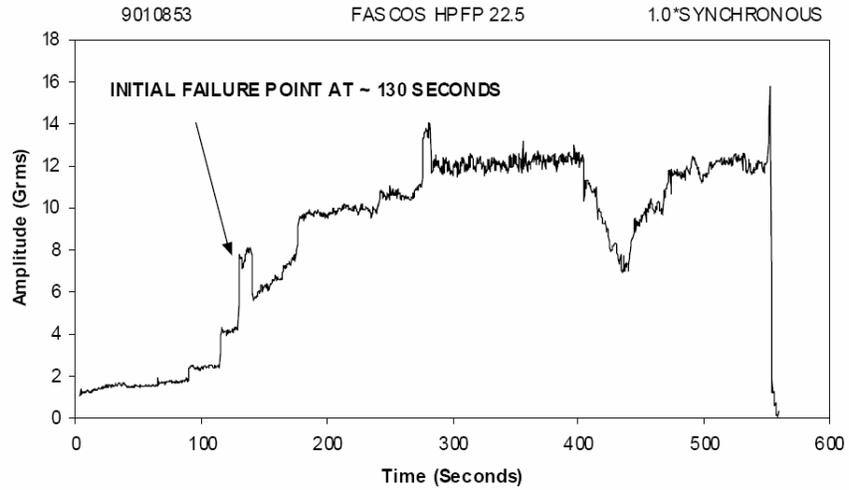
#### IV. Results

In our tests, the four algorithms successfully detected one major system failure, and several sensor failures. In this section, we review these results.

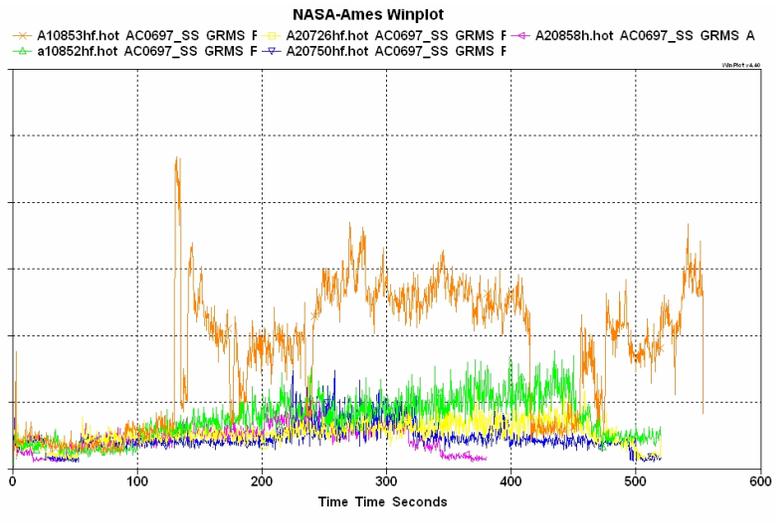
In each case described in this section, we used an anomaly detection algorithm to detect anomalies in data from one SSME test. In some cases, we used data from a different test as training data. Each test had 129 continuous variables, 18 discrete variables, and about 13,000 time steps. In some cases, we also included the first and second derivatives of the continuous variables with respect to time in the training and test data, so that the algorithms could detect unusual rates of change in the sensor values in addition to detecting unusual sensor values. GritBot took about 5 minutes to analyze the data from an SSME test on a 1.5 GHz Intel Pentium M laptop. Orca took about 3 minutes to analyze the data from an SSME test on a 500 MHz Sun Blade 100 workstation. IMS took about 3 minutes to analyze the data from an SSME test on a 2.6

GHz dual-core Intel Xeon. One-class SVMs took about 0.5 seconds to analyze the data from an SSME test on a 2.16 GHz Intel Core Duo laptop. Each SSME test lasted about 8 minutes, so all four algorithms were able to process the data faster than real time.

The first anomaly that we consider is a failure in the SSME's High Pressure Fuel Turbopump (HPFTP), which occurred during SSME static test 853 in January 1996, approximately 130 seconds into the test<sup>17</sup>. We tested the ability of the algorithms to detect this known failure by training them on data from test 851, which was nominal, and then testing them on tests 852 (which was also nominal) and 853 (which had the failure). Figure 1 shows the IMS outputs for tests 852 and 853. Both graphs plot the IMS output against time for each test. The IMS output is the distance in the normalized multidimensional space from the new data point at each time step to the nearest cluster in the training data. Larger IMS outputs indicate that the data is more anomalous. Notice that in test 852, the IMS output remained well below 50 for the entire test. In test 853, the IMS output remained well below 50 until 130



**Figure 3. Previously published detection of HPFTP failure.** *The HPFTP failure can be detected using a particular vibration sensor at a particular frequency. This figure is reproduced from Ref. 17.*



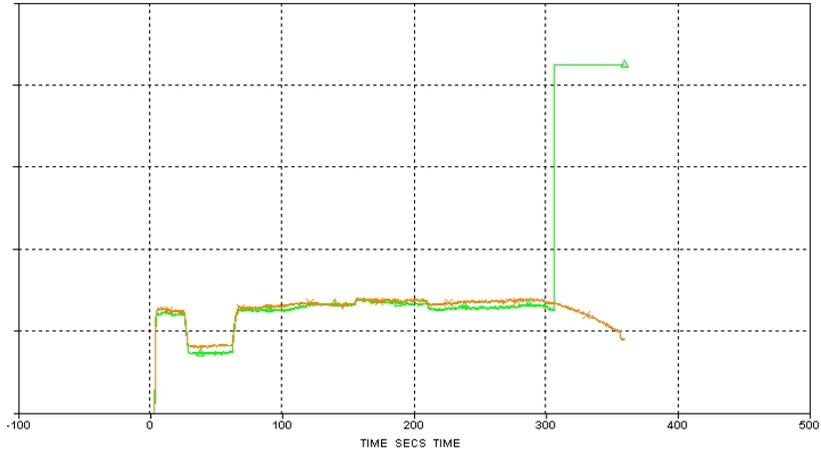
**Figure 4. Orca detection of HPFTP failure.** *Orca discovered that the HPFTP failure can be detected more easily using a different vibration sensor at a different frequency. The five curves represent the same vibration variable during five different SSME tests. The orange curve, for test 853, shows the failure at 130 seconds.*

seconds, and then became much larger. These results show that IMS successfully detected the failure. If a threshold of an IMS score of 50 (as shown in green in Figure 1) were used to signal an alarm, then IMS would signal the alarm at the correct time in test 853, without signaling any false alarms in test 852. Orca, GritBot, and one-class SVM were also able to successfully detect the HPFTP failure in test 853.

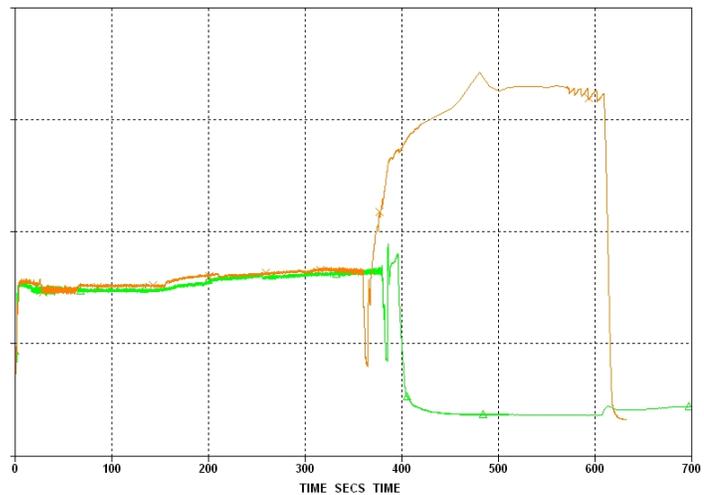
Figure 2 shows the output of the one-class SVM when it was tested on tests 852 and 853 after being trained on test 851. The horizontal axis displays the time in seconds from the start of the test just as in Figure 1. The vertical axis shows the output of the one-class SVM. Values greater than zero indicate that the one-class SVM thinks the system is behaving normally, with higher values indicating greater normality. Values less than zero indicate abnormal system operation, with lower values indicating more anomalous operation. For test 853, the one-class SVM first indicates abnormal operation at 115 seconds, which is 15 seconds before the HPFTP failure occurred. At 130 seconds, the output goes lower, indicating a greater level of abnormality. The SVM output then returns to a normal level, gradually decreases, and then at 276 seconds returns to indicating abnormality. The one-class SVM only indicates whether the system as a whole is operating normally or abnormally, and does not indicate specific variables or systems that may be operating abnormally. Future work will need to determine the exact source of the possible precursor at 115 seconds, the reason for the return to normality at 140 seconds, and the reason for the return to abnormality at 276 seconds.

Using Orca, we determined that it is possible to easily detect the HPFTP failure using a single sensor at a particular frequency. Figure 3 is a reproduction of a figure from Ref. 17, showing that the failure can be detected using a particular vibration sensor at a particular frequency. When Orca discovered the HPFTP failure, it reported that the strongest contribution to the anomaly was a different variable representing a different vibration sensor at a different frequency. This variable is shown in Figure 4, for five different SSME tests. (Note that in Figure 4 and in all subsequent figures, the values have been removed from the Y axis to protect the confidentiality of the data.) For test 853, there is a large increase in this vibration measure at 130 seconds, which is the time of the failure. By examining the other four tests, we see that the selected variable remains much lower during these other tests. We also note that in test 852—the previous test of the same engine on the same test stand, and the green curve in Figure 4—there appears to be a gradual increase in vibration. Our experts disagreed with each other regarding whether or not this increase represents a precursor of the HPFTP failure.

In addition to detecting the known failure in the HPFTP, the algorithms also discovered many sensor failures that were not previously known to us. We present four of them here.



**Figure 5. GritBot detection of temperature sensor failure.** *This graph shows two redundant temperature sensors for the same test of the same engine.*



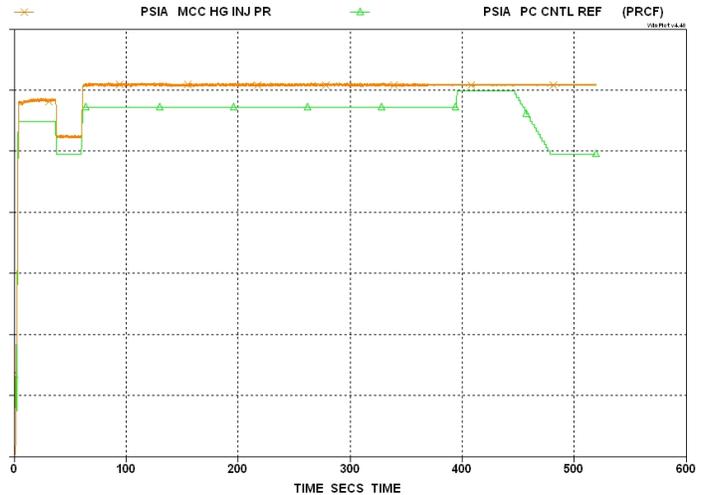
**Figure 6. Orca detection of temperature sensor failure.** *This graph shows the same temperature sensor across two different tests of two different engines.*

Figure 5 shows the values from two redundant temperature sensors for the same test. The two sensors return approximately the same value as each other until just after 300 seconds, when one of the sensors apparently has a failure that causes it to return a constant high value. GritBot detected this sensor failure.

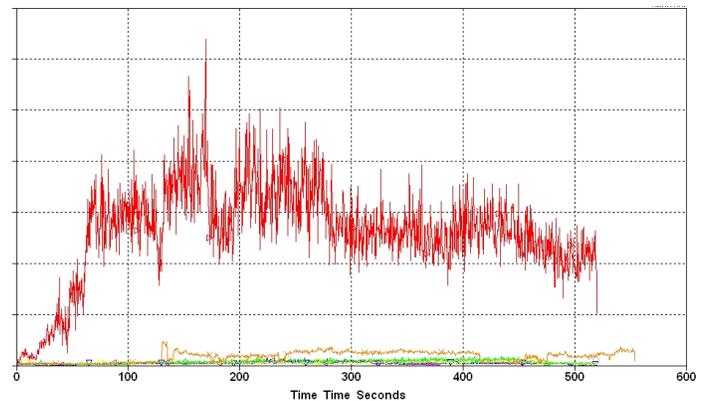
Figure 6 shows another temperature sensor failure that was detected by Orca. This failure, however, was detected not by comparing two redundant sensors during the same test, but by comparing two tests of two different engines. Orca was trained on one engine test and tested on the other engine test. It noticed that the temperature profiles from a particular sensor diverged sharply at around 400 seconds, indicating a sensor failure.

Figure 7 shows a pressure sensor failure detected by Orca. This particular failure is more difficult to detect than the sensor failures in the previous figures, because the values from the failed sensor remain within the normal range. Orca detected the failure by noticing that the main combustion chamber controller reference pressure (in green) and the main combustion chamber hot gas injection pressure sensor (in orange) are usually strongly correlated, but that the correlation is violated during this test. Our experts explained that this was a notorious sensor failure caused by moisture in the line freezing.

Figure 8 shows a vibration sensor failure detected by Orca. This figure shows the same vibration sensor as Figure 4, but includes an additional test that was excluded from Figure 4 because of the sensor failure. Figure 8 has a different scale from Figure 4; the five curves from Figure 4 can be seen at the bottom of Figure 8, each in the same color as in Figure 4, but they appear much smaller in Figure 8 than they do in Figure 4 because of the different scale. The HPFTP failure can still be seen in the orange curve, but it is greatly overshadowed by the red curve. This particular vibration sensor returned values approximately ten times higher in the test corresponding to the red curve than it did in the other five tests. Our experts confirmed that it is a sensor failure.



**Figure 7. Orca detection of pressure sensor failure.** This graph shows the main combustion chamber controller reference pressure (in green) and the main combustion chamber hot gas injection pressure sensor (in orange). The latter sensor apparently froze sometime before 400 seconds.



**Figure 8. Orca detection of vibration sensor failure.** This graph shows the same sensor that is shown in Figure 3, but with a sixth test added (in red). The orange curve shows the sensor failure, as before, at 130 seconds. The red curve represents a sensor failure.

## V. Related Work

In Ref. 18, Martin *et al.* present a comparison of six unsupervised anomaly detection algorithms, including the four algorithms discussed in this paper, plus a Gaussian Mixture Model and a Linear Dynamic System. They ran all six algorithms using SSME data from four space shuttle flights and two test stand firings for training, and eight shuttle flights and four test stand firings for validation. Although they acknowledge that they do not have enough data to make statistically significant comparisons of the relative performance of the six algorithms, they conclude that the algorithm with the best accuracy appeared to be either Orca or the one-class SVM, depending on how they classify ground truth in the validation data.

Park, *et al.* applied the BEAM (Beacon-based Exception Analysis for Multi-Missions) system to anomaly detection in SSME data<sup>19</sup>. BEAM has nine components that use nine different approaches to anomaly detection. The

work reported in Ref. 19 only used one of the nine components: the Dynamical Invariant Anomaly Detector (DIAD). DIAD is an unsupervised anomaly detection algorithm, like the algorithms described in this paper. DIAD differs from the algorithms described in this paper in that DIAD only considers one variable at a time, whereas the algorithms described in this paper all consider all of the variables together and look for anomalies in the relationships among the variables, in addition to anomalies in individual variables. Park, *et al.* trained DIAD using data from 16 nominal tests, and tested it using data from seven tests that contained known failures. It detected all of the major failures in these seven tests, although it missed some minor failures and had some false alarms.

Iverson's Inductive Monitoring System (IMS)<sup>15</sup> has also been used in another Space Shuttle application. After the STS-107 Space Shuttle Columbia disaster, Iverson applied IMS to data from four temperature sensors inside the Shuttle's wings. He trained it using data from five previous Space Shuttle flights, and then tested it using STS-107 data. It detected an anomaly in data from the temperature sensors on the Shuttle's left wing shortly after the foam impact, suggesting in retrospect that with the aid of IMS, flight controllers might have been able to detect the damage to the wing much sooner than they did.

In Ref. 20, Srivastava describes a data-driven system that detects anomalies in sequences of discrete data using envelope detection methods and dynamic Hidden Markov Models. This system has been applied to data from the switches in an aircraft cockpit. In this application, it detects anomalies in the sequence of switch flips made by the pilot, including detecting if the switches are flipped in an unusual order. For example, it can detect if the pilot lowers and raises the landing gear multiple times, instead of just once (as is usually the case in the training data).

Many of the existing approaches to data-driven systems health monitoring have used artificial neural networks to model the system. Artificial neural networks are a type of nonlinear model based loosely on the neural structure of the brain, in which the weights of the connections among neurons are automatically adjusted to maximize the fit of the model to the data<sup>21</sup>. Guo and Musgrave<sup>22</sup> applied neural networks to sensor validation for the SSME. He and Shi<sup>23</sup> found that support vector machines produced better accuracy than artificial neural networks when applied to a pump diagnosis problem. One disadvantage of neural network approaches is that most humans are unable to understand or interpret the neural network models. Models based on decision trees, decision rules, or nearest neighbors are generally easier to understand, and therefore more likely to be accepted by human experts.

## VI. Conclusion

This paper presents four sensor failures and one system failure that were detected by applying four unsupervised anomaly detection algorithms to data from the Space Shuttle Main Engine. Although all of these anomalies were either previously known or minor, we believe that they demonstrate that the algorithms have the ability to detect the kind of unusual phenomena in the data that correspond to significant anomalies. We have also demonstrated that these algorithms have the potential to process real rocket propulsion sensor data in real time.

An important point to make is that although some anomalies were detected by multiple algorithms, other anomalies were only detected by one algorithm out of the four. The reason for the difference is that the different algorithms use different definitions of an anomaly. Because of these differences, it can be useful to run multiple anomaly detection algorithms on a data set.

## VII. Future Work

This paper presents early results; there are many directions for future research. The preliminary results presented in this paper consist of five examples of anomalies that were detected by four unsupervised anomaly detection algorithms. Unfortunately, we will probably never have enough data to calculate false positive and false negative error rates with any statistical significance, but testing the algorithms on a larger amount of data than we have used so far should shed additional light on the performance of the algorithms.

All four algorithms described in this paper output anomaly scores that measure the degree of anomalousness of the data. Before the algorithms could be used in an automated fashion, alarm thresholds would need to be determined so as to minimize false positive and false negative error rates. Setting these thresholds is difficult because of the small number of available examples of verified anomalies.

In the research described in this paper, we used the four algorithms individually and compared the results. As we note in the conclusion, different algorithms within the set of four did a better job of detecting different anomalies. An important direction for future research is the automatic combination of the outputs of the different algorithms in a way that produces better results than any of the algorithms individually.

Because of the small number of examples of anomalies available to us, we decided to first try unsupervised anomaly detection algorithms. We believe that the small number of examples of known anomalies will present a challenge for supervised anomaly detection algorithms. A direction for future research is to try to find ways to make

supervised learning algorithms perform adequately given such a small number of examples of anomalies. One approach would be to use each time step from an anomaly that spans multiple time steps as an example of an anomaly.

In the results presented in this paper, the anomaly detection algorithms did not make use of any expert knowledge. We plan to explore ways of using background knowledge within the automated anomaly detection context. One possible source of knowledge is the determinations made by the domain experts that certain anomalies detected by the algorithms are not significant. One way to make use of this knowledge would be to use semi-supervised learning algorithms, and to provide them with examples that are labeled as nominal for each candidate anomaly that the experts judged to be nominal. The algorithm would then avoid incorrectly signaling an anomaly in the future when similar patterns appear in the data.

The data sets used in this research are all time series. The algorithms described in this paper, however, do not explicitly make use of time. Instead, time is treated just like any other variable. We plan to explore algorithms that explicitly model time, such as Kalman filters and Hidden Markov Models.

### Acknowledgments

This work would not have been possible without the collaboration of our domain experts. For providing SSME data and expertise, we thank Matt Davidson, Al Daumann, and John Stephens of Pratt & Whitney Rocketdyne, and Anthony Kelley and John Butas of NASA Marshall Space Flight Center.

We also thank the other members of the liquid propulsion health management team at NASA Ames Research Center: Ashok Srivastava, Rodney Martin, and Richard Watson. In addition, we thank Ashok Srivastava and Rodney Martin for reviewing a draft of this paper. We thank David Iverson of NASA Ames Research Center for providing the IMS software.

The research described in this paper was funded by the NASA Exploration Systems Mission Directorate (ESMD) under the Technology Maturation Program as part of the ISHM Testbeds and Prototypes Project, by ESMD's Advanced Space Technology Program as part of the Collaborative Decision Systems Project, and by ESMD's Exploration Technology Development Program as part of the Integrated Systems Health Management Project.

### References

- <sup>1</sup>Williams, B. C., and Nayak, P. P., "A Model-based Approach to Reactive Self-Configuring Systems," *Proceedings of the National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Menlo Park, CA, 1996.
- <sup>2</sup>Kurien, J., and Nayak, P. P., "Back to the Future for Consistency-based Trajectory Tracking," *Proceedings of the National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Menlo Park, CA, 2000.
- <sup>3</sup>Narasimhan, S., Dearden, R., and Benazera, E., "Combining Particle Filters and Consistency-based Approaches for Monitoring and Diagnosis of Stochastic Hybrid Systems," *15th International Workshop on Principles of Diagnosis (DX04)*, Carcassonne, France, June 2004.
- <sup>4</sup>Williams, B. C., Ingham, M., Chung, S., Elliott, P., and Hofbauer, M., "Model-based Programming of Fault-Aware Systems," *AI Magazine*, Fall 2003.
- <sup>5</sup>TEAMS-RT Web page. <http://www.teamqsi.com/RT.html> [cited 26 April 2007].
- <sup>6</sup>RODON Web page. <http://www.sorman.se/products/rodon/> [cited 26 April 2007].
- <sup>7</sup>Colgren, R., Abbott, R., Schaefer, P., Park, H., Mackey, R., James, M., Zak, M., Fisher, F., Chien, S., Johnson, T., and Bush, S., "Technologies for Reliable Autonomous Control (TRAC) of UAVs," *19th Digital Avionics Systems Conference*, October 2000.
- <sup>8</sup>Barrett, A., "Model Compilation for Real-Time Planning and Diagnosis with Feedback," *Proceedings on the International Joint Conference on Artificial Intelligence*, 2005.
- <sup>9</sup>Main Propulsion System. NASA Kennedy Space Center Web page. <http://science.ksc.nasa.gov/shuttle/technology/sts-newsref/sts-mps.html> [cited 26 April 2007].
- <sup>10</sup>Constellation Program: America's Fleet of Next-Generation Launch Vehicles - The Ares I Crew Launch Vehicle. NASA Fact Sheet FS-2006-07-85-MSFC, 2006. [http://www.nasa.gov/pdf/151419main\\_aresI\\_factsheet.pdf](http://www.nasa.gov/pdf/151419main_aresI_factsheet.pdf) [cited 26 April 2007].
- <sup>11</sup>Constellation Program: America's Fleet of Next-Generation Launch Vehicles - The Ares V Cargo Launch Vehicle. NASA Fact Sheet FS-2006-07-84-MSFC, 2006. [http://www.nasa.gov/pdf/151420main\\_aresV\\_factsheet.pdf](http://www.nasa.gov/pdf/151420main_aresV_factsheet.pdf) [cited 26 April 2007].
- <sup>12</sup>Schwabacher, M., "Machine learning for rocket propulsion health monitoring," *Proceedings of the SAE World Aerospace Congress*, Vol. 114-1, Society of Automotive Engineers, Warrendale, PA, 2005.
- <sup>13</sup>Bay, S. D. Schwabacher, M., "Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule," *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, 2003.
- <sup>14</sup>GritBot. RuleQuest Research Web site. <http://www.rulequest.com> [cited 26 April 2007].

<sup>15</sup>Iverson, D. L., "Inductive System Health Monitoring," *Proceedings of the International Conference on Artificial Intelligence, IC-AI '04*, Vol. 2 & *Proceedings of the International Conference on Machine Learning; Models, Technologies & Applications, MLMTA '04*, June 21-24, 2004.

<sup>16</sup>Tax, D. M. J. and Duin, R. P. W., "Support Vector Domain Description," *Pattern Recognition Letters*, 20(1113): 1191–1199, 1999.

<sup>17</sup>Fiorucci, T. R., Larkin II, D. R., and Reynolds, T. D., "Advanced Engine Health Management Applications Of The SSME Real-Time Vibration Monitoring System," AIAA paper 2000-3622, *36<sup>th</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, AIAA, Washington, DC, 2000.

<sup>18</sup>Martin, R. A., Schwabacher, M., Oza, N., and Srivastava, A., "Comparison of Unsupervised Anomaly Detection Methods for Systems Health Management Using Space Shuttle Main Engine Data," *Proceedings of the Joint Army Navy NASA Air Force Conference on Propulsion*, 2007 (to be published).

<sup>19</sup>Park, H., Mackey, R., James, M., Zak, M., Kynard, M., Sebghati, J., and Greene, W., "Analysis of Space Shuttle Main Engine Data Using Beacon-based Exception Analysis for Multi-Missions," *Aerospace Conference Proceedings*, Vol. 6, IEEE, Washington, DC, pp 6-2835–6-2844, 2002.

<sup>20</sup>Srivastava, A. N., "Discovering System Health Anomalies Using Data Mining Techniques," *Proceedings of the Joint Army Navy NASA Air Force Conference on Propulsion*, 2005.

<sup>21</sup>Bishop, C. M., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

<sup>22</sup>Guo, T.-H. and Musgrave, J., "Neural Network Based Sensor Validation for Reusable Rocket Engines," *Proceedings of the American Control Conference*, Vol. 2, 1995, pp. 1367–1372.

<sup>23</sup>He, F., and Shi, W., "WPT-SVMs Based Approach for Fault Detection of Valves in Reciprocating Pumps," *Proceedings of the American Control Conference*, 2002.