

Information Theoretic Anomaly Detection: A Space Shuttle Main Engine Application

Author 1 ¹, Author 2 ²

¹ School 1

² School 2

ABSTRACT

Automated model-free anomaly and fault detection using large collections of sensor suites is vital to increasing safety and reducing maintenance costs of complex aerospace systems, such as the Space Shuttle Main Engine. Current anomaly and fault detection methods are deficient in that they either require a huge amounts of laborious expert analysis or rely on models that fail to capture unmodelled anomalies. To overcome these deficiencies, model-free statistical approaches to this analysis are needed that do not require significant user input.

This paper presents two general automated analysis methods that detect anomalies in sensor data taken from large sets of sensors. The first approach uses entropy analysis over the entire set of sensors at once to detect anomalies that have broad system-wide impact. The global nature of this approach reduces its sensitivity to faulty sensors. The second approach uses automated clustering of sensors combined with intra-cluster entropy analysis to detect anomalies and faults that have more local impact.

Results derived from the application of these approaches to sensor data recorded from test-stand runs of the Space Shuttle Main Engine show that they can be effective in finding faults and anomalies. With test-stand data consisting time-series derived from 147 sensors, the system-wide approach was able to reveal an anomalous mixture ratio programmed by the test-engineers, but not revealed to the authors. Using similar data from a different engine test, the localized clustering approach revealed a fault in the high pressure fuel turbo-pump early in the test-run and subsequent cascaded faults later in the test run. In addition the clustering approach was able to separate sensors that contained little analytic value from more important sensors, potentially reducing the burden of subsequent expert analysis.

CONTENTS

1 Introduction	1
2 SSME Sensor Data	2
3 System-Wide Entropy	2
3.1 Approach	3
3.2 Results	3
4 Cluster Entropy	4
4.1 Approach	4
4.2 Results	5
5 Discussion	7

1 INTRODUCTION

Effective operation and analysis of the Space Shuttle Main Engine (SSME) is critical to the safety and long term success of NASA's manned space program (12). To help analyze performance of the engines, and to provide a snapshot of the health of each subsystem, dozens of sensors record SSME operating conditions on each shuttle flight. In addition, data from hundreds of sensors is recorded each time an SSME is fired on its test stand. Both in-flight and test-stand data can be used for long term issues such as maintenance, whereas the in-flight data is crucial for range-safety operations and automated abort attempts (4). In this paper, we focus on test-stand data which because of its more detailed nature is particularly useful for:

- Identifying anomalies that were previously unknown.
- Identifying occurrence of known intermittent anomalies.
- Gaining insight into known, but poorly understood anomalies.

of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

This is an open-access article distributed under the terms

The ability to automatically identify anomalies from large sets of sensor data can both greatly reduce the burden of analysts and lead to the discovery of subtle interactions among subsystems. The benefits of such an automated approach is to increase the safety of the shuttle and to decrease the maintenance costs of the SSME program. However, while highly desirable, the automated analysis of SSME test-stand data is difficult for the following reasons:

- The data set is hundreds of megabytes to gigabytes;
- The important sensor subsets for particular event detection are unknown;
- The time window of the first occurrence of an anomaly is generally unknown;
- The signals from dominant systems (e.g. throttle control) swamp most sensor readings;
- The sensors have random noise; and
- The sensors can fail or provide intermittent data.

Due to these difficulties, analysis of such complex data traditionally necessitates a tremendous amount of domain knowledge. Domain engineers have to identify important sensors and carefully filter data. This input by domain experts is laborious, and even worse introduces biases, which may prevent unknown anomalies from being detected. In this paper, we present two entropy-based methods that overcome the difficulties of analyzing complex sensor data, while minimizing domain expert input. The first method, *system-wide entropy*, involves looking at all of the sensors at once. This method detects anomalies that affect a significant part of the system, while being robust against sensor failure and avoiding the need to identify important sensors. The second method, *cluster entropy*, uses clustering methods on the sensor data, to partition sensors into related groups. This method is most useful for detecting failures that only affect a small subset of the sensors.

Section 2 describes the sensor data and the data collection process. Section 3 presents the system-wide entropy method, along with results showing how system-wide entropy detected an anomalous fuel mixture ratio from the SSME test-stand sensor data. Section 4.1 presents the cluster entropy method, and shows results where an engine anomaly that had only local impact was detected. Finally Section 5 provides a discussion on the impact of this work and identifies extensions of this work.

2 SSME SENSOR DATA

Before every mission, each Space Shuttle Main Engine to be used on a mission is tested at Stennis Space Center. Each test consists of firing the engine for 520 seconds while taking readings from sensors located on the engine. The sensor readings are recorded as time series in a database. Each test ultimately culminates in hundreds of separate time series, sampled at small fractions of a second. This paper uses a subset of this data composed of a 147 time series sample every 0.04 seconds. Each time series has a total 13,000 samples, with the entire data set having 1,911,000 values. Each time series comes from a single sensor, though there

may be multiple time series for a sensor corresponding to multiple readings of the sensor data coming from separate wires (4).

The data used in this paper comes from sensors measuring important properties at many different locations within the SSME. These properties include temperatures, pressures, turbo-pump rpms and vibration. The goal of this paper is to find anomalies by identifying “unusual” though not necessarily alarming activity in the test-stand data. In particular, we are interested in how the correlations among the different sensors vary across time. The hypothesis we investigate is that the subtle interactions among the sensors provide early warning, before the anomalies fully manifest themselves. However, these subtle interactions are usually masked by many known phenomena that affect nearly all of the time series. One of these is the command throttle level of the engine (see Figure 1). As such, any analysis method that will identify unknown events has to first sift through the known events with large impact on all the sensors. Furthermore, not all sensors are reliable throughout a test firing. Therefore analysis methods cannot be over-dependent on individual sensors. Finally, there is inherent random noise in the sensor readings, so analysis methods cannot depend on extremely small changes in individual sensors, since small changes will be dominated by the noise. The challenge is to tease out small subtle changes in system behaviors occurring across many sensors, from both the noisy fluctuations of individual sensors and the large scale impact of a handful of known system-wide phenomena.

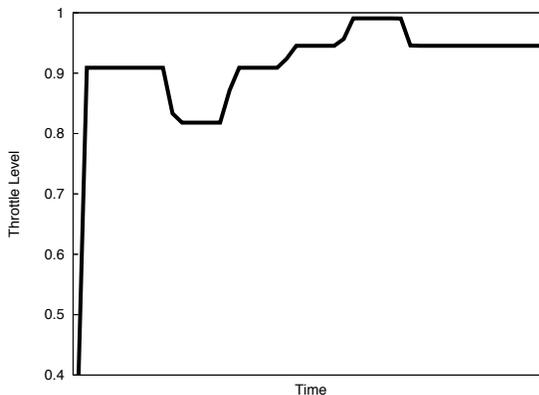


Figure 1: **Commanded Throttle Level of SSME.** Events such as a change in throttle level dominate activity in time-series data.

3 SYSTEM-WIDE ENTROPY

Both of the proposed methods are based on Shannon entropy, which intuitively provides a measure of the uncertainty remaining in the system after an observation has been made (3). The entropy of a continuous random variable X with probability density function $p(x)$, is given by:

$$H(X) = \int p(x) \log_2 p(x) dx. \quad (1)$$

Though a valuable statistical measure, the entropy of a string is a poor predictor of anomalies in the system generating that string, because it focuses on randomness. As a consequence a sensor reading pure white noise would have the highest entropy and a sensor reading a flat line would have zero entropy. Unfortunately, most interesting phenomena occur somewhere between these two extremes, meaning that neither a rise nor a drop in entropy necessarily mean that an anomaly is detected. Furthermore, entropy only focuses on the frequency of the observed symbols, not their ordering. As such a periodic binary string and a random binary string containing the same number of ones and zeros have the same entropy.

3.1 Approach

To overcome some of these limitations of entropy, in our first method of analyzing Space Shuttle Main Engine sensor data, we examine all of the 147 sensor values at a particular time at once. This method uses entropy to provide a “signature” for the system at that instant of time. The assumption of this analysis is that when an anomaly occurs that affects the values of many sensors, the signature of the sensor signals will change, even if the change in each sensor is within the noise threshold for that sensor. Because it processes all of sensors at once, this method is robust against failures of individual sensors and resistant to random noise present in the data.

Before computing the system wide entropy for the system, we perform a series of pre-processing operations as shown in Figure 2, which consists of:

1. Smoothing the time-series data for each sensor. This is done by averaging time-series values together in groups of twenty. This operation results in time series that have values every 0.8 seconds instead of the original 0.04 seconds.
2. Normalizing each time-series by dividing each value by the average value in the series.
3. Discretizing the time-series values into sixty-four uniformly sized bins.

Once each time-series has been preprocessed, the Shannon Entropy is computed for each time-step (10). This is done by first forming a vector S of size 147, containing all the pre-processed sensor readings for a time-step. The Shannon Entropy for the discrete string S of length n (obtained as described above), is given by:

$$H(S) = \sum_{i=1}^n \frac{n_i}{n} \log_2 \frac{n_i}{n} \quad (2)$$

where n_i is the count of number of times value i appears in S . When most of the elements in S are equal to a small subset of the total possible values, the entropy is low. This often happens when most of the sensors are in highly correlated groups, since the normalized values in these groups will tend to be the same. When the elements in S tend to take a wide range of possible values, the entropy is high. This often happens when many variables become uncorrelated, or there is lots of noise induced in the system. Note that entropy is insensitive to noise coming from a small subset of sensors, since each term in the summation

is weighted by $\frac{n_i}{n}$. If a sensor erroneously takes on a common value, then n_i is large and the single sensor has little impact on the term. If a sensor erroneously takes on a uncommon value, then $\frac{n_i}{n}$ is small and the entire term has little impact on the entropy.

3.2 Results

We tested the performance of the system-wide entropy method using three data sets from three separate test-stand firings. All three test-stand firings were conducted for the same amount of time, but each test used different throttle levels. On the first test no known anomalies occurred, but there were some sensor failures. On the second test there was a failure in the high pressure fuel turbo pump about one quarter of the way through the test, that caused an additional cascade of failures later in the test. There where no known sensor failures in the second test. In the third test there were no sensor failures and no unintended anomalies. However, there was an unusual commanded change in the fuel to oxidizer ratio.

Figure 3 shows the values of system-wide entropy for the first two tests. In these tests the system-wide entropy responds mostly to changes in the command throttle control. This is not surprising since the throttle level is the most important control in the engine, and it has a system-wide impact over most of the sensors. Unfortunately, because of the dominance of the throttle control, anomalies close to changes in throttle level can go undetected. This is the case in the second test, where the failure in the high-pressure fuel turbo pump went undetected, since it occurred close to a throttle transition.

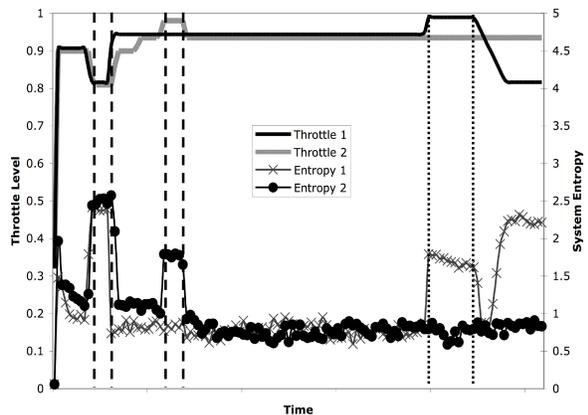


Figure 3: **Results for System-Wide Entropy.** System-wide entropy is strongly influenced by changes in throttle level, but has trouble detecting failures close to these changes.

However in the third test, the system-wide entropy detected an anomaly as shown in Figure 4. In this test the oxidizer to fuel ratio was changed from 6.0 to 5.7. Even though this constituted a small change, it had a large system-wide impact and was clearly detected by the system-wide entropy. The importance of this discovery is in that this change in mixture ratio was unusual, and was not included in the known anomalies for this data set. Only after detecting it, discussing

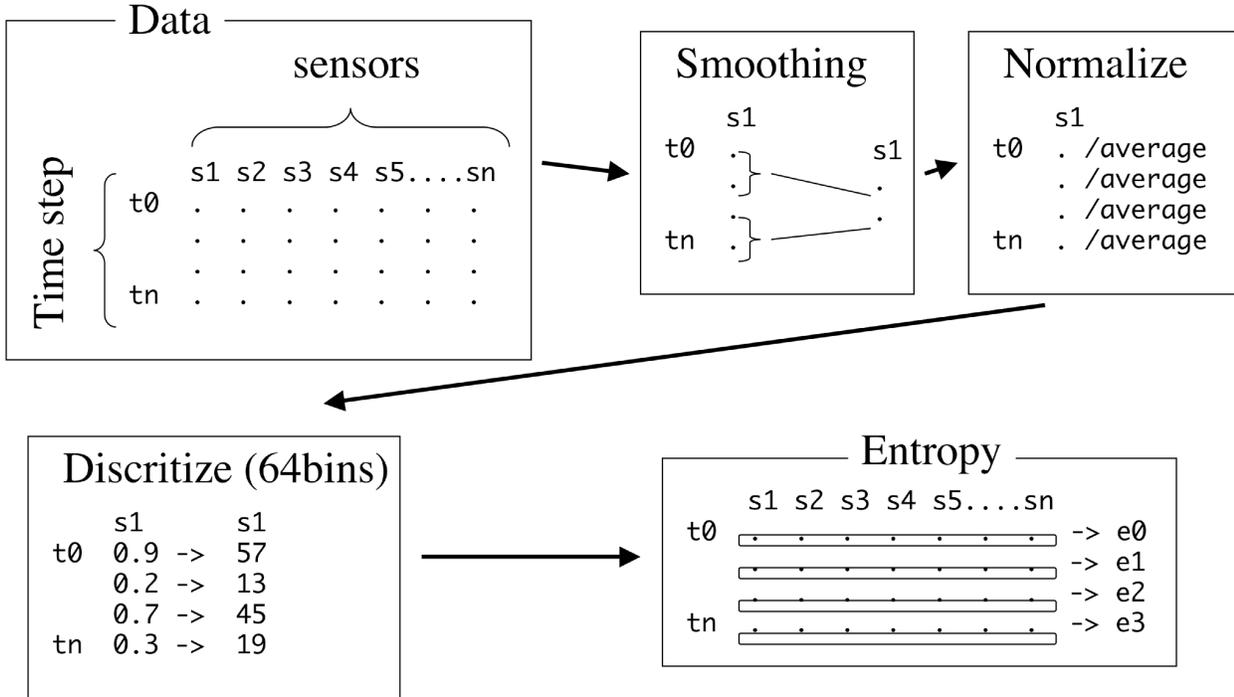


Figure 2: **System-Wide Entropy**. After preprocessing, the system-wide entropy for each time-step is computed by taking the entropy of all of the discretized sensor values.

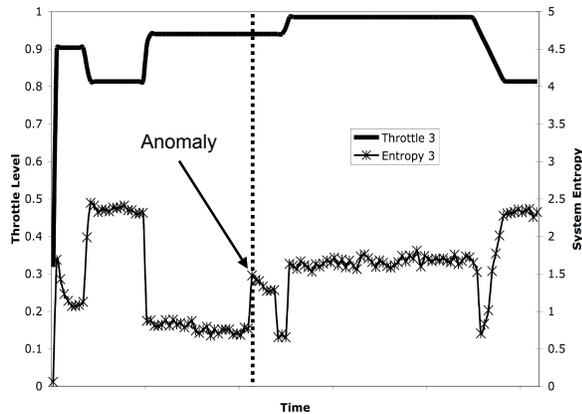


Figure 4: **Results for System-Wide Entropy**. System-wide entropy detects an anomalous change in the fuel-to-oxidizer ratio.

it with the domain experts and looking at the internal records of that particular test did we identify it as a fuel mixture change event. As a consequence, this event was an “unknown” anomaly that was detected by the method.

4 CLUSTER ENTROPY

While system-wide entropy can detect events that have system-wide impact, it has difficulties detecting events that have a more localized impact. In addition since in many cases there may be many system-wide events occurring simultaneously, system-wide entropy may not

be able to discover anomalies that co-occur with other events. To address these issues we introduce cluster entropy, which is designed to find anomalies that are specific to certain subsystems and their impact is spread over a small subset of correlated sensors. Furthermore, a secondary benefit of cluster entropy is in identifying sensors that are most relevant in finding specific anomalies.

4.1 Approach

Cluster-entropy is based on clustering groups of related sensors together to isolate their impact (8; 2; 5). In a good clustering, two sensors from the same cluster are more correlated with each other (by some metric) than sensors coming from different clusters. In this paper the distance between two sensors is computed by computing the Pearson Correlation between the time-series data coming from the sensors (9).

The process for computing cluster-entropy is summarized in Figure 6, and consists of:

1. For each of the n sensors, compute the correlation between each pair of sensors and form an n by n distance matrix. This matrix represents an adjacency graph, where sensors that are closer together are linked by edges of greater weight.
2. Cluster the sensors by performing a graph-partitioning of the adjacency graph. The graph-partitioning algorithm used is Metis which tries to minimize the aggregate weight of edges between partitions (6). This partitioning method produces a clustering that has strongly weighted edges between nodes in the same cluster and

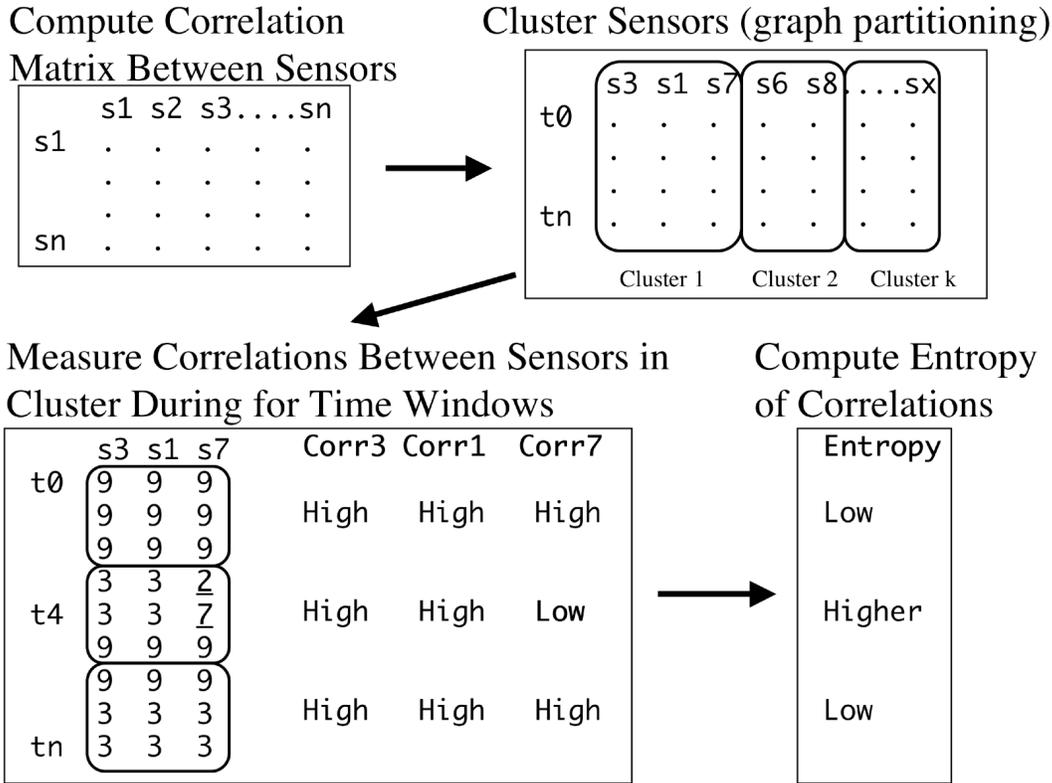


Figure 5: **Cluster Wide Entropy.** First a correlation matrix is computed between each pair of sensors. Second graph partitioning is performed from the matrix to cluster the sensors. Third correlations are computed within a cluster for a window of time. Fourth the entropy of these correlations within a cluster is computed.

weakly weighted edges between nodes of different clusters. Since the edge weight represents the correlation between sensors, the partitioning algorithm produces our desired results of having sensors within a cluster being highly correlated.

3. Perform a “time-windowed correlation” within each sensor cluster. For a cluster of m time-series, this operation is done by computing m correlations between an arbitrary time-series in the cluster and all the time-series in the cluster (including itself, which is always equal to one). These correlations are computed only using the parts of the time-series that are within the time window.
4. Discretize the m correlations
5. Compute the entropy of the m discrete values to provide the cluster entropy for the system at time moment in time.

Because all the time-series within a cluster should be highly correlated, we expect the time-series within a particular time-window would also be highly correlated. However, if an anomaly occurs during the time window, some of the time series could become uncorrelated, which would be detected in the entropy measurement.

4.2 Results

To test the effectiveness on cluster entropy we conducted three experiments using the second data set used in the system-wide entropy test. This data set came from a test-stand firing where there was a failure in the high pressure fuel turbo pump (HPFTP). In each of these experiments the method for forming the clusters were different. In particular, the three experiments were based on:

1. Forming a single clustering by using all the time-series;
2. Forming five clusters based on the sensor correlations across full time-series; and
3. Forming five clusters based on the sensor data from time periods before and after changes to throttle levels.

The objective of the first experiment was to investigate whether the turbo pump failure was detectable without using graph partitioning and sensor clustering. The second experiment aimed to determine whether forming sensor clusters did indeed separate the high impact system-wide effects from specific anomalies. Finally, the last experiment aimed to determine whether there are critical times in the operation of the SSME where the correlations among the sensors are more significant than at other times and whether forming clusters based on correlations at those critical times improves anomaly detection.

The results of the first experiment are shown in Figure 7. They indicate that this entropy value is highly dominated by the throttle transitions. This is to be expected since the single cluster contains data from all of the sensors, many of which are influenced by the throttle transitions. Not surprisingly, the entropy values computed from this single cluster were not able to detect the failure in the HPFTP because the effects of this anomaly were dominated by the effects of the throttle level transitions. What is needed is a clustering that will isolate the impact of the throttle and allow sensors containing information pertinent to the HPFTP to detect the anomaly.

In the second experiment, the sensor readings were clustered into five clusters. Figure 8 shows the values of the entropy computations for each cluster. The figure shows that clusters 2 and 3 were dominated by the throttle transitions. Upon examining the sensors placed in these clusters, it was clear that the clustering algorithm had successfully put similar sensors into the same clusters. In this case the clusters included sensors readings, such as turbo-pump pressures that were indeed strongly affected by the throttle level. In contrast, clusters 4 and 5 exhibit little activity at all. These clustered showed little activity because they contained sensors that did not change much, such as valves that opened once and stayed opened. On the other hand, cluster 1 showed activity, but was not dominated by the throttle level. As a consequence, this cluster detected the HPFTP failure in that the entropy of this cluster rose well above the fluctuations at the moment that pump failed. In fact the highest activity shown in this cluster corresponds to a fault in the high pressure turbo pump.

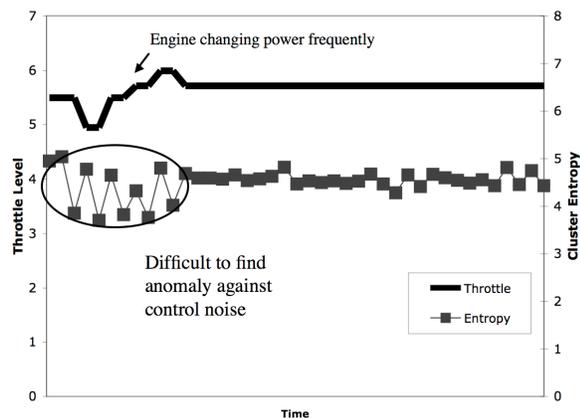


Figure 6: **Results for Cluster Entropy Method, with One Cluster.** Here the cluster entropy is computed, but all of the sensors are in one cluster. Values from this single cluster are not useful since they are dominated by the changes in throttle level.

Figure 9 shows results for the third experiments where only sensor readings before and after throttle changes are used in clustering the data. Interestingly, this method of clustering indeed reveal additional anomalies. In this case the times of high activity of clusters 1 and 2 show anomalies in the data. These anomalies corresponded to additional failures in

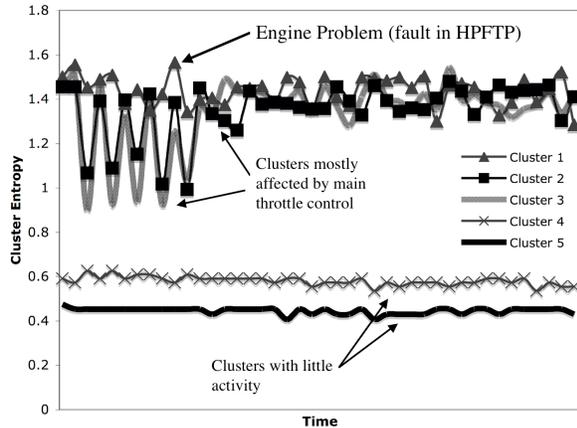


Figure 7: **Results for Cluster Entropy Method, with Five Cluster.** Sensors are divided into five clusters and time-windowed entropy is computed for each cluster. Sensors in clusters 2 and 3 are clearly dominated by the change in throttle. Clusters 4 and 5 show little activity. Cluster 1 shows most activity at time of HPFTP failure.

the HPFTP caused by the initial failure. Also shown is how these increased activity levels in the automatically generated clusters match the time of the activity shown in a sensor variable, hand picked by a domain expert. This result suggests that the cluster-entropy method can be used not only to detect faults, but also to identify sets of sensors are useful in detecting faults.

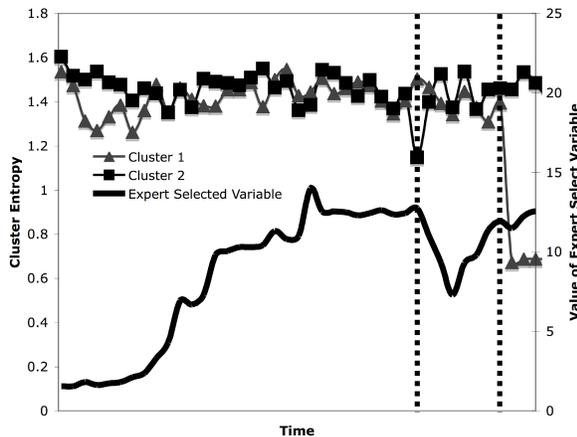


Figure 8: **Results for Cluster Entropy Method** Clustering algorithm divides sensors into five clusters, using only data coming from time periods close to throttle transitions. Two of the clusters show high levels of activity matching activity levels of sensor chosen by an expert in being important in revealing anomaly. This activity corresponds to secondary failures in HPFTP.

5 DISCUSSION

This paper demonstrated the value of entropy based data analysis for detecting anomalies in complex aerospace systems such as the Space Shuttle Main Engine. The first method presented, system-entropy,

showed that anomalies could be detected when they had a system-wide impact. Since system-wide entropy examines all of the sensors at once, it may be the best method when there is a high level noise in the data, or when sensors fail. The second method, cluster-entropy, showed how anomalies that had a more local impact could be detected against the noise of other known events that have a large impact.

In addition to detecting anomalies, the cluster-entropy method is also useful in finding sets of sensors that are helpful in finding anomalies. Since these methods are automated, they require minimal human intervention, alleviated the effort required by data analysts. In addition, since little input was given to the methods, they tend to be unbiased and can potentially find anomalies that would not be found otherwise.

REFERENCES

- (1) Y. Bar-Yam, editor. *The Dynamics of Complex Systems*. Westview Press, 1997.
- (2) Christopher M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1996.
- (3) T. Cover and J. Thomas. *Information Theory*. John Wiley & Sons, New York, NY, 1991.
- (4) M. Davidson and J. Stephens. Advanced health management system for the space shuttle main engine. In *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Fort Lauderdale, FL, July 11-14 2004.
- (5) R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd ed)*. John Wiley & Sons, New York, NY, 2000.
- (6) G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.
- (7) B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 42(2):291–307, 1970.
- (8) J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.
- (9) W. H. Press, B. P. Flannery, S. A Teukolsky, and W. T. Vetterling. *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, chapter 14.5, pages 630–633. Cambridge University Press, Cambridge, England, second edition, 1992.
- (10) C.E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois, 1949.
- (11) A. Srivastava and W. Buntine. Predicting engine parameters using the optical spectrum of the space shuttle main engine exhaust plume. In *Computing in Aerospace Conference*, pages 29–34, San Antonio, TX, 1995.
- (12) Jim Dumoulin (web contact). *NSTS 1988 News Reference Manual*. Kennedy Space Center - <http://science.ksc.nasa.gov/shuttle/technology/sts-newsref/>, 1988.
- (13) B. Whitehead and W. Hoyt. A function approximation approach to anomaly detection in propulsion system test data. In *AIAA/SAE/ASME/ASEE 29th Joint Propulsion Conference*, Monterey, CA, June 1993.
- (14) Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2005.