Data Mining For Space Applications

Wendy Foslien, Valerie Guralnik, Karen Zita Haigh

Honeywell Laboratories, 3660 Technology Drive, Minneapolis, MN 55418

1 Introduction

The complexity of operations, the nature of the space environment and the inherent inaccessibility to the spacecraft make early detection of any off-nominal condition critical to mission success. Currently, most spacecraft contain Diagnostics and Failure Detection Isolation and Recovery (FDIR) software to detect and contain anomalies. However, most of these applications detect only pre-defined failure conditions; they are triggered when pre-set limits or conditions are violated.

Monitoring and diagnosing evolving off-nominal conditions that may become critical anomalies is normally assigned to system experts and analysts on the ground. Space Shuttle Launch Control Center controllers, for example, monitor real-time data and compare its performance to previous flights while the Shuttle is on the launch pad minutes before launch. If any data deviates from previous performance, the controllers must detect it, then determine if it is a significant event or within normal parameters. When the off-nominal condition is propagated across several systems, this task becomes more complex.

In crewed spacecraft, like the International Space Station (ISS) or Space Shuttle, the problem is more acute. The on-board crew does not have time to perform continuous system monitoring or the necessary system expertise to diagnose complex and evolving off-nominal conditions. Success depends on the knowledge and expertise of the mission control center and engineering analysts monitoring and operating the spacecraft from Earth.

To achieve space program goals ranging from reducing the cost and complexity of mission operations to sending manned spacecraft to places where the real-time operation of the spacecraft from Earth is not possible, we must reduce dependency on human experts. Automating Early Event Detection (EED) is necessary for reducing this dependency. The algorithms used in EED are not domain specific; domain expertise is needed to select appropriate data and prioritize the importance of detected events. Models for EED can be derived from operational data in archives, such as the telemetry data from the ISS.

This paper describes how we used data mining and modelling methods to explain and explore anomalous conditions in the ISS's Beta Gimbal Assembly (BGA). The BGA is the subsystem that points the solar panels toward the sun. We received 500 hours of BGA data that includes baseline data plus 25 abnormal events. The data consists of 73 Program Unique Identifiers (PUIs) from the two BGAs. Nominally, the motor current should be 0.2 amps; abnormal events occur when the current is above 0.5 amps. The motor trips and turns off at 1.1 amps.

2 Data Mining Applicability to Early Event Detection

The core of many EED systems is a set of algorithms that monitor system state. An EED system may have goals that detect anomalies, localize them to specific areas, and identify abnormalities. EED algorithms are usually based on engineering knowledge of the problem and hand generated. They cannot identify problems outside prescoped knowledge. Data mining algorithms can sort through massive amounts of data to locate key features, identify key data segments, or directly train models for event detection. Data mining techniques can better use the vast archive of operational data to formulate models for early event detection. Data mining is essentially the process of automatically extracting valid, useful, previously unknown, and comprehensible information from large databases and using it to make crucial decisions. Data mining techniques are drawn from machine learning and artificial intelligence, pattern recognition, statistics, database systems, and data visualization. They are geared toward applications where traditional techniques may not be suitable, for example because of the enormity of the data, the high dimensionality of the data, or the heterogeneous, distributed nature of the data. Data mining techniques examine the collected data for unusual situations and the results can be used as a basis for an EED algorithm to detect unexpected problems. Usually, a data mining technique will learn a model of the normal behavior of the system, and then the run-time algorithm will detect situations outside normal. Models can also be built for known, abnormal conditions. Key challenges for data mining algorithms in this domain include:

Sampling rate: different parameters are sampled at different rates. The sampling rate is not always consistent.

Noise: sometimes sensors will report incorrect values, or communications streams will be garbled.

Lag time: often, there is a delay between the anomaly occurrence and when it is reported, especially if the data is sent from space to a ground station.

Missing values: sensors may go off-line.

Volume: this data generally has high dimensionality, much of it not relevant.

Coverage: it is unlikely that all modes of all parameters can or will be collected.

3 Processing the BGA data

The data mining process includes five tasks: data selection, cleaning to make the data consistent and remove outliers, reduction and transformation to reduce the effective number of variables under consideration, analysis to detect patterns and extract knowledge, and interpretation to verify the results and possibly use them in context. A common rule of thumb is that data preparation (selection, cleaning, and reduction) will be about 60% (or two-thirds) of the total effort expended.

3.1 Data Selection and Visualization

Since analysts are key players in selecting the right data and the right tool for the job [3,8,12], they must understand the data. There are many techniques for visualizing data. One example of a visualization method is Honeywell's Visual Query Language (VQL) [7], which we used to visualize the BGA data. VQL is a technology for locating time series patterns in historical or real-time data. VQL integrates user specifications of visual patterns with an automated search of a historical database for those patterns. With VQL, users can define qualitative patterns in the data graphically or select previously defined templates and then express the selections as search directives. The VQL search engine uses trend-oriented algorithms to find similar patterns in the data stream and returns a ranked list of matches.

VQL allows the user to specify how insensitive the matcher should be to variations in temporal length and feature amplitude. For example, if the feature shape were a single cycle of a sine wave, the VQL engine would locate all instances of a sine wave cycle in the data set. This would include data segments where the period and amplitude were somewhat different from the original feature.

Patterns may exist on single variables or multiple variables with temporal constraints on segment positions. Figure 1 is an example of the BGA data read into VQL. Two PUIs are shown on the same time axis. We select the feature of interest (grey box), and want VQL to find other places in the data set where we can see similar behavior in the same variable. The best match for the original template is shown in Figure 2, and the tree view (left) indicates that there are at least 13 other similar matches.

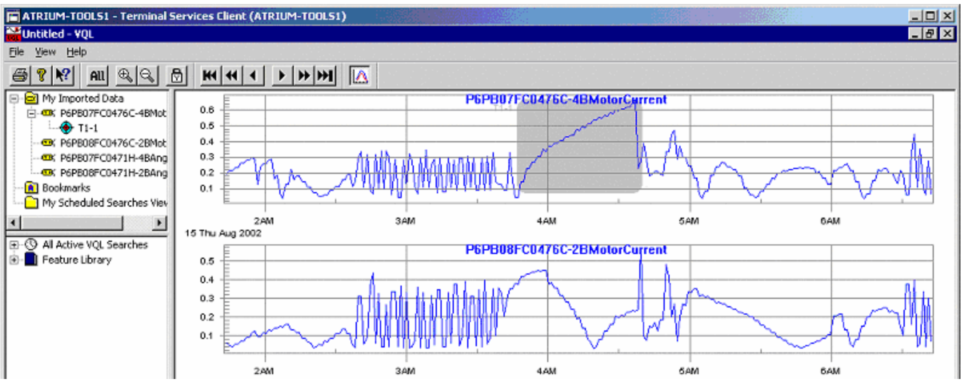


Figure 1: BGA data with selected feature

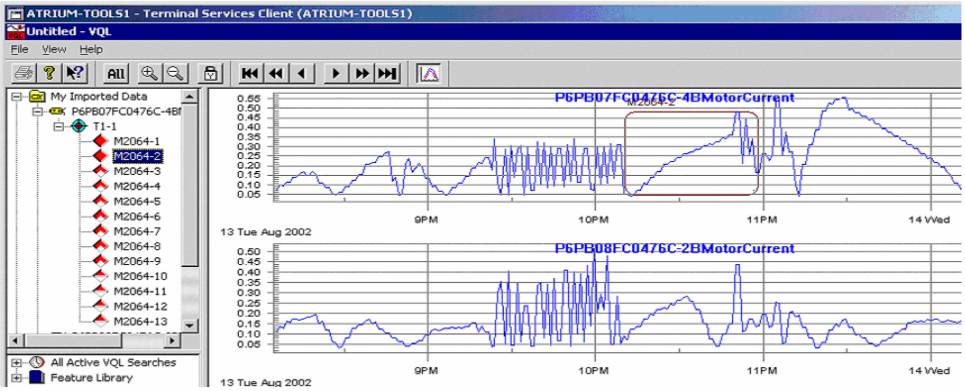


Figure 2: Best match to the feature selected in Figure 1.

In our work, we used VQL to identify segments of data that contained high-current events. In addition to confirming existence of known high-current events, VQL found two events in data considered ``nominal'' - that is, our domain expert had informed us that the data was collected from before the first problem was noticed. After reviewing our findings, the domain experts agreed that the events found by VQL were indeed high-current events.

Once the analyst has an understanding of the data, she can select which PUIs to record for analysis, at what sampling rate and over which time windows. She can also make a decision about if and how to segment the data (e.g. modes of operation, ab/normal, etc).

3.2 Cleaning

Data cleaning makes the selected data consistent and removes outliers. Time series data requires special cleaning considerations. Dynamic models generated in the analysis will rely on consistent time sampling in the data. In the BGA data, the sampling varies considerably throughout the data sets. Methods that rely on past information will give erroneous results when skewed or varying sample rates are used. Missing data points can cause havoc to some prediction engines.

To clean the data, we determine the criteria for assessing whether each data entry makes sense. We then decide how to handle entries that fail the test against our criteria: Do we throw out the entire row containing the entry? Do we replace the single entry with a value from a model? Do we replace the entry with the last sensible value we observed for that entry? These decisions depend on our knowledge of the data. For example, if we knew that the process of interest followed a repeatable pattern, e.g. daily cycles, we could create a model of “typical” daily behavior from the valid data we have. This “typical” model can then be used to fill in erroneous or missing data. In contrast, if we don't have a repeatable pattern in the data, we might fill in the data with the last (recent) valid point.

We know that different PUIs have different sampling rates and that the sampling rates vary over time. We also were fortunate to know the (official, approximate) sampling rates. After examining the data, we discovered three predominant rates. We then used linear interpolation or a zero order hold to regularize the data at the official, unvarying rates by: (1) replacing missing data with NaN, (2) keeping the values in as NaN values for the original interpolation, (3) running a function to substitute values for the NaN entries, and (4) using the last good value to fill in the NaN entries. This process produced flat line data over the ranges where NaN entries were found. We made this choice based on knowledge of how selected tools would interpret the results--other analysis methods have different requirements.

The cleaning process presented here is a simple one, which has some drawbacks, particularly in the method of filling bad data points. In future work, a thorough analysis of how the data should be prepared for analysis is critical. We advise using a more detailed study of methods for resampling and imputation of missing data points for this application.

3.3 Transformation and Reduction

Data transformation and reduction limits the effective number of variables under consideration both for presentation to the user and for analysis by data mining algorithms. One way to transform and reduce data is to detect relationships between PUIs.

Correlation analysis is a simple method that measures relationships between two variables, e.g. motor current and angle error. Highly correlated variables can be treated as redundant in the analysis step. However, blindly applying correlation analysis may give misleading results. For example, two PUIs that strongly correlate during normal operation may not correlate during the fault mode. Moreover, different modes of operation of BGA may affect relationships between PUIs. Correlating variables from different modes may lead to a conclusion that that BGA PUIs measure completely independent processes because of the low correlation between variables. Another drawback of correlation analysis is that it compares only two variables at a time, while in many processes it is likely that PUIs may belong to groups of strongly correlated variables.

Principal Component Analysis (PCA) [13] is a natural way to group PUIs. PCA has been successfully applied to time series data from industrial processes [18,14]. We found two issues in using PCA for transforming and reducing BGA data. First, PCA assumes that the data set is linear, has a stationary, normal distribution, and has comparable magnitude and coverage for variable-variances. These assumptions are not true for much of the BGA data. We can ignore some PUIS, detrend the selected PUIs, partition PUIs into sub-systems, and scale values to address some of these issues; however, the process of creating PCA models can become too human-involved and defeats the purpose of data mining. Second, PCA is well suited to continuous processes, where changes in measurements are gradual. For these gradual changes, the data can be detrended using a moving average. This is not true in the BGA, which operates in different modes. Partitioning the data into subsets corresponding to distinct modes solves this problem, but creates the problem of data coverage.

3.4 Analysis: PCA

We experimented with PCA for BGA data analysis as well as dimensionality reduction. A PCA model constructed for normal data can identify abnormal behavior in new data. A metric, called the Q statistics, measures how well the PCA approximates the correlation structure in the observed data. An unacceptably high Q statistic indicates aberration in the measurements (which might be caused by a fault or process problem). The PCA model can then be used to identify the major contributors to the excursion in the Q statistic. We did not use Hotelling's T2 [13] to detect abnormal situations.

For the BGA data, the principal component model was constructed using ``normal" data from the BGA using two days of data with no high current events. The validation set consisted of three days of data from the BGA with high current events included. The objective of the PCA model was to predict the “abnormal" behavior as measured by deviations in the Q statistic.

We built a series of PCA models, each time removing parameters that caused high prediction error measured by the Q statistic. In the best model, higher values of the Q statistic generally corresponded to higher values of the motor current. However, this was not consistent, with some cases of the high Q residual corresponding to low values of motor current. Thus, none of the PCA models tested were able to provide consistent prediction of the high current event.

3.5 Analysis: Tree Learning

Classification and Regression Tree (CART) [4] algorithms are a form of binary recursive partitioning on the data: each branch in the tree is a test, and each leaf node is a prediction. Classification trees usually have classifications of data at the leaf nodes, while regression trees usually have values.

CART makes no assumptions about the underlying distribution of the values of the predictor variables and therefore can handle skewed or multi-mode data well. It is also relatively simple to use, and very easy to interpret; in particular, rules can easily be generated from the trees.

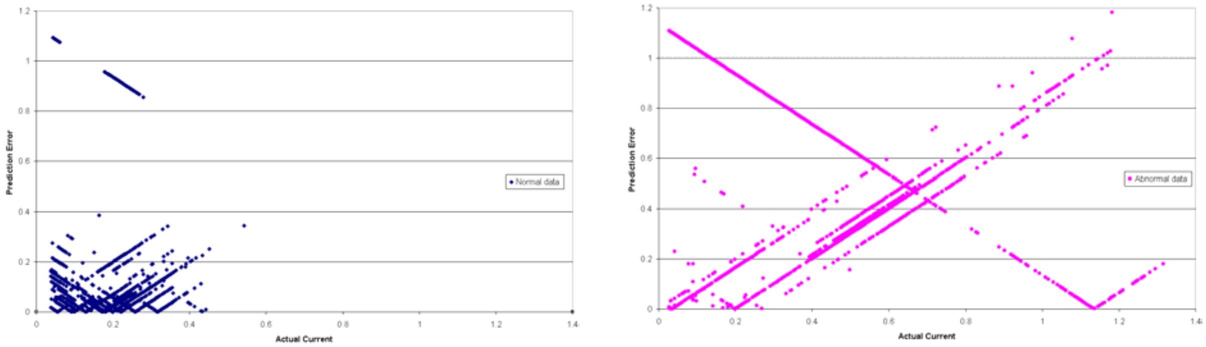
	Total Error	*N*	*Avg Error*
Normal Data	22904.4395	290017	0.078976 Amps
Abnormal Data	34945.3387	65579	0.532874 Amps
All Data	57849.7782	355596	0.162683 Amps

Table 1: The tree trained on both normal and abnormal data could not identify key predictors for abnormal data.

The structure of the regression tree we built on normal data matched the PID controller for the motor. It predicted normal currents with very high accuracy (less than 0.05 Amps error). However, when we built a regression tree from both normal and abnormal events, it did not discriminate among events, generating an average error of 0.16 Amps. Table 1 shows that almost all the error comes from predictions on the abnormal data, but Figure 3 shows that we cannot use high prediction error as an indicator of an upcoming failure because normal data frequently has high prediction error, and abnormal data frequently has low prediction error.

The implication of these results is that key predictor features (notably, those that show root cause of the problem) were not available in the data set.

This technique quickly homed in on the relevant features (essentially reconstructing the PID controller). Because of this capability, we expect regression trees to be useful for pre-processing to significantly reduce the data set. The greatest positive effect for our analysis was that, from raw data, we discovered the controller's basic operating principles-without first knowing its internals.



(a) Tested with normal data (b) Tested with abnormal data

Figure 3: Predictive error for a regression tree built from both normal and abnormal data.

3.6 Analysis: Regression Models

Linear Regression builds a function *y = f(x)*, where *x* is the set of predictor features, and y is the single value we'd like to predict [3,16].

We built a linear regression on our first data set. To verify the accuracy of the model, we examined the residual plots, the normal probability plots, and the histogram. Each of these graphs showed significant problems in the model.

Figure 4 shows the residual values between the actual current and the model's prediction. In a ``perfect" model (i.e. one that predicts the exact value every time), both graphs would show a horizontal

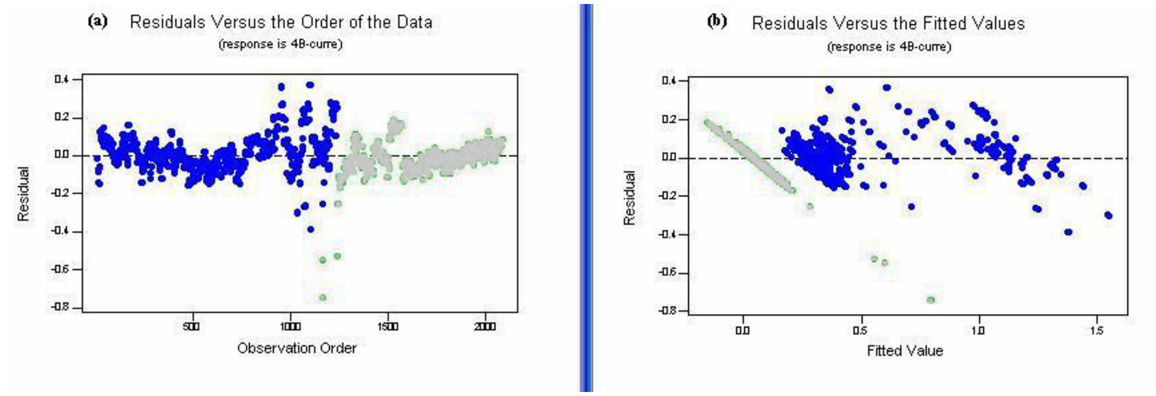


Figure 4: Residual error plots show an incomplete model.

line centered at zero (i.e. no error). In a ``good" model, you would expect to see an apparently random scattering of points. Since neither of these graphs met expectations, we took a closer look. Note that the apparent upper-left-to-lower-right slope of the data in Figure 4b particular the grey points, but also elsewhere. In Figure 4a the same points are marked grey, and you can see that there is a close correspondence to the trip event shown in Figure 5.

These observations strongly indicate missing features, and so we requested more data from our NASA expert. The additional data contained the Orbit Time of the station, which is a cyclic value. The new regression model contained this feature and completely eliminated the sine wave. However, we could not eliminate the slope problem.

In general, we cannot expect every predictive feature to be available, so our next step was to determine whether this incorrect model could be useful, i.e. that it has different prediction results for normal and abnormal events. Figure 6 shows the prediction error--dark points show error for normal operations, while grey points show abnormal events. The conclusion is that we can predict that the motor current will trip well before the current reaches 1.1 amps; even at 0.2 amps (x axis), when all hand-built tests would say things look nominal, we have a significant difference in prediction error.

These results show that even if the model is not correct, we can determine error conditions that hand-built modelling techniques are unable to detect.

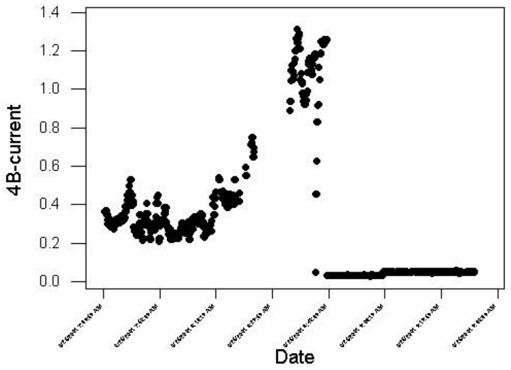


Figure 5: Motor current with a trip event.

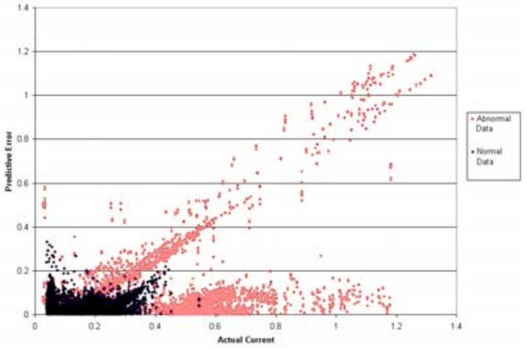


Figure 6: Even an inaccurate regression model can identify abnormal events.

3.7 Analysis: Temporal relationships

Temporal relationships between (sets of) variables can be established through sequential patterns discovery [1]. Sequential patterns are sub-sequences that occur in a sufficient number of sequences of data set. Discovered sequential patterns can be used to find rules that distinguish normal and abnormal operation of a device [19]. Unfortunately, sequential patterns can only be learned from categorical data. Time series need to be pre-processed into data sets of sequences of discrete events. Some domains lend themselves naturally to meaningful discretization, for example, by using a probability distribution function to identify bursts of activity in the Independent LifeStyle Assistant [10]. In BGA data, values of PUIs are usually uniformly distributed. Under such conditions PUI values can be discretized by diving sorted values into sets of equal size. This method creates a very dense data set in which most permutations of events are frequent. Not only does this approach make sequential patterns discovery intractable, but it does not create a data set in which significant temporal relationships can be discovered. A more sophisticated approach can be applied to convert continuous BGA data into event sequences. The candidates are change-point detection [2] to identify events responsible for changes in data and clustering data partitioned through a sliding window [6]. Unfortunately, all these methods require the user to set parameters; changes in the parameters significantly affect results.

4 Needs and Lessons Learned

Our experience with the BGA data demonstrated that while we could gain some insights into the data, significant challenges remain. We set out with the following goals:

- minimizing the amount of human guidance required
- generating models that are easy to understand (and use)
- handling both categorical data and continuous time-series data

We highlight here several categories of related and potentially relevant work. One category of related work includes those methods that make important statistical assumptions about the data being mined. For example, Liu et al [17] augments Box-Jenkins seasonal ARIMA models [3] with automatic verification and outlier detection to analyze and forecast the time series. This approach relies heavily on the predictable periodicity of the data. The BGA data does contain periodic elements related to orbit, but unfortunately these effects are complicated by the addition of mode changes on top of the periodic behavior. The PCA based process monitoring approach in Mylaraswamy [18] relies on slow movement of the process of the process mean relative to abnormal behavior. Linear dynamic model approaches, e.g. [17], also rely on consistent sampling rate, which we found to be problematic for the BGA data.

A second class of related work comes from the temporal data mining community. In general, the technique is to transform time-series into a categorical data-set by labeling features in time-series data, and to then use a categorical data mining technique to find significant patterns in the data. Höppner [11] developed a method that finds patterns from labeled time-series data, Das et al [6] used a sliding window to partition time-series and cluster partitons. Geurts & Wehenkel [9] developed temporal decisions in which nodes represent thresholds with delays.

These techniques are extremely attractive because they do not make any assumptions about the data. However, they have numerous limitations. One limitation is that these techniques are highly-dependent on user-set parameters (e.g. temporal window length and magnitude limits). In addition, none of these techniques take into account the distance between events. For example, two events pattern ignores the time difference between α and β. They are also limited by labelling; for example two local minima will be given the same label, even though they have different properties. Figures 1 and 2 show a particular shape and its best match; this ``degree'' of similarity is ignored by these algorithms. Exacerbating the problem is the fact that these labels need to be defined a priori. Another serious limitation is that these techniques learn frequent patterns; in our data, we are usually interested in detecting and avoiding situations that are infrequent.

One interesting approach for timeseries forecasting is data-centric modelling [15]. This approach builds local models from relevant data, rather than building a single global model that covers all the data. This technique has been extremely effective in forecasting energy use, but does not handle categorical data, and suffers from the usual issue of being unable to model & predict uncommon situations.

It has also been shown that different techniques work on different types of data [5,12]. Each separate strategy may predict different modalities of the time series (e.g. frequency, rhythm), or be more effective at predicting different modes within one time series. Fusion of outputs of different techniques can significantly increase reliability of data analysis and resolve conflicting outputs.

The data that we analyzed included multiple modes, inconsistently sampled data points, time series sequences, and was of moderately high dimension. To summarize, we need a method that does not make assumptions about the underlying data, but still exploits the capabilities of the statistical approaches. The method needs to handle both categorical data and timeseries data, without losing information about the timeseries data. It needs to be able to detect anomalous conditions. Additional attention needs to be focused on finding temporal patterns in the data, using those temporal patterns to create categorical data, and employing time series models to address autocorrelated data.

5 Conclusion

Our experiments with this data showed that existing techniques can help gain insight into patterns and relationships in space applications. While the regression trees were not predictive, they were extremely informative and helped to reduce the parameter space. It is very significant that we were able to discover the controller's basic operating principles from raw data. Even though the regression models were not completely correct, were were able to determine error conditions that hand-built modeling techniques are unable to detect.

In general, we cannot expect that every predictive feature will be available: sensors may be missing, the problem may not be sensable, and the data space is too large. No algorithm will be able to build a completely correct model when relevant parameters are missing, but even given this handicap, our explorations were fruitful. One important caveat for this work is to be cautious about inconsistent sampling in the data. Inconsistent sampling severely limits the development of time series models.

Data mining techniques will become more and more important in space applications as systems become more and more complex. Humans cannot continue to handle the monitoring and diagnosing of problems in these systems, and hand-generated monitoring models will never capture the complex nature of system interactions. Our early work, while promising, shows that more effort needs to focussed in this area.

6 Bibliography

1. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering (ICDE)*, pages 3-14, Taipei, Taiwan, 1995.
2. B. Bakshi and G. Stephanopoulos. Representation of process trends-III multiscale extraction of trends from process data. *Computers and Chemical Engineering*, 18(4):267-302, 1994.
3. G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: Forecasting and control*. Prentice Hall, Englewood Cliffs, NJ, 1994.
4. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmon, CA, 1984.
5. T. Brotherton and R. Mackey. Anomaly detector fusion processing for advanced military aircraft. In *IEEE Aerospace Conference*, March 2001.
6. G. Das, K. I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In *4th Intl Conference on Knowledge Discovery and Data Mining*, pages 16-22, August 1998.
7. W. Foslien, S. A. Harp, K. Lakshminarayan, and D. Mylaraswamy. Content based retrieval of time series data. United States patent pending 09/346,245, filed 7/2/2002.
8. N. A. Gershenfeld and A. S. Weigend. The future of time series: Learning and understanding. In *Time Series Prediction: Forecasting the Future and Understanding the Past*, pages 1-70, 1994.
9. P. Geurts and L. Wehenkel. Early prediction of electric power system blackouts by temporal machine learning. In *AAAI Workshop ``Predicting the Future: AI Approaches to Time-Series Problems''*, pages 21-28, 1998.
10. V. Guralnik and K. Z. Haigh. Learning models of human behaviour with sequential patterns. In *Proceedings of the AAAI-02 workshop ``Automation as Caregiver''*, pages 24-30, 2002.
11. F. Höppner. Learning dependencies in multivariate time series. In *Proceedings of the ECAI'02 Workshop on Knowledge Discovery in (Spatio-) Temporal Data*, pages 25-31, 2002.
12. W. Hsu, S. Ray, and D. Wilkins. A multistrategy approach to classifier learning from time series. *Machine Learning*, 38(1-2):213-236, 2000.
13. J. E. Jackson. *A User's Guide to Principal Components.* John Wiley and Sons, New York, 1991.
14. T. Kourti. and J. MacGregor. Multivariate statistical process control methods for monitoring and diagnosing process and product performance. *Journal of Quality Technology*, 28:409-428, 1996.
15. R. Kulhavy, J. Lu, and T. Samad. Emerging technologies for enterprise optimization in the process industries. In *Preprints of Chemical Process Control 6*, pages 411-422, 2001.
16. M. H. Kutner, C. J. Nachtsheim, and J. Neter. Applied Linear Regression Models. McGraw-Hill/Irwin, 2003. 4th edition.
17. L. Liu, S. Bhattacharyya, S. Sclove, R. Chen, and W. Lattyak. Data mining on time series: An illustration using fast-food restaurant franchise data. *Computational Statistics and Data Analysis*, 37:455-476, 2001.
18. D. Mylaraswamy. Addressing the problem of early abnormal event detection. In *Intl. Symposium on Process Systems Engineering*, pages 240-246, Mumbai, India, 2003.
19. M. J. Zaki, M. J. Lesh, and N. Ogihara. PLANMINE: Sequence mining for plan failures. In *4th International Conference on Knowledge Discovery and Data Mining*, 1998.