Data Mining to Detect Abnormal Behavior in Aerospace Data

José M. Peña
DATSI, Universidad Politécnica de Madrid
Campus de Montegancedo S/N, 28660
Madrid, Spain
jmpena@fi.upm.es

Fazel Famili
Institute for Information Technology, National Research Council
Bldg. M-50, Montreal Rd., K1A 0R6
Ottawa, Ontario, Canada
fazel.famili@iit.nrc.ca

Sylvain Létourneau
Institute for Information Technology, National Research Council
Bldg. M-50, Montreal Rd., K1A 0R6
Ottawa, Ontario, Canada
sylvain.letourneau@iit.nrc.ca

ABSTRACT

The operation and maintenance of today's aircraft is a complex task. It requires use of some state-of-the-art data mining facilites that are not currently available. This paper is about development and use of data mining techniques to detect abnormal situations in aircraft operation. Using historical sensor data, that is normally generated during the operation of aircraft, we induce models to predict abonormal situations in aircraft engines. The method involves creating new features from raw data and identifying trends in particular parameters of interest. We describe how models generated from individual aircraft with abnormal situations can be combined to generate a single model. We evaluate our approach using over 5 years of historical data from the operation of engines of 34 Airbus A-320's.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Managements, Database Applications, Data Mining; J.7 [**Computer Applications**]: Computers in Other Systems, Industrial Control

General Terms

Trend monitoring, Data partitioning, Machine learning

1. INTRODUCTION

Today's commercial aircraft are quite sophisticated and equipped with many sensors and on-board computers. Their operation and maintenance therefore involves dealing with an enormous amount of data that needs to be analyzed by a number of departments throughout the airline company. These departments have to be able to analyze data that is in the form of: (i) text generated by staff (such as observations related to the conditions of the aircraft and related actions), (ii) numerous types of automatically generated messages explaining situations such as deviation of aircraft parameters, and (iii) parametric data acquired through hundreds of sensors and on-board computers (whic hare mostly numeric with some being symbolic).

One of the main issues in the aerospace industry is that aircraft are getting more and more sophisticated as a result of which huge amounts of data are generated without even half of this data being properly analyzed [1, 5]. A large portion of this data is even destroyed after a short time. Although vendors of aerospace equipment provide software to use some of the above mentioned data (especially parametric data), much more needs to be done to intelligently detect abnormal situations in the operation of an aircraft. This is specially true when airline engineers and technical staff have to manage the operation of a large fleet of aircraft (sometimes more than 100).

Like many other industries, in aerospace proper use of parametric data for equipment monitoring and in-depth data analysis is rarely performed. This is due to the following reasons: (i) all the data may not be integrated into one database management system, (ii) engineers and operators do not have sufficient time to analyze huge amounts of data, unless there is an urgent requirement, such as equipment breakdown, (iii) complexity of the data analysis process is in most cases beyond the ordinary tools that they have access to, (iv) there is no well defined automated mechanism to extract, pre-process and analyze the data, and summarize the results so that the engineers and technicians can use it, and (v) even when there is a data analysis tool available, it may be too specific for the operation of certain equipment.

The most important objective in the operation and maintenance of today's aerospace industry is that all systems (e.g. auxiliary power units, main engines) for which data is available, can be continuously monitored. The monitoring is performed so that the specialists are informed of conditions: (i) where there is deviation in the range of any performance or condition parameters comparing to an expected level (e.g. fuel consumption is an example of a performance parameter and engine vibration is an example of a condition parameter), (ii) when there is an abnormal behavior in the

operation of a system (such as an upward or a downward trend in certain performance parameters) that may cause performance deterioration of one or more systems. The engineers and fleet specialists will benefit by paying attention only to the aircraft that are operating abnormally. They can then investigate the alerts, prevent possible damages to any system or component, rectify the problem and if possible, find out the reason for trends that are presented in the form of alerts. In this paper, we introduce an approach for recognizing the aircraft or engines that behave abnormally. Abnormal behavior may sometimes be related to specific aircraft components. We address this problem in a related work in which we explain how machine learning can be used to build models that predict problems with aircraft components before they become non operational [4]. The two approaches provide complementary solutions to support engineers and aircraft fleet specialists in the maintenance of commercial aircraft.

This paper presents the application of a data mining strategy to generate models for abnormal behavior detection in the aerospace domain. This domain, like many other domains, generates very complex data from the perspective of using data mining techniques. This complexity means that instead of directly running a data mining algorithm, the data requires a more elaborated solution. In our problem, there is little chance to apply a single algorithm to the data and obtain appropriate results. In other words, we need to plan a data mining strategy; a sequence of data mining techniques and algorithms, as well as some preprocessing and postprocessing operations to deal with the complexity of this data.

In our research, we encountered the following issues that represent problems we faced. These issues are:

❶ **Data selection**: Retriving the appropriate data from databases. These databases store sensor measurements data acquired from the aircraft operations over several years.

❷ **Labeling method**: Splitting the data instances into positive and negative cases. This classification is necesary in order to apply supervised learning methods.

❸ **Feature generation**: Adding new attributes to the existing features. All these features (original and newly created) are used to generate the models.

❹ **Model fusion**: Combining models from different data. The resulting model will describe all the data used to generate these models.

The structure of the paper is as follows. In Section 2 we briefly explain the related work and discuss other approaches to similar problems. In Section 3 we give an overview of our approach to the problem and in Section 4 we explain the model induction phase. Section 5 consists of model combination phase and in Section 6 we describe our model evaluation process. In Section 7 we show the results of our experimentation using real world data from this domain and Section 8 has conclusion and future work.

2. RELATED WORK

There is only a limited number of research papers that are relevant to this research. Guralnik and Srivstava [2] focus on extracting interesting patterns from time-series data. Their research is about identifying time points at which the behavior of a system changes, which in the statistics literature is called change-point detection. They propose an iterative algorithm that fits a model to a time segment (window) and uses a likelihood criterion to determine if the segment should be partitioned further.

Han, Gong and Yin [4] emphasize on the importance of mining through segment-wise or point-wise periodicity in time-related data. They propose an approach to integrate data cube and a-priori data mining techniques for mining segment-wise periodicity in time-series data. They concluded that data cube was an efficient structure for interactive mining of multiple level periodicity.

Other related works are by Mannila and Toivonen [6], Padmanabham and Tuzhilin [7], and Guralnik, Wijesekera and Srivastava [3] in which they have developed languages for specifying temporal patterns. They also propose algorithms that take advantage of the specified patterns to increase the computational efficiency of the mining process. While the above listed works are examples of the theoretical research efforts, there are also a number of projects that are directly aiming at the application itself [8, 10].

3. PROCESS OVERVIEW

We briefly explained the operation of modern aircraft that are equipped with hundreds of sensors to monitor different operation during flight. These aircraft generate a lot of data that is stored in a database. Attached to this data is additinal information provided by aircraft and ground technicians who describe abnormal situations or problems detected when the plane is in the air and when it is landed at the gate. These reports are usually textual descriptions of the problem, as well as brief information about when and where it happens. There are two kinds of these reports: (i) the aircraft maintenance process records all the components repaired or replaced identifying the exact component or piece removed or repaired because it was not working properly, (ii) reports that describe abnormal situations detected by other technicians (not maintenance engineers) with no specific information about the exact system that is operating abnormaly.

Appropriate models induced to predict the situations described by these two kinds of reports have to be developed in different ways:

❒ Repair reports provide details on how a specific system was repaired or replaced. Therefore, the model generated to predict when a component fails may use all of the maintenance reports. The only exception is for rare cases when the component replacement was not really required, but it was performed anyway.

❒ Abnormal situations describe the problems in terms of warning lights on the panels of the cockpit or other information observed by the crew. These kinds of warnings are sometimes solved by the pilots adjusting the

operation of the aircraft according to the safety procedures recommended. When these reports are analyzed there are extra difficulties because the information provided by them is not as detailed as the information in the other reports. This means that two identical warnings may be the result of two completely different problems.

The procedure presented in this paper divides the problem of generating models for the prediction of these abnormal situations into: (i) model induction and (ii) model fusion phases.

During the model induction phase, the data is selected (issue ❶) and prepared (issues ❷. ❸). A model is then generated for each of the different abnormal behavior reports selected. An iterative procedure is perfomed to find the best model for each report.

The second phase deals with the combination of two or more models into clusters. Each cluster contains models that are related to similar problems. Therefore, all the models can be described by a more general model generated by the fusion of all individual submodels (issue ❹). This phase is also performed iteratively.

4. MODEL INDUCTION PHASE

The goal of this phase is to extract prediction models for each of the reports selected. Each of these reports defines a dataset of records from the last operation of the aircraft before the problem date. Each of the records is called an *instance*. The instances are defined by a set of parameters that are measured by different aircraft sensors. The models are induced from the datasets that consist of values for these parameters for a number of flights before the problem was detected. To achieve this goal three main tasks are perfomed: (i) Data selection, (ii) Data preparation and (iii) Rule induction and validation.

4.1 Data Selection

Abnormal behavior reports contain identification attributes (e.g. aircraft number and date of report, etc.), as well as a textual description of the problem observed by the crew. The first task is selection of the reports containing a similar problem description. The selection process uses identifying keywords and phrases to be matched by the preselected reports. This process provides a reduced set of records compared with the total number of records stored in the database (five years of aircraft operation, in our case), but it also contains reports for different problems. All irrelevant reports are filtered out by a final review of output generated by the keyword-matching procedure.

Once the reports are selected for a particular problem, all the available sensor data is then retrieved from the database. This sensor information belongs to a time window of few months before the report date, from the same aircraft and component. Each of these datasets is called a *case*.

4.2 Data Preparation

All supervised induction algorithms require data to be classified as positive or negative instances. This involves partitioning the dataset into 2 equivalence classes. The description of these equivalence classes using the rest of the parameters is the target of the rule induction procedure. Since we want to get prediction rules, our positive instances represent the situations in which we want to warn the operator about a possible problem in the near future. The attribute created to classify the instances is called *label* and the process itself *labeling*.

In our experiments two different labeling methods are proposed: (i) *time labeling*, which labels the operations perfomed in the last few days before the problem happened as positive instances, and the other operation as negative instances, and (ii) *trend labeling* which takes into account information about trends of a specific parameter such as when an upword or downward trend is present just before the problem date. Both methods are shown in figure 1.b.

When trend labeling is applied, a statistical function is defined to detect when the parameter increases or decreases. This function gets a sequence of consecutive instances called a *window* and performs the following calculation:

1. The window is divided into a small number of partitions (N = 6-12). Each partition has the same number of consecutive instances.
2. For each of the partitions ($P_i, 0 < i < N$), the number of instances with parameter value greater than a specific threshold (NR_i) is calculated:

$$NR_i = |X| / X = \{x \in P_i, : PARAM(x) > \vartheta\} \quad (1)$$

   Let P_i be the i-th partition, x an instance from the partition and $PARAM(x)$ the value of the parameter for instance x.
3. Each partition is compared with the following partitions in order to check if the number of instances over ϑ increase for each specific pair of partitions. $\forall i, j / i < j$:

$$Inc_{ij} = \begin{cases} 0 & NR_j = 0 \\ 1 & NR_i \leq NR_j \\ 0 & NR_i > NR_j \end{cases} \quad (2)$$

4. Finally, this window is labeled as positive if:

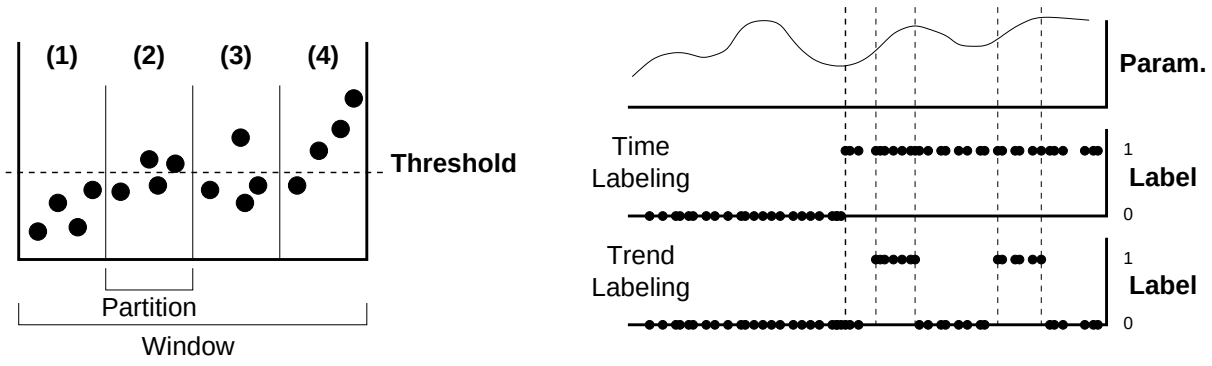
$$\frac{1}{NP} \sum_{i<j}^{N} Inc_{ij} \geq \gamma_{th} \quad (3)$$

$$\text{Let } NP = \sum_{i<j}^{N} 1 = \sum_{i}^{N-1} i \quad (4)$$

For example, Figure 1.a presents the division of the sequence into 4 partitions with 4 instances per partition.

This example is labeled (for $\gamma = 50\%$) as positive because:

$$\begin{aligned} Inc_{12} &= \ 1 \quad NR_1 = 0 \leq NR_2 = 2 \\ Inc_{13} &= \ 1 \quad NR_1 = 0 \leq NR_3 = 1 \\ Inc_{14} &= \ 1 \quad NR_1 = 0 \leq NR_4 = 3 \\ Inc_{23} &= -1 \quad NR_2 = 2 > NR_3 = 1 \\ Inc_{24} &= \ 1 \quad NR_2 = 2 \leq NR_4 = 3 \\ Inc_{34} &= \ 1 \quad NR_3 = 1 \leq NR_4 = 3 \end{aligned}$$



a) Trend Detection *b) Labeling Methods*

Figure 1: Data labeling

$$\frac{1}{6}\sum_{i<j} Inc_{ij} = \frac{5}{6} \geq \gamma = 0.50$$

The use of either of the labeling methods depends on the problem we are analyzing. For example, if the problem is a red light alert due to a high value of an aircraft performance or condition parameter (e.g. high exhaust temperature), the time labeling method would provide a description of the situation previous to this alert. However, the trend labeling would warn about the same situation only during the period that the parameter is continuously increasing.

4.2.1 Additional Features

An additional purpose of the data preparation task is the generation of new features for each of the instances of the dataset. These new features help the induction algorithm to achieve a better model. These features are numerical or statistical functions defined using the original parameter values from the instances of a window. Example of additional features are: statistical mean, standard deviation and any trend definition or variation detection functions.

During our experiments a large number of additional features were proposed. However, only those with semantical interpretation that were appropriate for human understanding were used. The reason being that models and rules containing these features are provided to the operator when the actual fleet monitoring process matches a warning condition defined by the rules. Even if the monitoring system handles complex features as conditions of the rules, technicians can hardly understand the meaning of numerical functions if they do not represent initial attributes or their derivatives, such as increments, decrements or variations.

4.3 Model Induction and Validation

The last step of the model induction phase is generation of the rules belonging to the model and their evaluation in terms of accurancy and error rate. The C4.5 algorithm [9] was applied to generate the rules, using labels generated by either of the labeling methods and taking original and additional parameters as possible rule descriptors.

The inducer is trained using 75% of the instances, saving the other 25% for testing. The division between training and testing instances was chosen randomly. A confusion matrix was created in the evaluation process. The contents of the evaluation matrix provided the number of instances classified successfully as well as the positive instances classified as negative (missing instances) and the negative ones classified as positive (false alerts). The percentage of missing instances plus the percentage of false alerts is called *error rate*.

The evaluation method is not as elaborate as cross-validation of similar strategies, because the training and testing data are randomly created from the same group of instances. In addition, there may be some dependency between the instances used for creating the model and the ones used for validating it. The importance of this drawback is therefore reduced because this validation method is only used for the individual models and not for models generated by the combination of different cases. The other validation method is presented in the following sections.

5. MODEL COMBINATION PHASE

At this point of experimentation, a model has been generated for each of the cases. The model usually has high accuracy when describing the case. However, it might not be appropriate for a different case, even though the cases share a similar textual description of the report. There is a balance between the use of different models for every single case and the induction of a global model for all the cases. The first option extracts very specific rules and some times only valid for the case they were generated from. The second one, as we have said, might induce very weak models with a high error rate from complex real-world data. In order to avoid this problem, once the models have been generated they are grouped in clusters of similar cases. The criteria to group two cases are based on the similarity between them. The measure of similarity is made by calculating the error rate while applying the models extracted from one case to another.

$$A \sim_\alpha B \Longleftrightarrow \alpha = \frac{\varepsilon_A(B) + \varepsilon_B(A)}{2} \qquad (5)$$

Where $\varepsilon_A(B)$ is the error rate when the model induced from case A is applied to the data in case B.

Using this metric, the clustering algorithm has the following steps:

① Compute the distance between every pair of cases, and build a triangular matrix with the results.

② Select the minimum value of the matrix (α_{min}) that represents the two cases with highest similarity.

③ If $\alpha_{min} > MAX\_ERROR$ then finish the algorithm.

④ Else combine the two cases to create a new case and remove from the original set both cases and insert the new one.

⑤ Go back to ①.

This algorithm has two variations that depend on how the combination of the cases is achieved:

- **Combination by Learning**: This option groups the data from both cases and runs the induction algorithm again. Then a new model is learned.
- **Combination by Fusion**: This alternative method does not require to run the inducer again, instead it uses some heuristics to combine the rules from each of the models and creates a new set of rules. The heuristics available for this option may be:
  - *Best Support*: Selecting the top rules with best support.
  - *Predicate Distance*: Using a special metric function to measure the distance between rules and then select the rules with highest support adding the support of all the rules within a specific distance (refer to 5.1).

5.1 Predicate Distance Fusion

The predicate distance method used for rule fusion defines a metric between rules using predicate information. It is then applied to a group of similar rules.

The rule metric is defined as:
Let r_i be a rule, defined as the triplet $\langle P_i, V_i, f_i \rangle$:

$$\text{Let } P_i \subseteq (A \times R) \tag{6}$$
$$\text{Let } A \text{ set of all attributes} \tag{7}$$
$$\text{Let } R \text{ set of all relational operations} \tag{8}$$
$$\text{Let } V_i \subseteq \mathbb{R} \tag{9}$$
$$\text{Let } f_i : P_i \longrightarrow V_i \tag{10}$$

The set of all the rules $R = \{r_1, \ldots r_n\}$ is the defined as a Metric Space with the Distance function $d : R \times R \longrightarrow \mathbb{R}^+ \cup \infty$ defined as:

$$d(r_i, r_j) = \begin{cases} \infty & \text{if } P_i \cap P_j = \emptyset \\ d_{aux}(r_i, r_j) & \text{if } P_i \cap P_j \neq \emptyset \end{cases} \tag{11}$$

$$d_{aux}(r_i, r_j) = \frac{|P_i| - |P_j|}{2} - |P_i \cap P_j| + \frac{|P_i \setminus P_j|}{|P_i| - |P_i \setminus P_j|} + \frac{|P_j \setminus P_i|}{|P_j| - |P_j \setminus P_i|} \tag{12}$$

This function also satisfies:

$$d(r_i, r_j) = 0 \Leftrightarrow r_i = r_j \tag{13}$$
$$d(r_i, r_j) = d(r_j, r_i) \tag{14}$$
$$d(r_i, r_j) + d(r_j, r_k) \geq d(r_i, r_k) \tag{15}$$

To match condition 13 (left to right implication), it is necesary to adjust the second case of the formula to deal with the predicate values V_i. The formula should be:

$$\frac{|P_i| - |P_j|}{2} - h(r_i, r_j) + \frac{|P_i \setminus P_j|}{|P_i| - |P_i \setminus P_j|} + \frac{|P_j \setminus P_i|}{|P_j| - |P_j \setminus P_i|} \tag{16}$$

Defining $h : R \times R \longrightarrow \mathbb{R}$ as

$$\sum_{(attr,rel) \in P_i \cap P_j} compare_{attr}(f_i(attr, rel), f_j(attr, rel)) \tag{17}$$

With $compare_{attr} : \mathbb{R} \times \mathbb{R} \longrightarrow (0, 1]$ we compare two values of a specific attribute that must satisfy:

$$compare_{attr}(v_i, v_j) = 1 \Leftrightarrow v_i = v_j \tag{18}$$

6. EVALUATION OF THE MODELS

When cases are grouped in clusters, the final accuracy of the model from each of the clusters is evaluated. This value is calculated taking all the cases from the cluster but one. These cases are then used to build the model (by learning or fusion) and then use the other one for validation. The process is then repeated by leaving aside a different one. The error rate of the cluster is then calculated as the mean of the error rate of each of the evaluations.

In order to contrast the model a different group of cases is used to evaluate the model. These cases belong to *healty* components with no abnormal behavior report for a long period of time. All instances of this dataset are considered negative and the model extracted from the cluster should not generate false alerts when it is tested on this dataset.

7. EXPERIMENTAL RESULTS

Our approach has been tested and evaluated for the prediction of abnormal behavior conditions in the operation of aircraft engines of Airbus A-320. Selected cases were those related to high EGT (exhaust gas temperature) problems detected by the technicians.

The keyword-driven search and the manual review of the reports provided 13 cases of this problem. In addition to these cases, a group of 12 cases were selected to represent healty engines without any report entry for several months of operation. For both case groups sensor information from the last three months before the report was recovered from the database.

7.1 Naive Approach

The simplest approach towards construction of a model for this problem is: (a) to label all the cases (either with time labeling or trend labeling methods) and (b) to group all the cases together and calculate the model with them. This approach provides the results shown on tables 2 [A] and 3.

The i-th column represents the validation of the model generated using all the cases with the data from case number i. These results show how inappropriate this approach is for this problem. In most of the problem cases the error rate is around 50% which means that classification is almost random. For the healthy engines the error rates in some cases are high.

7.2 Model Building

As an alternative to the previous approach, our strategy is to build a model for each of the cases and then combine them in a second phase, creating a cluster of similar cases. The induction of the individual models may be done using only the original features provided by the on-board sensors and computers. Table 2 [B] and [C] show the results when only the original parameters are used.

Development of these models using new features like (mean, standard deviation and other trend functions) provides a better approach, but their calculation requires tunning different parameters, like the number of instances in the window (window size), the overlap factor that represents how many instances share two consecutive windows, etc. For the calculation of the best values for these factors a different range of values is scanned for each of the factors and then the combination with minimum error rate is selected. Some of these factors are compared on the 2 plots of figure 2 and the error rates for the best combination are reported on the table 2 [D] and [E]. The accuracy of these models is by far better than the previous ones, this is also due to the stronger dependency present between two consecutive instances from the data set because of the sharing of common instances when new features are calculated (the number of shared instances depends on the overlaping factor).

7.3 Model Fusion

When individual models are generated, the last step is to combine the cases that are similar into clusters. In our experiments we have compared the results when using both labeling methods and using either combination by learning (generating new models when two cases are combined) or combination by fusion (using heuristics like predicate distance for the fusion). The results are given in table 1.

8. CONCLUSION AND FUTURE WORK

Two main issues are important in our work. First, the inclusion of new features drastically increases the quality of the results obtained by traditional algorithms for these kinds of problems (on Table 2, compare rows [B] and [C] with rows [D] and [E]). Labeling strategy is also important but it directly depends on the semantic meaning of the extracted model and its implications. Proper labeling would allow us to predict problems (some days before they happen) or it may help us to generate alerts for dangerous situations when one of the important parameters contains an important trend.

Second, when models are combined, fusion and learning techniques achieve results with different characteristics. These characteristics are not influenced by the mechanisms used to induce individual models. Following are examples of how these characteristics are related to the strategies that we used to build a new model:

- The combination of results by fusion reduces the error rate for healthy engines where all the cases are negative but increases the number of missing cases. The models generated by fusion are very conservative, describing stronger rules (with high support and accurancy) but they do not cover all the positive cases.

- When combining by learning, the models cover more positive cases but there is also a number of negative cases classified as possitive.

Our approach shows very interesting contributions related to the applications of data mining techniques in aerospace that may be applicable to other domains as well. There are, however, some issues that require further study and research. On one hand, the combination process was only able to group half of the cases (in the best experiment), keeping the rest of the cases apart. Defining larger clusters and getting also a good accuracy factor may be achieved either by other combination strategies or induction of different individual models. Both of these issues require further investigation.

On the other hand, labeling methods are an important element of model induction and working with different methods would provide different results. The new methods should be both useful (in terms of accuracy and support) for the induction algorithm and comprehensive for the human user that handles the rules extracted by the process.

9. ACKNOWLEDGMENTS

J.M. Peña thanks IIT group at NRC for support and help during my visit, as well as Dr. C. Fernández and Dr. E. Menasalvas for additional comments about experimentation and many paper revisions. The authors would also like to thank Alan Barton for reviewing an earlier version of this paper. We also thank Air Canada for providing the data and useful feedback.

10. REFERENCES

[1] F. Famili and S. Letourneau. Monitoing of aircraft operation uning statistics and machine learning. In *IEEE-Tools with AI-99*, pages 278–286, 1999.

[2] V. Guralnik and J. Srivastava. Event detection from time series data. In *ACM KDD-99*, pages 33–42, 1999.

[3] V. Guralnik, D. Wijesekera, and J. Srivastava. Pattern directed mining of sequence data. In *ACM KDD-98*, pages 51–57, 1998.

[4] J. Han, W. Gong, and Y. Yin. Mining segment-wise periodic patterns in time related data-bases. In *ACM KDD-98*, pages 214–218, 1998.

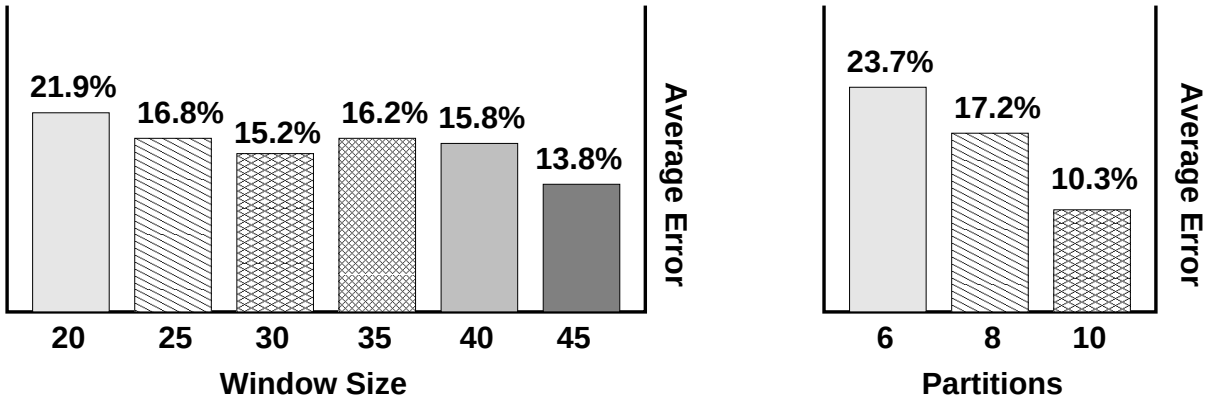


Figure 2: Average error for different experiment factor values

	Time Labeling		Trend Labeling	
	Learning	Fusion	Learning	Fusion
Number of clusters	2	3	1	1
Max Cases in a cluster	4	2	2	2
Min Cases in a cluster	3	2	2	2
Cases un-grouped	6	7	11	11
Error	12.3 %	23.7 %	31.2 %	31.2 %
Error (Health engines)	14.5 %	3.1 %	4.6 %	4.6 %

Table 1: Case combination results

Case #	1	2	3	4	5	6	7	8	9	10	11	12	13
[A] Naive Approach													
False alerts(%):	35.7	46.5	8.3	14.0	33.0	36.9	26.0	21.7	23.2	32.3	16.7	29.3	15.2
Missing(%):	7.1	3.6	43.8	41.2	18.7	14.7	23.9	25.4	38.6	13.7	47.6	21.6	40.7
Error Rate(%):	42.8	50.2	52.1	55.2	51.7	51.7	50.0	47.2	61.8	46.0	64.4	51.0	55.9
[B] Individual models (Time labeling, original features)													
Error Rate(%):	13.2	8.0	38.3	20.0	19.3	40.2	28.1	23.0	17.6	42.5	19.6	25.8	25.4
[C] Individual models (Trend labeling, original features)													
Error Rate(%):	11.9	15.9	39.2	19.2	19.3	33.3	28.1	36.6	18.0	25.0	21.5	23.3	24.1
[D] Individual models (Time labeling, new features)													
Error Rate(%):	0.0	5.6	10.1	1.8	2.8	0.0	1.5	2.5	1.8	2.1	8.1	1.5	5.1
[E] Individual models (Trend labeling, new features)													
Error Rate(%):	2.0	1.9	6.1	5.5	0.0	5.7	1.5	3.5	21.4	6.2	5.5	0.0	8.6

Table 2: Summary of results

Case #	1	2	3	4	5	6	7	8	9	10	11	12
[A] Naive Approach												
False alerts(%):	83.9	54.6	7.0	2.1	34.3	73.0	56.1	12.9	3.1	70.1	2.3	0.1

Table 3: Sumary of results for negative cases

[5] S. Letourneau, F. Famili, and S. Matwin. Data mining for prediction of aircraft component failure. *IEEE Intelligent Systems: Special Issue on Data Mining*, 14(6):59–66, 1999.

[6] H. Mannila and H. Toivonen. Discovering generalized episodes using minimal occurrences. In *ACM KDD-96*, pages 146–151, 1996.

[7] P. Padmanabham and A. Tizhilin. Pattern discovery in temporal databases: A temporal logic approach. In *ACM KDD-96*, pages 351–354, 1996.

[8] V. C. Patel et al. Gas turbine engine condition monitoring using statistical and neural network methods. In *IEE-Colloquium-(Digest). n 260, IEE*, pages 1/1–1/6, 1997.

[9] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California, 1993.

[10] G. Torella and G. Lombardo. Neural networks for the diagnostics of gas turbine engines. In *ASME Turbo Asia Conference, American-Society-of-Mechanical-Engineers*, pages 1–11, 1996.