Using Tree Augmented Naïve Bayes Classifiers to Improve Engine Fault Models

Daniel L.C. Mack
EECS Dept.
Vanderbilt University
Nashville, TN 37212

Gautam Biswas
EECS Dept.
Vanderbilt University
Nashville, TN 37212

Xenofon D. Koutsoukos
EECS Dept.
Vanderbilt University
Nashville, TN 37212

Dinkar Mylaraswamy
Honeywell Aerospace
1985 Douglas Drive N
Golden Valley, MN 55422

Abstract

Online fault diagnosis is critical for detecting and mitigating adverse events that arise in complex systems such as aircraft, automobiles, and industrial processes. A typical fault diagnosis system consists of a reference model that mathematically links diagnostic monitors providing partial evidence to potential fault hypotheses. A reasoning algorithm operated on this model uses a set-covering scheme to establish likely fault candidates and their rankings. However, incompleteness in the reference model and simplifying assumptions affect the accuracy of the reasoning algorithms. In this paper, we describe a Tree Augmented Naïve Bayes Classifier (TAN) approach to systematically extend a reference model structure using data from system operations. We compare the performance of the TAN models against a typical reference model, and demonstrate that the TAN improves classification accuracy by finding new causal links among the system monitors.

1 Introduction

Aircraft are complex systems containing several interacting components and subsystems such as propulsion, electrical, flight management, avionics, and bleed subsystems. Smooth and integrated operation of these subsystems is essential to keep the aircraft operating safely. However, any operating system degrades over time and monitoring the system online for detecting the onset of unfavorable conditions and intrinsic faults is essential for increasing aviation safety.

The current state of online fault diagnosis is focused on installing a variety of sensors onboard an aircraft along with reasoning software to automatically interpret the evidence generated to explore the presence of faults. One such state-of-the-art system is the Aircraft Diagnostic and Maintenance System (ADMS) (Spitzer, 2007) that is used on the Boeing B777. The ADMS uses an expert-derived fault propagation model, called the *system reference model* that captures the interactions between aircraft components under various operating modes. Generation of this reference model is a manual process and often the step results in incompleteness and inaccuracies in the development and deployment of an ADMS.

Some of the incompleteness and inaccuracies can be overcome as the engineering teams acquire additional knowledge from an operating fleet, and generate heuristics rather than a systematic upgrade to the original reference model. In other words, a gap exists for systematic upgrades and increments to the reference model even though vast amount of operational data is collected by operating airlines. Closing this gap using advances in data mining methods is the focus of this paper. We describe a specific data mining approach for augmenting an existing aircraft engine reference model as an alternative to ad hoc approaches. We demonstrate the effectiveness of our work on data generated from a realistic aircraft engine simulator.

Statistical analysis and designing classifiers for discovering knowledge from real-world data has been studied extensively. For example, Witten (Witten & Frank, 1999) describes several data mining approaches for producing black box models. Unfortunately, such models are very difficult to verify, making them almost impossible to certify for airworthiness. Further, the lack of transparency in these models makes it difficult to *append* this new knowledge to existing ADMS reference models. For practical purposes, data mining approaches for aircraft reference models have to "build upon" existing model structures rather than create something new, which will incur considerable engineering overhead cost.

The proposed approach to combing data mining with

fault models is somewhat unique. The data mining does not start from a clean slate, but builds up from an existing ADMS reference model structure. In section 2, we describe a typical reference model structure along with the reasoning algorithm (called the W-algorithm). Next, we systematically enumerate the missing or partially correct information in this state-of-the-art reference model. These gaps formalize the data mining problem described in Section 3. We discuss the use of Tree-Augmented Naïve Bayes Networks (TANs) as a data driven modeling structure for diagnosis with causal probabilistic models in section 4. The data mining approach is illustrated using data from a high fidelity simulator. Section 5 discusses the CMAPS-S simulator and the data selection task for our experiments. Section 6 describes the experimental results using the CMAPS-S data set, with a comparison of a Naïve Bayes classifier that replicates a system reference model against a TAN classifier model derived from a learning algorithm. Metrics are defined for evaluating classifier performance, and a number of different experiments are run to examine the addition of evidence to these models. Section 7 presents a summary of our approach, and outlines our directions for future work for diagnostic and prognostic reasoning using the data mining algorithms.

2 Reference Models and Reasoning

Model-based strategies for diagnosing large, complex, real-world systems rely on domain experts to craft the *reference models* used for monitoring and isolating faults. The complexity of the system makes it almost impossible to create complete physics-based models with reasonable resources. A more pragmatic solution is to rely on expert-generated cause-effect models. In simple terms, the reference model of the system being monitored can be represented as a bipartite graph consisting of two types of nodes: failure modes and evidence. The set F defines all *distinct* failure modes defined for the system under consideration. A failure mode $fm_i \in F$ may be present or absent in the system. This is defined as the state of the failure mode. In the primary model, we allow only binary (occurring or not-occurring) states for the failure mode. We use the following shorthand notations regarding these assertions.

$$\begin{aligned} fm_i = 0 &\Leftrightarrow \text{The failure mode is not present} \\ fm_i = 1 &\Leftrightarrow \text{The failure mode is present} \end{aligned} \tag{1}$$

Every failure mode has an a priori probability of occurring in the system. This probability is denoted by $P(fm_i = 1)$. A failure mode fm_k can occur (or not occur) independently of another failure mode fm_j occurring, that is, $P(fm_k = 1|fm_j = 1) = P(fm_k = 1)$.

To isolate and disambiguate failure modes, the model also defines an entity called "evidence". The jth evidence is denoted by e_j and the set E denotes all distinct monitors defined for the system under consideration. The diagnostic monitor associated with the ith evidence can either *indict* or *exonerate* a subset of failure modes called its *ambiguity group*. The monitor m_i can take three mutually exclusive values allowing a monitor to express indicting or exonerating or unknown support for the failure modes in its ambiguity group. The notations are described in equation (2).

$$\begin{aligned} m_i = 0 &\Leftrightarrow \text{Exonerating evidence} \\ m_i = 1 &\Leftrightarrow \text{Indicting evidence} \\ m_i = -1 &\Leftrightarrow \text{Unknown evidence} \end{aligned} \tag{2}$$

Ideally, we want a monitor associated with evidence e_i to fire only when the failure modes in its ambiguity group are occurring. Given the fact that the ith failure mode is occurring in the system, d_{ji} denotes the probability that there will be a monitor providing an indicting evidence under this condition.

$$d_{ji} = P(m_j = 1|fm_i = 1), \tag{3}$$

d_{ji} is called the detection probability of failure mode monitor fm_j with respect to the ith evidence. A monitor may fire when there is no failure mode present in the system. False alarm probability is the probability that an indicting monitor is present when there are no failure modes occurring in the system. That is,

$$\epsilon_i = P(m_i = 1|fm_j = 0, \forall fm_j \in F) \tag{4}$$

A reference model describes the relation between failure modes and monitors. The reference model is a 6-tuple defined as: $[\, E, F, D, Pr, \epsilon \,]$,where: E is evidence set, F is failure mode set, D is detection probabilities, Pr is a priori probability of failure modes, ϵ is false alarm rate for monitors.

Figure 1 illustrates an example reference model graphically, with fault modes (hypotheses) as nodes on the left, and diagnostic monitors (DM) on the right. Each link has an associated detection probability, i.e., conditional probability $P(m_j = 1|fm_i = 1)$. In addition, fault nodes on the right contain the a priori probability of fault occurrence, i.e., $P(fm_i)$. Probabilities on the DM nodes indicate the likelihood that a particular monitor would indicate a fault in a nominal system, which as defined above is ϵ_i. Bayesian methods are employed to combine the evidence provided by multiple monitors to estimate the most likely fault candidates.

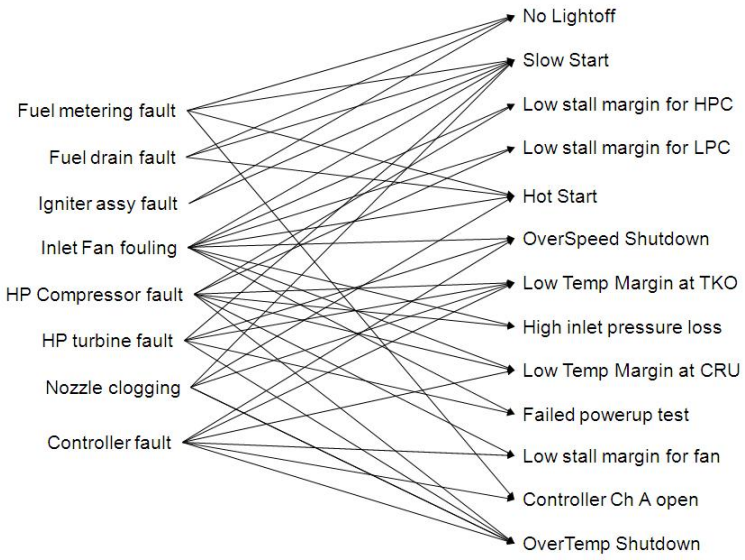

Figure 1: Example Reference Model

The reasoner algorithm (called the W-algorithm) combines an abductive reasoning algorithm with a forward propagation algorithm to generate and rank possible failure modes. This algorithm operates in two steps: (1) *Abductive reasoning step*: Associated with each DM is an *ambiguity set*, $AG = \{fm_1, fm_2, \cdots fm_k\}$. This step assumes that the firing of the DM implies at least one of the faults in the ambiguity set has occurred; and (2) *Forward reasoning step*: For each fm_i belonging to the AG, we extract other DMs that support fm_i. We call this set the supporting DMs, or the monitors of interest, i.e., $S - DM_i$ for fm_i. As these additional monitors fire, fm_i without that monitor in $S - DM_i$ are removed from the AG. Over time as the monitors fire, AG reduces in size, and ideally, to a single fm_i. Additional details about the reasoning algorithm is described in (Honeywell, 2010).

The reasoning algorithm generates multiple single fault hypotheses, each hypothesis asserting the occurrence of exactly one failure mode in the system. The basic probability update rules assume independence of monitor firing events. In other words, $P(m_j, m_k|fm_i) = P(m_j|fm_i)\ P(m_k|fm_i)$ for all monitors m_j and m_k. The two independence assumptions on: (1) Fault modes, and (2) monitors implies that the reasoning algorithm treats the reference model as a set of Naïve Bayes classifiers. The direct correspondence between the reference model for diagnosis and the simple Bayesian structure provides opportunities to use a class of generative Bayesian model algorithms to build these model structures from data and enhance the existing structures produced by a domain expert.

This reasoning algorithm assumes the DMs used in the reference model are strictly binary. The DMs are often derived by applying a threshold to other real valued features known as condition indicators(CIs). These CIs are built as functions of sensors to provide more information about the health of the system. The thresholds applied to create DMs are selected by a domain expert. Data collected from these systems more often contain raw sensors and the CIs rather than the DMs. This creates an issue when trying to examine structures built with data and comparing them to the expert crafted models. Our approach utilizes the idea of the abstracted CIs when constructing models from data. Models built with data and containing CIs or other select sensors are only missing the thresholding, and as such, when the the probabilities are calculated, a Naïve Bayesian model is in essence approximating the reasoning algorithm above. No fault modes are removed from consideration, but the probabilistic ranking of all failure modes will render many with a probability at or near 0. The inference used in Bayesian networks is calculated in the context of discretized values (Conditional Probability Tables). Any necessary discretization of these values is providing a thresholding that acts similar to the reasoning algorithm on an expert model. We believe these similarities are enough to warrant comparisons in the analysis of our results. We utilize this similarity in computation of learned models and their metrics for evaluation.

3 The Data Mining Problem

The reasoning algorithm may not reduce the ambiguity group to a single fault element. For example, all of the evidence (i.e., DMs) required to isolate the single

fault may not fire, leaving the size of the ambiguity set to be greater than 1. In this case, the reference model is incomplete. This gap can be addressed by employing heuristic rules or systematically discovering new diagnostic monitors from vast amount of historical data.

A second source of error arises from the "independence assumptions". The assumption of independence between (1) different pieces of evidence and (2) different fault modes may lead to certain hypotheses being assigned higher likelihood than the evidence truly implies. This assumption is made primarily because, causality (or correlation) between evidence in the system is not easily discernible while the system is being designed and assembled. Furthermore, deriving conditional probability tables with joint probabilities such as when nodes have multiple parents is a difficult task for human experts, and can be derived from data. Therefore, the knowledge required to overcome the simplifying (but erroneous) assumptions of independence are best derived by analyzing data from an operating fleet.

As implied above, the reference model that does not make the simplifying independence assumptions can be interpreted as a Noisy-OR classifier, which is a simplified form of a standard Bayes Network. A number of Machine Learning techniques for building Bayes networks from data have been reported in the literature (Friedman, Geiger, & Goldszmidt, 1997) We have studied a number of these approaches in the framework of diagnostic and prognostic reasoning. Among the important considerations have been the notion of independence among the monitors that support the diagnostic reasoning. Our choice for a Bayesian model and for the data mining algorithms that build these models has been guided by:

1. The data mining algorithms should be designed to provide information that supplements existing expert-generated reference models. It is very important that the experts be able to interpret the results of the data mining algorithms, and characterize them as:
   (a) new relations between monitors and fault hypotheses that will improve the reference model;
   (b) additional monitors (both simple and advanced) that help differentiate and provide support for specific diagnostic hypotheses;
   (c) refinements to the conditional probability values between hypotheses and monitors.
2. The computational complexity of the data mining algorithms should be manageable, so that they can be used as exploratory analysis tools by the domain experts. We envision a successive refinement process, where the expert requests a sequence of experimental runs, each built from their observations and interpretations from previous results generated by the algorithms. They can interpret the causal relations between faults and monitors, and discover the dependence among the monitors for different fault situations. The expert may also consider different analysis scenarios to estimate methods for increasing the accuracy (while reducing false positives) in the diagnostic reasoner.

After considering these factors and staying within the Bayes net paradigm, we selected Tree Augmented Naïve Bayes(TAN), a model that could address the factors in a reasonable fashion, as well as challenge the independence assumption in limited ways.

4 Data Mining with Tree Augmented Naïve Bayes Networks

The choice of the data driven techniques to apply to a particular class of problems is very much a function of the nature of the data and the problem(s) to be solved using the data. For example, using data we can systematically test and relax the independence assumptions employed in the reference model, especially if it is useful for diagnosis. There are several interesting alternatives, but one that fits well with our reference model structure is the Tree Augmented Naïve Bayes (TAN) Method (Friedman et al., 1997). The TAN structure is a simple extension of the Naïve Bayes network. Like Naïve Bayes, the root node is the class node, corresponding to one or more fault modes, is causally connected to every evidence (monitor) node. In addition, the TAN structure relaxes the assumption of independence between the evidence nodes, and allows most evidence nodes to have a second parent, which can be a related evidence node. This maintains the directed acyclic graph requirements and produces a tree that captures relationships among the monitors. Generation of this structure is not as computationally expensive as a general Bayesian network.

An example TAN structure is illustrated in Figure 2. The class node is the fault hypothesis under consideration. The other nodes represent supporting evidence for the particular fault hypotheses. In this structure, the only node connected to the class node, is the root observational node. Dependencies among the monitors are captured as additional causal links in the TAN structure.

The TAN Structure can be generated in several dif-

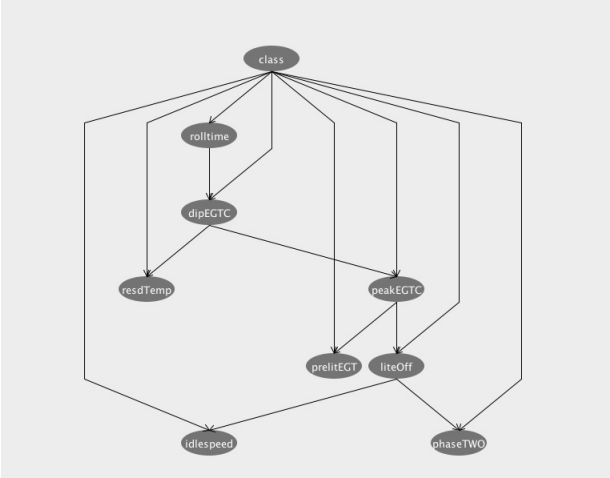

Figure 2: Example TAN Structure

ferent ways that includes (1) a *greedy search* with the constraint that *illegal edges* (i.e., a node having more than one parent from the evidence nodes) are disallowed (Cohen, Goldszmidt, Kelly, Symons, & Chase, 2004); and (2) a *Minimum Weighted Spanning Tree* (MWST) approach that builds a minimum spanning tree to capture the dependencies among monitors, and then connects the class (fault mode) to all of the monitor nodes (Friedman et al., 1997). In either case, a decision has to be made about the monitor node to use as the observational root node in the derived tree structure. The derived TAN structure is static, i.e., it does not include explicit temporal information through causality.

A standard algorithm (e.g., Kruskal's algorithm (Kruskal, 1956)) is applied to generate the MWST. The edge weights of the MWST structure are a log likelihood function, e.g., Bayesian value (Chickering, Heckerman, & Meek, 1997) or the Bayesian Information Criterion (BIC) (Schwarz, 1978). The Bayesian likelihood metric is preferred for discrete data, wheras the BIC measure works better for continuous distributions. The algorithm we use calculates the BIC value for every pair of evidence nodes (note that directionality matters, therefore, for nodes A and B, two BIC values are computed from A to B and B to A). The values are stored in a matrix, which facilitates the application of Kruskal's algorithm to generate the MWST.

The MWST version of this algorithm is implemented in the data mining toolkit called Weka (Hall, Eibe, Holmes, Reutemann, & Witten, 2009) It does not handle continuous features, and instead uses a discretization algorithm to bin each of the features into sets that best discriminate among classes. This produces better classifiers, but it may create very fine splits for features that result in excessive binning(thus building very large conditional probability tables).

5 The CMPAS-S Data

The CMAPS-S data set is generated from a simulator developed at NASA's Glenn Space Center (Frederick, DeCastro, & Litt, 2007). The engine simulator takes into account the wear and tear on a turbine engine over multiple flights, and it can produce data for a number of sensors for climb, cruise, and descent modes of operation. The simulator parameters can be set to run in nominal and faulty modes of operation.

As a first step, we select appropriate sensor measurements as features and transform them into a sequence of values for the data mining task. Since the reference model structure and the reasoner do not directly include temporal information, the data is separated into the different modes of operation. For this study, all of the data for fault analysis was extracted from the cruise mode of operation. In this mode, most sensor values remain steady, except for measurement noise. Therefore, for this study each flight was represented as a datapoint consisting of a vector of sensor values, and the entire dataset was made up of n data points corresponding to n flights.

Table 1 shows the different features in the CMAPS-S data set. Some features are marked as a "condition indicator"(CI), which is a term for complex features that can be used to indicate when an engine is experiencing abnormal behavior. A threshold on these values would produce the *health indicator* (also called a diagnostic monitor, DM) that a reference model would relate to a fault mode.

The reference model as defined above is in terms of DMs which in this data would be HIs. Since the data contains only the CIs for the engine and an expert crafted reference model was unavailable, we used a Naïve Bayes structure based on CIs as the "base" reference model. This represents an approximation, but the approximation is a good one. As mentioned, experts avoid complex relationships in these models (such as between monitors and faults) they often implicitly assume independence. We find a close approximation of this as a Naive Bayes classifier.

The rest of the features extracted from the data represent the sensors, and thus, features that would most likely be available in data from other complex systems of this nature. These features are selectively added to determine if the reasoner can generate more accurate results with the added information and the refined structures that the learning algorithm generates.

The CMAPS-S data was generated in a way that the

Sensor	Notes
Altitude	$\mathbb{R}$, unit is feet
Mach Number	$\mathbb{R}$, the unit is Mach
Throttle Angle	$\mathbb{R}$, measured in degrees
Fuel Flow	$\mathbb{R}$, measure in percent
Stall Margin of HPC	CI
Stall Margin of LPC	CI
Stall Margin of Fan	CI
Temp. of High Pressure Turbine	$\mathbb{R}$, measured in Centigrade
Temp. of the Fan Inlet	$\mathbb{R}$, measured in centigrade
Temp. of the Low Pressure Turbine	$\mathbb{R}$, measured in centigrade
Pressure of Fan Inlet	$\mathbb{R}$, measured in PSI
Phys. Fan Speed	$\mathbb{R}$, measured in RPM
Phys. Core Speed	$\mathbb{R}$, measured in RPM

Table 1: Sensor values and Monitors (Conditional Indicators) for the CMAPS-S Engine Data

fault(s) and their time of introduction was known, so it was easy to assign *nominal* and *faulty* labels for each data stream. The CMAP-S data models three faults: (1) a fan fault (Fan), (2) a High Pressure Compressor fault (HPC), and (3) a High Pressure Turbine fault (HPT). The reference model for the three faults could be constructed in different ways. For example, one could construct three different models – each model defining a classifier that differentiated a fault condition from nominal behavior. Another possibility was to treat the model building as a multi class learning problem. The result would be a single classifier structure that distinguished between four hypotheses that included the three faults and nominal operations. This structure as the model would likely produce insights on how to differentiate between several faults hypothesis. Given that we were adopting an exploratory framework to study the effectiveness of different classifier models, it made sense to compare between different classifier structures and analyze the discriminating evidence provided by each model. Furthermore, the availability of the CMAPS-S simulator facilitated this approach, since in real situations it may be hard to collect sufficient amounts of fault data to build robust classifiers that include multiple fault hypotheses.

6 Experiments

To initially evaluate the ability of the data mining techniques to improve over the Naïve Bayes based reference models, we have conducted and evaluated a set of experiments using the data from the CMAPS-S engine system to establish whether the TAN-based model produces a better diagnostic classifier than a reference model that is implemented as a Naïve Bayes Classifier. Our experiments compare the performance results of the Naïve Bayes versus the TAN models.

In the CMAPS-S data, we utilize two feature sets. The first experiment uses the feature set defined as the baseline reference model(only CIs), and extracts a classifier structure by running our machine learning algorithms. The next experiment adds additional sensors to the baseline that are not conditional indicators, to see if using these sensors can improve diagnostic accuracy while reducing false alarms.

A systematic study of the performance of the algorithms requires running of n-Fold Cross Validation experiments. Dividing the data into n equally sized and distinct sets of samples, each with the balance of classes maintained as in the original set allows for the creation of $n-1$ training sets with the last set being held out as the test set. This is done n times, and the metrics generated are then averaged over each of the n runs. This experimental style helps test the robustness of the classifier and keeps the metrics from being overly optimistic or pessimistic depending on the random construction of one hold out set. The experiments include: (1) derivation of models for the individual faults, and (2) derivation of a model for the multi-fault case. The metrics reported in Tables 2 and 3 are the average of 10-Fold Cross Validation runs.

6.1 Experimental Results

The data generated for the experimental study included the three faults discussed previously, and the analysis was conducted in the cruise mode with the aircraft flying at an altitude of 35,000 feet. The data mining algorithms were run to derive individual models for the three single fault modes, as well as a combined model with all three faults. Tables 2 and 3, summarize our experimental results in terms of the accuracy metrics, i.e., overall accuracy (Acc), false positives (FP), and false negatives (FN).

6.1.1 Experiment 1

The Naïve Bayes model with only the CIs represents the reference model for analysis of core engine anomalies. The TAN structure with additional causal relations results in a model with better accuracy. The results in Tables 2 and 3 demonstrate that the TAN Structure for the FAN Fault and the multi-fault classifier have higher accuracy. Their superior performance shows that even with a small number of fea-

	Fan			HPC			HPT			All Three		
	Acc	FP	FN	Acc	FP	FN	Acc	FP	FN	Acc	FP	FN
Naïve Bayes Network	67.9	15.4	36.7	71.4	0	35.3	94.2	0	9.3	82.1	15.5	19.6
TAN	99.4	0.4	0.7	80.8	36.7	0	94.7	8.9	2.9	97.4	1.1	3.8

Table 2: Cruise Mode: Model with Only Conditional Indicators

	Fan			HPC			HPT			All Three		
	Acc	FP	FN	Acc	FP	FN	Acc	FP	FN	Acc	FP	FN
Naïve Bayes Network	68.8	12.5	49.5	72.9	0	56.7	93.8	3.6	9.9	84.9	1.1	23.2
TAN	99.8	0	0.4	87.96	23.0	0	96.6	5.4	0.5	98.0	0.8	0.7

Table 3: Cruise Model: Model with Conditional Indicators + Sensor Measurements

tures(3), introduction of two new causal links, the results improved considerably(67.9% to 99.4% for the Fan and 82.1% to 97.4% for multi-fault). Figure 3 shows the representative TAN used in the multi-fault scenario(the NB Model on the right is for comparison). The CI corresponding to stall margin for the Low Pressure Compressor provided the best discriminating evidence between different faults when only conditioned by the class variable. For the single fault classifiers, the Fan and HPC TANs outperformed the Naïve Bayes, but the HPT classifier provided minimal improvement. The HPT Classifier seems to require a simple classifier and both models achieved over 90% accuracy. The classifiers for the HPC fault were the lowest performing set. Although the TAN did better than the NB classifier by over 8%, this would indicate that the reference model for the engine may not be able to detect and isolate this fault, particularly from cruise data.

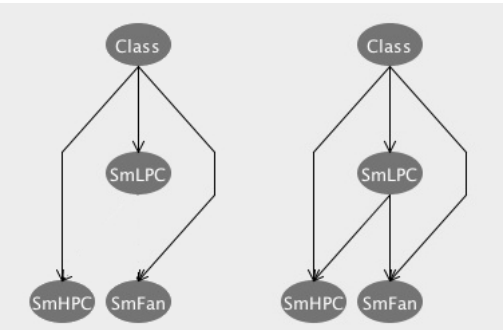

Figure 3: NB Model on the left and the TAN Model on the right for the Multi-Fault Scenario with Only CI

6.1.2 Experiment 2

For the second set of experiments, we consider the additional sensors. From Table 3, there is an improvement in the accuracy numbers for all of the TAN models. This is highlighted by the HPC fault scenario, which was problematic in first experiment, but the accuracy increased significantly. This improved the False Positive rate, while not increasing the corresponding false negative metric. This additional information improved it significantly over its Naïve Bayes counterpart as well as the models in the first experiment. This improvement without a negative cost to the error rates is true for the TAN models across all scenarios. As interesting observation is that the additional information seems to have had a small negative impact in a few cases of the Naïve Bayes models. In summary, the additional information provided an advantage to the TANs , which were able to generate additional causal relations and information to improve diagnostic accuracy.

Figure 4 displays the TAN model structure generated for the HPC scenario. This TAN model with additional features has an accuracy metric of 88% as compared to the original TAN model that produced an accuracy of 80.8%. The Naïve Bayes Model using the additional sensors improved to 72.9%, from the original Naïve Bayes model at 71.4%. The accuracy results clearly indicate: (1) additional sensor information increases diagnostic accuracy and (2) Switching from a Naïve Bayes to a TAN model improves diagnostic accuracy.

This improvement can be examined visually in Figure 4, where in place of the three CIs, the Mach Number sensor becomes the observational root node. The new causal structure, captured in Figure 4 shows the Fuel Flow sensor as a parent to two of the CIs. Network structures such as the one for the HPC fault explicitly illustrate how additional sensor information can be included to enhance the accuracy of the reference model. In general, the new causal relations suggested can be examined by a domain expert who in turn can construct new and improved indicators to use in a reference model. The results generated by these data driven models can provide numbers on how the new in-

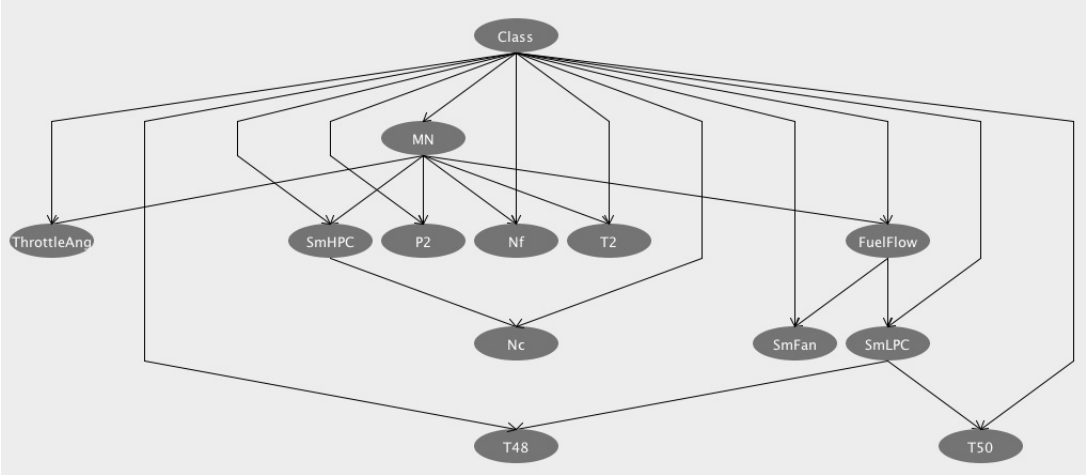

Figure 4: TAN Model for HPC Scenario with Conditional Indicators and Extra Sensors

formation can improve the accuracy of the diagnoser, and how it may impact the error rates.

7 Conclusions and Future Work

The results on experiments conducted with the CMAPS-S data illustrate the promise of the methodology and process we have been developing. To further validate our work, we have identified a number of directions and tasks we need to pursue as we move forward in this project.

- The Naïve Bayes Classifier is an approximation to the expert built reference models. We would like to perform a more thorough experiment and use actual models constructed by domain experts.
- Simulation systems, such as CMAPS-S study particular systems, like the core engine functions in greater detail than any information that can be derived from sensors and monitors in current aircraft configurations. We are looking to develop methods by which detailed simulation data may be combined with actual aircraft flight data to carry on extensive analyses of diagnostic and prognostic events and their propagation through the aircraft system.

Acknowledgements

This project has been supported by NASA NRA NNL09AD44T.

References

Chickering, D. M., Heckerman, D., & Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. In *In Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann.

Cohen, I., Goldszmidt, M., Kelly, T., Symons, J., & Chase, J. S. (2004). Correlating instrumentation data to system states: a building block for automated diagnosis and control. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6* (pp. 16–16). Berkeley, CA, USA: USENIX Association.

Frederick, D., DeCastro, J., & Litt, J. (2007). *Users Guide For the Commercial Modular Aero-Propulsion System Simulator* (Tech. Rep.). NASA.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, *29*, 131–163.

Hall, M., Eibe, F., Holmes, B., Geoffrey amd Pfahringer, Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, *11*(1), pp. 10-18.

Honeywell. (2010). Vehicle Integrated Prognostic Reasoner. *NASA Contractor Report to appear*, *NNL09AD44T*.

Kruskal, J., Joseph B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, *7*(1), pp. 48-50.

Schwarz, G. (1978). Estimating the Dimension of a Model,. *Annals of Statistics*, *6*.

Spitzer, C. (2007). Honeywell Primus Epic Aircraft Diagnostic and Maintenance System. *Digital Avionics Handbook*(2), pp. 22-23.

Witten, I., & Frank, E. (1999). *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations*.