Learning Bayesian Network Structures to Augment Aircraft Diagnostic Reference Models

Daniel L. C. Mack, Gautam Biswas, *Fellow, IEEE*, Xenofon D. Koutsoukos, *Senior Member, IEEE*, and Dinkar Mylaraswamy

***Abstract*—Fault detection and isolation schemes are designed to detect the onset of adverse events during operations of complex systems, such as aircraft and industrial processes. The state-of-the-art fault diagnosis systems on aircraft combine an expert-created reference model of the associations between faults and symptoms, and a Naïve Bayes reasoner. For complex systems with many dependencies between components, the expert-generated reference models are often incomplete, which hinders timely and accurate fault diagnosis. Mining aircraft flight data is a promising approach to finding these missing relations between symptoms and data. However, mining algorithms generate a multitude of relations, and only a small subset of these relations may be useful for improving diagnoser performance. In this paper, we adopt a knowledge engineering approach that combines data mining methods with human expert input to update an existing reference model and improve the overall diagnostic performance. We discuss three case studies to demonstrate the effectiveness of this method.**

***Note to Practitioners*—This paper takes a first step toward combining information from adverse event logs matched with real flight data to improve the accuracy and timeliness of diagnoser systems used on commercial aircraft. We have developed a knowledge engineering approach, which uses the results derived from machine learning classifier algorithms to inform experts about changes and additions that could be made to the existing reference model, created by human experts, to improve diagnostic performance. One of the primary constraints we face in this work is not to alter the structure of the diagnostic reference model, which would require changes in the reasoning algorithm for fault diagnosis. With this in mind, we address a number of challenges in developing our methodology. First, we extend the Naïve Bayes learning schema by adopting the Tree Augmented Naïve Bayesian (TAN) learning algorithm that captures some of the dependencies among the monitors in the aircraft diagnostic system. This provides us with more accurate diagnostic results, and we then apply a transformation schema to generate classifier structures that can be matched against existing reference model structures, thus providing the experts a better understanding of the implications of adding new knowledge and detectors to the reference model. Second, we use real flight data to validate the new reference model structure by determining the improvements in diagnostic accuracy and timeliness of isolation using well-defined metrics. Our overall approach shows promise for targeted fault analysis that may lead to faster detection, and, therefore, avoidance of adverse events such as an engine shutdown during flight. However, the task of studying and refining large, centralized reference models for aircraft systems is complex, especially for quantifying diagnostic accuracy and false alarm rates across multiple fault modes. We will address this larger task along with detection of previously undetected faults (anomaly detection) in future work.**

***Index Terms*—Aviation safety, classification algorithms, diagnosis, knowledge engineering, tree augmented Bayesian networks (TANs).**

Manuscript received August 25, 2014; revised May 08, 2015, and December 30, 2015; accepted February 27, 2016. Date of publication March 30, 2016; date of current version January 04, 2017. This paper was recommended for publication by Associate Editor H. Hu and Editor M. P. Fanti upon evaluation of the reviewers' comments. The work of D. L. C. Mack, G. Biswas, and X. D. Koutsoukos was supported by NASA NRA under Grant NNL09AA08B. *(Corresponding author: Gautam Biswas.)*

D. L. C. Mack was with the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37235 USA. He is now with Baseball Analytics/Research Science for the Kansas City Royals, Kauffman Stadium, Kansas CIty, MO 64129 USA (e-mail: daniel.mack@gmail.com).

G. Biswas and X. D. Koutsoukos are with the Institute for Software Integrated Systems and the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235 USA (e-mail: gautam.biswas@vanderbilt.edu; xenofon.koutsoukos@vanderbilt.edu).

D. Mylaraswamy is with Honeywell Aerospace Advanced Technology, Honeywell Aerospace, Golden Valley, MN 55422 USA (e-mail: dinkar.mylaraswamy@honeywell.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2542186

I. Introduction

An IMPORTANT challenge for aviation safety is the early detection and mitigation of potential adverse events caused by degradation and failures in system components. Consider an aircraft with several interacting subsystems, such as the propulsion, avionics, bleed, and flight control subsystems. Degradation and faults in one component may affect other components during flight operations. As a result, multiple fault symptoms may be generated, i.e., sensors spread across the system may report anomalous or faulty behaviors. Combining this information in a way that leads to accurate and timely fault detection and isolation is a challenging task.

Current Aircraft Diagnostic and Maintenance Systems (ADMS) [1] use: 1) a system reference model that describes causal relations between potential faults and symptoms that are derived from sensor measurements and 2) reasoning software that combines abductive [2] and Naïve Bayesian reasoning [3] to infer and rank potential fault hypotheses. A widely used ADMS in operation today is the Boeing 777 Central Maintenance System (CMC) [4].

Separating the reference model from the reasoner software allows subsystem manufacturers to encode proprietary fault models for individual subsystems into the reference models. The system integrator (i.e., the aircraft manufacturer) designs the integrated solution that combines information from the subsystem reasoners to make global diagnostic inferences [5]. Bayesian reasoning methods address the uncertainty in the diagnostic relations and improve reasoner robustness in the presence of missing evidence [6]. This results in a better overall ranking of the potential fault hypotheses based on

their likelihood of occurrence. However, the accuracy, robustness, and timeliness of the reasoner are very much a function of the accuracy of the system reference model. Experts use their knowledge of subsystems and experiences derived from previous aircraft operations, but gaps may arise because: 1) components are periodically updated as newer, more improved versions become available and 2) complex interactions between subsystems are hard to model *a-priori*. Often, such knowledge comes from years of experience, and abnormal situations are typically understood only when a pattern emerges after multiple occurrences.

Data mining methods applied to large sets of operational data collected by the airlines and equipment manufacturers provide means for targeted anomaly detection and fault diagnosis in aircraft systems [7]–[9]. Similar methods have been developed for diagnostics applications in other domains, e.g., [10]–[13]. This paper develops an approach that employs targeted search techniques with a Bayesian learning algorithm to detect and analyze the onset of faults that may lead to adverse events during aircraft flight operations. The challenges we address in this approach are: 1) finding the right flight data segments that can inform the diagnostic system about specific faults that occur during flight; 2) using machine learning to generate diagnostic structures that are compatible with existing reference models; and 3) providing the information derived by machine learning to the aircraft experts in a format that makes it easy for them to integrate the information into existing reference model structures. The effectiveness of this methodology is demonstrated using three case studies: 1) engine overheating problem caused by a leak in a fuel metering hydromechanical (HMA) unit; 2) engine shutdown triggered by the fire alarm system of the engine; and 3) excessive vibration that led the crew to shut down the engine manually.

Many flight management and flight control functions on aircraft are now handled by software [14]. This software has to meet stringent certification requirements (DO-178 or Level 1 certification). In contrast, ADMS systems play an advisory role during flight, therefore, they require less stringent Level 4 certification. Level 4 certification implies that only the ADMS reasoner code is certified, and requires recertification only if changes are made to that code. On the other hand, the system reference model associated with the reasoners is treated as data, and can be updated by system experts without recertification. In this work, we operate under the constraints of improving the ADMS, while avoiding expensive recertification costs. Therefore, our data mining solutions are designed to support system experts in their knowledge engineering tasks that can be implemented without requiring reimplementation or updating of the reasoner code.

In this paper, we focus on updating reference models for aircraft engine systems using *five* years of flight data from a U.S. regional airline that operated a number of identical aircraft. The 0.7 terabytes of flight data was collected from a large number of aircraft monitors and sensors, many of them associated with the four engines on the aircraft.[1] Data curation methods were developed to extract relevant data for our targeted knowledge engineering application [5]. Our knowledge engineering framework includes four steps: 1) select relevant data to derive new knowledge for targeted diagnostic analysis; 2) apply data mining algorithms to build targeted models associated with specific faults; 3) utilize the derived models to help domain experts update reference models to improve diagnostic performance; and 4) perform experiments to demonstrate that the augmentations lead to overall improvements in diagnostic performance.[2]

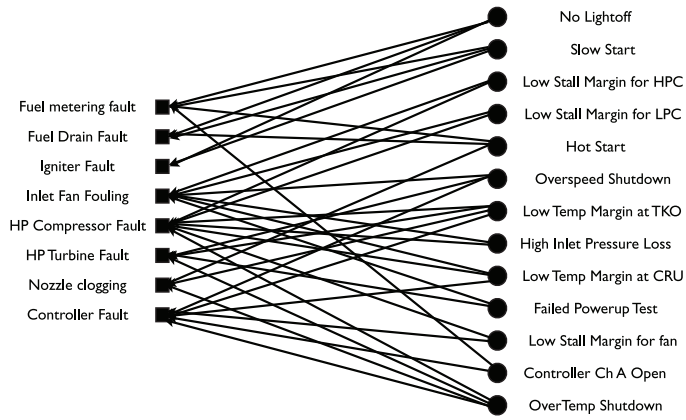

Fig. 1. Example reference model.

The rest of this paper expands on the knowledge engineering process. Section II briefly reviews on-board model-based diagnostic reasoner systems. Section III describes the implementation of the knowledge engineering approach for incorporating information from learned Bayesian models to the ADMS reference model in a way that does not violate the assumptions and properties of the reasoner. Section IV presents the results of our three case studies that demonstrate how human experts utilized the framework to interpret the information generated by our Tree Augmented Bayesian Network (TAN) structures to update existing reference models. Section V presents related work and Section VI outlines possible extensions for generalizing the case studies to other diagnostic applications.

II. Aircraft Reference Model Structure and Diagnostic Reasoners

A traditional system reference model, such as the one used in the Boeing 777 Central Maintenance Computer (CMC) [15]), can be represented as a flat bipartite graph with two types of nodes: 1) failure modes or hypotheses and 2) sensor and monitor nodes as evidence variables. Fig. 1 shows an example reference model for an engine subsystem.

Diagnostic monitors represent the evidence nodes in the system. Designing a monitor often requires deep domain knowledge about the component or subsystem, but the component manufacturer may not reveal this information to the system integrator. The abstract view of a monitor exposed to the system integrator is shown in Fig. 2. With few exceptions, the output of a diagnostic monitor is derived by applying a threshold to a time-series signal. This signal can be a raw sensor value or be derived from a set of one or more sensor values. The intermediate derived signals are labeled as *condition indicators*

[1]The aircraft flight data set used in this study can be accessed at https://c3.nasa.gov/dashlink/projects/85/

[2]A more detailed description of this work appears in D. Mack's Ph.D. dissertation: http://etd.library.vanderbilt.edu/available/etd-04092013–182409/

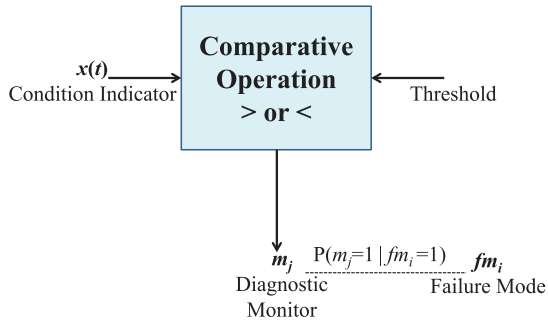

Fig. 2. Abstraction of diagnostic monitor assuming conditional independence of monitors given an fault.

(CIs), $x(t)$. Assuming a predefined threshold value θ, we set $m = 1 \Leftrightarrow x(t) \lesseqgtr \theta$. The binary output of the monitor makes the computational framework of the corresponding diagnostic reasoner easier to implement.

Given the set of distinct failure modes in the system, F, each failure mode variable $fm_i \in F$ takes a binary value[3]:

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

The *a priori* probability of failure mode fm_i is denoted by $P(fm_i = 1)$. Failure modes are assumed to be independent, i.e., given any two failure modes fm_k and fm_j, $P(fm_k = 1|fm_j = 1) = P(fm_k = 1)$.

For a set of diagnostic monitors, DM, monitor $m_j \in DM$ may *indict*, *exonerate*, or provide *no evidence* for or against a subset of failure modes. This defines a monitor's *ambiguity group*, and each monitor m_j in the system expresses indicting, exonerating or unknown support for failure modes in its ambiguity group, i.e.,

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

Ideally, a monitor m_j fires when at least one failure mode in its ambiguity group is occurring. Given that the i^{th} failure mode, fm_i, is occurring in the system, d_{ji}, the conditional probability of the j^{th} monitor evidence given the failure mode is

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

False alarm probability, the probability that an indicting monitor fires when the corresponding failure modes in its ambiguity group are not occurring in the system, is given by

$$\epsilon_j = P(m_j = 1|fm_i = 0, \forall fm_i \in Ambiguity\ Set). \tag{4}$$

Typical ADMS reasoner algorithms simplify calculation by making the Naïve Bayes assumption [3]. Therefore, fault hypotheses (fm_i's) are independent of one another, and given fm_i, monitors (m_j's) that support fm_i are considered to be independent of one another. As monitors fire, the reasoner algorithm first performs an elimination step where failure modes exonerated by newly activated monitors are removed from the set of probable failure hypotheses. For the remaining fault hypotheses, the likelihood of fault hypotheses are updated using Bayes rule and the Naïve Bayes assumption

$$\begin{aligned} P(fm_i|m_j, \ldots, m_p \cdots) &= \alpha \times P(m_j, m_k, m_l \cdots |fm_i) \\ &= \alpha \times P(m_j|fm_i) \times \cdots \\ &\quad \cdots \times P(m_p|fm_i) \end{aligned}$$

where α is a normalizing constant. As additional monitors fire, the failure mode set becomes smaller, and may reduce to a single hypothesis. In situations where more than one failure mode remains active, the reasoner ranks the active hypotheses in the order of their likelihood of occurrence. The probability of false alarms calculated in parallel, indicates the level of uncertainty in the inferred fault modes, given the set of monitors that have fired. This reasoning algorithm for aircraft fault diagnostics is very similar to the one adopted for probabilistic diagnosis of disease hypotheses in the Internist-1/QMR system [16]. More recent work, e.g., [9], have relaxed the Naïve Bayes assumption, to design diagnostic reasoners based on general Bayes net schemes, but these approaches have only been applied in small case studies, e.g., software signal handling faults in aircraft navigation systems. There has also been research on Dynamic Bayes nets for modeling the evolving dynamics of faults in continuous and hybrid systems (e.g., [17] and [18]), but these methods have not been scaled up to apply to large, complex systems.

III. The Knowledge Engineering Approach for Augmenting Reference Models

In this section, we elaborate on the four step knowledge engineering process outlined in Section I.

A. Building Relevant Flight Segments

The set of monitor and condition indicator values relevant to the fault was obtained by analyzing the reference model and by seeking expert input for additional features. The flight data segments with failures, identified by information gleaned from the FAA Aviation Safety Information Analysis and Sharing (ASIAS)[4] database, provided information about the aircraft tail number, the date, and the flight when the adverse event occurred, and additional information about the fault mode that caused the adverse event. To ensure that we captured indicators that imply early onset of the failure, we went back N flights from the flight where the adverse event was reported to facilitate early detection and repair, and thus avoid adverse events during flight. The number N depended on the nature and type of fault.

1) Aircraft Flight Data: The aircraft flight data was generated from a fleet of 30+ identical four engine aircraft that composed a U.S. regional airline. The data covered about five years of flight operations, with each aircraft involved in 2–5 flights each day. The Aircraft Condition Monitoring System (ACMS)

[3]A value of -1 may be used to denote an unknown failure mode.

[4]http://www.asias.faa.gov/pls/apex/f?p=100:1:13807616980905

TABLE I
STARTUP FEATURES TRANSFORMED FROM THE RAW DATA

CI Name	Description
StartTime	Time the engine takes to reach its idling speed. Appropriate threshold generates the *no start* diagnostic monitor.
IdleSpeed	Steady state idling speed. Appropriate threshold generates the *hung start* diagnostic monitor.
peakEGTC	Peak exhaust gas temperature within an engine start-stop cycle. Appropriate threshold generates the *overtemp* diagnostic monitor.
N2atPeak	Speed of the engine when the exhaust gas temperature achieves its peak value. Appropriate threshold generates the *overspeed* diagnostic monitor.
timeAtPeak	Dwell time when the exhaust gas temperature was at its peak value. Appropriate threshold generates the *overtemp* diagnostic monitor.
Liteoff	Time duration when the engine attained stoichiometry and auto-combustion. Appropriate threshold generates the *no lightoff* diagnostic monitor.
phaseTWO	Time duration when the engine controller changed the fuel set-point schedule. There are no diagnostic monitors defined for this CI.
prelitEGTC	Provides the engine combustion chamber temperature before the engine attained stoichiometry. Appropriate threshold generates the *hot start* diagnostic monitor.
tkoN1, tkoN2, tkoEGT, tkoT1, tkoPALT	Fan speed, engine speed, exhaust gas temperature, inlet temperature and pressure altitude, respectively, averaged over the time interval when aircraft is in take off mode. There are no diagnostic monitors associated with these CIs.
tkoMargin	Temperature margin for the engine during take off conditions. Appropriate threshold generates the *medium yellow* and *low red* diagnostic monitors.
Rolltime	Time duration of the engine's roll down phase. Appropriate threshold generates the *abrupt roll* diagnostic monitor.
resdTemp	Engine exhaust gas temperature at the end of the engine's roll down phase. Appropriate threshold generates the *high temp* diagnostic monitor.
N2atDip, dipEGTC	Engine speed and the exhaust gas temperature at the halfway point in the engine's roll down phase. There are no diagnostic monitors defined for these CI.
N2cutoff	Rate of change of the engine speed at the halfway point in the engine's roll down phase. There are no diagnostic monitors defined for these CI.

associated with these aircraft collected sensor data from the propulsion, airframe, aircraft bleed, and flight management systems in a central location on the aircraft during flight to support fault analysis by the on board diagnoser and maintenance operations on the ground. The sensors had different precision levels and different sampling rates, therefore, the data samples collected by each sensor per flight was different. This data was stored in raw, uncompressed form as binary files. On landing, the ACMS recorded data was transferred to permanent storage (in our case, the data was stored on CDs). This initial step indexed the flight data by the aircraft tail identification number and date and time of flight.

In addition to the flight data, we had independent access through the ASIAS database to a collection of adverse events reported by various airline operators to the FAA. Examples of adverse events in our flight data included events, such as *loss of an engine* and *engine on fire*. Many of these incidents resulted in the affected aircraft abandoning their flight plan, and making an emergency landing at the nearest airport. Our three case studies were derived from incidents reported in the ASIAS database.[5]

2) Brief Overview of Case Studies: Two of our three case studies were *computer-aided engine shutdown* events during flight, and the third was an *excessive engine vibration* event that resulted in a crew-initiated shutdown of that engine. From the ASIAS records, we identified the tail numbers of the affected aircraft and the exact flight in which the adverse event occurred. The first case study is an *engine overheating problem*, which triggered an engine shutdown event on the belief that the engine was in imminent danger of catching fire. The second event involved *excessive vibration in an engine*, which forced the crew to shut the engine down manually. The third event was an engine shutdown triggered by the fire alarm system on the engine. After the fact, FAA investigators determined that the cause was a *leaking fuel manifold* but the leak was not detected by existing sensors and monitors on the aircraft.

3) Curation of Flight Segments: Our case studies, focused primarily on the aircraft engine subsystem and fuel flow into the engines. A set of condition indicators (CIs) related to engine health were extracted as time series data, and then annotated by the three primary modes of operation of the engines: 1) startup; 2) takeoff; and 3) shutdown. Our experts surmised that the engines were most stressed during takeoff, and knowing the state of the engine at the start and end of a flight, was more important for diagnosing incipient faults. Therefore, we did not include data from the other phases: climb, cruise, and descent/landing in our analyses. The flight segment data was obtained in two steps: 1) data from all flights for the selected condition indicators was collected into the curated database for all four aircraft engines and 2) the labeled flight segments, representing nominal and faulty situations were extracted into individual data sets for the classifier studies. Table I lists the CI's used for each flight segment.

B. Classifier Methods for Deriving Diagnostic Relations in the System

1) Learning Tree Augmented Naïve Bayesian Networks: For aircraft systems, it is well known that the CIs may not be independent given a fault hypothesis: 1) CIs may be based on

[5]In developing our case studies, we ignored ASIAS events like sprinkler incidents in the main cabin, because they did not have serious implications on aircraft flight safety.

dependent measurements, e.g., a CI derived from a pressure measurement at the end of a pipe is not independent of a second CI whose value is derived from a pressure measurement at the inlet of the pipe and 2) two CIs may share one or more sensor measurements, e.g., two different measures of engine health state may use the engine temperature in their computations. In other words, the Naïve Bayes assumptions of the reasoner and the conditional independence assumption on CI's given the fault hypothesis are not true in reality. However, given our discussion earlier, we were not at liberty to make changes to the reasoner algorithm. To overcome this problem, we chose a Bayesian learning algorithm that was not computationally expensive, but the independence assumptions could be relaxed to capture additional diagnostic evidence. Our hypothesis was that this information could be used to improve the diagnostic results.

The Tree Augmented Naïve Bayesian learning algorithm [19], also called the Bayesian TAN classifier meets these requirements. The TAN structure provides a simple extension to the Naïve Bayes model. In our derived TAN structures, the fault hypothesis, the root or class node, is causally linked to every CI, which represent the evidence nodes that support the hypothesis. In addition, an evidence node (CI) can have at most two parents: 1) the class node and (2) a causal connection to another evidence node (CI). These constraints maintain the directed acyclic graph requirement of Bayesian networks, and produce a more nuanced tree that captures additional dependency relationships among the CIs without allowing arbitrary graphical structures that would make it harder for the expert to interpret, and extract relations to enhance the reference model.

TAN structures can be generated in several ways. A traditional approach uses a greedy search that constrains the graph from building "illegal" edges from the evidence nodes[6] [20]. In our work, we developed a systematic procedure to build the TAN structure for a fault hypothesis. Our approach first derives the Minimum Weighted Spanning Tree (MWST) for all of the evidence (CI) nodes using Kruskal's algorithm [21], and then completes the TAN structure by connecting the fault node (root) to all of the evidence nodes in the tree [19]. We used the Mutual Information (MI) metric for pairwise edge weight computations when constructing the MWST [19]. The MI measure is not directional. Directionality of the causal links was established by selecting one of the evidence nodes in the MWST as the *observational root node* and recursively directing all edges from this node outward. The observational root node is defined as the evidence node with the highest likelihood given the fault mode.

A TAN structure generated using the MWST algorithm is illustrated in Fig. 7. The root or class node of the TAN, corresponds to the fault mode under study, i.e., the *FuelHMA* fault. Next, Rolltime, a monitor associated with the shutdown phase of the aircraft is selected as the observational root node from the constructed MWST. As discussed, the fault hypothesis node (class) is then linked to all of the monitor nodes that support the class node. Dependencies among other monitors, e.g., Rolltime and dipEGTC correspond to links from the MWST. The TAN represents a static Bayes net structure; it does not capture temporal relations among the evidence nodes. The selection of the observation root node, the only evidence node that has one parent (the class node) is based on the fact that it provides the strongest evidence in support of the class node among all of the evidence nodes of the TAN structure.

The choice of the observational root node also provides a heuristic ranking of the evidence nodes. Much like Information Gain in a decision tree [22], the mutual information calculation for each class node to CI node edge is used to create an ordering of CIs from larger to smaller impact on the class node. In other words, generated TAN structures point the domain expert to the observational root node as the primary evidence that supports the fault mode, and as one moves down the tree hierarchy (see Fig. 7), the corresponding CI nodes have a smaller impact in establishing the fault mode.

We used an implementation of the TAN algorithm from the Weka [23] toolkit for our case studies. Weka uses Conditional Probability Tables (CPT) and preprocesses the data using a discretization algorithm. The discretization algorithm bins the individual features into ranges that create the biggest unbalance in the class labels for each feature value (or pairs of feature values when there is a dependency between features), to generate CPTs that provide the most differentiation between classes. The choice of the observational root node is determined by the CI node that provides the best discrimination among the nominal versus faulty class as calculated by the mutual information measure. The value of the CPT and more specifically the ranges found by the preprocessing algorithm are essential for updating existing monitor thresholds and adding new links between monitors and the fault hypotheses in the reference model.

2) Using N-Fold Cross Validation to Validate the TAN Classifier Results: We divided up the data segments into training and test sets, and ran N-fold cross validation studies to estimate the accuracy of classification and the false-alarm rate for the derived TAN structures, and to determine if the error rates met the requirements of the aircraft diagnosis task.

C. Updating the Reference Model

The domain experts used the learned TAN structures to update the reference model. Since the example reference model in Fig. 1 is reasonably complex with a number of multiply-connected nodes, we demonstrate the types of information our experts extracted to augment in the reference model with a simpler example illustrated in Fig. 3. We limited our approach to the three types of reference model updates discussed in Section I. The first was related to scaling problems for conditional probability distributions for large models. Consider the example where the conditional probability between FM_2 and DM_2 has to be updated because the TAN structure implies a better threshold for monitor DM_2. DM_2 is a shared monitor between fault hypotheses FM_1 and FM_2, which means the two faults are causally dependent. Therefore, to reason about the likelihood of FM_1 being indicted by the evidence, i.e., $P(FM_1|DM_1, DM_2)$, we have to consider marginalization of the joint distribution $P(FM_1, FM_2, DM_1, DM_2, DM_3)$ with respect to nodes FM_2 and DM_3. Generating the joint probability distribution table requires more information than the domain expert may be able to provide, and it is also hard

[6]An illegal edge is created when more that two edges are created from an evidence node to parent nodes. Note that one of the parent nodes has to be the class node.

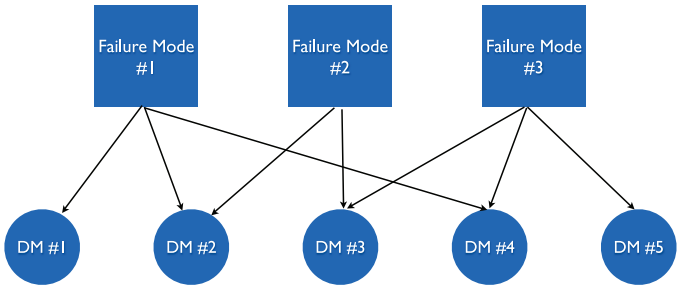

Fig. 3. Graphical representation of a reference model.

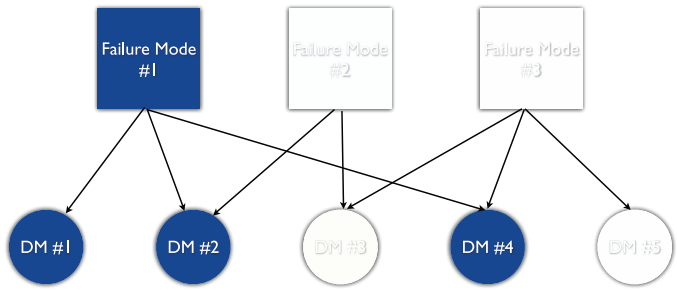

Fig. 4. The relevant structure after isolating a failure mode.

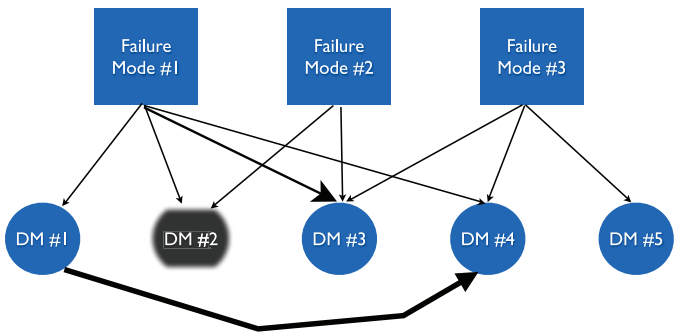

Fig. 5. Additional Information derived from data: (a) update to monitor threshold DM_2 with respect to fault FM_1; (b) finding a new relation between FM_1 and DM_3; and (c) discovering that monitors DM_1 and DM_4 are causally related.

to directly derive this information from data [24]. Preserving the Naïve Bayes model structure assumptions, i.e., the independence of the fault hypotheses and the independence of the monitors associated with a fault hypotheses, simplifies this task of deriving the conditional probabilities. In our example, the discovery of a new link between FM_1 and DM_3 makes all of the failure modes dependent, which greatly increases the number of parameters needed to specify the joint probability distribution. The Naïve Bayes assumption allows for a simplified refactoring of the problem, making the conditional probability tables easier to specify. Fig. 4 shows the local structure used for failure mode FM_1.

A second challenge arose from dependencies among monitors, such DM_1 and DM_4 in Fig. 5. This clearly violates the assumption of independence of monitors given the fault mode. We address this problem by defining the notion of a "super monitor." To accommodate the dependency between DM_1 and DM_4 while retaining the Naïve Bayes modeling framework, the two monitors are combined to form a *Super Monitor*, and

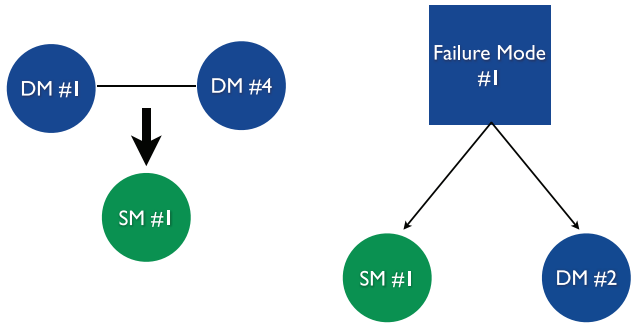

Fig. 6. The construction of a super monitor.

the substructure between FM_1, DM_1, and DM_4 is replaced by a new node SM_1 and a link from FM_1 to SM_1, as shown in Fig. 6. In general, combining existing monitors, M_i and M_j implies stronger indictment evidence for the failure mode FM_k. That is

$$P(DM_i = 1, DM_j = 1 | FM_k = 1) > P(DM_i = 1 | FM_k = 1) \times P(DM_j = 1 | FM_k = 1).$$

Note that monitors DM_i and DM_j are not removed from the reference model because they may provide supporting evidence for other faults. This illustrates yet another local update method applied to the reference model. The three reference model update procedures are summarized below.

1) Update Monitor Thresholds: Updating the threshold θ associated with a diagnostic monitor should make the monitor i more sensitive to failure mode j (allowing the fault can be detected earlier) without degrading the false alarm rate. As an example, consider a change in the threshold for monitor DM_2 with respect to fault FM_1 (see Fig. 5). The threshold value may be made lower to make the fault mode more sensitive to the monitor value, or it may be increased to decrease the false alarm rate.

In more detail, updating monitor thresholds require further analysis of the discretization of the CI used to create the CPTs. Applying marginalization to the CI parent (if one exists) will produce general probabilities for each set of ranges found through the discretization. The fault range is established from the range that has the highest probability of the failure mode given the marginalized CPT. The value that defines the border between the nominal range and faulty range is taken as the new threshold for the monitor. Given the data associated with the structure in Fig. 7, the derived CPT for the Rolltime CI is given in Table II. The table indicates that the fault node, Fuel HMA failure, is more likely when Rolltime is > 34.875. Using these results, the experts update the threshold in the reference model with the goal of improving the accuracy and time of detection for the FuelHMA fault.

2) Add New Links Between Monitors and Failure Modes: Discovering new relations between monitors and fault hypotheses is equivalent to deriving a new CI in the TAN structure. With expert guidance, this creates a new monitor that is added to the reference model. Proper choice of threshold (similar to the Update Monitor approach) helps to improve the fault detection accuracy. If, for example, $resdTemp$ appears in the TAN structure of Fig. 7, but it does not exist in the reference model, the experts and the data mining researchers study the

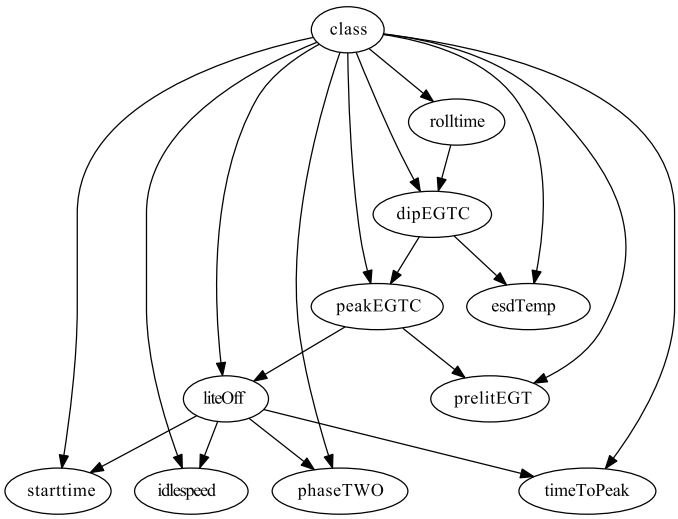

Fig. 7. TAN structure generated using data from all 50 flights.

TABLE II
EXAMPLE CPT FOR FINDING THRESHOLDS

Class	(-inf-34.875]	[34.875-inf)
Nominal	.823	.177
Fault	.227	.773

CPT for this CI and add a new monitor with a threshold based on the value discovered in the CPT.

A second possibility is that the threshold associated with an existing CI contradicts the threshold value of a monitor that already exists. For example, the CPT associated with this CI indicates the higher likelihood of a fault when the CI values exceeds a threshold, but the existing monitor is designed to generate an alarm when the CI value falls below a second threshold. After careful examination, the domain experts may conclude that the addition of a new diagnostic monitor defined by the new threshold helps to improve the detection performance. Using the example of the threshold for Rolltime in Table II, a new monitor is defined with the threshold of greater than 34.875 because the previous Rolltime monitor was designed to generate an alarm for values less than a threshold value $\theta = 34.875$. The earlier monitor is replaced by the new monitor for online detection and isolation.

3) Create Super Monitors: If a new relation between an observational root node and a child node in the TAN structure is deemed important by the experts, this becomes the basis for developing a *super monitor*. The interrelationship among the CIs implied by the TAN model can be transformed into a new monitor that computes its value across adjoining flights segments. For example, if the TAN structure showed a possible relationship between monitors in flight segment n followed by flight segment $n+1$, the causal implications of this could be captured in a new monitor that fires only when the two original monitors fire in sequence, first for flight n and then for flight $n+1$. Not only does this super monitor combine the results from other monitors, but it also indicates cyclic behaviors that again provide additional diagnostic information not originally captured by the reference model.

In general, super monitors can model complex interactions that improve the isolation function of the reasoner. The creation of a super monitor results in the links from the individual monitors to the failure mode being removed (they remain active for other failure modes in the original reference model). The new super monitor uses logic, such as AND and OR to combine results from the original monitors. Also, a new monitor may be subsumed into a super monitor relation. As an example, our experts used the TAN in Fig. 7 and the monitors associated with Rolltime and dipEGTC, and decided that the combined relationship was strong enough to produce a super monitor that indicts a fault if and only if, both the monitors for Rolltime and dipEGTC would indict the fault. Super monitors are likely to improve diagnostic accuracy while decreasing the false alarm rate.

D. Validating the Impact of the Expert Updates to Diagnostic Accuracy

It is imperative to determine if the updates made to the reference model truly improve the detection time and diagnostic accuracy. Traces generated from the data of the flights leading to the faulty incident are used as input to the reasoner with both the original model as well as the updated reference model. Examining when the reasoner identifies the fault determines whether the augmented reference model provides sufficient improvements in detecting and isolating the correct fault. An improved performance implies earlier maintenance decisions therefore, improved overall safety. The output will either be confirmation of the approved changes, or empirical proof to reject the changes.

IV. CASE STUDIES

Three case studies demonstrate the effectiveness of our approach to updating the subsystem reference model for improving diagnostic performance. Domain experts played an integral role in interpreting the TAN structures derived from flight data and updating the reference model. A tenfold cross validation approach along with two standard metrics: 1) the classification accuracy and 2) the false positive rate were used to evaluate the TAN models. After updating, the system reference model was tested along with the reasoner to determine if it provided an improvement in diagnostic performance, i.e., higher accuracy and faster detection time.

A. Case Study 1: Fuel HMA Fault

The FuelHMA fault resulted in engine overheating and eventual shutdown. The TAN classifier was derived by comparing the data from the faulty engine against the three other engines on the aircraft, which were assumed to operate normally during the period of 50 flights before the adverse event.

1) Experiment 1: Classification Accuracy of the Generated TAN Structure: Experiment 1 was used to study the effectiveness of the generated TAN classifier structure in isolating the FuelHMA fault condition. The values for the CIs chosen by our experts, was calculated for 50 flights before the engine shutdown event. Data from the three aircraft engines that showed no abnormalities (1, 2, and 4) was labeled as nominal, and the

TABLE III
ACCURACY, FALSE POSITIVE RATE FROM DIFFERENT DATA SEGMENTS

Bin	Flights	Acc.	FP%
1	1 to 10	97.65%	2.30%
2	11 to 20	93.90%	5.70%
3	21 to 30	94.65%	5.30%
4	31 to 40	96.62%	3.50%
5	41 to 50	96.06%	4.10%

TABLE IV
OBSERVATIONAL ROOT NODE AND IMMEDIATE CHILD NODE FOR CLASSIFIERS CREATED FROM DIFFERENT DATA SEGMENTS

Bin	Flights	Obs. Root Node	Children of ORN	Notes
1	1 to 10	IdleSpeed	startTime	Thresholds Chosen from this Bin due to low FP
2	11 to 20	peakEGTC	liteOff,dipEGTC	peakEGTC Important Node
3	21 to 30	peakEGTC	liteOff,dipEGTC	peakEGTC Important Node
4	31 to 40	startTime	peakEGTC	Links startTime and PeakEGTC
5	41 to 50	liteOff	phaseTwo,RollTime	Links Startup and Rolldown CI

data associated with engine 3, for which the shutdown incident occurred, was labeled as faulty.

The average classification accuracy of the derived TAN structures after running tenfold cross validation was 99.5% with a .7% false positive rate. This clearly implies that the set of CIs chosen were appropriate for detecting and isolating the Fuel HMA fault. As a next step, we checked whether the generated classification structure was an artifact of engine position, i.e., engine three versus the other engines on the aircraft. This required generating the TAN classifier using training data from engine 3 (faulty) and one of the nominal engines (engines 1, 2, or 4). The data from the other two nominal engines was used as test data. The high classifier accuracy (at or above 90%) for the test data sets indicated that the TAN structure was not an artifact of engine position on the aircraft.

2) Experiment 2: Using the TAN Structure to Update the Reference Model: The domain experts examined the TAN structure derived in Experiment 1 (Fig. 7). The expert's attention was drawn to the relationships between different pairs of CI's for different phases of the flight, viz.: 1) Rolltime and dipEGTC during the Shutdown phase and 2) PeakEGTC and startTime during the Startup phase. The experts reasoned that a likely dependence between the shutdown phase of flight n and the startup of the next flight, $n + 1$, i.e., an incomplete or inefficient shutdown for flight n created situations where the startup phase of flight $n + 1$ was affected. The expert hypothesized that this cycle of degradation from previous shutdown to the next startup resulted in the fault effect growing with each flight, and eventually impacted a number of CIs of the faulty engine. This phenomena indicated a causal relation that was not captured in the original reference model. The experts suggested introducing a super monitor that combined the CIs associated with a landing and subsequent takeoff would make the diagnostic reasoner more sensitive to the fault. But the experts wanted to study the data further to gain a better understanding of how to express this relationship between monitors evolved over multiple flights.

To address this, we developed a binning procedure where the 50 flights were divided into five bins of ten consecutive flights each. The data from the ten flights for a bin was used for training, and the data from the other 40 flights was used as test data. Additional test data was also generated from flights after engine three was repaired after the adverse event. Table III shows the accuracy and false positive rate (FP%) metrics reported for the five experiments corresponding to five bins of 10 flights each (for a total of 50 flights). The observation root node, and its immediate child in the generated TAN structures are listed in Table IV.

The conventional wisdom is that the accuracy and false positive metrics will have their best values for the classifiers that are generated from data close to the adverse event occurrence, and performance will deteriorate for the TAN structures derived from bins that are further away from the incident. The results showed partial agreement. The bin 1 experiment produced the highest accuracy and lowest false positive rate, but the next best result came from the bin 4 data. The high performance of the TAN in bin 1 meant that the discretization used in the CPTs derived from that bin should be used for threshold updating and adding any new monitors to the reference model.

While performing this threshold updating, the domain expert discovered additional information. The expert used the startTime CI discretization derived by the TAN algorithm to determine that faster than nominal startTime values produced a higher probability for the fault. The original monitor for this CI was based on a greater than relationship threshold for a slowStart monitor. This new relation derived from the CPT implied a new monitor called fastStart could be added to detect the failure mode. The fastStart monitor, which triggers when the start time exceeds the threshold specified in the CPT, was added to the enhanced reference model.

The results of bin 1 and bin 4 prompted the domain expert to study the bin 1 to bin 4 TANs more closely. The expert concluded that two CIs, startTime and peakEGTC showed a strong causal connection in bin 4, and startTime had a high ranking in the bin 1 TAN. On the other hand, PeakEGTC was the root node for bins 2 and 3. This study led the domain expert to conclude that a new monitor that combined startTime and peakEGTC would further enhance the reference model with better detection and isolation capabilities for this fault. This new diagnostic monitor combines information from the newly formed fastStart monitor and the HighTemp monitor to improve detection of the fuelHMA fault. To accommodate the super monitor, the connection from the FuelHMA fault hypothesis to the individual monitors was deleted to avoid redundancy and preserve the Naïve Bayes structure. Therefore, the updated reference model also includes improved threshold values for some monitors, as well as the new super monitor.

3) Experiment 3: Verifying Improvement in Reasoner Performance: This experiment verified whether the reasoner performance improves with the updated reference model. These results from the reasoner simulations are shown for the original reference model in Fig. 8 and the augmented reference model in Fig. 9. The traces illustrate the reasoner's inferences through a progression of flights before the incident occurred. A green shade on a failure mode indicates that there is a likelihood of the fault given evidence and the number in the box indicates the calculated relative likelihood of the fault. A failure mode

Fig. 8. Trace of the reasoner on the original reference model.

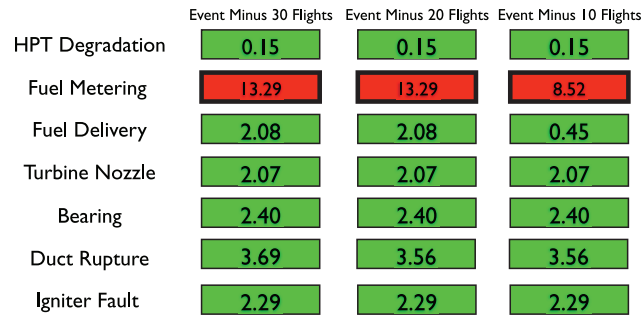

Fig. 9. Trace of the reasoner with the improved reference model.

shaded red, indicates a high likelihood for that hypothesis, and when the failure mode is marked in bold, "Fuel Metering" in this case, it indicates that the failure mode has a very high likelihood, and added to a report for the mechanics. In this case study, the red indicator appeared about 30 flights before the adverse event, which would give the mechanics plenty of opportunities to perform maintenance actions and avoid the adverse event. Verification experiments of this kind are critical not just to establish the fact that the early detection metric is improved, but also that the new information does not create side-effects, such as increasing the number of potential diagnostic hypotheses, which would complicate the mechanics decision making process. In this case, the expert deemed the verification test a success, and the updated reference model is accepted as an improved version of the previous model.

B. Case Study 2: Broken Turbine Blade

The broken turbine bucket blade fault labeled as the *HPT degradation* failure mode caused excessive vibration that resulted in an engine shutdown and the aircraft had to make an emergency landing. This failure, though engine-related, illustrates the analysis of a fault that is physically different from the Fuel HMA failure. We discuss the results of the three experimental steps and additional experiments that were conducted to show that the false alarm rates remain low when the Fuel HMA and HPT degradation faults were compared.

1) Experiment 1: This case study used the same CIs as case study 1 and employed the same tenfold cross validation method for the 50 flights before the engine shutdown incident. The experiments showed an average accuracy of 92.18% and a false positive rate of 2.1% for the derived TAN classifier.

2) Experiment 2: The same binning procedure as case study 1 was applied, and the accuracy and false positive values for

TABLE V
ACCURACY AND FALSE POSITIVE RATE FOR CLASSIFIERS CREATED FROM DIFFERENT DATA SEGMENTS FOR CASE STUDY 2

Bin	Flights	Acc.	FP%
1	1 to 10	90.625%	4.2%
2	11 to 20	92.50%	2.5%
3	21 to 30	87.5%	5%
4	31 to 40	88.125%	12.50%
5	41 to 50	85.625%	11.7%

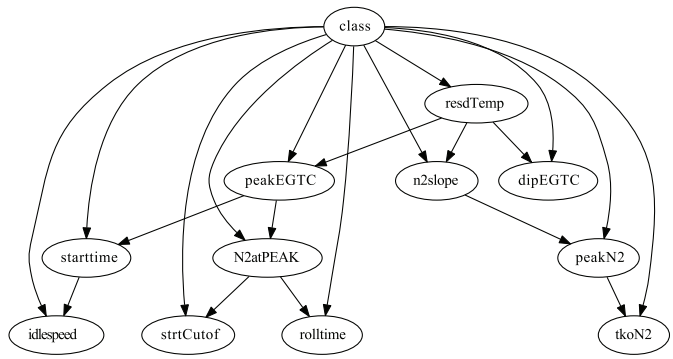

Fig. 10. TAN structure generated using data from case study 2.

the different bins are listed in Table V. Results from bin 2 were chosen for updating thresholds and looking for new monitors. The startTime monitor indicating a slow start turned out to be the most important monitor for fault detection and isolation. There was no overlap between the thresholds for this fault mode and the Fuel HMA fault. This means that in spite of shared monitors, there is no ambiguity in determining the fault mode.

The experts found that the TAN structures generated from the different bins were similar, and, therefore, decided to focus on the TAN generated from all 50 flights. From the structure shown in Fig. 10, the experts focused on the connection between resdTemp, the residual temperature of the engine at shutdown, and the peakEGTC, which is the peak temperature of the engine after startup. The causal direction of this relationship implies that the residual temperature is causally related to the peak engine temperature. The experts decided that this was most likely a relation between the resdTemp of flight n and the startup temperature in flight $n+1$. They used this relation to design a super monitor that indicts the fault, if and only if, the high temperature monitors associated with resdTemp of flight n, fires, and the high-temperature monitor connected to peakEGTC of flight $n+1$ also indicts the fault. Therefore, this super monitor captures temporal information between flights for diagnostic reasoning. Like before, the updated reference model included updated thresholds and the new super monitor.

3) Experiment 3: We generated traces by running the reasoner on the original (Fig. 12) and the updated reference model (Fig. 11). Twelve flights before the adverse event occurred, there were clear indications from the updated reference model pointing to degradation in the high pressure turbine (HPT). Typically, *HPT degradation* will trigger a maintenance request, and the mechanics use a special camera called a borescope to visually inspect the damage and determine if the engine should be removed for maintenance to avoid safety incidents in future flights. The maintenance procedures replace the broken blade

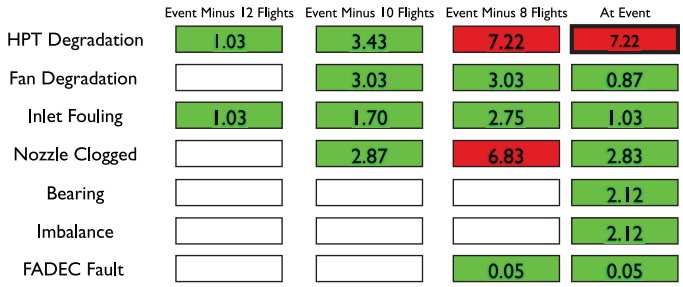

Fig. 11. Trace of data from case study 2 with the reasoner using the augmented reference model.

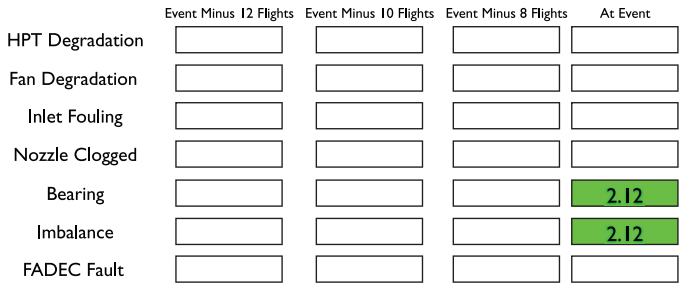

Fig. 12. Trace of data from case study 2 with the reasoner using the original reference model.

before the engine is put back into operation. In comparison, the original reference model reported a bearing failure mode just before the flight where the adverse event occurred. In this case, the maintenance crew's actions would have been triggered by an incorrect fault hypothesis, therefore, a good chance that the HPT blades would not be checked before the adverse event occurred.

While the HPT degradation was hypothesized 12 flights before the adverse event by the updated reference model, its likelihood increased through subsequent flights and eight flights prior to the event it was listed as the most likely candidate. However, another failure mode, *fuel nozzle clogging* was also a high likelihood candidate. Our domain experts surmised that the reasoner would have generated a maintenance alert about eight flights before the adverse event, although the fault isolation did not generate a unique result. However, a borescope inspection triggered by one of the two highly ranked fault hypotheses associated with the alert would clearly identify the broken turbine blade.

4) Robustness Experiment: Comparing Fuel HMA and Broken Turbine Blade Faults: Using the updated reference models for the two faults, we run a robustness experiment comparing the performance of one fault against the other. For the TAN classifier generated using the Fuel HMA data, an experiment run with the Turbine Bucket Blade (TBB) fault data produces a no-fault hypothesis with 95.93% accuracy and a false positive rate of 4.10%. The TBB TAN achieves 85% accuracy with a false positive rate of 15% when the experiment is run with the faulty Fuel HMA fault data. This shows that the Fuel HMA TAN is better tuned to detecting the Fuel HMA fault without increasing the false alarm rate, but the TBB TAN is less precise. The experts conclusion after these experiments was that additional CIs, such as a vibration detector, was needed to isolate the TBB fault with greater accuracy. This second case study establishes the generality of our approach for different faults in the engine subsystem. It also shows that data-driven robustness analysis helps the experts gain a better understanding of the nature of the failures and the feature sets being used to distinguish between those failures.

C. Case Study 3: Fuel Manifold Leak

The *fuel manifold leak* fault caused an engine shutdown event in flight, leading to an emergency landing. This failure differs from the first two in that the cause is not isolated to a specific subsystem. The fuel manifold includes the fuel lines that supply two of the four engines of the aircraft, therefore, the manifold leak should affect the performance of more than one engine. However, when one looked at the individual engine monitors the manifold leak produced effects that are similar to other engine failures. A direct analysis of the engine CIs would imply faults in one or more engines. However, our classifier analyses helped the experts realize that this fault could not be associated with one of the engine subsystems, and, therefore, required analysis by the system level diagnoser to isolate the true fault.

1) Experiment 1: Our experiment with the system-level Fuel Manifold TAN provided an accuracy value of 90.31% and a false positive rate of 5.4% using tenfold cross validation. A second experiment with the other two data sets as the test set revealed more about the nature of this fault. The Fuel HMA data produced a 77.5% accuracy and 22.5% false positive rate. The broken blade failure scored an accuracy rate of 44.4%. Overall, the accuracy and false positive rates were weaker than case studies 1 and 2, which implied that the Fuel Manifold TAN is not sufficiently discriminatory in isolating the manifold leak fault from other engine failures. On further reflection, our experts realized that this failure could not be reliably isolated at the engine subsystem level.

This case study revealed that the data mining methods are useful not only for finding additional relations and monitors to augment subsystem reference models, but they also provide useful indicators to knowledge engineers and system experts, when the approach being used is not a good fit for the fault being analyzed.

V. Related Work

In the past, a number of model- and data-driven approaches have been developed for aircraft fault diagnoses. However, most model-based methods have focused on smaller subsystems of the aircraft, e.g., the work on aircraft sensor and actuator fault diagnosis using a bank of Kalman filters [25], aircraft avionics diagnosis using Bayesian Belief Networks [26], and diagnosis of the cabin pressure outflow valve actuator in the passenger aircraft using parameter estimation methods [27]. A good review of data-driven methods for diagnostics and prognostics in aircraft and spacecraft systems appears in [28] and [29]. Examples of data-driven methods for diagnosis include support vector machines for detecting valve and pump failures using vibration data [30], neural network methods for predicting remaining useful life of actuator components [31], and decision tree based fault detection and classification of solar photovoltaic cells [32].

The above methods have been primarily developed for offline analyses, and deal with small signal streams as opposed to the

large volumes of flight data that we have analyzed for improving existing diagnoser performance for existing aircraft reasoners. Besides, our focus has been to improve existing monitors, discover new monitors, and update the aircraft reference models in ways that the information can be uploaded and used on aircraft without requiring recertification. In future work, we will extend our classifier approaches to develop semi-supervised methods for anomaly detection in aircraft flight.

Our case studies clearly demonstrate that working with our domain experts to make local changes in existing model structures improved the accuracy of an existing reference model by: (1) *making relevant evidence more sensitive* to specific fault hypotheses; 2) *discovering new relations* between existing monitors and fault hypotheses, and 3) *creating new monitors* by exploiting the dependency between existing monitors to provide stronger support for fault hypotheses. The novelty in our approach comes from: 1) our data curation methods for selecting the relevant data and flight segments from the large flight data set to discover new information for characterizing faults; 2) the easy interpretation of the structures generated by the learning algorithm to make it easier for our domain experts to update existing diagnostic reference models; and 3) the ability to evaluate the resultant improvements to the diagnostic reasoner quantitative metrics.

VI. Conclusion

The data mining method presented in this paper derives Bayesian TAN classifiers from selected segments of aircraft flight data, and with the help of domain experts, augments the existing ADMS reference models to improve overall time to detection and detection accuracy. Experiment 3 in studies 1 and 2 demonstrated that the knowledge engineering processes developed not only improved fault isolation capabilities, but faults could be detected earlier in the flight sequence, thus aiding mechanics in their maintenance tasks, and contributing to overall safety by helping to mitigate the occurrence of adverse events. Case study 3 demonstrated how the classifier performance alerted the knowledge engineers and experts by showing that the fault under consideration did not fit the existing reference model structure. This led the experts to better understand the nature of the fault, i.e., this was a system-level fault, and could not be analyzed within a subsystem diagnoser.

It is important to note that this method is developed with the rarity of failure data in mind. It may be difficult to find enough data to build a robust classifier that works across a number of different single faults. As more data on different faults becomes available, better discretization bounds can be derived from the TAN CPTs, resulting in more precise thresholds for the condition indicators. This leads to more accurate and quicker detection, and less false positive rates in fault isolation.

However, the lack of sufficient data makes the generality of the classifiers hard to test. Our present approach does not provide a mechanism to reliably update the conditional probabilities used by the reasoner to rank the potential fault hypotheses. There is no systematic approach by which the conditional probability tables (CPTs) from the individual TANs can be translated into the conditional probabilities of the reference model. In future, we will study approaches to address these problems. We will also extend our studies to semi-supervised methods for anomaly detection that utilize the entire flight data set to find fault situations not previously discovered by the human experts.

References

[1] C. Spitzer, "Honeywell primus epic aircraft diagnostic and maintenance system," *Digital Avionics Handbook*, no. 2, pp. 22–23, 2007.

[2] D. Poole, "Explanation and prediction: An architecture for default and abductive reasoning," *Comput. Intell.*, vol. 5, no. 2, pp. 97–110, 1989.

[3] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[4] M. J. Aslin and G. J. Patton, "Central maintenance computer system and fault data handling method," U.S. Patent US 4943919, 07 24, 1990.

[5] "Vehicle integrated prognostic reasoner," NASA Contractor Report Honeywell, 2010, vol. NNL09AD44T, to be published.

[6] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1988.

[7] S. Budalakoti, S. Budalakoti, A. Srivastava, M. Otey, and M. Otey, "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety," *IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev.*, vol. 39, no. 1, pp. 101–113, Jan. 2009.

[8] S. Das, B. Matthews, and R. Lawrence, "Fleet level anomaly detection of aviation safety data," in *Proc. IEEE Conf. Prognostics Health Manage. (PHM)*, Jun. 2011, pp. 1–10.

[9] J. M. Schumann, T. Mbaya, and O. J. Mengshoel, "Bayesian software health management for aircraft guidance, navigation, and control," presented at the Proc. Annu. Conf. Prognostics Health Manage. Soc. (PHM-11), Montreal, QC, Canada, 2011.

[10] B. Efron, *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Cambridge, U.K.: Cambridge Univ. Press, 2010, vol. 1.

[11] I. Kononenko, "Inductive and bayesian learning in medical diagnosis," *Int. J. Appl. Artif. Intell.*, vol. 7, no. 4, pp. 317–337, 1993.

[12] O. J. Mengshoel, M. Chavira, K. Cascio, S. Poll, A. Darwiche, and S. Uckun, "Probabilistic model-based diagnosis: An electrical power system case study," *IEEE Trans. Syst., Man, Cyberne., Part A: Syst. Humans*, vol. 40, no. 5, pp. 874–885, Sep. 2010.

[13] E. Kiciman, D. Maltz, and J. C. Platt, "Fast variational inference for large-scale internet diagnosis," in *Proc. Advances in Neural Inf. Process. Syst.*, 2008, pp. 1169–1176.

[14] D. Sharp, A. Bell, J. Gold, K. Gibbar, D. Gvillo, V. Knight, K. Murphy, W. Roll, R. Sampigethaya, V. Santhanam, and S. Weismuller, "Challenges and solutions for embedded and networked aerospace software systems," *Proc. IEEE*, vol. 98, no. 4, pp. 621–634, Apr. 2010.

[15] T. Felke, "Application of model-based diagnostic technology on the boeing 777 airplane," in *Proc. 13th AIAA/IEEE Digital Avionics Syst. Conf., DASC*, 1994, pp. 1–5.

[16] M. A. Shwe, B. Middleton, D. Heckerman, M. Henrion, E. Horvitz, H. Lehmann, and G. Cooper, "Probabilistic diagnosis using a reformulation of the internist-1/qmr knowledge base," *Methods Inform. Med.*, vol. 30, no. 4, pp. 241–255, 1991.

[17] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian fault detection and diagnosis in dynamic systems," in *Proc. AAAI Conf.*, 2000, pp. 531–537.

[18] I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Comprehensive diagnosis of continuous systems using dynamic Bayes nets," in *Proc. 19th Int. Workshop Principles Diagnosis*, 2008, pp. 151–158.

[19] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, pp. 131–163, 1997.

[20] I. Cohen, M. Goldszmidt, T. Kelly, J. Symons, and J. S. Chase, "Correlating instrumentation data to system states: A building block for automated diagnosis and control," in *Proc. 6th Conf. Symp. Oper. Syst. Design Implementation*, Berkeley, CA, USA, 2004, vol. 6, pp. 16–16.

[21] J. Kruskal and B. Joseph, "On the shortest spanning subtree of a graph and the traveling salesman problem," in *Proc. Amer. Math. Soc.*, 1956, vol. 7, pp. 48–50, 1.

[22] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, pp. 81–106, 1986.

[23] M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[24] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Education, 2010.

[25] T. Kobayashi and D. L. Simon, "Aircraft engine sensor/actuator/component fault diagnosis using a bank of Kalman filters" Glenn Research Center, Cleveland, OH, USA, NASA Tech. Rep. NASA/CR—2003-212298, Mar. 2003.
[26] C. S. Byington, P. W. Kalgren, R. Johns, and R. J. Beers, "Embedded diagnostic/prognostic reasoning and information continuity for improved avionics maintenance," in *Proc. IEEE Syst. Readiness Technol. Conf., AUTOTESTCON*, 2003, pp. 320–329.
[27] R. Isermann, "Model-based fault-detection and diagnosis-status and applications," *Annu. Rev. Control*, vol. 29, no. 1, pp. 71–85, 2005.
[28] M. Schwabacher, "A survey of data-driven prognostics," in *Proc. AIAA Infotech Aerosp. Conf.*, 2005, pp. 1–5.
[29] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2226–2238, Nov. 2013.
[30] F. He and W. Shi, "WPT-SVMs based approach for fault detection of valves in reciprocating pumps," in *Proc. Amer. Control Conf.*, 2002, vol. 6, pp. 4566–4570.
[31] C. S. Byington, M. Watson, and D. Edwards, "Data-driven neural network methodology to remaining life predictions for aircraft actuator components," in *Proc. IEEE Aerosp. Conf.*, 2004, vol. 6, pp. 3581–3589.
[32] Y. Zhao, L. Yang, B. Lehman, J.-F. De Palma, J. Mosesian, and R. Lyons, "Decision tree-based fault detection and classification in solar photovoltaic arrays," in *Proc. 27th Annu. IEEE Appl. Power Electron. Conf. Expos. (APEC)*, 2012, pp. 93–99.

Daniel L. C. Mack received the B.S. degree in computer science from the University of Notre Dame, Notre Dame, IN, USA, in 2006, and was a teaching assistant while completing the M.S. degree in computer science from Columbia University, New York, NY, USA, in 2008, with a concentration in machine learning. He received the Ph.D. degree in computer science from Vanderbilt University, Nashville, TN, USA, in 2013, where his dissertation focused on machine learning and anomaly detection. While pursuing his doctorate, he worked as a Research Assistant at the Institute for Software Integrated Systems where he and his research group won the NASA Associate Administrator Award for Technology and Innovation for work combining machine learning with fault diagnosis.

He is the Director of Baseball Analytics/Research Science, for the Kansas City Royals. He works closely with the entire Baseball Analytics staff to assist with quantitative research and development of analytics in support of all areas of baseball operations.

Gautam Biswas (S'78–M'82–SM'91–F'14) received the BTech. degree in electrical engineering from the Indian Institute of Technology (IIT), Mumbai, India, in 1977, and the M.S. and Ph.D. degrees in computer science from Michigan State University, East Lansing, MI, USA, in 1980 and 1984, respectively.

He is a Professor of Computer Science, Computer Engineering, and Engineering Management with the Department of Electrical Engineering and Computer Science and a Senior Research Scientist with the Institute for Software Integrated Systems (ISIS), Vanderbilt University, Nashville, TN, USA. He has over 450 refereed publications. He conducts research in intelligent systems with primary interests in hybrid modeling, simulation, and analysis of complex embedded systems, and their applications to diagnosis, prognosis, and fault-adaptive control. More recently, he has been working on data mining algorithms for diagnosis and prognosis, and developing methods that combine model-based and data-driven approaches for diagnostic and prognostic reasoning.

Prof. Biswas and his colleagues received the NASA 2011 Aeronautics Research Mission Directorate Technology and Innovation Group Award for Vehicle Level Reasoning System and Data Mining Methods to improve aircraft diagnostic and prognostic systems. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, PROGNOSTICS AND HEALTH MANAGEMENT, and the IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES.

Xenofon D. Koutsoukos (S'96–M'00–SM'07) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 2000.

He is a Professor with the Department of Electrical Engineering and Computer Science and a Senior Research Scientist with the Institute for Software Integrated Systems (ISIS), Vanderbilt University, Nashville, TN, USA. He was a Member of Research Staff at the Xerox Palo Alto Research Center (PARC) (2000–2002), working in the embedded collaborative computing area. He has published numerous journal and conference papers and he is co-inventor of four U.S. patents. His research work is in the area of cyber-physical systems with emphasis on formal methods, distributed algorithms, security and resilience, diagnosis and fault tolerance, and adaptive resource management.

Prof. Koutsoukos was the recipient of the NSF Career Award in 2004, the Excellence in Teaching Award in 2009 from the Vanderbilt University School of Engineering, and the 2011 NASA Aeronautics Research Mission Directorate (ARMD) Associate Administrator (AA) Award in Technology and Innovation.

Dinkar Mylaraswamy received the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1997.

He joined Honeywell Aerospace, Golden Valley, MN, USA, in 1997, after completing his Ph.D. degree. His Ph.D. thesis on blackboard-based architectures was adopted by the Abnormal Situation Management Consortium as the basis of an operator tool for addressing the $16B loss suffered by the petrochemical industry from abnormal situations and equipment malfunctions. He is the Technology Fellow for condition-based maintenance within Honeywell's Advanced Technology Organization. His area of expertise is fault diagnosis, process monitoring, modeling and control. In his current role, he is responsible for identifying and maturing strategic health management technologies that cut across multiple products and services, providing inputs for strategic technology investments, and mentoring. He spent the first six years in Honeywell developing and deploying an Early Abnormal Event Detection application at six refinery sites in North America. On the Aerospace side, he was the technical lead for Honeywell's Predictive Trend Monitoring Program, a web-based application for monitoring aircraft engines. He continues to serve as the technical lead on various health management programs – within Honeywell as well the U.S. Army, NASA, UKMOD, and Navair – to support the Aero services, engines, mechanical components and avionics business within Honeywell. As the Technology Fellow, he routinely works with academic institutes and small businesses, seeking cutting-edge technologies to support the condition-based service business within Honeywell. He has authored over 30 papers and holds 23 patents in the area of fault diagnosis and its applications.