Factoring a priori classifier performance into decision fusion

Kai Goebel, Shreesh P. Mysore

Kai Goebel, Shreesh P. Mysore, "Factoring a priori classifier performance into decision fusion," Proc. SPIE 4731, Sensor Fusion: Architectures, Algorithms, and Applications VI, (6 March 2002); doi: 10.1117/12.458386

Event: AeroSense 2002, 2002, Orlando, FL, United States

Factoring in a priori classifier performance into decision fusion

Kai Goebel[*a], Shreesh P. Mysore[**b]
[a]GE Corporate Research & Development; [b]Dept. of Control and Dynamical Systems, Caltech.

ABSTRACT

In this paper we present methods to enhance the classification rate in decision fusion with partially redundant information by manipulating the input to the fusion scheme using a priori performance information. Intuitively, it seems to make sense to trust a more reliable tool more than a less reliable one without discounting the less reliable one completely. For a multi-class classifier, the reliability per class must be considered. In addition, complete ignorance for any given class must also be factored into the fusion process to ensure that all faults are equally well represented. However, overly trusting the best classifier will not permit the fusion tool to achieve results that rate beyond the best classifiers performance. We assume that the performance of classifiers to be fused is known, and show how to take advantage of this information. In particular, we glean pertinent performance information from the classifier confusion matrices and their cousin, the relevance matrix. We further demonstrate how to integrate a priori performance information within an hierarchical fusion architecture. We investigate several schemes for these operations and discuss the advantages and disadvantages of each. We then apply the concepts introduced to the diagnostic realm where we aggregate the output of several different diagnostic tools. We present results motivated from diagnosing on-board faults in aircraft engines

Keywords: Classification; Diagnostics; Information Fusion; Decision Fusion; A priori Information; Confusion Matrix

1. INTRODUCTION

To satisfy the need for high classification performance or a need for increased class coverage, different classification tools are sometimes developed either in parallel or sequentially. Often times, it is difficult or impossible for any one given classifier to deal with all the classes of interest at the desired level of accuracy. This motivates the parallel use of several classifiers. In addition, other classifiers are developed to be able to overcome expansion limitations of existing tools and the lack of adaptability to system changes and environmental changes. While the resulting patchwork approach might achieve optimization at a particular local level, it might also cause new problems due to the inevitable introduction of conflicting information. However, there is a potential benefit to be gained by taking a system-level view. This system-level scheme gathers and combines the results of different classification tools to maximize the advantages of each one while minimizing the disadvantages. Such a fusion scheme holds the promise to deliver a result that is better than the best result possible by any one tool employed. In part this can be accomplished because redundant information is available, which when combined correctly improves the estimate of the better tool and compensates for the shortcomings of the less capable tool. In addition, there is a gamut of secondary information that can potentially be folded into the fusion scheme to boost the overall classification performance. However, there is no substitute for a good classification tool and, ordinarily, multiple, marginal-performance tools do not necessarily combine to produce an improved result and in fact may worsen the outcome[1].

There are several traditional approaches that deal with aggregation of information. Weighted averaging attempts to compensate for poor tool decisions by smoothing out the inferior performers. However, the trade off is that good information succumbs to the bad information in the process and a particular tool's superior performance for some classes is sacrificed. In voting schemes, the tools decide jointly on the final output through a majority vote but encounter similar problems as the weighted averaging because several poor performers can outvote a good tool. Bagging and boosting[2] try to address some of those problems. Pattern recognition approaches such as neural networks can be employed to recognize patterns of behavior that may lead to correct decisions. However, if the input to the tools is not available to the

[*] goebelk@crd.ge.com; phone 1 518 387-4194; fax 1 518 387-6104; http://best.me.berkeley.edu/~goebel/kai.html; GE Corporate Research & Development, K1-5C4A, 1 Research Circle, K1-5C4A, Niskayuna, NY 12309, USA;
[**] shreesh@caltech.edu; phone 1 626 390 2849; http://www.its.caltech.edu/~shreesh; Dept. of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125

fusion tool and the output pattern looks similar for different input scenarios, the neural network fusion may not perform satisfactorily. Dempster-Shafer reasoning is widely used for fusion tasks where several information sources are aggregated using Dempster's rule of combination. However, it is imperative to properly fix the meaning of the underlying belief functions because the suitability of the rule depends on the correct context[3]. Model-based approaches consist of a sequence of steps for validation and conflict resolution, among others. The method shown by Nelson and Mason[4] uses multiple models of known (or suspected) behavioral patterns to establish degrees of compatibility between data and hypothesis. It enforces preferences over the set of candidates (by removing candidates that violate these preferences), and iterates through a cycle of merging and deleting within a set of associated hypotheses for that conflict. Sequential and parallel multi-layered configurations[5] employ a number of diagnostic tools in a sequential and parallel fashion for the refinement of the initial class found utilizing a priori probabilistic performance information about the tools which is used to calculate an error probability. The individual classifiers have the current input pattern as well as the class index of the preceding layer as input variables. A fuzzy fusion scheme described in Loskiewicz-Buczak and Uhrig[6] utilizes the generalized mean and an α-cut. The fusion scheme fuses the first two sensors, defines the confidence of the fused decision, and then continues to fuse additional sensors. If the confidence drops, the step is reversed. Finally, an α-cut (depending on the particular class) determines the exact class. Rao[7] discusses methods that provide performance guarantees based on finite samples from a statistical perspective. These approaches are constrained by performance bounds which could be improved by suitably incorporating application specific details.

We address the overall fusion problem using a multi-layered solution approach that focuses on the incorporation of a priori and external information in addition to the information provided by the individual classifiers. The approach specifically avoids statistics based approaches. Rather, it is a compilation of heuristics gleaned from expert reasoning. This method presented breaks the problem down into the three major components pre-processing, analysis, and post-processing. Each component contains several sub-modules.

2. INFORMATION USED IN FUSION SCHEMES

2.1 A priori Information

The fusion tool makes use of the output coming from the classifiers, non-classification information sources, and a priori tool performance information. The latter corresponds to information that is attainable through experiments or simulations.

	$\hat{C}_0$	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_3$	$\hat{C}_4$	$\hat{C}_5$	$\hat{C}_6$
C_0	0.833	0.023	0.039	0.035	0.035	0.013	0.023
C_1	0.258	0.696	0.019	0.012	0.005	0.005	0.005
C_2	0.313	0.011	0.582	0.029	0.027	0.014	0.024
C_3	0.325	0.010	0.029	0.573	0.052	0.007	0.004
C_4	0.382	0.007	0.027	0.041	0.496	0.007	0.041
C_5	0.094	0.001	0.013	0.005	0.012	0.848	0.028
C_6	0.234	0.007	0.032	0.004	0.058	0.026	0.640

Table 1: Confusion Matrix used as Input for both Design and IFM run-time version

The confusion matrix of the classification tools is the primary source of a priori information for the information fusion scheme described herein. The confusion matrix is a performance measure for the individual classification tools. It lists the observed classes versus the estimated classes. Because all classes are enumerated, it is possible to obtain information not only about the correctly classified states, but also about the false positives (FP), false negatives (FN), and false classified (FC) states. In our representation of the confusion matrix, the rows list the actual classes, the columns list the estimated classes. Note that the class C_0 represents the normal ("null") condition. The diagonal entries of the confusion matrix represent the correctly classified cases. The first row – except the first entry – contains the FP. The first column – except the first entry – contains the FN. The off-diagonal elements – except the FP and FN – are the FC. Table 1 shows the normalized confusion matrix for a classification tool where the result was divided by the number of experiments for each class. The classes are denoted as C_n where n={0, … , 6}.

2.2 Fusion Input

Primary input to the information fusion is the output of the classifiers. The information fusion tool is built on the premise that it can utilize information that led to the classification. In other words, it will not only consider the final fault assignment but also the underlying relevant fault strength. Depending on the diagnostic tool employed this can be a distance measure (for example for a k-means classifier), probability (for example for a Bayesian Belief Net), weight (for example for a neural net), membership (for example for a fuzzy knn), etc. This individual assignment criterion is then scaled between zero and one using an appropriate classifier-specific non-linear function. Therefore, the final output of a classifier that recognizes n classes is a column of n (corresponding) fault strengths. Sorting these strengths in descending order gives the explicit fault assignment. The implicit interpretation is that a fault strength closer to one means that the fault is increasingly more likely while a strength less than 0.5 is increasingly not likely. Thereby we avoid the step of needing a parametric model for fusing heterogeneous data[8] and instead impose this task on the designer of the diagnostic tools who has to provide the mapping from diagnostic output to confidence level.

Other system information not diagnostic in nature (that can be used to support a diagnostic opinion) is also provided as evidential input for the information fusion tool. This is information that would not in itself give rise to an action but helps the diagnostician in understanding and confirming a diagnostic opinion.

3. FUSION ARCHITECTURES

Generally, our fusion tool is divided into three components: 1.) pre-processing, 2.) analysis or core fuser, and 3.) post-processing. We have designed the architecture in such a manner that a maximum amount of external information can be integrated. In addition, we attempted to keep the design modular to allow for later addition of domain-specific modules. Each component consists of several modules that are designed to improve the fusion task at hand. We have explored different schemes that can deal with the fusion scheme. Data are first conditioned in the data pre-processing component.

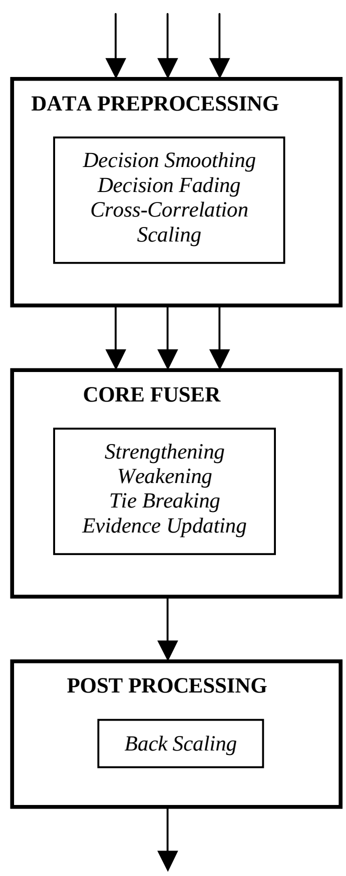

Figure 1: Fusion Components of scheme 1

This includes changes to the classifier output through smoothing, outlier eliminations, capturing temporal effects, and integrating a priori classifier performance information. The outputs of the pre-processing component are modified class estimates. Next, the modules of the hierarchical core fuser aggregate the modified inputs. Finally, the results are polished – where necessary – in the post-processing component to allow for better user interpretation and to account for unequal fault representation. Elsewhere[9,10,11,12,13], we reported about a hierarchical architecture ("scheme 1") as shown in Figure 1. In this paper, we contrast findings from scheme 1 with a modified version which results in increased performance of the classifier fusion scheme. We will first briefly introduce the components of scheme 1 before addressing the changes made to the new scheme ("scheme 2"). The focus of this paper will be on the changes between the two schemes and in particular to the Relevance Equalization and Inter Tool Fusion of scheme 2.

3.1 Scheme 1

Preprocessing - Averaging

This stage deals with temporal information aggregation. Although plain averaging ensures smoothing of information with time, it can stifle the influence of new information. Hence adaptive averaging is used using an adaptive filter parameter that is adjusted to be low when 'changes' are high and vice-versa[10].

Preprocessing – Fading

This module serves to aggregate conflicting information across tools and simultaneously acts on the temporal information also. For instance, if tool X indicates class A at time t_1 and tool Y indicates class B at time t_2, (t_2>t_1), then we need to account for the fact that B might have occurred in the time interval t_2-t_1. So we need to fade the past information with a fading factor that is a function both of the time elapsed and the confidence in the earlier tool's decision[10].

Cross-Correlation

This module makes use of the preferred misclassification of tools (off-diagonal entries of the confusion matrix) to discount the tool output for each class. The purpose is to factor out cross correlation effects. In this scheme, information about preferred misclassifications is used in a manner that discounts the output of a certain class based on the entries in the association matrix[11].

Scaling

The inequity in the representation of faults by tools is addressed here by boosting the diagonal entries of the confusion matrix. In this process, the module uses a 'relevance matrix' [$r_{t,i}$], where $r_{t,i}$ is 1 if tool t is built to recognize fault class i and 0 otherwise. Once this is done, the diagonal entries are used to scale the tool outputs so that more reliable tools are 'trusted' more[12].

Strengthening

If tools agree on a certain class, then this module strengthens the output for that class (by a simple addition operation) [12].

Weakening

This module performs conflict resolution by discounting the entries of the classes in conflict[12].

Tie-breaking

Fault criticality and fault frequency information are used to break ties.

Evidence updating

Any evidentiary information available is used to modify the fused output. We note that evidence plays only a supporting role and is used only to reinforce a decision, not to weaken it. Evidence information is available in the form of the evidence vector and it is multiplied by the evidence matrix (a binary matrix which captures the relevance of an evidence item to a fault class).

Back scaling

This is the final module and it converts the internally coherent information to a form that is externally interpretable. A [0,1] normalization and a dilation operation also constitute this stage.

3.2 Scheme 2

Scheme 2 is divided into pre-processing and core fusion (Figure 2). The post-processing was folded into the modules of the core fusion. Decision Averaging and Decision Fading functions as well as evidence updating was largely retained. Changes were made to Cross-Correlation, Scaling, Strengthening, Weakening, and Back Scaling. These functions were consolidated and newly structured to better accommodate the particular demands on the overall fusion task. In particular, a new definition of relevance was devised which uses some of the concepts of the scaling in scheme 1. Cross-Correlation, Scaling, Tie Breaking concepts, and tasks from the Back Scaling of scheme 1 were moved into a new module "Intra Tool Fusion".

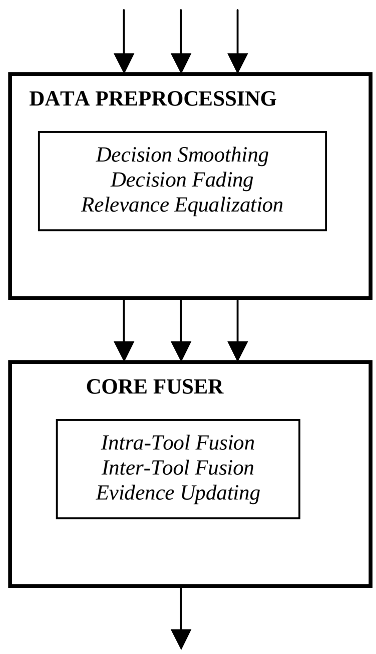

Figure 2: Fusion Components of scheme 2

Strengthening, Weakening, and concepts from Back Scaling were moved into a module "Inter Tool Fusion". The changes are depicted in Figure 3.

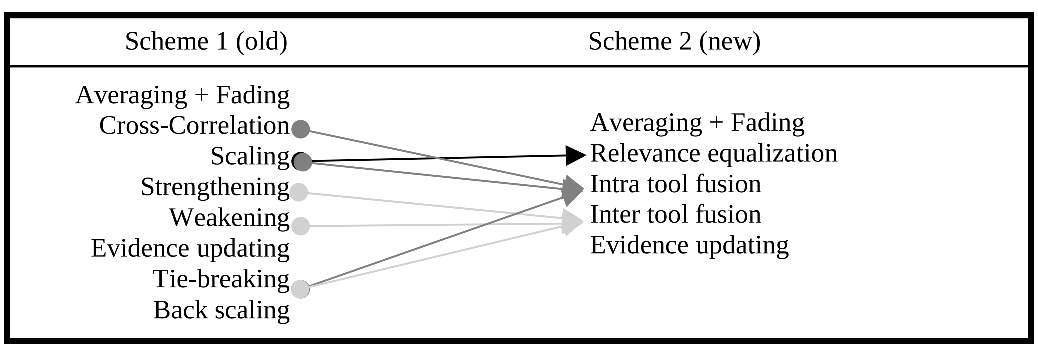

Figure 3: Changes from scheme 1 to scheme 2

Preprocessing – Averaging

The spirit of this module is the same as before, but a few changes have been made to the implementation. Firstly, α is now tool specific. Therefore, we have

$$aggregated\_decision(t,n) = \alpha(t,n) \cdot aggregated\_decision(t,n) + (1 - \alpha(t,n) \cdot new\_decision(n)) \quad (1)$$

Secondly, the value of α is now determined by a simple fuzzy inference sytem. A sample rule is

IF (changes/window) is small THEN α is large

Preprocessing – Fading

This is the same as before.

At this point, some non trivial changes have been made to the existing scheme. In the scheme 1, the scaling module performs two tasks – a.) relevance equalization and b.) modification of the tool outputs. In the scheme 2, relevance is defined in an entirely different way. In addition, the relevance equalization part of scaling is performed in a separate module. Finally, the modification of the tool outputs that was part of scaling in the existing scheme is moved over to a new module, along with the cross-correlation functions performed in the scheme 1. The rationale behind these changes is made clear in the following paragraphs.

Relevance equalization

In the scheme 1, the entries of the relevance matrix are binary valued. The entry ($r_{t,i}$) corresponding to tool i and fault class i is 1 if c_{ii} in the confusion matrix for t is 'high' (presumably greater than .6 or .7). Since the proposed association scheme aims at exploiting the preferred misclassification of a tool to strengthen the confidence in a class, we wish to define relevance matrix entries based not only on the diagonal entries of the confusion matrices, but also the off diagonal entries. The following heuristic scheme is proposed to ensure equal representation of all classes.

We define $c\_\max_{t,i,1}$, $c\_\max_{t,i,2}$, and $c\_\max_{t,i,3}$ to be the top three values (in descending order) in the ith row of the confusion matrix for tool t.

R is called the relevance equalization matrix and is of the form

$$R = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \quad (2)$$

Each r_i (defined below in equation (4)) is the relevance equalization factor for class i and this operation suitably boosts the confusion matrix entries corresponding to those classes that are weakly represented by the tools. Each r_i is computed by using the information about the extent of representation of class i by the tools and is derived from the confusion matrices as follows. Let *r(t,i)* be the "relevance" or "extent of representation" of class i by tool t. Then define

$$\begin{aligned}
r(t,i) &= 1 && \text{if } c_\max_{t,i,1} \geq 0.7 && \text{where } c_\max_{t,i,1} = \max_{j}(c_{ij}) \text{ for tool t; and } \alpha_1 \equiv \arg(c_\max_{t,i,1}) \\
r(t,i) &= \sum_{k=1}^{2} c_\max_{t,i,k} && \text{if } 0.5 < c_\max_{t,i,1} \leq 0.7 && \text{where } c_\max_{t,i,2} = \max_{j \neq \alpha_1}(c_{ij}) \text{ for tool t; and } \alpha_2 \equiv \arg(c_\max_{t,i,2}) \\
r(t,i) &= \sum_{k=1}^{3} c_\max_{t,i,k} && \text{if } 0.3 < c_\max_{t,i,1} \leq 0.5 && \text{where } c_\max_{t,i,3} = \max_{j \neq \alpha_1 \neq \alpha_2}(c_{ij}) \text{ for tool t; and } \alpha_3 \equiv \arg(c_\max_{t,i,3}) \\
r(t,i) &= \sum_{k=1}^{4} c_\max_{t,i,k} && \text{if } c_\max_{t,i,1} \leq 0.3 && \text{where } c_\max_{t,i,4} = \max_{j \neq \alpha_1 \neq \alpha_2 \neq \alpha_3}(c_{ij}) \text{ for tool t}
\end{aligned} \quad (3)$$

The new module consists only scaling the confusion matrix entries for those classes that have insufficient number of tools recognizing them. The quantifier used to decide whether or not a class is well represented is the ratio of the sum of the relevances for a given fault ($\Sigma_t\, r(t,i)$) to the total number of tools ($\Sigma_t\, 1$). This is used in the following manner to modify the diagonal entries of the confusion matrices. We finally define the r_i in equation (2) as

$$r_i = M / R(i) \quad (4)$$

where

$$R(i) = \sum_t r(t,i) \quad (5)$$

$$M = \max_i (R(i)) \tag{6}$$

Now relevance equalization is achieved by pre-multiplying the confusion matrix with R. The resulting matrix is termed the cross-correlation matrix, A. That is,

$$A=R.C \tag{7}$$

This ensures that all the faults are equally well-represented.

This stage does not involve the modification of the tool outputs. It only involves modification of the confusion matrix entries. The modification that was earlier being performed in the "Scaling" module has now been integrated into "Intra-tool fusion".

Intra-tool Fusion

The zero thresholding operation in method 1 can, in some cases, skew the relative confidences in the classes. To circumvent this issue, we introduce a modified different scheme with a claim that it is more intuitive. With the cross-correlation matrix defined as above (equation (7)), and using the notation $A=[a_{ij}]$, the modified outputs are calculated as

$$w^{\sim} = (Av^T)^T. \tag{8}$$

That is,

$$w_1^{\sim} = a_{11}v_1 + a_{12}v_2 + a_{13}v_3 \tag{9}$$

$$w_2^{\sim} = a_{21}v_1 + a_{22}v_2 + a_{23}v_3 \tag{10}$$

$$w_3^{\sim} = a_{31}v_1 + a_{32}v_2 + a_{33}v_3 \tag{11}$$

Consider $w_1^{\sim}$. One of the terms that constitute it is $a_{12}v_2$. v_2 represents the 'confidence' of the tool in the hypothesis that the fault in question belongs to class 2. c_{12} represents the historic 'probability' that the tool misclassifies class 1 as class 2. So the modifying factor $c_{12}v_2$ strengthens the confidence in class 1 based on the probability of misclassification of class 1 as class 2. The above operation is smooth in the sense that the relative confidences in the various classes are continuously modified based on available information.

We note that the diagonal entries in the confusion matrix are already modified in the previous stage, to reflect the relative number of tools for each class.

Inter-tool Fusion

Once fusion has been performed within each tool, the next step is to do this across the tools. To this end, we propose to calculate the fused output corresponding to each class. This serves to strengthen class confidences where appropriate. We also wish to use the information available about criticality of faults to the mission and the frequency of occurrence of faults. We propose the following scheme to cover our requirements.

Given class 'i', the confusion matrices $C(t)=[c_{ij}(t)]$ and the tool outputs $V(t)=[v_i(t)]$, the fused outputs are given by:

$$Fused(i) = \sum_t \frac{c_{ij}(t)\cdot v_i(t)}{1-\frac{prior(i)}{\sum_i prior(i)}} \tag{12}$$

where

$$prior(i) = 0.7\cdot fault\_criticality + 0.3\cdot fault\_frequency \tag{13}$$

with each of *fault_criticality* and *fault_frequency* being real numbers between 0 and 1. Here, we assume that we are working in a potential multiple fault scenario, i.e., it is possible for multiple faults to occur at the same time. Therefore, instead of weakening conflicting outputs, we simply rank order the fused outputs in the descending order. We note that at this point, a normalization operation is performed to ensure that the fused decisions are values between 0 and 1.

Evidence updating

This module remains the same as the existing one except for a minor difference. We allow the entries of the evidence matrix to be continuous values between 0 and 1, thus allowing for uncertainty in the expert's opinion regarding the relevance of a certain item of evidence to a certain class.

4. APPLICATION TO DIAGNOSTIC INFORMATION FUSION

Our work was motivated by the diagnostic task of aircraft engine gas path faults. On a very coarse level, service providers to aircraft engines – both commercial and military – are strongly interested in reducing off-wing time and shop time for engines. There are several benefits in savings for the actual repair cost as well as the increased up-time. In addition, improved system reliability leads to a higher success rate for missions in case of the military engine. To accomplish that goal, it would be desirable to obtain reliable in-flight diagnosis that can perform system state estimation throughout the operation of the engine and deliver the results to a maintenance crew during the landing phase thus avoiding lengthy diagnosis after landing. A realistic goal was determined for this particular case to be fault detection capability of greater than 95%, i.e., less than 5% false negatives (missed faults) in addition to less than 1% false positives (false alarms)[14]. Based on traditional tool performance, it was anticipated that this goal could not be met by any one diagnostic tool alone. However, it was expected that a scheme aggregating the information from several diagnostic tools would be able to achieve the desired performance. To that end, project IMATE (Intelligent Maintenance Advisor for Turbine Engines) set out to tackle the aforementioned issues. Relevant diagnostic and other on-board information sources were designed to produce diagnostic estimates and secondary supporting information covering all faults of interest with maximum overlap of fault coverage and time of diagnosis. Irrespective of those goals, the final diagnostic tool suite exhibited sometimes substantial differences in the fault coverage (not all tools covered all faults of interest), fault diagnosis performance (some tools were better than others at performing the diagnosis), flight operation regimes (some tools operated during certain phases, e.g., climb & cruise vs. takeoff only), and operating rates (e.g., 1 Hz, 30Hz, once per flight). Other design requirements were that the operation had to be performed in real-time and on-board the aircraft during flight.

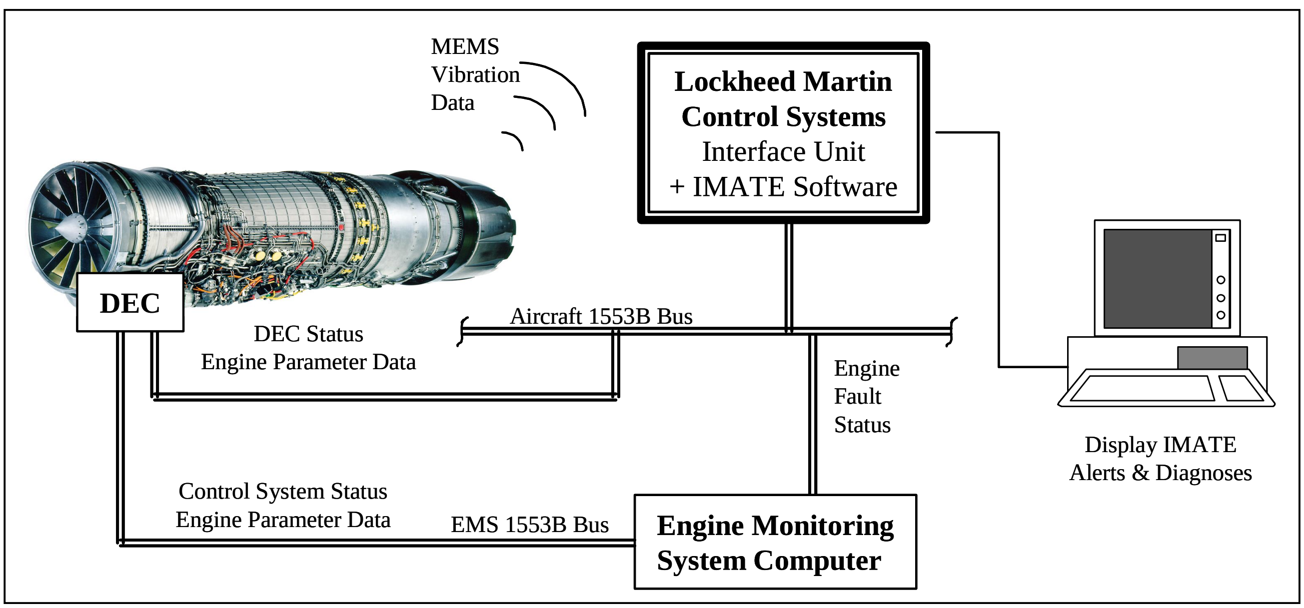

Figure 4: System Configuration for IMATE [14]

Since this work was performed as a dual application for both a commercial and military engine, it necessitated a flexible design that allowed the use of the fusion tool for both applications. In addition, it was the intent to be able to add diagnostic tools at a later time to the fusion scheme thus calling for a modular design. The system configuration for IMATE is displayed in Figure 4.

The main goal was to provide in-flight health monitoring capability for gas path faults. Gas path faults are faults that can occur anywhere within system components subject to the flow of the air and exhaust gases, i.e., the fan, the high pressure compressor, and the high and low pressure turbines. The faults considered were:

- Fan blade damage,
- Compressor blade damage or abnormal compressor operation,
- Partial loss of one or more blades on high pressure turbine,
- Partial loss of one or more blades on low pressure turbine,
- Leakage in excess of the desired bleed level at the Customer Discharge Pressure (CDP) valve,
- Variable Bleed Valve (VBV) doors getting stuck in a particular position,
- Holes burnt in combustor liner,
- Manual errors in installation resulting in small misalignments in vane angles of the Variable Stator Vanes (VSV), and
- Manual errors in installation resulting in small misalignments in vane angles of the Inlet Guide Vanes (IGV).

The combustor leak, VSV, and IGV faults are applicable to the military engine, while the CDP leak and VBV faults are applicable to the commercial engine only; otherwise, the faults are applicable to both engines.

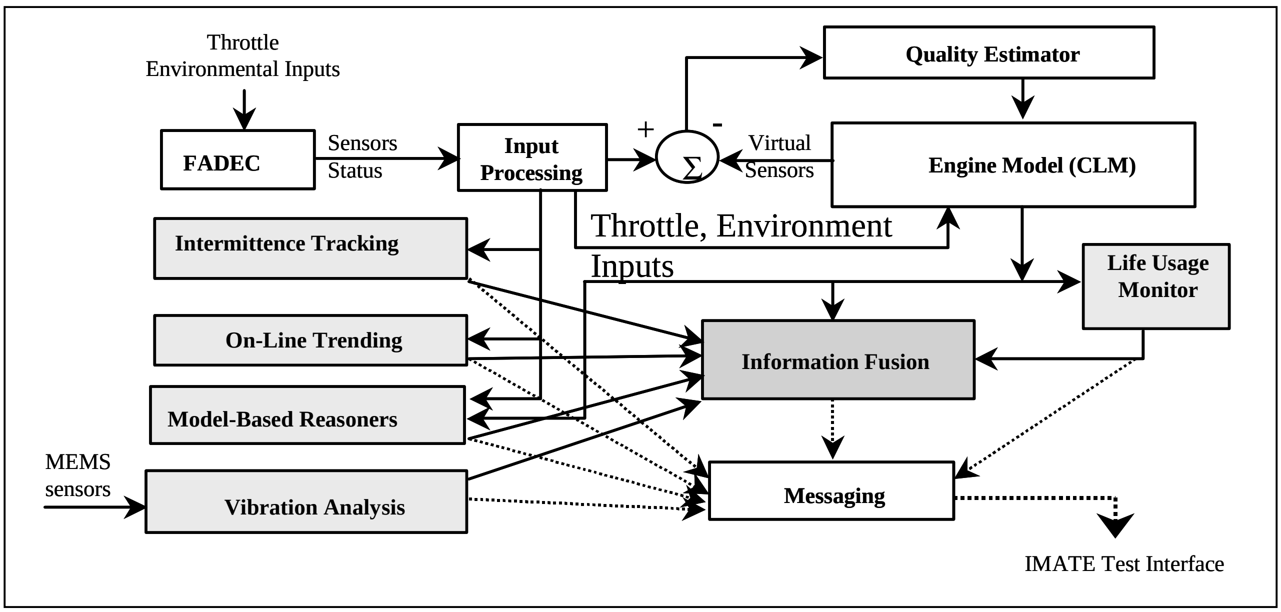

Figure 5: IMATE Functional Architecture[14]

Currently, diagnostic and condition monitoring systems generate information that, while unambiguous in their specific intended application, will be less accurate as more fault coverage is demanded from the tool and less definite as new diagnostic tools are added to either enhance capability or address new faults. This may lead to: 1) ambiguity in troubleshooting, 2) maintenance personnel making uninformed decisions, 3) erroneous component removals, and 4) high operating costs. The fusion effort is one part of an overall project that addresses these problems within IMATE[14]. The overall goal of the information fusion was to combine the relevant diagnostic and other on-board information to produce a fault diagnosis estimate to mitigate each of the aforementioned problems in order to achieve a more accurate and reliable diagnosis than any individual diagnostic tool.

Several diagnostic tools (model based diagnostic tools, neural nets, etc.) as well as non-diagnostic information sources (vibration, fault codes, etc.) were selected for information aggregation. The functional architecture of IMATE is shown in Figure 5. We carried out extensive Monte Carlo simulations and rig tests to validate the tools as well as the fusion module.

3.3 Comparison of the two schemes

We tested the performance of the two schemes via exhaustive Monte Carlo simulation with diagonal heavy confusion matrices. That is, the classifiers are assumed to work fairly well (reliability > 70% for any class). The results are presented with help of a performance indices and a time ratio that capture the difference in performance and computing time for the two schemes, respectively. First, a benchmark algorithm that performed a maximum wins strategy on transformed and normalized input scaled by the a priori reliability of the tool was used for reference. An overall performance index was created by weighing the individual components false positives (FP), false negatives (FN), and false classified (FC) where the weightings of the individual components were driven be the application at hand as shown in equation (1).

$$performance\_index = 0.6 \cdot (1 - FP) + 0.3 \cdot (1 - FN) + 0.1 \cdot (1 - FC) \tag{14}$$

The benchmark performance index was set to zero. An increase in performance of scheme 1 compared to the benchmark algorithm is measured as the fraction of improvement from that baseline to perfect performance, expressed in percent. In contrast, the comparison of scheme 2 to scheme 1 is defined as:

$$performance\_change = \frac{performance_{scheme1} - performance_{scheme2}}{1 - performance_{scheme2}} \cdot 100 \tag{15}$$

In addition, a time ratio was recorded in the comparison of scheme 2 with scheme 1. The time ratio is defined as:

$$time\_ratio = \frac{time_{scheme2}}{time_{scheme1}} \tag{16}$$

The results are tabulated in Table 2. The first seven comparisons (Comp 1 – Comp 7) show that scheme 1 shows significant performance improvement when compared to the benchmark algorithm. In particular, strengthening and weakening show strong improvements with 39% each. The combination of all scheme 1 modules results in a 94.71% performance improvement.

Table 2: Relative performance changes of the different schemes

Schemes Used & Result Type	*Modules*	*Comp 1*	*Comp 2*	*Comp 3*	*Comp 4*	*Comp 5*	*Comp 6*	*Comp 7*	*Comp 8*	*Comp 9*	*Comp 10*
Benchmark Algorithm		X	X	X	X	X	X	X			
Scheme 1	Decision Smoothing/ Decision Fading	X						X			X
	Association		X					X	X	X	X
	Scaling			X				X		X	X
	Strengthening				X			X			X
	Weakening					X		X			X
	Evidence Updating						X	X			X
Scheme 2	Relevance Equalization									X	X
	Intra-Tool Fusion								X	X	X
	Inter Tool Fusion										X
Performance		10%	9%	6%	39%	39%	1%	94.71%	0%	53.96%	53%
Time Ratio		N/E	N/E	N/E	N/E	N/E	N/E	N/E	0.222	0.56	0.74

These seven comparisons are followed by 3 comparisons of scheme 2 to scheme 1 (Comp 8 – Comp 10). These results show that the intra tool fusion does not improve the result but processes the information considerably (about 4.5 times) faster (time_ratio = 0.222). Relevance equalization of scheme 2 improves the relative performance by 53.96% when compared to Association and Scaling of scheme 1 with a favorable time ratio of 0.56 for scheme 2. When Relevance Equalization coupled with Intra-Tool Fusion and Inter-Tool Fusion is compared with the full suite of modules of scheme 1, a 53% improvement is observed with a time ratio of 0.74. The performance increase is even more remarkable since the combination of all scheme 1 modules performed already in mid-90th percentile.

5. SUMMARY AND CONCLUSIONS

We compared two schemes that make heavy use of a priori information encoded in the confusion matrices of classifiers. This information is used to manipulate classifier output by either discounting or rewarding tool output for each class, depending on the module employed. Differences of the two schemes revolve around how the relevance of fault coverage is interpreted and how it is used to manipulate the classifier output. Both schemes lead to a significant improvement of classifier output compared to a benchmark algorithm where scheme 2 holds an edge with a combination of modules that use relevance equalization and intra-tool fusion. Both methods use weight manipulation approaches that lend themselves to amendment with domain specific components and expert heuristics. To ensure that only components are added that truly add to an improvement of the overall performance index, it is imperative to use a design approach that forces implementation of advances only[9]. A hierarchical weight manipulation approach constitutes a potential to surpass the performance boundaries of statistics based approaches. However, there is no guarantee that a weight manipulation approach meets these performance boundaries.

Future work should address (within the pre-processing setting) a discounting penalty for directly redundant output across tools. Currently we only have an implicit assumption that tools must be sufficiently different to be a productive contributor to the fusion module. Other future work should address the aggregation of information from different domains. Because we could influence the design of the classifiers, we postulated the desired output format thus circumventing the need to deal with aggregation of information in different domains. If the fusion task is to be performed for existing tools, there will be a need to perform the tool output transformation. Finally, the particular application may drive the need to address specific challenges which may be encoded as separate layers.

ACKNOWLEDGMENTS

This research was in part supported by DARPA project MDA 972-98-3-0002. The authors also greatfully acknowledge the comments of Malcolm Ashby, Kiyoung Chung, Vivek Badami, Michael Krok, and Hunt Sutherland.

REFERENCES

1. D. Hall and A. Garga, "Pitfalls in Data Fusion (and How to Avoid Them*)", Proceedings of the Second International Conference on Information Fusion (Fusion '99)*, pp. 429-436, 1999.
2. Y. Freund and R. Schapire, "A Short Introduction to Boosting", *J. Japanese Society Artificial Intelligence*, **Vol. 14,** No.5, pp. 771-780. 1999.
3. P. Smets, "What is Dempster-Shafer's model?" *Advances in the Dempster-Shafer Theory of Evidence*, Yager, R., Fedrizzi, M., and Kacprzyk, J., (Eds.), John Wiley & Sons, New York, pp. 5-34, 1994.
4. M. Nelson and K. Mason, "A Model-Based Approach to Information Fusion". *Proc. Information, Decision, and Control*, pp. 395-400, 1999.
5. A. Rahman and M. Fairhurst, "Towards a Theoretical Framework for Multi-Layer Decision Fusion", *Proc. IEE Third Europ. Workshop on Handwriting Analysis and Recognition*, pp. 7/1-7/7, 1998.
6. A. Loskiewicz-Buczak, R. Uhrig, "Decision Fusion by Fuzzy Set Operations", *Proc. third IEEE Conf. Fuzzy Systems*, Vol. 2, pp.1412-1417, 1994.
7. N. S. V. Rao, Finite sample performance guarantees of fusers for function estimators, Information Fusion, **Vol. 1**, no. 1, pp. 35-44, 2000.

8. R. Hathaway, J. Bezdek, and W. Pedrycz, "A Parametric Model for Fusing Heterogeneous Fuzzy Data", *IEEE Trans. on Fuzzy Systems*, **Vol. 4**, No. 3, pp. 270-281, 1996.
9. K. Goebel, "Architecture and Design of a Diagnostic Information Fusion Tool", *AIEDAM: Special Issue on AI in Equipment Service*, pp. April 2001.
10. K. Goebel, "Decision Forgetting and Decision Smoothing for Diagnostic Information Fusion in Systems with Redundant Information", *Proc. SPIE: Sensor Fusion: Architectures, Algorithms, and Applications IV*, Vol. 4051, pp. 438-445, 2000.
11. K. Goebel and S. Mysore, "Taking Advantage of Misclassifications to Boost Classification Rate in Decision Fusion", *Proc. SPIE: Sensor Fusion: Architectures, Algorithms, and Applications V*, pp. 11-20, 2001
12. K. Goebel, "Conflict Resolution using Strengthening and Weakening Operations in Decision Fusion", *Proc. 4th Annual Conf. Information Fusion, Fusion 2001*, pp. ThA1-19 - ThA1-25, 2001.
13. K. Goebel, M. Krok, and H. Sutherland, "Diagnostic Information Fusion: Requirements Flowdown and Interface Issues", Proc. IEEE Aerosense Conference, 11.0303, 2000.
14. M. Ashby and W. Scheuren, Intelligent Maintenance Advisor for Turbine Engines (IMATE), Proceedings of the IEEE Aerospace Conference, 11.0309, 2000.