PROCEEDINGS OF SPIE

Taking advantage of misclassifications to boost classification rate in decision fusion

Kai Goebel, Shreesh P. Mysore

SPIE.

Taking advantage of misclassifications to boost classification rate in decision fusion

Kai Goebel[*a] and Shreesh P. Mysore[**b]

[a]GE Corporate Research & Development, 1 Research Circle, K1-5C4A, Niskayuna, NY 12309
[b]Dept. of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125

ABSTRACT

This paper presents methods to boost the classification rate in decision fusion with partially redundant information. This is accomplished by utilizing the information of known misclassifications of certain classes to systematically modify class output. For example, if it is known beforehand that tool A misclassifies class 1 often as class 2, then it appears to be prudent to integrate that information into the reasoning process if class 1 is indicated by tool B and class 2 is observed by tool A. Particularly this preferred misclassification information is contained in the asymmetric (cross-correlation) entries of the confusion matrix. An operation we call "cross-correlation" is performed where this information is explicitly used to modify class output before the first fused estimate is calculated. We investigate several methods for cross-correlation 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 show how the proposed approach fits into an information fusion architecture and finally present results motivated from diagnosing on-board faults in aircraft engines.

Keywords: Classification; Diagnostics; Information Fusion; Decision Fusion; Correlation; Cross-Correlation

1. INTRODUCTION

To accommodate particular classification requirements, different classification tools are sometimes developed either in parallel or sequentially resulting from a need for increased class coverage or for better classification performance. The stipulation to use several tools arises in part because often times any one tool cannot deal with all classes of interest at the desired level of accuracy. Other reasons include the historic use of particular tools that do not have expansion capabilities, limited resources to develop a comprehensive classification tool addressing all issues, etc. In addition, tools' classification capabilities are impacted differently by environmental changes. While the resulting patchwork approach might achieve optimization at particular local level, it might blur the picture for other existing classifiers due to conflicting information. There is a potential benefits gained by taking a system-level view. Such a system-level scheme gathers and combines the results of different classification tools to maximize the advantages of each one while at the same time minimizing the disadvantages. This 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. 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 always 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 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

---

[*] goebelk@crd.ge.com; http://best.me.berkeley.edu/~goebel/kai.html;
[**] shreesh@caltech.edu; http://www.its.caltech.edu/~shreesh

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 schemes 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.

We address the overall fusion problem using a multi-layered solution approach. This approach breaks the problem down into the three major components pre-processing, analysis, and post-processing. Each component contains several sub-modules. In this paper we propose a method that manipulates the input to the analysis part of the fusion tool as part of the preprocessing component.

2. INFORMATION USED AND FUSION ARCHITECTURE

In the sections below we will discuss the information used for the fusion tool and the architecture of the proposed fusion scheme.

2.1 Information used in fusion scheme

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.

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}.

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

	$\hat{C}_0$	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_3$	$\hat{C}_4$	$\hat{C}_5$	$\hat{C}_6$
C_0	0.7525	0.0010	0.0150	0.0290	0.0320	0.0020	0.0710
C_1	0.0020	0.6733	0.1440	0.0290	0.0860	0.0010	0.0080
C_2	0.0580	0.0010	0.7921	0.0190	0.0190	0.0020	0.0160
C_3	0.0260	0.0010	0.0340	0.4851	0.4220	0.0020	0.0030
C_4	0.1170	0.0010	0.0060	0.1600	0.6337	0.0010	0.0460
C_5	0.0040	0.0010	0.0590	0.0620	0.0270	0.7327	0.0280
C_6	0.1410	0.0010	0.0200	0.0040	0.0150	0.0200	0.7228

2.2 Fusion architecture

Generally, our fusion tool is divided into three components - pre-processing, analysis, and post-processing. Each component consists of several modules that are designed to improve the fusion task at hand. Figure 1 shows the general fusion tool layout. Within the data pre-processing component, the Decision Smoothing and Decision Forgetting modules perform functions to

eliminate outliers, feed more reliable data to the core fuser, and deal with temporal decision discord[7]. The cross-correlation module (shaded) is discussed in this paper. The outputs of the pre-processing component are modified class estimates that reflect partially integrated information about a priori tool performance. Next, the modules of the hierarchical core fuser aggregate the modified inputs. Finally, the results are polished in the post-processing to allow better interpretation for the user. The individual modules and their design are discussed in Goebel[8].
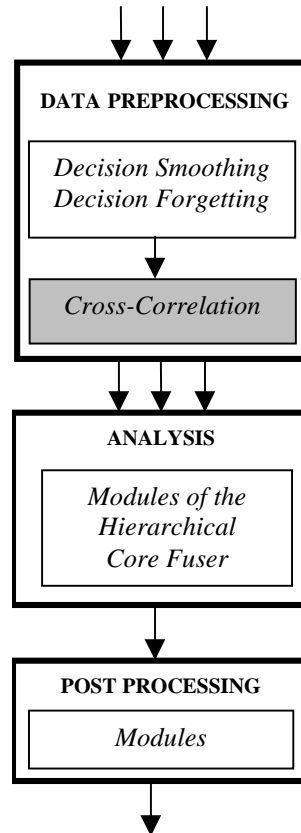

Figure 1: Fusion Components

3. CROSS-CORRELATION METHODOLOGY

We postulate that there is information in a preferred misclassification. That is, if it is known beforehand that tool A misclassifies class 1 often as class 2, then it appears to be prudent to integrate that information into the reasoning process if class 1 is indicated by tool B and class 2 is observed by tool A. This information is contained in the asymmetric (cross-correlation) entries of the confusion matrix. An operation we therefore call "cross-correlation" is performed where this information is explicitly used to modify class output. To perform cross-correlation, we first compute a cross-correlation matrix A for each classifier and then pre-multiply the classifier output with it. Three methods are discussed below for the computation of A.

3.1 Method 1: Factoring out cross-correlations

Here, A is calculated by using the negative normalized confusion matrix with the diagonal elements set to one. Let the confusion matrix C for a certain tool in a 3 class situation be

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{1}$$

Then A is obtained as

$$A = [a_{ij}] = \begin{bmatrix} 1 & -c_{12} & -c_{13} \\ -c_{21} & 1 & -c_{23} \\ -c_{31} & -c_{32} & 1 \end{bmatrix} \tag{2}$$

Let the output vector from the tool in question be $[v_1, v_2, v_3]$. The modified outputs corresponding to the three classes are computed as $w = (Av^T)^T$ yielding:

$$w_1 = 1 \cdot v_1 - c_{12}v_2 - c_{13}v_3 \tag{3}$$

$$w_2 = -c_{21}v_1 + 1 \cdot v_2 - c_{23}v_3 \tag{4}$$

$$w_3 = -c_{31}v_1 - c_{32}v_2 + 1 \cdot v_3 \tag{5}$$

Further, any w_i that is negative as a result of the above calculation is set to zero. To understand the effect of this operation, consider w_1. One of the terms that constitute it is $-c_{12}v_2$. v_2 represents the 'confidence' of the tool in the hypothesis that the class 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 $-c_{12}v_2$ *discounts* the confidence in class 1 based on the negative of the probability of misclassification of class 1 as class 2. The effect of this operation is that side-effects of cross-correlation are factored *out* before the first fused estimate is calculated. After this, the tool outputs are multiplied by the corresponding class reliabilities which are suitably scaled to ensure that classes that are recognized by only a few tools are not at a disadvantage when compared to classes recognized by many tools. The aim of the latter operation is to ensure a more balanced comparison.

For illustrative purposes, an example case with 2 tools and 3 possible classes will be presented. The scaled readings from the tools are (arranged as class 0, class 1, and class 2):

Table 2: Tool outputs

	Class 0	Class 1	Class 2
Tool A	0.6	0.8	0.3
Tool B	0.5	0.4	0.7

The output from the cross-correlation module is then:

Table 3: Output from the cross-correlation module (method 1)

	Class 0	Class 1	Class 2
Tool A	0.582	0.720	0.100
Tool B	0.227	0.195	0.700

This modification needs to be viewed in the context of the confusion matrices, which for this example are shown in Tables 4 (tool A) and 5 (tool B).

Table 4: Confusion Matrix for tool A

Tool A	Est. Class 0	Est. Class 1	Est. Class 2
Class 0	0.95	0.01	0.04
Class 1	0.10	0.80	0.10
Class 2	0.20	0.10	0.70

and for tool B:

Table 5: Confusion Matrix for tool B

Tool B	Est. Class 0	Est. Class 1	Est. Class 2
Class 0	0.60	0.01	0.39
Class 1	0.39	0.60	0.01
Class 2	0.01	0.01	0.98

Using the results from tool B as an example, we can see that the input estimate for class 2 (Table 2) was the highest (0.7), followed by class 0 (0.5), and class 1 (0.4). The confusion matrix for tool B (Table 5) shows that class 2 is correctly classified in 98% of the cases. Class 1 is correctly classified only in 60% of the cases and class 1 is misclassified as class 0 in 39% of the time. As a consequence, the cross-correlation procedure recognizes this "pattern" and downgrades the estimates for class 0 and class 1 proportional to their misclassification (while also taking into account the classification and misclassifications for class 0), resulting in a significant downgrading of the confidence for class 0 and class 1 (see Table 3), but not for class 2. The same procedure is undertaken for tool A.

3.2 Method 2: Factoring in cross-correlations

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 $C=[c_{ij}]$ being the confusion matrix as before, we define the cross-correlation matrix $A =[a_{ij}]$ as

$$A=R*C \tag{6}$$

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

$$R = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \tag{7}$$

Each r_i 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

$$r(t,i) = 1 \qquad\qquad \text{if } c\_\max_{t,i,1} \geq 0.7 \qquad \text{where } c\_\max_{t,i,1} = \max_j(c_{ij}) \text{ for tool t; and } \alpha_1 \equiv \arg(c\_\max_{t,i,1})$$

$$r(t,i) = \sum_{k=1}^{2} c\_\max_{t,i,k} \qquad \text{if } 0.5 < c\_\max_{t,i,1} \leq 0.7 \quad \text{where } c\_\max_{t,i,2} = \max_{j\neq\alpha_1}(c_{ij}) \text{ for tool t; and } \alpha_2 \equiv \arg(c\_\max_{t,i,2})$$

$$r(t,i) = \sum_{k=1}^{3} c\_\max_{t,i,k} \qquad \text{if } 0.3 < c\_\max_{t,i,1} \leq 0.5 \quad \text{where } c\_\max_{t,i,3} = \max_{j\neq\alpha_1\neq\alpha_2}(c_{ij}) \text{ for tool t; and } \alpha_3 \equiv \arg(c\_\max_{t,i,3})$$

$$r(t,i) = \sum_{k=1}^{4} c\_\max_{t,i,k} \qquad \text{if } c\_\max_{t,i,1} \leq 0.3 \qquad \text{where } c\_\max_{t,i,4} = \max_{j\neq\alpha_1\neq\alpha_2\neq\alpha_3}(c_{ij}) \text{ for tool t}$$

$$\tag{8}$$

The representation of class i by all the tools is the computed as $\Sigma_t\, r(t,i)$. The equalization factor r_i for class i is then computed as

$$r_i = \frac{\max_i \sum_t r(t,i)}{\sum_t r(t,i)} \tag{9}$$

By defining r_i in this way and using equation (6), we ensure equal representation of all classes. Once the $r_i s$ are obtained, we get $a_{ij} = c_{ij*}r_i$ and the modified outputs are calculated as $\tilde{w} = (Av^T)^T$.

$$\tilde{w_1} = a_{11}v_1 + a_{12}v_2 + a_{13}v_3 \tag{10}$$

$$\tilde{w_2} = a_{21}v_1 + a_{22}v_2 + a_{23}v_3 \tag{11}$$

$$\tilde{w_3} = a_{31}v_1 + a_{32}v_2 + a_{33}v_3 \tag{12}$$

Consider $\tilde{w_1}$. One of the terms that constitute it is $a_{12}v_2 = r_1c_{12}v_2$. v_2 represents the 'confidence' of the tool in the hypothesis that the class in question belongs to class 2. a_{12} represents the historic 'probability' (suitably modified) that the tool misclassifies class 1 as class 2. So the modifying factor $a_{12}v_2$ *strengthens* the confidence in class 1 based on the probability of misclassification of class 1 as class 2 by factoring *in* cross-correlations. Note that the operation results in an

overall reduction of the class confidences because they are scaled between zero and one. The above operation is smooth in the sense that the relative confidences in the various classes are continuously modified based on available information.

Using the confusion matrices described in the illustrative example above, (with $r = \begin{bmatrix} 1,1,1 \end{bmatrix}$) the output for the two tools is:

Table 6: Output from the cross-correlation module (method 2)

	Class 0	Class 1	Class 2
Tool A	0.59	0.730	0.410
Tool B	0.577	0.442	0.695

It becomes immediately apparent that this method is not as drastic as the first method.

3.3 Method 3: Fuzzy implementation of method 2

With the proposed scheme as the basis, we developed a fuzzy version of the cross-correlation stage. Each of the three components in equations (6 - 8) is termed a modification component. The fuzzy scheme entails performing the modification based on a simple fuzzy rule based system. Given a tool t, and given the class c_i whose output we are currently modifying and given the class c_j whose output is being used to perform the modification, we compute the modification component using a fuzzy inference system (FIS). This FIS is defined by the tool outputs and the confusion matrix entries. The definitions of the fuzzy sets are 'very very low', 'very low', 'low', 'medium' and 'high', 'very high', and 'very very high'. The plots of the membership functions are shown in Figure 3. The seven fuzzy sets are labeled 1 through 7 for convenience of representation. Since the tool outputs and the confusion matrix entries are both bounded between 0 and 1, their product varies as shown in Figure 2. This is used as a guideline to define the rule consequents for the 49 rules (Table 7).

In words, one of the forty nine rules (which corresponds to entry (3,5) in Table 7) is:

IF tool_output is **low** AND confusion_matrix_entry is **high** THEN product is **very low**

We use the product inference method to determine the degree of fulfillment of a rule, the normed weighted sum method for rule response combination and the mean defuzzification method to arrive at a crisp final answer[9].

The pseudo-code for implementing the above computations is given below.

```
FOREACH tool (t)
   FOREACH class (i)
      ModificationComponent=zeros(1,totalClasses)
         FOREACH class (j)
            Apply AssociationFIS to toolOutput(t, j) and confusionEntry(t,i,j);
            Get modificationComponent(j);
         END
      ModifiedDecision=sum(modifcationComponent);
   END
END
```

We note that at this point, a normalization operation is performed to ensure that the modified decisions are values between 0 and 1.

The output of the fuzzy tool for the illustrative example is:

Table 8: Output from the cross-correlation module (method 3)

	Class 0	Class 1	Class 2
Tool A	0.372	0.374	0.254
Tool B	0.313	0.249	0.438

Table 7: Schematic representation of the 49 rules in the fuzzy inference system.

Confusion matrix entry → Tool output ↓	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	1	2	2	2	2	3
4	1	2	2	2	3	3	4
5	1	2	2	3	3	4	5
6	1	2	2	3	4	4	6
7	1	2	3	4	5	6	7
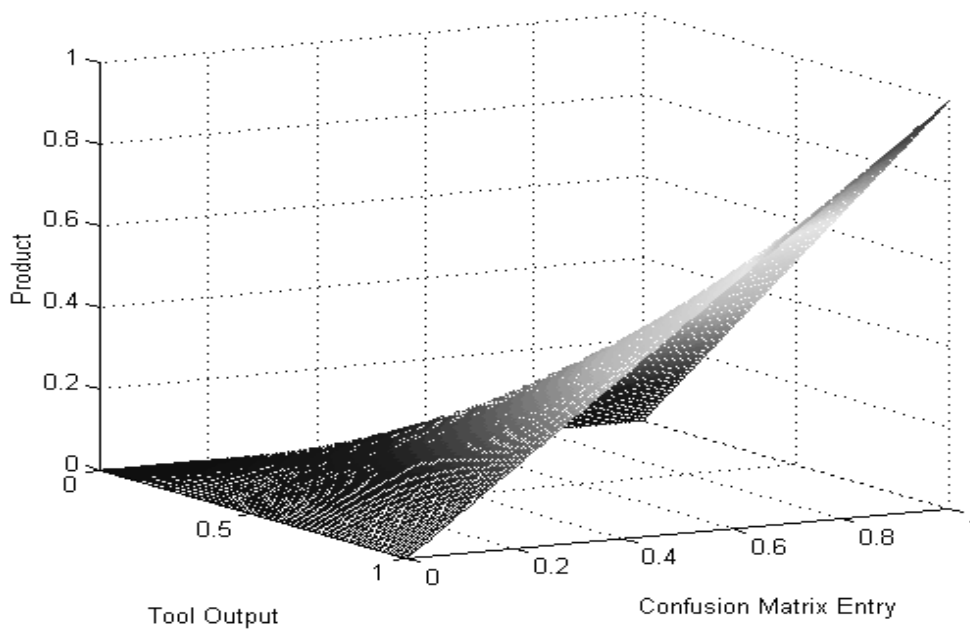
Figure 2: Plot of product of tool output and confusion matrix entry
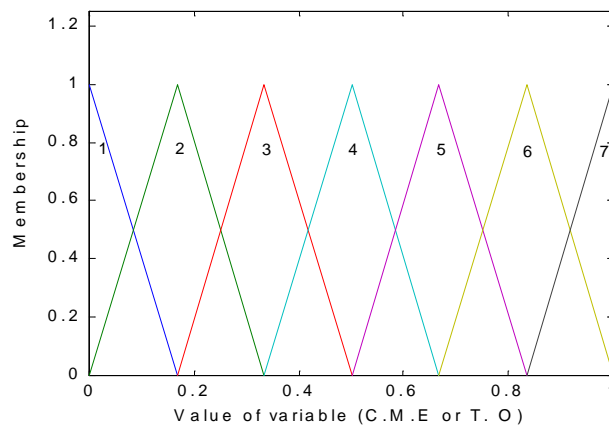
Figure 3: Plots of membership functions of the seven fuzzy sets

3.4 Comparative results

We tested the performance of the three cross-correlation methods via exhaustive Monte Carlo simulation and before applying them to the application as described below. The contribution of the first method to the fusion methodology was a 9% performance improvement over the benchmark algorithm. The second method typically outperformed the first in all categories FP, FN, and FC and posted an improvement of 2-3% over the first method. It also ran about 40% faster than the first method. The third method was able to post improvements over the second by less than 1% (all percentages are given as absolute measures) but at the price of a heavy penalty for computational burden which increased with the chosen implementation by more than an order of magnitude.

4. APPLICATION TO DIAGNOSTIC INFORMATION FUSION

We applied the principles of cross-correlation to the diagnostic information fusion task for a gas turbine engine. 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 & low pressure turbines. The information fusion module (IFM) is intended to demonstrate dual use capability by being designed, and tested on both a commercial and a military engine (CFM56 and F110, respectively)[10]. The faults considered are:

- Fan fault – Fan blade damage, typically occurring due to bird strikes or other Foreign Object Damage (FOD) during takeoff.
- Compressor fault – Compressor blade damage or abnormal operation
- High Pressure Turbine (HPT) fault – Typically a partial loss of one or more blades, most commonly during high power conditions.
- Low Pressure Turbine (LPT) fault – Typically a partial loss of one or more blades, most commonly during high power conditions. LPT blade faults are less frequent than HPT blade faults.
- Customer Discharge Pressure (CDP) fault – Leakage in excess of the desired bleed level commanded by the aircraft and communicated to the Full Authority Digital Electronic Control (FADEC). FADEC does not have control over the CDP valve. The CDP valve takes air off the HP compressor for use in various aircraft functions such as air-conditioning, cabin pressurization, and anti-icing.
- Variable Bleed Valve (VBV) fault – VBV doors not closing according to FADEC issued command, or one or more doors get stuck in a particular position. VBVs are intended to prevent low pressure compressor stalls.
- Combustor Leak (Leak) fault - Holes burnt in liner and hot gas leaks into the bypass duct.
- Variable Stator Vanes (VSV) fault – Manual errors in installation resulting in small misalignments in vane angles. The VSVs control the amount of air that goes through the high pressure compressor.
- Inlet Guide Vane (IGV) fault – Manual errors in installation resulting in small misalignments in vane angles. The IGVs control the amount of air that goes into the fan.
- 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.

Current 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 through the design and test of a condition-based Intelligent Maintenance Advisor for Turbine Engines (IMATE) system[11]. 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. The vision is 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. The overall classification task of the integrated main fusion module with the pre-processing tool (including the cross-correlation module) improved the result by about two orders of magnitude compared to any individual tool alone from an average of 10% misclassifications to 0.1% misclassifications. The effect of the cross-correlation module was estimated to contribute about 9-10% performance improvement to the overall fusion scheme.
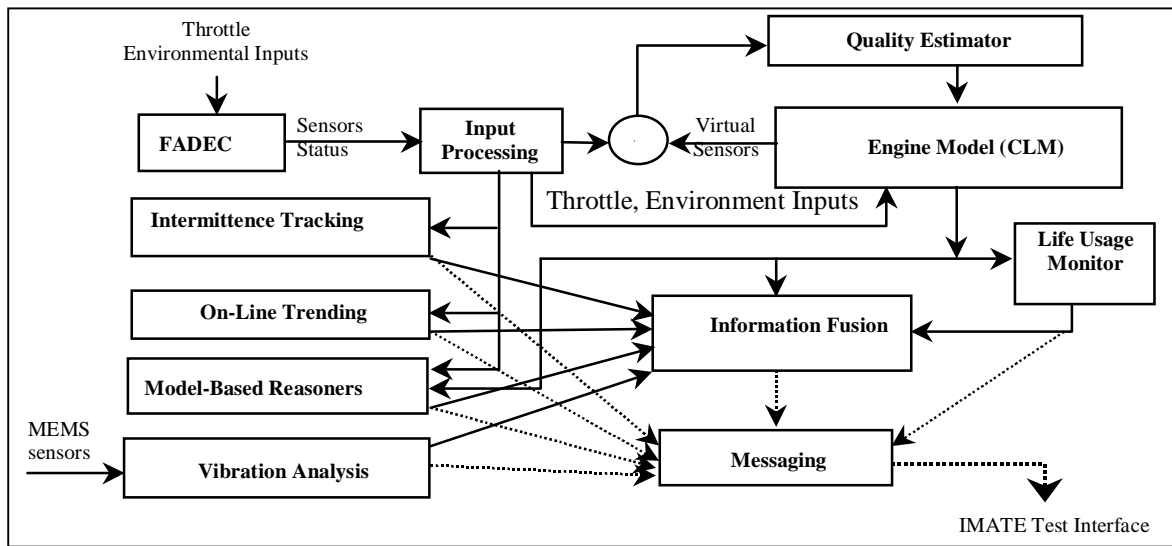
Figure 4: IMATE Functional Architecture[11]

5. SUMMARY AND CONCLUSIONS

We introduced a method that makes use of the preferred misclassification of tools which are encoded as off-diagonal entries of the confusion matrix. The purpose is to use cross-correlation information to boost the fusion performance. This information is either to discount or strengthen the tool output for each class, depending on the method employed. In particular, we show three methods: one that discounts the tool output, and two that strengthen the tool output. One of the strengthening methods is cast in the framework of fuzzy logic. We show that all three methods lead to an improvement of the fusion performance in the categories false positives, false negatives, and false classified classes. The cross-correlation module has been designed in a flexible manner that allows the use within a hierarchical fusion architecture as shown in Goebel[8].

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

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*, **14**(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. 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.

8. K. Goebel, "Architecture and Design of a Diagnostic Information Fusion Tool", to appear in*: AIEDAM: Special Issue on AI in Equipment Service*, April 2001.

9. A. Bárdossy and L. Duckstein, *Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological and Engineering Systems*, CRC Press, Boca Raton. 1995.

10. K. Goebel, M. Krok, and H. Sutherland, "Diagnostic Information Fusion: Requirements Flowdown and Interface Issues", Proc. IEEE Aerosense Conference, 11.0303, 2000.

11. M. Ashby and W. Scheuren, Intelligent Maintenance Advisor for Turbine Engines (IMATE), Proceedings of the IEEE Aerospace Conference, 11.0309, 2000.