Proceedings of the 4th Annual Conference on Information Fusion, Fusion 2001, pp. ThA1-19 - ThA1-25, 2001

Conflict Resolution Using Strengthening and Weakening Operations in Decision Fusion

Kai F. Goebel
Information & Decision Technology Lab
GE Corporate Research & Development
Niskayuna, NY, USA
goebelk@crd.ge.com

Abstract – *We discuss in this paper how to aggregate output from different classifiers and associated uncertainty measures. The fundamental question in the presence of conflicting information is to which degree individual signals or information should be believed. This paper tackles the fusion task by sequentially reinforcing and discounting information using a hierarchical fusion approach. The focus of this paper is on two modules of the hierarchical approach: Decision strengthening and decision weakening. Decisions are reinforced when individual tools agree; decisions are discounted when tools disagree or when several cases are indicated at the same time. Boundary conditions and operators to accomplish this are discussed and an algorithm that shows the desired properties is introduced. We show how the methods work in the context of real-time gas turbine diagnosis. We were able to record substantial performance improvement over the best diagnostic tool employed and over a benchmark algorithm.*

Keywords: Information fusion, decision fusion, diagnosis, classification, diagnostic information fusion, conflict resolution.

1 Introduction

In today's systems, there is an increasing use of redundant sensors and reasoning systems. This redundancy is sometimes direct (i.e., several systems measuring or reasoning about the same information) or indirect (i.e., several systems measuring or reasoning about information having some functional or analytical relation). By successively implementing reasoning systems service providers and manufacturers attempt to address the need for better class coverage and for better classification performance. The stipulation to use several tools arises because often times any one tool cannot deal with all classes of interest at the desired level of accuracy. In part this results from system changes or environmental changes, which have a different impact on the classification capabilities of individual classification tools. In addition, some tools cannot easily be expanded to include new classes. Finally, while some tools are good detecting certain classes they might be virtually worthless for others. In any case, the information is hardly ever the same at some level of granularity. Where information is expressed in different domains, has only partially overlapping signal (or class) coverage, is acquired at different sampling periods, or experiences significant time delays between updates, the complexity of the fusion task rises significantly. The resulting patchwork approach achieves optimization at the local level, but ignores benefits gained by taking a system-level view. Therefore, it seems logical to take the next step and use a system-level scheme that gathers and combines the results of different classification tools to maximize the advantages of each one while at the same time 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 used. In part this can be accomplished because redundant information is available that 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].

2 Decision fusion

We propose a hierarchical architecture that sequentially manipulates the information from the classification tools (initially) and the fused estimate (later) until the most likely candidate class has been refined. In particular, this is a process that increases and decreases the weight given to the classes according to the heuristics implemented in the respective layers of the fusion process. This implies that it is possible for some classes to emanate as winners from one layer, only to be overruled in the next layer.

2.1 Fusion architecture

The architecture displayed in Figure 1 gives an overview over how the fusion is performed conceptually. Generally, the fusion tool is divided into the components pre-processing (layers 1-3), aggregation in the core fuser (layers 4-7), and post-processing (layer 8). Each component consists of several modules that are designed to improve the fusion task at hand. Within the data pre-processing component, the modules "Temporal Aggregation" and "Decision Fading" perform functions to eliminate outliers, feed more reliable data to the core fuser, and deal with temporal decision discord. Cross-correlations are factored into the fusion process in the "Association" layer and a priori tool performance is considered in the "Scaling" layer. Output 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, acting on heuristics that reward and penalize class assignment based on their distribution across classification tools and with classification tools (Strengthening, Weakening). In addition, evidential information is integrated into the reasoning process in the layer "Evidence Updating". Finally, layer "Tie-Braking" merges information about criticality and frequency of occurrence. The results are then polished in the post-processing layer "Back-Scaling" to allow better interpretation for the user. The layers were initially arranged according to expert reasoning. In this paper, we report on the modules "Strengthening" and "Weakening". Some of the other modules are described in Goebel [2, 3] and Goebel and Mysore [4].

2.2 Information used in fusion scheme

Our fusion tool uses a priori information and the output coming from the classification and non-classification information sources. The proposed scheme relies heavily on information about tool performance implying that this information is attainable through experiments or simulations.

2.2.1 Confusion matrix

The confusion matrix of the classification tools is the primary source of a priori information for the information fusion. 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. False positives are false alarm conditions, false negatives are missed faults, and false classifieds are reported alarms that are improperly assigned. In our representation of the confusion matrix, the rows list the actual classes, the columns list the estimated classes. The diagonal entries 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. Given is a set of

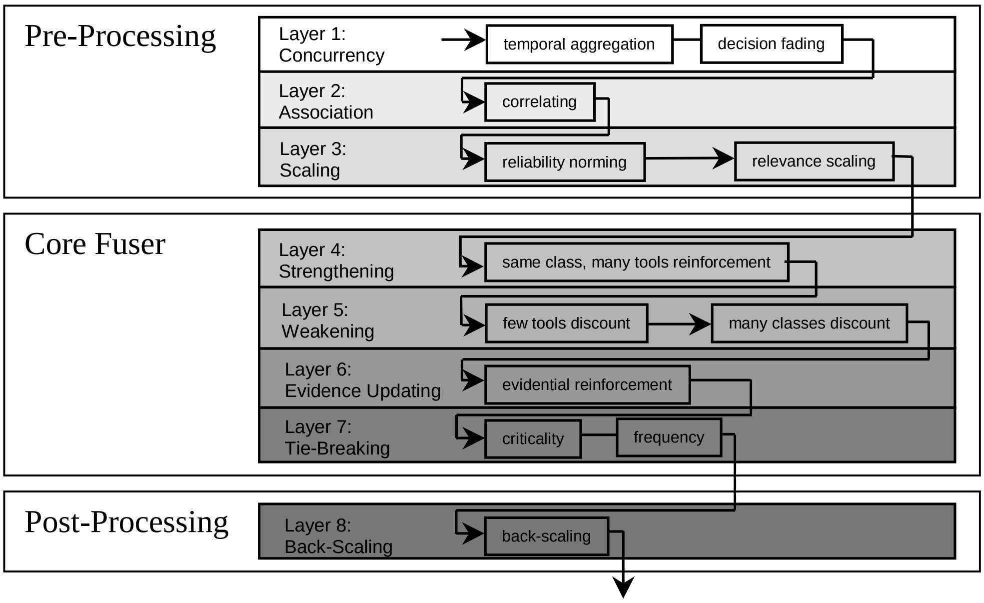

Figure 1: Fusion Architecture

classes C with classes C_n where n = {0, … , 6} and a set of estimated classes $\hat{C}$ with classes $\hat{C}_n$ 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	0.833	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

2.2.2 Relevance matrix

Relevance is an assignment of whether a tool can perform a classification on a per class basis. It can be derived from the confusion matrices but can also be used in the conceptual design phase of the classification tools to determine which tool will or can cover which class. Relevance is indicated by "1", no relevance by "0". Relevance is summarized for all classification tools in a relevance matrix *R*. Table 2 shows an example for 3 tools t_1 through t_3 and 7 classes $C_0 - C_6$.

Table 2: Relevance assignment of tools

	C_0	C_1	C_2	C_3	C_4	C_5	C_6
t_1	1	1	1	1	1	0	0
t_2	1	1	1	1	1	1	1
t_3	1	1	1	1	1	0	1

2.2.3 IFM input

Primary input to the information fusion is the output of the classifiers, i.e., the class vector of each classification tool for the respective classes considered. 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 class assignment but also the underlying relevant class strength. Depending on the classification 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. The implicit interpretation is that a level closer to one means that the class is increasingly more likely while a confidence level less than 0.5 is increasingly not likely. Thereby we avoid the step of needing a parametric model for fusing heterogeneous data [5] and instead impose this task on the designer of the classification tools who has to provide the mapping from classification output to confidence level. Figure 2 shows the classification output to information fusion mapping.

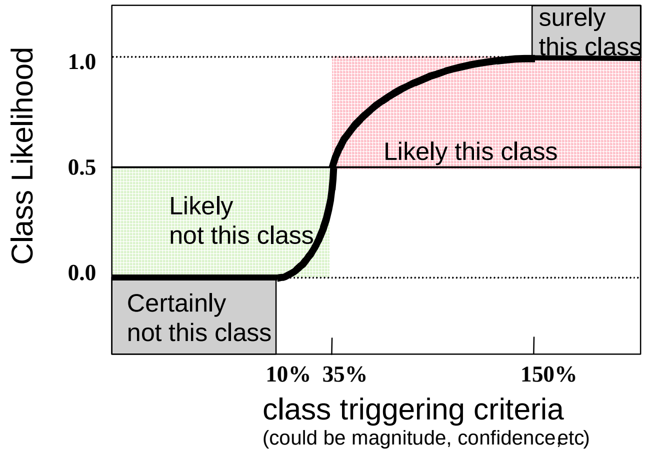

Figure 2: Mapping of classification tool output into 0-1 domain

Other system information that does not stem from a classifier per se (but can be used to support a class opinion) is also provided as input for the information fusion tool. We call this information "evidential information". In a diagnostic context, this is information that would not in itself give rise to an action but helps the diagnostician in understanding and confirming a diagnostic opinion.

2.3 Strengthening and weakening

Fundamentally, the proposed fusion scheme operates by manipulating the classifier outputs and the fused estimate. This manipulation is performed by increasing and decreasing the likelihood of class assignment. In part, this operation starts at the pre-processing level by factoring in cross-correlation effects, temporal effects etc., which may result in substantial fusion performance improvement [3]. In addition, likelihood manipulation in the core-fuser also improves performance. We report here on two layers of the core-fuser: Strengthening and Weakening. Essential to this manipulation is the normalized format of the classifier output. The preferred classifier output is $C_i \in [0,1]$, i.e., $0 \le C_i \le 1$. This is necessary to take advantage of the operators proposed.

2.3.1 Operators for fusion estimate increase and decrease

There are several operators that can be employed in a weight manipulation approach. For operations meant to increase the weight of a class decision, the power operator is useful. The function $s(c)$ is defined by $s : C \to [0,1]$ with the properties

$$s(C) = c^p \text{ where } 0 < p < 1 \qquad (1)$$

where c is the fused estimate of a class. This operation results in an increase of the class estimate provided that the old estimate is kept within the 0-1 boundaries. The fusion tool designer must provide a proper number for the exponent s which is not necessarily a trivial task because of the need to translate the specific conditions surrounding the desired increase. Tuning can be performed using Monte Carlo approaches [3] or using other optimization tools such as genetic algorithms.

Similarly, the power operator can be employed in decreasing operations. The function $w(c)$ is defined by $w: C \rightarrow [0,1]$ with the properties

$$w(C) = c^{p} \text{ where } p > 1 \qquad (2)$$

c is again the fused estimate of a class. This operation results in a decrease of the class estimate provided that the old estimate (the input) is kept within the 0-1 boundaries. Note that the output is also bounded by the interval [0,1];

Another simple operator is the multiplier defined by function $m: C \rightarrow [0,1]$ with the properties

$$m(C) = c \cdot m_d \text{ where } 0 < m_d < 1 \qquad (3)$$

In some sense, this is a more convenient operator because of the linear effect on the outcome which is easier to visualize. However, tuning issues remain mainly the same as mentioned above. One drawback of these operators may be for classification systems that do not provide the output in the required format scaled between 0 and 1. In those cases, methods to scale the output have to be employed first. The operator will also have trouble with binary classifier output. However, we have shown that our scheme can deal with these cases through the manipulations at the preprocessing steps. These steps result in the conversion of binary output into continuous variables as a result of the concurrency, association and scaling modules [2, 4].

2.3.2 Strengthening (layer 4)

The idea of the Strengthening layer is to reinforce specific class assignments. In particular, if class opinions expressed by different tools agree (where each tool is from a set of tools $t \in T$), the result should lead to a more confident assessment of the system state. This is the trivial case where coinciding opinions are rewarded. The rewarding scheme is accomplished by calculating the fused value as the sum of the classifier outputs that are in agreement. Whether classes agree is established on the basis of whether they surpass a common threshold t_s. Recall that the classification is a value expressing a likelihood for that particular class. That is, it is typically a weighted value where the sum over the output from the tool $t_i, i \in [1,...,t]$ for a class j is scaled such that $\sum_{i=1}^{t} x_{i,j} \leq 1$. The threshold t_s needs to be tuned to take that scaling into account. It has to be noted that when the architecture of Figure 1 is used, the classifier output has been further scaled at this point to take a priori information, temporal effects, as well as cross-correlation into account [2, 3]. The resulting set of scaled classifier output C' has the same properties as the original set C', in particular $C'_i \in [0,1]$. The resulting strengthening equation $s: C' \rightarrow [0,1]$ aggregates the scaled classifier outputs that are greater than threshold t_s, $c'_i \geq t_s$, i.e., $c'_i = \begin{cases} c_i \text{ if } c_i \geq t_s \\ 0 \text{ otherwise} \end{cases}$ produces the fused value c_{f_i}.

$$c_{f_i} = \sum_{j=1}^{t} c'_{i_j} \qquad (4)$$

For illustrative purposes, an example case with 2 tools and 3 possible class cases will be presented. The classifier tool outputs were scaled in the pre-processing layer and are listed in Table 3.

Table 3: Scaled input to strengthening layer

	Class 0	Class 1	Class 2
Scaled tool A	0.496	0.380	0.024
Scaled tool B	0.047	0.047	0.646

The output from the strengthening layer is given in Table 4.

Table 4: Output of strengthening layer

	Class 0	Class 1	Class 2
Fused value	0.543	0.427	0.670

From Table 3 we can see that the class estimates disagree between the two tools. While tool A assigns class 0 the highest likelihood, tool B assigns class 2 the highest likelihood. The fused output are then the aggregated values between the two tools. Note again that the outputs of the two classifiers were scaled by their respective a priori reliability. The output of this layer produces a disagreement between the two classes.

2.3.2 Weakening (layer 5)

The weakening layer performs part of the conflict resolution in cases where tools disagree on the classification state of the system. This is performed – in contrast to the strengthening layer – by discounting entries. Discounting was chosen because conflicting information lowers the conviction in a correct outcome. Therefore the fused value for the conflicting classes will be decreased. The implementation of this module checks whether the sum of the tools' outputs for a class are less than the sum of the relevance assignments (i.e., check whether this tool is allowed to make a statement about a class) multiplied by a thresholding scale (here chosen to be t_w=0.5). In essence, this heuristic discounts the fused value if the cumulative class opinion did not have sufficient strength. The effect is a thresholded contrast intensification operation. The weakening equation is $w: C_f \rightarrow [0,1]$:

$$x_{f_j} = \begin{cases} x_{f_j} \cdot d_{\mathrm{w}} & \text{if } \sum_{\mathrm{i=1}}^{\mathrm{t}} \mathrm{c}_{\mathrm{i,j}} < t_w \cdot \sum_{\mathrm{i=1}}^{\mathrm{t}} \mathrm{r}_{\mathrm{i,j}} \\ x_{f_j} & \text{otherwise} \end{cases} \quad (5)$$

where $c_{,j}$ is the classifier value before pre-processing and $r_{i,j}$ is the relevance for classifier i and class j as discussed in section about the relevance matrix.

In addition, for scenarios with a one-class assumption (which is not uncommon in diagnostic environments), it is considered a conflict if the fused estimate contains several classes with high likelihood. The conflict is addressed by penalizing the contributions of conflicting information. In other words, if more than one class is greater than a threshold, then both estimates get discounted. This is expressed by the following equation $w_s : C_f \rightarrow [0,1]$ where the threshold $\mathrm{t_c}$ was set to $\mathrm{t_c} = 0.5$ and the discount $\mathrm{d_m} \in]0,1[$.

$$x_{f_j} = \begin{cases} x_{f_j} \cdot d_{\mathrm{m}} & \text{if } x_{f_j} > \mathrm{t_c} \wedge \underset{\mathrm{k=1:c,\, k\neq j}}{x_{f_k}} > \mathrm{t_c} \\ x_{f_j} & \text{otherwise} \end{cases} \quad (6)$$

Note that in both weakening operations the multiplier operator was used. The power operator could have been used instead.

For the illustrative example, using the output of the strengthening layer as input to the weakening layer, the output of the weakening layer becomes as shown in Table 5:

Table 5: Output from weakening layer

	Class 0	Class 1	Class 2
Fused value	0.543	0.341	0.536

Note that – in contrast to the input – class assignment of class 2 is now lower than class 0. Their relative similarity is modified further downstream in layers Evidential Updating and Tie-Breaker.

3 Application to diagnostic information fusion

We applied the principles of information fusion via weight manipulation – including Strengthening and Weakening – to a diagnostic task of a gas turbine engine. The main goal was to provide in-flight health monitoring capability for gas path faults. 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 was 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 [6]. 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 was to achieve a more accurate and reliable diagnosis than any individual diagnostic tool.

Key system components considered for this health monitoring scheme were the fan, the high pressure compressor, the high & low pressure turbines, and bearings. In addition, wireless micro electro-mechanical systems (MEMS) measured and processed vibration data from the bearings. This sensing technology offered enhanced turbo-machinery vibration diagnostics without an accompanying weight penalty. The information fusion module (IFM) demonstrated dual use capability by being designed, and tested on both a commercial and a military engine (CFM56 and F110, respectively) [7]. The faults considered were:

- 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

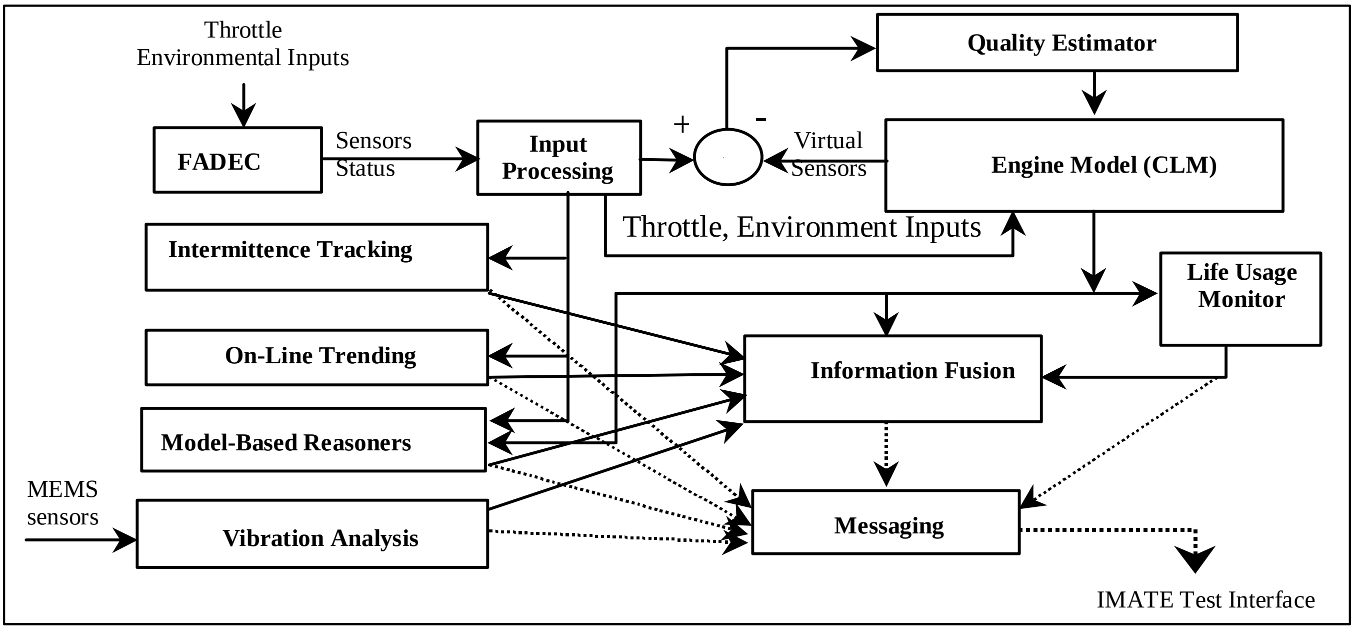

Figure 3: IMATE functional architecture [7]

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.

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 3. We carried out extensive Monte Carlo simulations to validate the classification tools as well as the fusion modules individually and working together as outlined in the architecture of Figure 1. In addition, we performed instrumented rig tests. The overall classification task of the integrated main fusion module with the pre-processing tools, core fuser (of which strengthening and weakening are a part of), and post-processing improved the result by about two orders of magnitude compared to any individual tool alone. In particular, improvements were recorded from an average of 10% misclassifications of the individual classification tools to 0.1% misclassifications of the fusion tool. The effect of the Strengthening and Weakening modules alone was evaluated to contribute 39% performance improvement each to the overall fusion scheme. It has to be noted that the cumulative improvement of both Strengthening and Weakening was smaller because the effects are of course not additive. Details of the training and testing methodology are described in [3].

4 Summary and Conclusions

We introduced a system for manipulation of fused values for decision strengthening and decision weakening. Both are part of the core-fusion module within the hierarchical model introduced. These modules allow partial conflict resolution and improve fusion performance through the concept of information aggregation and discounting based on several criteria where conflicts are encountered. We tackle in particular situations where class estimates differ within and across classification tools and the fusion tool. Operators suitable for the manipulation were introduced and discussed. Strengthening and weakening depend on processed information that integrate a priori performance information and temporal information with the classification tool outputs. Results from a diagnostic task in an aircraft engine context were mentioned. Strengthening and weakening contribute a large portion to the overall performance improvement.

We view these techniques shown here as applicable not only to decision fusion. Rather, they are equally useful for processing on the feature or data level. In addition, if a hierarchical model is not used, the individual layers and also the discounting and increasing operators can be part of stand-alone solutions or other integrated fusion tools.

Acknowledgements

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

References

[1] D. Hall, A. Garga, *Pitfalls in Data Fusion (and How to Avoid Them)*, Proceedings of the Second International Conference on Information Fusion (Fusion '99), 1999.

[2] K. Goebel, *Decision Forgetting and Decision Smoothing for Diagnostic Information Fusion in Systems with Redundant Information*, Proceedings of the SPIE: Sensor Fusion: Architectures, Algorithms, and Applications IV, Vol. 4051, pp. 438-445, 2000.

[3] K. Goebel, *Architecture and Design of a Diagnostic Information Fusion Tool*, accepted for publication in the special edition of AIEDAM: AI in Equipment Service, April 2001.

[4] K. Goebel and S. Mysore, *Taking Advantage of Misclassifications to Boost Classification Rate in Decision Fusion*, Proceedings of the SPIE: Sensor Fusion: Architectures, Algorithms, and Applications V, pp. 11-20, 2001.

[5] 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.

[6] K. Goebel, M. Krok, H. Sutherland, *Diagnostic Information Fusion: Requirements Flowdown and Interface Issues*, Proceedings of the IEEE Aerosense Conference, pp. 11.0303, 2000.

[7] M. Ashby, W. Scheuren, *Intelligent Maintenance Advisor for Turbine Engines (IMATE),* Proceedings of the IEEE Aerospace Conference, 2000.