An evidence-based approach to damage location on an aircraft structure

K. Worden, G. Manson, T. Denœux

PII: S0888-3270(08)00307-5
DOI: doi:10.1016/j.ymssp.2008.11.003
Reference: YMSSP 2337

To appear in: *Mechanical Systems and Signal*

Received date: 8 April 2008
Revised date: 22 October 2008
Accepted date: 1 November 2008

Cite this article as: K. Worden, G. Manson and T. Denœux, An evidence-based approach to damage location on an aircraft structure, *Mechanical Systems and Signal* (2008), doi:10.1016/j.ymssp.2008.11.003

An Evidence-Based Approach to Damage Location on an Aircraft Structure

[1]**K Worden**, [1]**G Manson** & [2]**T Denœux**

[1]Dynamics Research Group, Department of Mechanical Engineering,
University of Sheffield, Mappin Street, Sheffield S1 3JD, UK.

[2]U.M.R. CNRS 6599 Heudiasyc, Université de Technologie de Compiègne,
BP 20529-F-60205 Compiègne Cedex, France.

Abstract

This paper discusses the use of evidence-based classifiers for the identification of damage. In particular, a neural network approach to Dempster-Shafer theory is demonstrated on the damage location problem for an aircraft wing. The results are compared with a probabilistic classifier based on a multi-layer perceptron neural network and shown to give similar results. The question of fusing classifiers is considered and it is shown that a combination of the Dempster-Shafer and MLP classifiers gives a significant improvement over the use of individual classifiers for the aircraft wing data.

1 Introduction

The recent past has seen considerable use of machine learning techniques for Structural Health Monitoring (SHM). The basic idea of the approach is to use data measured from undamaged and damaged structures in order to train a learning machine to assign a condition label to previously unseen data. The simplest problem of SHM is arguably *damage detection*. This is most easily carried out in the machine learning context by using a novelty detector [1]. Novelty detection involves the construction of a model of the normal condition of a system or structure, which can then be used in a hypothesis test on unseen data to establish whether the new data corresponds to normal condition or not. The advantage of the novelty detection approach is that it can be carried out using unsupervised learning, i.e. with only samples of undamaged data. If a more detailed diagnosis of a system is required, e.g. it is necessary to specify the type or location of damage in a structure, this can still be done using machine learning methods. For higher levels of diagnostics, algorithms based on classification or regression are applicable; however, these must be applied in a supervised learning context and examples of data from both the undamaged and damaged conditions can be used [2].

The most popular classifiers for damage location and quantification so far have been those based on Multi-Layer Perceptron (MLP) neural networks [3] (although there is growing popularity for classifiers based on the concepts of statistical learning theory - like support vector machines [2]). Training of MLP networks as classifiers is usually accomplished by using the *1 of M* strategy [3] which implicitly assumes a Bayesian probabilistic basis for the classification. While probability theory is only one (but arguably the most important) of a group of theories which can quantify and propagate uncertainty, other theories of uncertainty, perhaps with the exception of fuzzy set theory, have been largely unexplored in the context of damage identification. The object of this paper is to design a classifier for damage location based on the Dempster-Shafer theory of evidence [4]. To the knowledge of the authors, DS theory was first used in the context of SHM by researchers at El Paso as in [5]. The theory has also been used in the context of machine condition monitoring [6]. The reason for exploring the possibilities of DS theory is that it *extends* probability in a number of ways which are potentially exploitable in an SHM context. For the moment though,

the current paper is looking only to demonstrate that the method is competitive with the probability-based MLP approach on an experimental case study of an aircraft wing. The DS classifier here is also implemented using a neural network structure [7].

The layout of the paper is as follows. Section Two provides a pedagogical introduction to DS theory and places its use in the context of SHM. Section Three briefly describes the neural network implementation of the DS Classifier. Section Four describes the case study discussed in this paper and the results of the analysis are given in Section Five and compared with existing results from a multi-layer perceptron neural network. Section Six discusses the fusion of the two classifiers in both the Bayesian and DS frameworks. The paper ends with some discussion and conclusions.

2 Dempster-Shafer Reasoning

2.1 Theory

Dempster-Shafer Theory is a means of decision-fusion which is formulated in terms of probability-like measures but extends probability theory in a number of important respects. The basic idea, that of *belief* was introduced by Dempster in [8] and extended in Shafer's treatise [4].

The basic model is formulated in similar terms to probability. In the place of the sample space is the *frame of discernment* Θ, which is the set spanning the possible events for observation $A_i \subset \Theta, i = 1, \ldots, N_e$. On the basis of sensor evidence, each event or union of events is assigned a degree of probability mass or *Basic Belief Assignment* (BBA) m such that,

$$0 \leq m(A_i) \leq 1 \qquad \forall A_i \subseteq \Theta \tag{1}$$

$$m(\phi) = 0 \tag{2}$$

$$\sum_{A_i \subseteq \Theta} m(A_i) = 1 \tag{3}$$

where ϕ is the empty set. (Note that normalisation, as in equation (3), is not always required in belief function theory. In particular, it is not the case in the Transferable

Belief Model (TBM), an important variant developed by Smets [9].) The difference between this *evidential* theory and probability theory is that the total probability mass need not be exhausted in the assignments to individual events. There is allowed to be a degree of *uncertainty* or *ignorance*. This is sometimes denoted by a probability mass assignment to the *whole* frame of discernment $m(\Theta)$ or $m(A_1 \cup \ldots \cup A_N)$.

The *belief* in an event B is denoted $\mathrm{Bel}(B)$ and is defined by,

$$\mathrm{Bel}(B) = \sum_{A_i \subseteq B} m(A_i) \tag{4}$$

and this is the total probability which is committed to the support of the proposition that B has occurred.

The *doubt* in the proposition B is denoted by $\mathrm{Dou}(B)$ and is defined by

$$\mathrm{Dou}(B) = \mathrm{Bel}(\overline{B}) \tag{5}$$

i.e. the doubt is the total support for the negation of the proposition B (negation is denoted by an overline).

One of the fundamental differences between Dempster-Shafer and probability theory is that the belief and doubt *do not necessarily sum to unity* i.e. it is not certain that $B \cup \overline{B}$ is true. This can be illustrated diagrammatically as in Figure 1.

The uncertainty Un in the proposition B is that portion of the probability mass which does not support B *or* its negation. If further evidence were provided, some of the uncertainty could move in support of B but the mass assigned to the doubt *cannot* move. This means that the possible belief in B is bounded above by the quantity $\mathrm{Bel}(B) + \mathrm{Un}(B) = 1 - \mathrm{Dou}(B)$ and this quantity is termed the *plausibility* of B and denoted $\mathrm{Pl}(B)$. The plausibility can also be defined by,

$$\mathrm{Pl}(B) = \sum_{A_i \cap B \neq \phi} m(A_i) \tag{6}$$

The belief and plausibility form the lower and upper bounds of an interval of uncertainty for a given event. A concrete example will be useful at this point.

Consider a composite structure which may have sustained damage at one of two internal sites A and B which are indistinguishable. It is known that

the only possible damage mechanism at site A is delamination (denoted D), but site B may fail by delamination or fibre fracture (denoted F) and the relative probabilities of the damage mechanisms are unknown. It is further known that failure at A is twice as likely as failure at B. What can one say about the likely damage type if a fault is found?

First of all, if damage occurs at A it is certainly by delamination, and as damage is twice as likely to be at A as at B this forces the mass assignment,

$$m(D) = \frac{2}{3}$$

the remaining mass cannot be assigned with certainty, so it is assigned to the frame of discernment,

$$m(\Theta) = m(D \cup F) = \frac{1}{3}$$

The belief in the delamination is simply $Bel(D) = 2/3$ *as this is the only basic mass assignment to B. There is no such assignment to F so the belief* $Bel(F) = 0$. *The plausibility in D is given by,*

$$Pl(D) = m(D) + m(D \cup F) = 1$$

and the plausibility of F is similarly calculated as $1/3$. *The uncertainty interval for D is* $[2/3, 1]$ *and that for F is* $[0, 1/3]$.

Note that it is not possible to use probability theory here directly as the relative probabilities at site B are not known. It is possible to construct bounds on the probabilities though. Suppose delamination were impossible at site B, then the overall probability of delamination would be 2/3 and this would be a lower bound. If delamination were certain at B, the overall probability would be 1. Note that these quantities are the belief and plausibility respectively. For this reason, the belief and plausibility are sometimes termed the lower *and* upper probabilities.

The interpretation of some common instances of the uncertainty interval for a proposition B is as follows.

[0,0] B is impossible.

[1,1] B is certain.

[0.75,0.75] There is no uncertainty, the belief in B (0.75) and the doubt in B (0.25) sum to unity.

[0,1] There is total ignorance regarding B.

[0.25,1] B is plausible, there is no support for $\underline{\underline{B}}$.

[0,0.75] $\underline{\underline{B}}$ is plausible, there is no support for B.

[0.25,0.75]. Both B and $\underline{\underline{B}}$ are plausible.

All this suffices to establish terminology, to explain how to compute belief functions and how to interpret the results. It does not provide a means of data fusion - that requires the use of *Dempster's combination rule*.

Suppose that one has two sensors 1 and 2. Basic probability assignments are possible on the basis of either sensor alone, denoted m_1 and m_2. Belief functions Bel_1 and Bel_2 can be computed. Dempster's rule allows the calculation of an overall probability assignment m_+ and a corresponding overall belief function Bel_+, where this *direct sum*,

$$\text{Bel}_+ = \text{Bel}_1 \oplus \text{Bel}_2 \tag{7}$$

is induced by m_+. Suppose that sensor 1 makes assignments $m_1(A_i)$ to the propositions A_i (which can, and usually do, include the frame of discernment), and sensor 2 makes assignments $m_2(B_j)$ to the propositions B_j, then Dempster's rule makes assignments as follows.

Consider a matrix with row entries labelled by i and column entries j, then the $(i,j)^{th}$ element of the matrix is an assignment of probability mass $m_1(A_i) \times m_2(B_j)$ to the proposition $A_i \cap B_j$.

This is again best understood by an example:

> *Suppose a classifier is required which can assign a damage type to data from a composite structure. The possible damage types are delamination D, fibre fracture F, matrix cracking M or fibre pullout P. Two different classifiers are trained, A and B, which produce different probability mass assignments. Classifier A makes the assignments,*
>
> $$m_A(D) = 0.25, \quad m_A(F \cup M) = 0.5 \quad m_A(\Theta) = 0.25$$

and classifier B returns,

$$m_B(D \cup M) = 0.3 \quad m_B(D \cup F) = 0.4 \quad m_B(\Theta) = 0.3$$

Dempsters rule induces a mass assignment matrix,

	$m_A(D)$	$m_A(F \cup M)$	$m_A(\Theta)$
$m_B(D \cup M)$	$0.25 \times 0.3 = 0.075$	$0.25 \times 0.4 = 0.1$	$0.25 \times 0.3 = 0.075$
$m_B(D \cup F)$	$0.5 \times 0.3 = 0.15$	$0.5 \times 0.4 = 0.2$	$0.5 \times 0.3 = 0.15$
$m_B(\Theta)$	$0.25 \times 0.3 = 0.075$	$0.25 \times 0.4 = 0.1$	$0.25 \times 0.3 = 0.075$

to the propositions,

	$m_A(D)$	$m_A(F \cup M)$	$m_A(\Theta)$
$m_B(D \cup M)$	D	M	$D \cup M$
$m_B(D \cup F)$	D	F	$D \cup F$
$m_B(\Theta)$	D	$F \cup M$	Θ

and the direct sum m_+ assigns support to the propositions D, M, F, $D \cup M$, $D \cup F$, $F \cup M$ and $\Theta = D \cup M \cup F \cup P$.

The belief in delamination is $Bel_+(D) = 0.075 + 0.15 + 0.075 = 0.3$; similarly, $Bel_+(M) = 0.1$, $Bel_+(F) = 0.2$ and $Bel_+(P) = 0.0$. The plausibility of delamination D is given by,

$$Pl_+(D) = m_+(D) + m_+(D \cup M) + m_+(D \cup F) = 0.3 + 0.075 + 0.15 = 0.525$$

similarly $Pl_+(M) = 0.275$, $Pl_+(F) = 0.45$ and $Pl_+(P) = 0.075$. Note that P receives no explicit mass assignments and therefore only acquires plausibility through its appearance in Θ.

In summary, the uncertainty intervals for the fused belief function are:

D	$[0.3, 0.525]$
M	$[0.1, 0.275]$
F	$[0.2, 0.45]$
P	$[0, 0.075]$

The most plausible diagnosis is clearly delamination.

In mathematical terms, Dempster's combination rule is expressed as

$$m_+(C) = \sum_{A_i \cap B_j = C} m_1(A_i) m_2(B_j) \tag{8}$$

and,

$$\mathrm{Bel}_+(C) = \sum_{B \subseteq C} m_+(B) \tag{9}$$

Unfortunately things are not quite as straightforward as this. Problems arise in using Dempster's rule if the intersection between supported propositions A_i and B_j is empty. In this circumstance a non-zero mass assignment will be made to the empty set ϕ and this contradicts the basic definition of the mass assignment which demands $m_+(\phi) = 0$. In order to preserve this rule, Dempster's rule *must* assign zero mass to non-overlapping propositions. However, if this is the case, probability mass is lost and the total mass assignment for m_+ will be less than unity, contradicting another rule for probability numbers. A valid mass assignment is obtained by *re-scaling* m_+ to take account of the lost mass. If the mass lost on non-overlapping propositions totals k, the remaining mass assignments should be re-scaled by a factor $K = 1/(1-k)$. The combination rule (8) is modified to,

$$m_+(C) = K \sum_{A_i \cap B_j = C} m_1(A_i) m_2(B_j) \tag{10}$$

Consider the last example. Suppose the assignments made by sensor A were as before, but those of sensor B were now,

$$m_B(D \cup M) = 0.3 \quad m_B(P \cup F) = 0.4 \quad m_B(\Theta) = 0.3$$

Dempster's rule gives the same assignments,

	$m_A(D)$	$m_A(F \cup M)$	$m_A(\Theta)$
$m_B(D \cup M)$	$0.25 \times 0.3 = 0.075$	$0.25 \times 0.4 = 0.1$	$0.25 \times 0.3 = 0.075$
$m_B(P \cup F)$	$0.5 \times 0.3 = 0.15$	$0.5 \times 0.4 = 0.2$	$0.5 \times 0.3 = 0.15$
$m_B(\Theta)$	$0.25 \times 0.3 = 0.075$	$0.25 \times 0.4 = 0.1$	$0.25 \times 0.3 = 0.075$

but this time to the propositions,

	$m_A(D)$	$m_A(F \cup M)$	$m_A(\Theta)$
$m_B(D \cup M)$	D	M	$D \cup M$
$m_B(P \cup F)$	ϕ	F	$P \cup F$
$m_B(\Theta)$	D	$F \cup M$	Θ

and a total mass of 0.15 is lost on the empty set. This means that the assignments should be re-scaled by a factor $K = 1/0.85 = 1.1765$ *(to four decimal places). The mass matrix becomes,*

	$m_A(D)$	$m_A(F \cup M)$	$m_A(\Theta)$
$m_B(D \cup M)$	*0.0882*	*0.1176*	*0.0882*
$m_B(P \cup F)$	*0.0*	*0.2353*	*0.1764*
$m_B(\Theta)$	*0.0882*	*0.1176*	*0.0882*

and the calculation for the belief functions and uncertainty intervals proceeds exactly as before.

The differences between the Dempster-Shafer approach and the probabilistic are manifest. First of all, probabilistic - or rather Bayesian - approaches are unable to accommodate ignorance. All probability must be assigned to the set of propositions under consideration. Secondly, the Bayesian approach is unable to meaningfully assign probabilities to the union of propositions. If the uncertainty for all propositions is zero and the mass assigned to unions is zero, Dempster-Shafer is reduced to Bayesian probability reasoning.

There are other frameworks which seek to extend Bayesian methods in a similar manner to Dempster-Shafer such as the Generalised Evidence Processing (GEP) approach of [11] and that proposed in [12].

3 The DS Neural Network

The object of this section is to briefly describe the neural network implementation of the DS-based classifier. Much more detail can be found in the original reference [7].

The basic idea will be to assign one of M classes $C_1,\ldots,C_M$ (these form the frame of discernment), to a feature vector $\underline{x}$ on the basis of a set of N_t training examples $\underline{x}_1,\ldots,\underline{x}_{N_t}$. Suppose the vector $\underline{x}$ is close to a training example $\underline{x}_i$ with respect to an

appropriate distance measure d ($d_i = ||\underline{x} - \underline{x}_i||$). It is then appropriate that the class of the vector $\underline{x}_i$ influences ones beliefs about the class of $\underline{x}$. One has evidence about the class of $\underline{x}_i$. The approach to the classification taken in [13] is to allocate belief to the event C_q (the possible class of $\underline{x}$), according to the distances $d_i(x)$.

$$m^i(C_q) = \alpha\phi_q(d_i) \tag{11}$$

where $0 < \alpha < 1$ is a constant and ϕ_q is an appropriate monotonically decreasing function. Each training vector close to $\underline{x}$ will contribute some degree of belief. For each training vector, a degree of belief is also assigned to the whole frame of discernment Θ as follows,

$$m^i(\Theta) = 1 - \alpha\phi_q(d_i) \tag{12}$$

The function ϕ_q used here is the basic Gaussian,

$$\phi_q(d_i) = \exp(-\gamma_q(d_i)^2) \tag{13}$$

where γ_q is a positive constant associated with class q. To simplify matters, one confines the construction of the belief assignment for the vector $\underline{x}$ to a sum of the beliefs induced by its nearest neighbours. The sum is computed using Dempster's combination rule as described in Section Two. Actually, a further simplification is made to speed up the processing. Rather than summing over the nearest neighbours from the whole training set in order to assign the belief, one sums over a set of *prototypes* constructed from the training set by a clustering algorithm. Each prototype $\underline{p}_i$ is assigned a degree of membership to the class q denoted by u_q^i with the constraint $\sum_{q=1}^{M} u_q^i = 1$. These are used to compute the belief in the class q for $\underline{x}$ given the distances d_i from the prototypes.

Although it is a gross simplification, the algorithm can be summed up as follows:

1. Construct the prototypes $\underline{p}_i$ from the training data using a clustering algorithm.
2. Given a vector $\underline{x}$, compute the distances from the vector to the prototypes. Using the parameters d_i and u_q^i assign a degree of belief for each class q.
3. Use Dempster's combination rule to compute the total belief in each class from all the contributing prototypes.

The algorithm extends the probabilistic classifier by also making an assignment to the frame of discernment and this quantifies the degree of uncertainty of the classification. The reference [7] explains how the algorithm can be implemented in terms of a four-layer neural network. The network has a feed-forward structure, but is not as simple as an MLP. Despite the fact that DS network is superficially more complicated than the MLP, it has essentially the same computational expense. It is shown in [13], that forward propagation of a given input pattern through the DS network involves the same order of arithmetic operations as for a MLP network. In terms of training time, the issue becomes the expense of making a gradient calculation, and it is shown in Appendix A of [7] that this expense is the same for the DS network and the MLP.

In order to assign a class to the vector $\underline{x}$, one selects that with the largest overall belief assignment induced by the training data (There are other decision strategies as in [14]).

4 A Damage Location Example

In the two papers [15, 16], methods of novelty detection were applied to the damage detection problem for experimental structures. In [15], the structure of interest was an idealised laboratory model of an aircraft wingbox, while in [16], the problem was to detect damage in an inspection panel of a Gnat aircraft wing. In both cases, a novelty detection approach was adopted based on the statistical method of outlier analysis. The next stage of the programme was to investigate the possibilities for damage location in the Gnat wing. Due to restrictions on actually damaging the structure, it was decided to investigate if a method could be developed to see which of a number of inspection panels had been removed. As there were a small number of distinct panels, the problem of damage location was cast as one of classification. Only a short summary of the study will be given here, the interested reader can refer to [17] for more details.

Due to the success of using novelty detectors for the damage detection problem, it was decided to attempt to extend this approach to see whether it could be used for the location problem. A network of sensors was used to establish a set of novelty detectors, the assumption being that each would be sensitive to different regions of the wing. Once the relevant features for each detector had been identified and extracted,

a neural network was used to interpret the resulting set of novelty indices.

4.1 Test Set-up and Data Capture

As described above, damage was simulated by the sequential removal of nine inspection panels on the starboard wing. Figure 2 shows a schematic of the wing and panels.

The area of the panels varied from about 0.008 m to 0.08 m with panels P3 and P6 the smallest. Measured transmissibilities were used as the basic features for construction of the novelty indices and were recorded in three groups, A, B and C as shown in Figure 2. Each group consisted of four sensors (a centrally placed reference transducer and three others). In each case, the transmissibility was the ratio of the acceleration spectrum at a receiver transducer divided by the acceleration spectrum at the reference transducer for the group. Only the transmissibilities directly across the plates were measured in this study. One 16-average transmissibility and 100 one-shot measurements were recorded across each of the nine panels for seven undamaged conditions (to increase robustness against variability) and the 18 damaged conditions (two repetitions for the removal of each of the nine panels).

4.2 Feature Selection and Novelty Detection

The feature selection process for the novelty detectors was conducted by inspecting the transmissibility functions to find small regions of the frequency range of each which distinguished between damage conditions. An exhaustive visual classification of potential features as weak, fair or strong was made with the intention of only selecting fair or strong features, the details can be found in [17]. In order to simplify matters, only the group A transmissibilities were considered to construct features for detecting the removal of one of the group A panels; similarly for groups B and C.

Initially 44 candidate features were evaluated using outlier analysis. The best features were chosen according to their ability to correctly identify the 200 (per panel) damage condition features as outliers while correctly classifying as inliers, those features corresponding to the undamaged condition. Figure 3 shows the results of the outlier analysis for a feature that was clearly able to recognise removal of inspection panel 4. Once the 44 features had been selected by the empirical approach, a Genetic

Algorithm was used to select the best subset of location features by optimising the classification error using an MLP as the classifier [18]. The first set of results given here will consider the case of the best 4 features; the reduction to 4 features was made to ensure that the MLP network used for comparison with the DS network later was unlikely to suffer from overtraining. The case of the optimal 9-feature set will also be considered later. In the latter case, the possibility of overtraining was more of a concern; a detailed discussion of the issues with the particular training sets can be found in [18]

The data was divided into training, validation and testing sets in anticipation of presentation to the classifier. As there were 200 patterns for each damage class, the total number of patterns was 1800. These were divided evenly between the training, validation and testing sets, so (with a little wastage) each set received 594 patterns, comprising 66 representatives of each damage class. The plot in Figure 3 shows the discordancy (novelty index) values returned by the novelty detector over the whole set of damage states. The horizontal dashed lines in the figures are the thresholds for 99% confidence in identifying an outlier, they are calculated according to the Monte Carlo scheme described in [19]. The novelty detector substantially fires only for the removal of panel for which it has been trained. This was the case for most panels but there were exceptions (e.g. there were low sub-threshold discordancies for the smaller panels and some novelty detectors were sensitive to more than one damage type).

Note that there are now two layers of feature extraction. At the first level, certain ranges of the transmissibilities were selected for sensitivity to the various damage classes. These were used to construct novelty detectors for the classes. At the second level of extraction, the 9 indices themselves were used as features for the damage localisation problem. This depends critically on the fact that the various damage detectors are local in some sense, i.e., they do not all fire over all damage classes. This was found to be true in this case.

5 Networks for Damage Location

The final stage of the analysis was to produce a classifier based on the DS neural network algorithm which could serve as a damage location system. As with a standard

MLP network, the specification of the DS network structure requires hyperparameters; in this case, the number of prototypes (analagous to the number of hidden units in the first layer of the network) and the starting values of the weights before training. These were computed by a cross-validation procedure as for the MLP [20]. Many neural networks were trained with the same training data but with differing numbers of prototypes and initial weights. Up to 30 prototypes were considered, and in each case 10 randomly chosen initial conditions were used. The best network was selected by observing which produced the minimum misclassification error on the validation set. The final judgement of the network capability was made by using the independent testing set.

The results for the presentation to the classifier are summarised in the confusion matrix given in Table 1. (The confusion matrix simply counts how many times the classifier makes a certain assignment when the data is from the true class indicated by the row; a perfect classifier would generate a diagonal matrix i.e. the predicted class is always the true class.) The best DS network used 29 prototypes. The probability of correct classification was 89.7%. There were 4 events associated with the frame of discernment, corresponding to probability mass of 0.007. (Note that there are other ways to implement rejection, [14].) This means that allowing for the fact that the network indicates when it has insufficient evidence to make a classification, the classification error is 9.6%.

The main source of confusion is in locating damage to the two smallest panels, 3 and 6, and of course this was anticipated.

In order to make a comparison with the standard approach, the algorithm chosen was a standard Multi-Layer Perceptron (MLP) neural network. The neural network was presented with 4 novelty indices at the input layer and required to predict the damage class at the output layer.

The procedure for training the neural network again followed the guidelines in [20]. The training set was used to establish weights, whilst the network structure and training time etc. were optimised using the validation set. The testing set was then presented to this optimised network to arrive at a final classification error. For the network structure, the input layer necessarily had four neurons, one for each novelty index, and the output layer had nine nodes, one for each class.

Prediction	1	2	3	4	5	6	7	8	9	Θ
True Class 1	54	5	5	0	0	0	2	0	0	0
True Class 2	0	63	0	0	2	0	0	0	0	1
True Class 3	6	1	56	2	0	0	0	0	0	1
True Class 4	5	0	1	55	0	3	0	2	0	0
True Class 5	0	0	0	0	65	0	0	1	0	0
True Class 6	2	2	2	4	0	54	1	0	0	1
True Class 7	0	1	1	0	0	0	61	2	1	0
True Class 8	0	0	1	0	1	0	0	62	1	1
True Class 9	0	0	0	0	0	0	0	3	63	0

Table 1: Confusion matrix for best DS network using 4 log features - testing set.

The training phase used the 1 of M strategy [3]. This approach is simple, each pattern class was associated with a unique network output; on presentation of a pattern during training, the network was required to produce a value of 1.0 at the output corresponding to the desired class and 0.0 at all other outputs.

It is known that MLP networks trained using a squared-error cost function with the 1 of M strategy for the desired outputs, actually estimate Bayesian posterior probabilities for the classes with which the outputs are associated [3]. This means that such a network actually implements a Bayesian decision rule if each pattern vector is identified with the class associated with the highest output.

The best neural network had 19 hidden units and resulted in a testing classification error of 0.118 i.e. 89.2% of the patterns were classified correctly. This means that the misclassification probability is of course 10.8%. The confusion matrix is given in Table 2. Again, the main errors were associated with the two small panels P3 and P6.

The next illustration used the optimal 9-feature set from the GA [18] in order to demonstrate a much better classification accuracy, albeit with the concerns about over-training discussed in [18]. The results for the test set are summarised in the confusion matrix given in Table 3. The best DS network used 28 prototypes. The probability of correct classification was 98.3%. This is approaching the level of classification which

Prediction	1	2	3	4	5	6	7	8	9
True Class 1	61	1	0	0	0	0	1	0	0
True Class 2	0	63	0	0	3	0	0	0	0
True Class 3	1	0	48	8	0	5	7	2	0
True Class 4	0	1	3	56	0	2	0	4	0
True Class 5	0	0	0	0	66	0	0	0	0
True Class 6	4	1	0	9	0	52	0	0	0
True Class 7	1	0	0	0	0	0	59	5	1
True Class 8	1	0	0	0	1	0	1	59	4
True Class 9	0	0	0	0	0	0	0	5	61

Table 2: Confusion matrix for best MLP neural network using 4 log features - testing set.

would be required for a credible SHM systems There were 10 misclassifications, but only one of these events was associated with the frame of discernment, corresponding to a probability mass of 0.0017.

Prediction	1	2	3	4	5	6	7	8	9	Θ
True Class 1	66	0	0	0	0	0	0	0	0	0
True Class 2	0	64	0	1	0	0	0	0	1	0
True Class 3	1	0	64	1	0	0	0	0	0	0
True Class 4	0	0	0	65	0	1	0	0	0	0
True Class 5	0	0	0	0	66	0	0	0	0	0
True Class 6	0	0	0	0	1	65	0	0	0	0
True Class 7	0	0	0	0	0	0	66	0	0	0
True Class 8	0	0	0	0	0	0	0	65	0	1
True Class 9	1	0	0	0	0	2	0	0	63	0

Table 3: Confusion matrix for best DS network using 9 log features - testing set.

The corresponding results from a MLP network are given in Table 4. The network concerned had 10 hidden units and gave a test classification rate of 98.2%. This

corresponds to 11 misclassifications. However, a close look at the confusion matrices shows that there are at most only two misclassified data points which are common. This raises the possibility that one might profitably fuse the two classifiers, and this possibility is explored in the next section.

Prediction	1	2	3	4	5	6	7	8	9
True Class 1	65	0	0	0	0	0	0	0	1
True Class 2	0	65	0	1	0	0	0	0	0
True Class 3	1	0	62	0	0	1	0	1	1
True Class 4	0	0	0	66	0	0	0	0	0
True Class 5	0	0	0	0	66	0	0	0	0
True Class 6	0	3	0	0	0	62	0	1	0
True Class 7	0	0	0	0	0	0	66	0	0
True Class 8	1	0	0	0	0	0	0	65	0
True Class 9	0	0	0	0	0	0	0	0	66

Table 4: Confusion matrix for best MLP neural network using 9 log features - testing set.

6 Data Fusion

As indicated in the last section, there might be some advantage in fusing the results of the MLP classifier and the DS classifier. As both networks see the same feature data, this is an example of fusion through *methodological diversity* [21]. Having decided to fuse the classifiers, one is presented with the question of which uncertainty framework to use: the (Bayesian) probabilistic or the evidence-theoretic; both will be considered here.

6.1 Bayesian Fusion

Using Bayes' rule, the fused *a posterio* probability will be,

$$p(C_i|MLP, DS) = \frac{P(MLP, DS|C_i)P(C_i)}{P(MLP, DS)} \tag{14}$$

The only way to progress at this point is to make an assumption of *conditional independence* of the classifiers. This is to say that,

$$P(MLP, DS|C_i) = P(MLP|C_i)P(DS|C_i) \tag{15}$$

which means that the assignments of the MLP and DS classifiers, given that the data belongs to class C_i are independent or uncorrelated; this is unlikely to be true, but is necessary. Note that this does not amount to an assumption that $P(MLP, DS) = P(MLP)P(DS)$. This assumption yields,

$$P(C_i|MLP, DS) = \frac{P(MLP|C_i)P(DS|C_i)P(C_i)}{P(MLP, DS)} \tag{16}$$

but Bayes' rule also gives,

$$P(MLP|C_i) = \frac{P(C_i|MLP)P(MLP)}{P(C_i)}$$

$$P(DS|C_i) = \frac{P(C_i|DS)P(DS)}{P(C_i)}$$

and substituting these into equation (16) yields,

$$\begin{aligned} P(C_i|MLP, DS) &= \frac{P(C_i|MLP)P(C_i|DS)P(MLP)P(DS)}{P(C_i)P(MLP, DS)} \\ &= \frac{P(MLP)P(DS)}{P(MLP, DS)} \times \frac{P(C_i|MLP)P(C_i|DS)}{P(C_i)} \\ &= k\frac{P(C_i|MLP)P(C_i|DS)}{P(C_i)} \end{aligned} \tag{17}$$

The normalisation constant k is not needed here as the decision rule selects the highest posterior probability over the classes. However, if needed, k is fixed by the condition,

$$\sum_i P(C_i|MLP, DS) = 1 = \sum_i k\frac{P(C_i|MLP)P(C_i|DS)}{P(C_i)} \tag{18}$$

or,

$$k = \sum_i \frac{P(C_i)}{P(C_i|MLP)P(C_i|DS)} \tag{19}$$

The only problem which remains now is the fact that equation (17) contains a probability $P(C_i|DS)$ and the DS theory has returned a probability mass assignment

rather than a probability. The answer here is to use the *pignistic probability* defined by [14],

$$BetP(C_i) = m(C_i) + \frac{m(\Theta)}{M} \tag{20}$$

where M is the number of classes.

At this point a note about the MLP probabilities is needed. Bishop [3] states that the MLP outputs can be interpreted as posterior probabilities when the cross-entropy function is used as an objective function for network training and a softmax function is applied to the outputs. In fact, in the original work by Richard and Lippman [22] it was shown that the interpretation holds if a squared-error cost function is used and the output transfer functions are hyperbolic tangents. This is the situation here; however, because of small fluctuations in the outputs, they can sometimes have small negative values and the sum of the outputs can show small deviations from unity. This is not a problem when the usual decision rule is used for the MLP classifier; however, if the results are to fused with other classifiers, one should force an interpretation of the outputs as probabilities. In order to do this here, the small negative values are zeroed and the outputs are scaled by a common factor to make them sum to unity.

When the classifiers for the 4-feature test data of the previous section are fused, the confusion matrix in Table 5 is obtained.

The classification rate is 92.6%, a small but significant improvement of 2.9% over the DS classifier, that which performed best on its own. When the classifiers on the 9-feature data were fused, the result was the confusion matrix given in Table 6.

The classification rate is 98.5%, an improvement of 0.2% over the best lone classifier.

6.2 Dempster-Shafer fusion

In order to fuse the classifiers in the DS framework, one uses Dempster's combination rule (10), re-stated here as,

$$m_+(C) = K \sum_{A_i \cap B_j = C} m_{DS}(A_i) m_{MLP}(B_j) \tag{21}$$

where one takes $m_{MLP}(C_i)$ to be $P(C_i|MLP)$. In this particular case, one has $\Theta = C_1 \cup C_2 \cup \ldots \cup C_N$. It follows from the assignment above that $m_{MLP}(\Theta) = 0$. As it is

Prediction	1	2	3	4	5	6	7	8	9
True Class 1	62	2	0	0	0	0	2	0	0
True Class 2	0	64	0	0	2	0	0	0	0
True Class 3	0	1	57	6	0	0	2	0	0
True Class 4	0	1	2	60	0	1	0	2	0
True Class 5	0	0	0	0	65	0	0	0	1
True Class 6	1	2	0	5	0	57	1	1	0
True Class 7	0	0	0	0	0	0	60	5	1
True Class 8	1	0	0	0	1	0	0	63	2
True Class 9	0	0	0	0	0	0	0	4	62

Table 5: Confusion matrix for Bayes-fused classifiers using 4 log features - testing set.

Prediction	1	2	3	4	5	6	7	8	9
True Class 1	65	0	0	0	0	0	0	0	1
True Class 2	0	65	0	1	0	0	0	0	0
True Class 3	1	0	63	0	0	1	0	0	1
True Class 4	0	0	0	66	0	0	0	0	0
True Class 5	0	0	0	0	66	0	0	0	0
True Class 6	0	3	0	0	0	63	0	0	0
True Class 7	0	0	0	0	0	0	66	0	0
True Class 8	0	0	0	0	1	0	0	65	0
True Class 9	0	0	0	0	0	0	0	0	66

Table 6: Confusion matrix for Bayes-fused classifiers using 9 log features - testing set.

assumed that the damage locations are disjoint here, it further follows that $C_i \cap C_j = \phi$ for $i \neq j$. The expression (21) reduces to,

$$m_+(C_i) = Km_{MLP}(C_i)m_{DS}(C_i) + Km_{MLP}(\Theta)m_{DS}(C_i) + Km_{MLP}(C_i)m_{DS}(\Theta)$$

$$= Km_{MLP}(C_i)[m_{DS}(C_i) + m_{DS}(\Theta)] \quad (22)$$

As the classifier assigns the class with the highest $M_+(C_i)$, the normalisation constant K is not needed here.

When the fusion rule in equation (22) is applied to the 4-feature data of the previous section, the results are as shown in Table 7.

Prediction	1	2	3	4	5	6	7	8	9
True Class 1	63	2	0	0	0	0	1	0	0
True Class 2	0	64	0	0	2	0	0	0	0
True Class 3	0	1	55	6	0	0	2	2	0
True Class 4	0	1	2	59	0	1	0	3	0
True Class 5	0	0	0	0	65	0	0	0	1
True Class 6	1	2	0	6	0	57	0	0	0
True Class 7	0	0	0	0	0	0	60	5	1
True Class 8	1	0	0	0	1	0	0	63	2
True Class 9	0	0	0	0	0	0	0	4	62

Table 7: Confusion matrix for DS-fused classifiers using 4 log features - testing set.

The fused classification rate is 92.3%, an improvement of 2.6% over the lone DS classifier. This is slightly below the result obtained using Bayes-based fusion. When the results for the classifiers were fused on the 9-feature data, the confusion matrix in Table 8 was obtained.

The results are almost identical to those obtained using Bayesian fusion, only one misclassification is different. The classification rate is 98.5%.

7 Conclusions

The main conclusion here is that the Dempster-Shafer approach to classification implemented as a neural network gives comparable results to the standard analysis using a MLP neural network. The data used is from a full-scale experimental test on an aircraft wing and is therefore a stringent test of algorithms from the point of view of SHM. In fact on 4-feature data the DS network shows a slight improvement, giving a

Prediction	1	2	3	4	5	6	7	8	9
True Class 1	65	0	0	0	0	0	0	0	1
True Class 2	0	65	0	1	0	0	0	0	0
True Class 3	1	0	63	0	0	1	0	0	1
True Class 4	0	0	0	66	0	0	0	0	0
True Class 5	0	0	0	0	66	0	0	0	0
True Class 6	0	3	0	0	0	63	0	0	0
True Class 7	0	0	0	0	0	0	66	0	0
True Class 8	1	0	0	0	0	0	0	65	0
True Class 9	0	0	0	0	0	0	0	0	66

Table 8: Confusion matrix for DS-fused classifiers using 9 log features - testing set.

classification probability of 89.7% compared to 89.2% for the MLP. On 9-feature data the DS network gives a classification rate of 98.3%, compared with 98.2% for the MLP.

One possible advantage of the DS approach is the fact that it can assign patterns to the frame of discernment and thus indicate to the analyst that there is insufficient evidence to make a classification. The effect is small here, with only 0.7% of the probability mass assigned to Θ in the case of the 4-feature data. Taking this effect into account, one can say that the probability of misclassification of the DS network is 9.6% compared to 10.8% for the MLP network. It is conceivable that situations in SHM could arise where this scale of difference could be important.

Another advantage of using the DS approach is to provide the possibility of methodological diversity in order to use data fusion with the results from a MLP classifier. When the two classifiers are fused on the 4-feature set here, the classification rate is increased by 2.9% to 92.6%. This is small but significant change in the context of SHM problems and is much larger here than the improvement in rate seen by exploiting the ignorance from the DS-classifier. When the classifiers are fused on the 9-feature data, the improvement is smaller, giving an overall classification rate of 98.5

The overall goal in SHM must be a zero misclassification rate, in the sense that the classifier will reject the data rather than produce an error. Further work on the

classifiers shown here will consider strategies for data rejection other than the value of the ignorance in the DS-classifier.

Acknowledgements

The authors would like to acknowledge the late Dr David Allman, without whom this work would not have been possible.

References

[1] Hayton (P.), Utete (S.), King (D.), King (S.), Anuzis (P.), Tarassenko (L.) 2007 *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **365** pp.593-514. Static and dynamic novelty detection methods for jet engine health monitoring.

[2] Worden (K.) & Manson (G.) 2007 *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **365** pp.515-537. The application of machine learning to structural health monitoring.

[3] Bishop (C.M.) 1998 *Neural Networks for Pattern Recognition*. Oxford University Press.

[4] Shafer (G.) 1976 *A Mathematical Theory of Evidence*. Princeton University Press.

[5] Osegueda (R.A.), Seelam (S.R.), Mulupuru (B.) & Kreinovich (V.) 2003 *In NDE for Health Monitoring and Diagnostics,SPIE, San Diego, CA* Paper No **5047-18**. Statistical and Dempster-Shafer techniques in testing structural integrity of aerospace structures.

[6] Yang (B.-S) & Kim (K.J.) 2006 *Mechanical Systems and Signal Processing* **20** pp.403-420. Application of DempsterShafer theory in fault diagnosis of induction motors using vibration and current signals.

[7] Denœux (T.) 2000 *IEEE Transactions on Systems, Man and Cybernetics* **30** pp.131-150.A neural network classifier based on Dempster-Shafer theory.

[8] Dempster (A.P.) 1967 *Annals of Mathematical Statistics* **38** pp.325-339. Upper and lower probabilities induced by a multi-valued mapping.

[9] Smets (Ph.) 1990 *IEEE Transactions on Pattern Analysis and Machine Iintelligence* **12** pp.447-458.The combination of evidence in the transferable belief model.

[10] Shafer (G.) & Logan (R.) 1987 *Artificial Intelligence* **33** pp.271-298. Implementing Dempster's rule for hierarchical evidence.

[11] Thomopoulos (S.C.A.) 1990 *In: Sensor Fusion III: 3D Perception and Recognition. Proceedings of SPIE 1383.* Theories in distributed decision fusion: comparison and generalisation.

[12] Yen (J.) 1986 *In: Proceedings of the 5th AAAI-86 national Conference on Artificial Intelligence* pp.125-131. A reasoning model based on an extended Dempster-Shafer theory.

[13] Denœux (T.) 1995 *IEEE Transactions on Systems, Man and Cybernetics* **25** pp.804-813. A k-nearest neighbor classification rule based on Dempster-Shafer theory.

[14] Denœux (T.) 1997 *Pattern Recognition* **30** pp.1095-1107. Analysis of evidence-theoretic decision rules for pattern classification.

[15] Worden (K.), Manson (G.) & Allman (D.J.) 2003 *Journal of Sound and Vibration* **259** pp.323-343. Experimental validation of a structural health monitoring methodology I: novelty detection on a laboratory structure.

[16] Manson (G.), Worden (K.) & Allman (D.J.) 2003 *Journal of Sound and Vibration* **259** pp.345-363. Experimental validation of a structural health monitoring methodology II: novelty detection on an aircraft wing.

[17] Manson (G.), Worden (K.) & Allman (D.J.) 2003 *Journal of Sound and Vibration* **259** pp.365-385. Experimental validation of a structural health monitoring methodology III: Damage location on an aircraft wing.

[18] Worden (K.), Manson (G.), Hilson (G.) & Pierce (S.G.) 2007 *Journal of Sound and Vibration* **309** pp.529-544 Genetic optimisation of a neural damage locator.

[19] Worden (K.), Manson (G.) & Fieller (N.R.J.) 2000 *Journal of Sound and Vibration* **229** pp.647-667. Damage detection using outlier analysis.

[20] Tarassenko (L.) 1998 *A Guide to Neural Computing Applications.* Arnold.

[21] Chandroth (G.O) 2000 *PhD Thesis, University of Sheffield, Department of Computer Science*. Diagnostic classifier ensembles: enforcing diversity for reliability in the combination.

[22] Richard (M.D.) & Lippmann (R.P.) 1991 *Neural Computation* **3** pp.461-483. Neural network classifiers estimate Bayesian *a posteriori* probabilities.

Figure Captions

Figure 1. The Dempster-Shafer uncertainty interval.

Figure 2. Schematic of the starboard wing inspection panels and transducer locations.

Figure 3. Outlier statistic for all damage states for the novelty detector trained to recognise panel 1 removal.

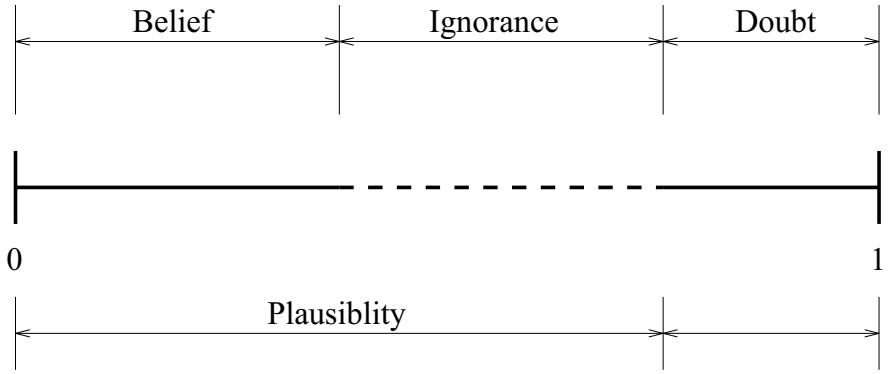

Figure 1: The Dempster-Shafer uncertainty interval.

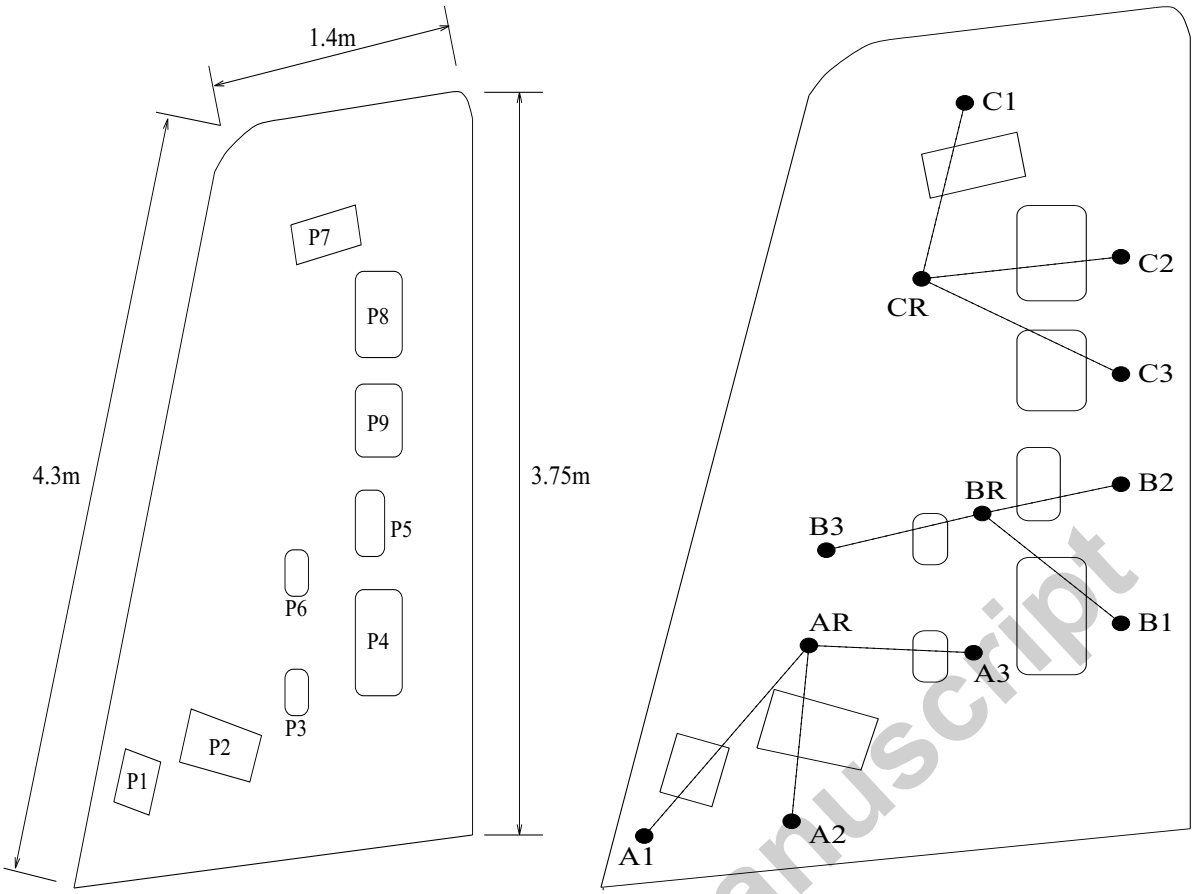

Figure 2: Schematic of the starboard wing inspection panels and transducer locations.

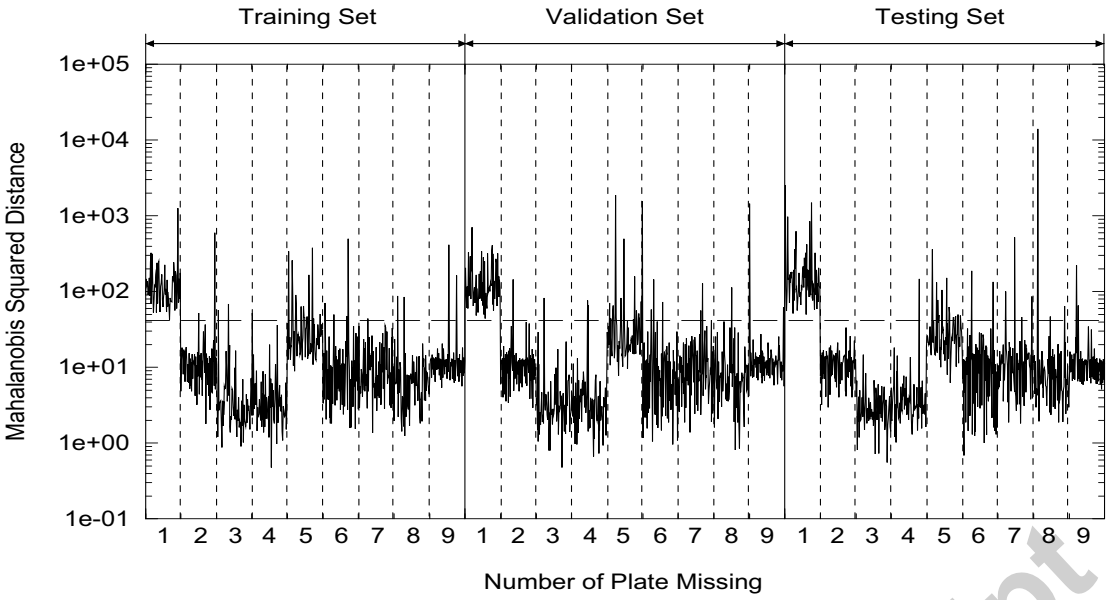

Figure 3: Outlier statistic for all damage states for the novelty detector trained to recognise panel 1 removal.