The application of machine learning to structural health monitoring

BY KEITH WORDEN* AND GRAEME MANSON

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

In broad terms, there are two approaches to damage identification. Model-driven methods establish a high-fidelity physical model of the structure, usually by finite element analysis, and then establish a comparison metric between the model and the measured data from the real structure. If the model is for a system or structure in normal (i.e. undamaged) condition, any departures indicate that the structure has deviated from normal condition and damage is inferred. Data-driven approaches also establish a model, but this is usually a statistical representation of the system, e.g. a probability density function of the normal condition. Departures from normality are then signalled by measured data appearing in regions of very low density. The algorithms that have been developed over the years for data-driven approaches are mainly drawn from the discipline of pattern recognition, or more broadly, machine learning. The object of this paper is to illustrate the utility of the data-driven approach to damage identification by means of a number of case studies.

Keywords: structural health monitoring; machine learning; pattern recognition; neural networks; statistical learning theory; support vector machines

1. Introduction

As the title of this paper suggests, it is concerned with a specific class of algorithms that are applicable to damage detection problems. Owing to space limitations, the paper will not attempt to discuss the desirability of structural health monitoring (SHM); the interested reader will be directed elsewhere within this theme issue. The assumption here is that SHM is a good thing and one should only be concerned with how it is to be accomplished. Even within this philosophy, the remit of this paper will be limited to a discussion of pattern recognition and machine learning algorithms, competing approaches will simply be indicated in the references.

The fundamental problem of SHM, the question of damage detection, is simply posed. The object is just to identify if and when the system departs from normal condition. This is the most basic question that can be addressed. At a slightly more sophisticated level, the problem of damage identification can be approached. This seeks to determine a much finer diagnosis and can even address issues of prognosis. The broader problem can be regarded as a hierarchy of levels which are as follows (Rytter 1993).

* Author for correspondence (k.worden@sheffield.ac.uk).

One contribution of 15 to a Theme Issue ‘Structural health monitoring’.
— **Level 1.** (Detection.) The method gives a qualitative indication that damage might be present in the structure.
— **Level 2.** (Localization.) The method gives information about the probable position of the damage.
— **Level 3.** (Assessment.) The method gives an estimate of the extent of the damage.
— **Level 4.** (Prediction.) The method offers information about the safety of the structure, e.g. estimates a residual life.

The main body of this paper will argue that machine learning theory offers a natural framework in which to address these problems (at least at levels 1–3). Before this can begin, it is necessary to specify the remit of machine learning. This is a body of knowledge that attempts to construct computational relationships between quantities on the basis of observed data and rules. It is characterized by the fact that computational rules are inferred or learned on the basis of observational evidence. This contrasts with the ‘classical’ view of computation, where the algorithmic rules are imposed in the form of a sequence of serially enacted instructions. It is sometimes stated that learning theory is designed to address the following three main problems (Cherkassky & Mulier 1998).

— Classification, i.e. the association of a class or set label with a set or vector of measured quantities. The set of observations may be sparse and/or noisy.
— Regression, i.e. the construction of a map between a group of continuous input variables and a continuous output variable on the basis of a set of (again, potentially noisy) samples.
— Density estimation, i.e. the estimation of probability density functions from samples of measured data.

A further division of learning algorithms may be made between **unsupervised** and **supervised** learning. The former is concerned with characterizing a set on the basis of measurements and perhaps determining underlying structure. The latter requires examples of input and output data for a postulated relationship so that associations might be learnt and errors corrected. Although it is not universally so, regression and classification problems usually require supervised learning, while density estimation can be conducted in an unsupervised framework.

It has been proposed (Vapnik 1998) that the three learning problems shown above are given in order of difficulty. A commonly stated rule of machine learning is that one should never replace a problem with one of a more difficult type. For example, one should not solve a classification problem by learning the densities of the individual classes. One might suspect that the hierarchical system proposed above for learning problems might be brought into correspondence with Rytter’s level-based system for damage identification on the basis of difficulty. In fact, this is not necessarily the case. Level 1 is often addressed by using density estimation techniques (albeit usually with restrictive assumptions), while level 3 is naturally posed as a regression problem in many cases.

The other important factor in using machine learning is that it requires an **organizing principle.** This is sometimes implicit in the analysis, but can be made explicit, for example, in the **data to decision** process of Lowe (2000) or in the embedding of the problem in the general framework of **data fusion** (Worden & Staszewski 2003). In general, the actual machine learning step may only be a part
of the required analysis. It is usually necessary to convert measured data into features, i.e. quantities that make the rule to be learned explicit. Alternatively, feature selection can be regarded as a process of amplification. In the context of damage detection, one transforms the data in such a way so as to retain only the information necessary for a diagnosis, any other redundant information is discarded. This is clearly desirable. Another frequent aim of feature selection is to produce quantities with low (vector) dimension. The reason for this is that the data requirements of learning algorithms usually grow explosively with the dimension of the problem—the so-called curse of dimensionality. Before feature selection, it may be necessary to clean the data or attend to it in other ways: filtering might be employed as a means of noise rejection, missing values may need to be estimated, etc. As an example of an organizing principle, one might cite the waterfall model (Bedworth & O’Brien 2000), as depicted in figure 1.

The waterfall model is one of the simpler structures in data fusion, more sophisticated examples can be found in Worden & Staszewski (2003). It should be noted that equally useful organizing principles can be constructed for the specific context of interest, e.g. the framework discussed in Sohn et al. (2001) which is specific to damage identification. In any case, the point of the foregoing discussion is that the classification or regression step is not necessarily the most complex part of the problem, and data pre-processing or feature selection themselves may prove very difficult. For example, careful feature selection may yield a simple classification problem that yields to an elementary algorithm.

The objective of this paper is to illustrate the application of machine learning algorithms to damage identification problems. This will cover the three main flavours of learning problems and will address the damage identification problem up to and including level 3 in Rytter’s hierarchy. Level 4 in the scheme cannot be addressed by machine learning methods in general. The estimation of useful or remaining life for a system or structure will usually involve an injection of context-specific physical
theory. For example, machine learning may tell us the position and length of a crack in a metallic component, but without a detailed knowledge of the underlying fatigue and fracture properties of the metal (and the particular structure), it will not be possible to extrapolate to failure. The problem of prognosis is one of the most important that structural engineers are facing at present, and the discussion on this point will be taken up again later. In illustrating the different problems and approaches, the authors will draw shamelessly on their own work. This is purely for convenience—the material is readily available. In order to convince oneself of the widespread application of these techniques, one needs only to consult the current literature. A good summary of the initial efforts in the field is given in Doebling et al. (1996) that exhaustively surveys the field of damage identification (using vibration data) up to 1995, citing 20 papers that apply neural networks alone. The sequel to this survey is also available (Sohn et al. 2003) and shows that more research effort has been expended in the field between 1995 and 2001 than was expended up to 1995. It will also show that researchers are now exploiting a considerably greater range of algorithms; neural networks are still popular, but systems like support vector machines (SVMs) from the statistical learning theory are beginning to appear more regularly. Their use will be discussed later in this paper.

As mentioned above, this paper is restricted to learning theory solutions. There are many competing approaches to damage identification that have different formulations, e.g. schemes based on finite element updating. These approaches, when used with care, are similar to those from learning theory. They are usually based on the concept of an inverse problem and employ solution methods drawn from computational linear algebra. Again, Doebling et al. (1996) is an excellent place to find out about these other methods. Another compact source of reference is Worden (2003), which describes the benchmarking activities of the European COST F3 Action on Structural Dynamics. Within this action, Working Group 2 concerned itself with SHM and carried out benchmarking work on two large-scale civil structures. Both machine learning and inverse problem approaches are shown and compared.

The layout of this paper is as follows: §2 is concerned with damage detection, i.e. the lowest level of diagnostics; §3 illustrates a solution to a damage localization problem; §4 shows a damage quantification application; and §5 describes the basic concepts of SVMs and illustrates their use on a classification problem described earlier in the paper. The paper concludes with some discussion on the merits and demerits of the machine learning approaches.

2. Level 1: damage detection

The work in §2a,b,c is concerned with an experimental validation programme for an SHM methodology based on novelty detection.

The work is reported in considerably more detail in Manson et al. (2003a,b) and Worden et al. (2003). The philosophy of the programme of work was to develop methods that are robust enough to be successful on real aircraft structures. The programme spanned a period of around 3 years. The structure of interest was a Gnat trainer aircraft or, more specifically, the starboard wing of the aircraft as shown in figure 2. The first phase of the work was concerned with level 1 in the damage hierarchy.

Phil. Trans. R. Soc. A (2007)
As it was not permitted to damage the aircraft, damage was effectively introduced into an inspection panel. This was accomplished by making 10 copies of the panel; one was left intact and the remaining nine received controlled damage. Figure 3 shows a schematic of the damage conditions. Damage states f1, f2 and f3 were holes of diameter 20, 38 and 58 mm, respectively. States f4, f5 and f6 were saw cuts across the panel width with f4 being an edge cut of 50 mm and f5 and f6 being central cuts of extent 50 and 100 mm, respectively. States f7, f8 and f9 were saw cuts along the longer axis of the panel with f7 being a 100 mm edge cut and f8 and f9 being central cuts of 100 and 200 mm long, respectively. The original point of introducing different damage types was to explore the possibility of classifying the different orientations. This could be, and sometimes is, added as an extra level in Rytter’s hierarchy. The importance of this level is actually for prognosis rather than diagnosis. If one is considering a ‘simple’ fatigue crack in a metal, the question of how the fault will grow in the future depends on its orientation with respect to the future loadings. If one is considering a composite, then each of the fault types (matrix cracking, delamination, fibre snapping, etc.) is likely to require different damage accumulation models.

(b) Data capture

Transmissibilities were used as the base measurements from which novelty detection features would later be selected. The transmissibility between two points i and j is defined here as the ratio of the acceleration spectra measured at those points, i.e. $A_j(\omega)/A_i(\omega)$. These spectra were obtained by Fourier transforming acceleration time data obtained from piezoelectric accelerometers. Appropriate windowing and averaging were employed. The reasons for the choice of transmissibilities were the success on a previous study on a laboratory structure (Worden et al. 2003), and because they were thought to be susceptible to local changes in the region between the relevant sensors (accelerometers). The sensitivity to local changes is important for detecting small damage. Many
other vibrational features like natural frequencies and modeshapes are possible; however, many are global quantities which are not sensitive to small damage. Four sensors were used in all: one pair to establish the transmissibility across the panel in the length direction and one pair across the width. The wing was excited with a white Gaussian excitation using an electrodynamic shaker attached directly below the inspection panel on the bottom surface of the wing. Transmissibilities were recorded in the 1–2 kHz range as this was found to be sensitive to the types of damage being investigated. In all the cases, 2048 spectral lines were recorded. Figure 4 shows two examples of the averaged transmissibility measured across the length of the panel area when the panel had been completely removed. (This shows the degree of variability in the measurements that is to be expected from environmental changes and instrumental drift. The degree of variability as a result of re-fixing the plate with the boundary screws was considerably higher. This is discussed later.)

For the undamaged panel, a 128-average transmissibility and 110 one-shot transmissibilities (100 for the novelty detector training and 10 held back for the testing set) were obtained. Next, for each damaged panel and for the undamaged panel again (for testing purposes), a 128-average transmissibility and 10 one-shot transmissibilities were obtained. Finally, a set of measurements was recorded with the panel completely removed and, for repeatability purposes, a further four tests were carried out to obtain 128-average transmissibilities for the undamaged panel. The panel was removed and replaced between each of the latter tests.
Feature selection

In many situations, there is a requirement for some pre-processing of the raw data signals before proceeding to the feature selection phase, which is represented by the signal processing box in figure 1. In this study, however, pre-processing of the transmissibilities was not deemed necessary.

It is the authors’ opinion that the process of selecting or extracting good features is probably the most important and most difficult phase in the data to decision process. Essentially, in the context of novelty detection, what is meant by a feature is some set of values drawn from or calculated from the measured (or pre-processed) data. The choice of feature will depend upon the novelty detector’s purpose. For damage detection, one desires a feature that is capable of distinguishing between the undamaged and damaged states. It is obvious that a poor choice of feature will probably result in a poor novelty detector. Conversely, a good feature will often result in a successful novelty detector irrespective of the underlying method used to construct the detector.

In the case of the Gnat damage detection study, the 128-average transmissibilities from all of the undamaged and damaged cases were compared. It was found that there was a significant variability between the undamaged transmissibilities due to the panel’s boundary conditions (23 screws). This raises the issue of robust features: a feature will clearly be of little value, even if it does distinguish between damaged and undamaged, if it results in a novelty detector which flags damage when there is merely a slight change in the boundary conditions. The issue of robustness against environmental variability will be discussed later and is highlighted elsewhere in this theme issue.

(c) Feature selection

Figure 4. Examples of averaged transmissibility measurements.
The procedure for selecting potential features for the detection of one or more of the damage states was straightforward: each of the 128-average transmissibilities measured from the various damage conditions was compared to the five 128-average transmissibilities measured from the undamaged structure. This resulted in 1–3 areas of interest being highlighted for each of the four main damage types (namely no panel, holes, width-spanning cuts and length-spanning cuts) for each of the two transmissibilities. Figure 5 shows an example of one of these features, selected to detect length-spanning cuts in the panel. In total, 10 areas of interest were highlighted from the transmissibilities across the panel length and eight from those recorded across the panel width. It is a simple matter to convert these areas of interest into feature patterns; the transmissibility function is simply subsampled over the required region to give an array of 50 sample points or a 50-dimensional pattern in multivariate statistics terminology. This means that there were 18 potential features in total, each of 50 dimensions.

One might argue that if data are available from the damaged and undamaged data, a better course of action would be to train a classifier. In fact, in this case, an optimal approach like the Fisher discriminant could be adopted. However, the intention here is to illustrate a level 1 approach. The damage data are only used here to define novelty detectors that are likely to work well for illustrative purposes. One of the main problems with data-driven approaches, and this will be discussed in more detail later, is that data for the damage cases are rarely available. In the absence of damage data, one conceivable strategy is to define novelty detectors for a number of features and then observe them all to see if there is any signal damage.

Figure 5. Novelty detection feature to detect length-spanning cuts.
Once the features have been selected, the next step is to construct a novelty
detector. This is simply an algorithm that learns—in an unsupervised
manner—the particular characteristics of a given dataset so that new examples
of data can be tested to see if they come from the same source or distribution.
The learning is unsupervised in the sense that only examples from the normal
condition are used, while there is no need for examples of data from the damaged
structure. There are many possible techniques that have been applied
successfully to novelty detection in the past, e.g. artificial neural networks
(Nairac et al. 1997) or, more recently, SVMs (Manevitz & Yousef 2001). The
method used in this study was outlier analysis. The restriction of the particular
algorithm used here is that it assumes that the training data can be represented
by a multivariate Gaussian distribution. If this is not the case, a method capable
of handling representations using a more complicated probability density function,
such as the Kernel Density Estimation, could be employed (Silverman 1986).

Details of outlier analysis and a description of how threshold levels are
calculated are given in Worden et al. (1999). For a more in-depth analysis,
readers are referred to Barnett & Lewis (1994).

Outlier analysis basically calculates a distance measure—the squared (sample)
Mahalanobis distance—\( D_z \), for each testing observation, \( x_z \), from
\[
D_z = (x_z - \bar{x})^T S^{-1} (x_z - \bar{x}),
\]
where \( \bar{x} \) and \( S \) are the mean vector and covariance matrix of the training set.
These measures are then compared to a statistically calculated threshold; above the
threshold the observation is declared an outlier, otherwise it is declared an inlier.

Novelty detectors were constructed for each of the 18 potential features
discussed above. This was done using 500 artificially noise-contaminated
unfaulted features. Six separate tests were conducted under normal condition.
However, although a 128-average transmissibility was obtained in each case, due
to time constraints the 110 one-shot measurements were only obtained in one
test (labelled uf). The test labelled uf2 below extracted 10 one-shot
measurements. In order to obtain more training data, the statistics of the one-
shot measurements were computed from the comprehensive test and 100 samples
were taken from the resulting distribution and added to the averaged
transmissibilities from the restricted tests (not uf2—this was used only for
testing). Principal component visualization of the extended training dataset
showed that it was not Gaussian and that the outlier analysis would therefore be
likely to be conservative. The use of five different normal condition sets allowed
robust features to be selected which were not subject to substantial variation as a
result of the boundary conditions. Testing sets were constructed using the final
10 of the unfaulted patterns (denoted uf in figure 6), followed by 10 of each fault
conditions f1–f9. The testing set was completed with 10 patterns drawn from the
panel removed condition (np) and 10 patterns with the unfaulted panel
reattached (uf2).

Four of the 18 features were capable of detecting some of the damage
conditions while correctly classifying the 20 unfaulted patterns. The rest
produced some false positives and were discarded. (In the event that no damage
data were available to allow a judgement of performance and all novelty
detectors were being monitored, a problem arises as to which novelty detectors to trust. A voting scheme might help, but this is a difficult problem which requires further research.) Figures 6 and 7 show the novelty detector results for two of these features which, when combined, are able to detect all nine damage types and the panel removed condition while correctly returning below-threshold values for the unfaulted patterns.

All the threshold values were calculated using the Monte Carlo method outlined in Worden et al. (1999) based upon the critical value of 1% test of the discordancy. There is, however, a school of thought (Tarassenko et al. 1995) which argues that it is acceptable practice to set a threshold value as being slightly greater than the highest discordancy value of all the testing patterns taken from the unfaulted structure. Accepting this argument results in a single feature being capable of detecting all the damage conditions while classifying the unfaulted test patterns as being below threshold. However, if this procedure is applied it could be argued that a fourth independent dataset would be needed to test generalization.

3. Level 2: damage location

This phase of the work investigated is the next level in Rytter’s damage hierarchy, namely damage location. Having detected that damage is present in the structure, there is generally a desire for further information regarding the location of the damage. This problem is often cast in a regression framework with the output being the coordinates of the damage. This was the framework used to detect the impacts to a composite panel in Worden & Staszewski (2000). In this
study, owing to the restrictions upon actually damaging the structure, the problem of damage location was cast as one of the classification. As in the first phase of the study, the Gnat aircraft was the experimental structure and inspection panels on the starboard wing were used to introduce ‘damage’ into the structure. The question on this occasion was concerned with identifying which of the nine inspection panels had been removed. Although the casting of the problem in a classification framework was imposed by restrictions, it could be argued that this may be a more robust approach to the damage location problem. Consider the problem of damage in an aircraft wing: it may be sufficient to classify which skin panel is damaged rather than give a more precise damage location. It is likely that, by lowering expectations, a more robust damage locator will be the result.

Owing 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 level 2 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. (A further reason for using novelty indices for localization features is that this substantially reduces the dimension of the feature vector. If all the points of the various transmissibility ranges were used, the required neural network would have many more weights and would possibly have difficulty in generalizing.) As the panels were of different sizes, the analysis gave some insight into the sensitivity of the method.

Figure 7. Outlier analysis results for feature from spectral lines 1900–2000 of T34.
As described above, damage was simulated by the sequential removal of nine inspection panels on the starboard wing; this also had the distinct advantage that each damage scenario was reversible and it would be possible therefore to monitor the repeatability of the measurements. Figure 8 shows a schematic of the wing and panels.

The area of the panels varied from approximately 0.008 to 0.08 m² with panels $P_3$ and $P_6$ the smallest. Transmissibilities were again used and were recorded in three groups, A, B and C, as shown in figure 8. Each group consisted of four sensors (a centrally placed reference transducer and three others). Only the transmissibilities directly across the plates were measured in this study. The excitation and recording of the transmissibilities were conducted in the same manner as during the first phase. One 16-average transmissibility and 100 one-shot measurements were recorded across each of the nine panels for the seven undamaged conditions (to increase robustness against variability) and the 18 damaged conditions (two repetitions for the removal of each of the nine panels).

(b) Feature selection and novelty detection

The feature selection process was essentially conducted in the same manner as previously with the only difference being the exhaustive visual classification of potential features as weak, fair or strong. 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.
Candidate features were then evaluated using the 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 9 shows the results of the outlier analysis for the feature that was designed to recognize removal of inspection panel $P_4$. The data are divided into training, validation and testing sets in anticipation of presentation to the neural network classifier. As there are 200 patterns for each damage class, the total number of patterns are 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 shows the discordancy 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 and are calculated according to the Monte Carlo scheme described in Worden et al. (1999). The novelty detector substantially fires only for the removal of panel $P_4$, 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).

(c) Network of novelty detectors for damage location

The final stage of the analysis was to produce a damage location system. The algorithm chosen was a standard multi-layer perceptron (MLP) neural network (Bishop 1998). The neural network was presented with nine novelty indices at the input layer and is required to predict the damage class at the output layer.

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 nine indices themselves are used as features for the damage localization 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.

The procedure for training the neural network followed the guidelines in Tarassenko (1998). The data were divided into a training set, a validation set and a testing set. The training set was used to establish weights, while the network structure and training time, etc. were optimized using the validation set. The testing set is then presented to this optimized network to arrive at a final classification error. For the network structure, the input layer necessarily had nine neurons, one for each novelty index, and the output layer had nine nodes, one for each class.

The training phase used the 1 of M strategy (Bishop 1998). This approach is simple, each pattern class is associated with a unique network output; on presentation of a pattern during training, the network is required to produce a value of 1.0 at the output corresponding to the desired class and 0.0 at all other outputs.

There is an important connection with statistical classifiers as discussed in Bishop (1998), in which it is shown 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 (things are a little more complicated than this, but see Bishop (1998)). 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 network had 10 hidden units and resulted in a testing classification error of 0.135, i.e. 86.5% of the patterns were classified correctly. The confusion matrix is given in Table 1. The main errors were associated with the two small panels \( P_3 \) and \( P_6 \) and the panels \( P_8 \) and \( P_9 \), whose novelty detectors sometimes fired when either of two panels was removed. The classification error could probably be improved using a more extended feature selection strategy.

### Table 1. Confusion matrix for testing data—Gnat damage location problem.

<table>
<thead>
<tr>
<th>true class</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>true class 1</td>
<td>62</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>true class 2</td>
<td>0</td>
<td>62</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>true class 3</td>
<td>0</td>
<td>1</td>
<td>52</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>true class 4</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>57</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>true class 5</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>47</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>true class 6</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>58</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>true class 7</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>57</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>true class 8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>53</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>true class 9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>56</td>
<td></td>
</tr>
</tbody>
</table>

At the second level of extraction, the nine indices themselves are used as features for the damage localization 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.

The procedure for training the neural network followed the guidelines in Tarassenko (1998). The data were divided into a training set, a validation set and a testing set. The training set was used to establish weights, while the network structure and training time, etc. were optimized using the validation set. The testing set is then presented to this optimized network to arrive at a final classification error. For the network structure, the input layer necessarily had nine neurons, one for each novelty index, and the output layer had nine nodes, one for each class.

The training phase used the 1 of M strategy (Bishop 1998). This approach is simple, each pattern class is associated with a unique network output; on presentation of a pattern during training, the network is required to produce a value of 1.0 at the output corresponding to the desired class and 0.0 at all other outputs.

There is an important connection with statistical classifiers as discussed in Bishop (1998), in which it is shown 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 (things are a little more complicated than this, but see Bishop (1998)). 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 network had 10 hidden units and resulted in a testing classification error of 0.135, i.e. 86.5% of the patterns were classified correctly. The confusion matrix is given in Table 1. The main errors were associated with the two small panels \( P_3 \) and \( P_6 \) and the panels \( P_8 \) and \( P_9 \), whose novelty detectors sometimes fired when either of two panels was removed. The classification error could probably be improved using a more extended feature selection strategy.

### 4. Level 3: damage assessment

The most recent phase of the work investigated damage assessment or severity (Manson et al. 2002). The problem was cast in a regression framework on this occasion with the output from a neural network being trained to indicate the
severity of the damage. This means that all three types of machine learning technique have been used in this study. (Given the small number of damage levels considered here, the problem could equally well have been considered as one of the classification. However, the regression approach is illustrated here as the authors believe this is the proper approach to severity problems.)

(a) Test set-up and data capture

Once again, the inspection panels on the starboard wing of the Gnat aircraft were used to introduce damage into the structure. This time, only the panels in group C from figure 8, along with its group of four transducers, were used in the study. For ease of discussion, the transmissibilities between sensors CR and C1, CR and C2, and CR and C3 are denoted $T_7$, $T_8$ and $T_9$, respectively, in this study. Four copies of each panel were produced. One of the copies was left undamaged to act as a reference, while the other three were subjected to three levels of damage: one-eighth, one-quarter and one-half of the panel area was excised (symmetrically from the centre). During the experimental phase, data were also recorded for a complete removal of each panel, giving four different damage states for each panel. These were labelled $D_1$–$D_4$ for one-eighth, one-quarter, one-half and complete removal of panel $P_7$. Labels $D_5$–$D_8$ and $D_9$–$D_{12}$ were used for damage to panels $P_8$ and $P_9$, respectively.

The measurement process was the same as in the two earlier phases. Ten sets of normal condition data and two repetitions of each of the 12 damage cases were recorded. Each test comprised 121 measurements: 120 one-shot transmissibility measurements followed by one 16-average transmissibility.

An initial study revealed that the one-shot measurements were too noisy to give good resolution of the damage extent. In order to overcome this problem, a type of bootstrapping technique was adopted to construct 120 16-average transmissibility measurements for the three transmissibilities for each of the training and testing conditions. These were constructed from the one-shot measurements by randomly sampling (with replacement) 16 one-shot measurements and averaging them.

(b) Feature selection and novelty detection

The methods used during the level 2 exercise were again repeated to select potential features and train the novelty detectors. Five features were highlighted from transmissibility $T_7$ as being potential features for detecting damage in panel 7. Eight $T_8$ features and seven $T_9$ features were highlighted for detecting damage in panels 8 and 9, respectively. (The same panel and transmissibility labels as given in figure 8 were used.)

Novelty detectors were formed, again using outlier analysis, for all of these features by using the 120 16-average bootstrapped measurements from the first eight normal condition sets as training data. The other two normal condition sets and all of the damage condition sets were used to form the testing sets.

In the case of each panel, two features that appeared to be capable of detecting some of the damage cases while correctly classifying the majority of the undamaged cases were selected.
Before proceeding to the damage assessment, the damage was located using the same method as in level 2. The neural network is supplied with the values of the six novelty detectors when the networks were trained, the one that gave the lowest validation error had three hidden units and gave a testing set classification error of 0.0104, i.e. 98.9% of the patterns were classified correctly. These are very good results and a significant improvement on the previous level 2 study, possibly due to the use of averaged data.

The final stage of the analysis was to produce a damage assessment system. The following methodology was used. All the 20 features would be recorded and tested against their relevant novelty detector. If any of these novelty detectors ‘fired’, indicating damage, the process would pass to the level 2 neural network classifier to identify which of the three panels were damaged. Once the damage had been successfully located, the required features would then be passed to the relevant Level 3 neural network to give a measure of the damage severity.

The networks used were again MLP neural networks. The idea this time was to try to map the novelty indices obtained from the transmissibilities to the damage severity. The neural network was supplied with the values of the 5, 8 or 7 novelty indices at the input layer (depending upon which panel is damaged) and required to predict the damage severity at the output layer. The data was again divided into a training set, a validation set and a testing set. For the network structure, the input layer had 5, 8 or 7 nodes, one for each novelty index, and the output layer had a single node, for the severity value (0 indicating no damage and 1 indicating the panel is missing). The number of hidden nodes for each case was determined during training as for the location. Instead of the network structure being determined based on the lowest misclassification error, it was based upon the lowest mean squared error between the predicted damage severity and the actual damage severity.

For damage in panel 1, the best network had five hidden units and gave a mean square error of 5.6% on the testing set. For panel 2, the best network had nine hidden units and gave a mean square error of 0.92% on the testing set. Finally, for damage in panel 3, the best network also had nine hidden units and this gave a mean square error of 2.0% on the testing set. Figure 10 shows the results of the panel 2 neural network when presented with the testing data. It should be noted that the damage conditions have been input as 0.25, 0.50 and 0.75 rather than the proportion of area missing in order to spread them out evenly. This amounts to a nonlinear transformation and may not be the most satisfactory approach.

5. Support vector machines

The foundational material in this section has been discussed in considerably more detail in recent monographs—notably Vapnik (1995), Cherkassky & Mulier (1998) and Cristianini & Shawe-Taylor (2000); it is included here in order to make the paper a little more self-contained as the methods and theory of SVMs.
are arguably less familiar to an engineering readership than the principles of pattern recognition using neural networks. The review paper by Burges (1998) is also a valuable source of information.

SVMs are algorithms motivated by Statistical Learning Theory (Vapnik 1995, 1998), which attempts to formalize much of the basic foundations of machine learning. The basic principle is that an algorithm is defined which learns a relationship between two sets of data. The first is defined on a $d$-dimensional input space and denoted $x$, and the second will be taken (without loss of generality) as one dimensional and denoted $y$. The relationship is induced by or constructed on the basis of training data $(x_k, y_k); k = 1, ..., N$. If $y$ is a continuous variable, the problem is one of regression, and if $y$ is a class label, the problem is one of classification.

(a) Basic theory

It is simplest to begin the discussion of SVMs with a consideration of linear discriminant analysis. This is a statistical technique that seeks to separate two classes of data using a hyperplane (i.e. a straight line in two dimensions). Suppose that the two classes of data are indeed linearly separable as shown in figure 11a. In general, there will be many separating hyperplanes. The problem with many of them is that they will not generalize well, the reason is that they pass too close to the data. The idea of an SVM for linear discrimination is that it will select the hyperplane that generalizes best, and as shown in figure 11b this means the one that is furthest from the data in some sense.

Suppose the hyperplane has the equation

$$D(x) = \langle w, x \rangle = 0,$$

Figure 10. Neural network prediction of damage severity for panel 2.
where \( < \sim, \sim > \) is the standard Euclidean scalar product, then the separation condition is given by

\[
\mathbf{x}_k \in C_1 \Rightarrow D(\mathbf{x}_k) = \langle \mathbf{w}, \mathbf{x}_k \rangle \geq 1,
\]

\[
x_k \in C_2 \Rightarrow D(x_k) = \langle \mathbf{w}, x_k \rangle \leq -1,
\]

(5.2)

(where \( C_1 \) and \( C_2 \) are the two classes), or more concisely

\[
D(x_k) = y_k \langle \mathbf{w}, x_k \rangle \geq 1,
\]

(5.3)

where \( y_k \) in this case is a class label. \( y_k = 1 \) (respectively, \( y_k = -1 \)) if \( x_k \in C_1 \) (respectively, \( x_k \in C_2 \)).

It is a simple matter to show that the distance of each point in the training set \( \mathbf{x}_k \) from the separating hyperplane is \( |D(x_k)|/||\mathbf{w}|| \). Now, \( \tau \) is a margin (an interval containing the hyperplane but excluding all data; figure 11b) if

\[
\frac{y_k D(x_k)}{||\mathbf{w}||} > \tau \quad \forall k.
\]

(5.4)

Note that the parametrization of the hyperplane is currently arbitrary. This can be fixed by specifying

\[
||\mathbf{w}|| \tau = 1,
\]

(5.5)
and this converts (5.4) into the separation condition (5.3). It is now clear that maximizing the margin will place the hyperplane at the furthest point from the data and this can be accomplished—in the light of (5.5)—by minimizing $||w||$ subject to the constraints (5.3). This is a large but well-understood quadratic programming problem. For details of the optimization procedure, the reader can consult Cristianini & Shawe-Taylor (2000).

(b) SVM results for Gnat location classification

Before training the SVM classifier, the data were divided into training, validation and testing sets. Each set had 594 training vectors, 66 for each location class as described in §3b.

The SVMs described in §5a are designed to separate two classes, i.e. to form a *dichotomy*. However, the problem of interest here is to assign data to nine classes. The usual method of using the SVM for multi-class problems is to train one SVM for each class so that each separates the class in question from all the others. This means that nine quadratic programming problems have to be solved for the current problem. There are strategies for multi-class problems that overcome the need for several SVMs, some are summarized in Bennett (1999). The software used in this study was SVMlight (Joachims 1999).

The SVM expresses the discriminant function in terms of a series of basis functions; in this study, it was decided to adopt a radial-basis kernel. The basis functions are

$$
\phi_i(x) = \exp\left(-\gamma ||x-x_i||^2\right),
$$

and the discriminant function is thus equivalent to a radial-basis function network (Bishop 1998) with the centres at each of the training data points and the radii of each of the basis functions are equal. The parameter $\gamma$ is to be specified for each of the classifiers, along with the strength of the margin $C$. (This is a parameter which decides how severely to punish misclassifications in the event that the data are not separable.) The ‘best’ values for these constants were determined by optimizing the performance of the classifiers on the validation sets. The final performance of the classifiers was determined by their effectiveness on the testing sets. A fairly coarse search was carried out with candidate values of $\gamma$ taken from the range between 0.01 and 10.0 and with $C$ taken from the range 1.0–1024.0 in powers of two.

The results from the nine classifiers on the testing set are given in table 2. Note this is different from the usual confusion matrices in one important respect: when a neural network is used to create a classifier with the $1$ of $M$ scheme, each data point is assigned to the one and only class of the $M$ possibilities. In the case of the SVM, each data point is presented to all of the $M$ SVMs, and could in principle be assigned to all or none of the classes. This means that the numbers in the rows of the class matrices do not add up to the number of points in the class.

Overall, on the testing set, the multiple SVM scheme gives 89.2% of correct classifications (including multiple classifications where the correct class is identified) and 7.2% misclassifications. As discussed above, the classifier does not respond to 5.6% of the data and 2.7% points give multiple class labels. Thus, the SVM appears to outperform the MLP neural network (table 1). (If one only counts unambiguous classifications, the SVM achieves 86.5%—the same as the
neural network.) In addition, the SVM indicates which data points are the sources of confusion. One of the other strengths of the SVM approach discussed here is that it is universal and the same algorithm embedded in the same software could just as easily have fitted polynomial or neural network discriminants. This offers the possibility of optimizing the performance over the model basis.

One of the often-cited strengths of the SVM approach is that the quadratic programming problem is convex and thus has a single global minimum. However, in the case described above, this is only true once appropriate values for $\gamma$ and $C$ have been fixed. If these values were left free for optimization, the problem would no longer be quadratic.

### 6. Discussion and conclusions

There are many lessons to be learned from the examples in the previous sections. The overwhelming message is that the diagnostic system will only perform as well as the data that has been used to train it, the adage garbage in–garbage out is particularly apt. The most important issue is already raised at the lowest level of detection. That is, how does one acquire data corresponding to damage states? The reason that this question is already pertinent at the detection level is that it is necessary to decide features that distinguish between the normal condition of the system or structure and the damaged conditions, and this is not possible without examples of the damage condition. In the example in §2, the features were selected regions of certain transmissibility functions. In the absence of examples from the damage cases, it is not possible to assess if a given transmissibility peak is sensitive to a given type of damage, or in fact any type of damage. In the case of damage detection, this problem can be overcome by training novelty detectors for each candidate feature and then monitoring all of them for threshold crossings on new data. This would be tedious but effective. In the case of a damage location system, data for each class of damage becomes essential. This can only be acquired in two ways, by modelling or from experiment. Both approaches have potential problems. If one considers modelling, one must hope that a low-cost model should suffice otherwise one simply invests all the effort that a model-driven approach like FE updating would require anyway. If one considers experiment, it will not generally be

<table>
<thead>
<tr>
<th>prediction</th>
<th>true class 1</th>
<th>true class 2</th>
<th>true class 3</th>
<th>true class 4</th>
<th>true class 5</th>
<th>true class 6</th>
<th>true class 7</th>
<th>true class 8</th>
<th>true class 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>true class 1</td>
<td>55</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>true class 2</td>
<td>0</td>
<td>60</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>true class 3</td>
<td>0</td>
<td>1</td>
<td>58</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>true class 4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>60</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>true class 5</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>61</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>true class 6</td>
<td>2</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>53</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>true class 7</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>63</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>true class 8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td>true class 9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>61</td>
</tr>
</tbody>
</table>
possible to accumulate data by damaging the structure in the most likely ways unless the structure is extremely cheap and mass-produced. It is obvious that a testing programme based on imposed damage could not be used on, say, an aircraft wing. If one cannot impose real damage, one might be able to experimentally simulate the effects of damage. This was the approach adopted in Worden et al. (2001) where local masses were added to a wingbox structure as a means of simulating damage. The rationale behind that approach was that local stiffness reductions will reduce the natural frequencies and an alternative way of reducing the frequencies might be to locally increase mass. This approach showed a certain degree of success on a comparatively simple structure. In any case, the acquisition of training data for diagnostics higher than level 1 is arguably the main obstruction to the widespread use of pattern recognition for damage identification.

The second major problem in damage identification was also raised in §2. Without careful feature selection, the variations in the measured data due to boundary condition changes in the structure swamped the changes due to damage. (This is also a problem for model-based approaches.) This is an observation that is particularly pertinent for civil engineering. If one wishes to carry out a program of automatic monitoring for an aircraft, it is conceivable that one might do it off-line in the reasonably well-controlled environment of a hangar. This is not possible for a bridge that is at the mercy of the elements. It is known that changes in the natural frequencies of a bridge as a result of daily temperature variation are likely to be larger than the changes from damage (Sohn et al. 1999). Bridges will also have a varying mass as a result of taking up moisture from rain, etc. There are two possible solutions to this problem. The first is to accumulate normal condition data spanning all of the possible environmental conditions. This is time-consuming and will generate such a large normal condition set that it will probably be insensitive to certain types of damage. The second solution is to determine features that are insensitive to environmental changes but sensitive to damage. This of course raises the first problem discussed above, where is the damage data coming from?

A third problem relating to data-driven approaches is that the collection or generation of data for training the diagnostic is likely to be expensive, this means that the datasets acquired are likely to be sparse. This puts pressure on the feature selection activities as sparse data will usually require low-dimensional features if the diagnostic is ever going to generalize away from the training set. There are possible solutions to this, e.g. regularization can be used in the training of neural networks in order to aid generalization and this can be as simple as adding noise to the training data. Other possibilities are to use learning methods like SVMs, which are implicitly regularized and therefore better able to generalize on the basis of sparse data.

One issue which applies equally to data- and model-driven approaches is that they are more or less limited to levels 1–3 in Rytter’s hierarchy. If one is to pursue damage prognosis, it is necessary to extrapolate rather than interpolate, and this is a problem for machine learning solutions. If prognosis is going to be possible, it is likely to be very context specific and to rely critically on understanding the physics of the damage progression. In certain simple cases, it is already possible to make calculations. For example, for a crack in a metallic specimen with a simple enough geometry to allow the theoretical specification of

Phil. Trans. R. Soc. A (2007)
a stress intensity, one can use the Paris–Erdogan law to predict the development of the crack given the loading history (or rather future). Even here there are problems. First of all, the loading future is uncertain and it may only be possible to specify bounds. Secondly, the constants of the Paris–Erdogan equation are strongly dependent on microstructure and would probably have to be treated as random variables in a given prediction. These observations are intended to show that prognosis is only likely to be possible in the framework of a statistical theory where the uncertainty in the calculation is monitored at all stages. Another major stumbling block in the application of prognosis is that most realistic situations will not be backed up by applicable theory, i.e. the laws of damage progression are not known for simple materials with complicated geometry, or for ‘complicated’ materials like composite laminates even with simple geometries.

The overall conclusion for this paper is that if the conditions are favourable, machine learning algorithms can be applied with great effect on damage identification problems. In the light of the comments above, ‘favourable’ conditions largely means that data are available in order to train the machine learning diagnostics. Even if the conditions seem to exclude such a solution, one should bear in mind that even a model-driven approach will need appropriate data for model validation.

References


Lowe, D. 2000 Feature extraction, data visualisation, classification and fusion for damage assessment. Oral Presentation at EPSRC SIDANET meeting, Derby, UK.


