
Automated Novelty Detection in Industrial Systems

David A. Clifton^{1,2}, Lei A. Clifton³, and Peter R. Bannister¹,
and Lionel Tarassenko¹

¹ Department of Engineering Science, University of Oxford, Oxford, UK,
davidc@robots.ox.ac.uk, prb@robots.ox.ac.uk, lionel@robots.ox.ac.uk

² Oxford BioSignals Ltd., Abingdon, Oxfordshire, UK

³ Nuffield Department of Anaesthetics, University of Oxford, Oxford, UK,
lei.clifton@nda.ox.ac.uk

1 Introduction

1.1 Novelty Detection

Novelty detection is the identification of abnormal system behaviour, in which a model of normality is constructed, with deviations from the model identified as “abnormal”. Complex high-integrity systems typically operate normally for the majority of their service lives, and so examples of abnormal data may be rare in comparison to the amount of available normal data. Given the complexity of such systems, the number of possible failure modes is large, many of which may not be characterised sufficiently to construct a traditional multi-class classifier [22]. Thus, novelty detection is particularly suited to such cases, which allows previously-unseen or poorly-understood modes of failure to be correctly identified.

Manufacturers of such high-integrity systems are changing the focus of their business such that they take on responsibility for provision of system maintenance [23]. Intelligent data analysis techniques are required to assess the “health” of system components, aiming to identify potential precursors of component failure in advance of actual system failure. This prognostic approach to condition monitoring is useful for types of fault that can be avoided if identified sufficiently early. In addition to providing early warning of critical failure, such techniques enable a “needs-based” approach to system maintenance, in contrast to the traditional dependence on maintenance scheduled at fixed intervals. Typically such warning systems require highly robust alarming mechanisms, with a minimal number of false positive activations, due to the cost involved of decommissioning and examining equipment following an alarm.

In order to provide early warning of this large set of potentially ill-defined possible failures, a novelty detection approach may be adopted. Novelty detection is alternatively known as “one-class classification” [24] or “outlier detection” [25].

This chapter describes recent advances in the application of novelty detection techniques to the analysis of data from gas-turbine engines. Whole-engine vibration-based analysis will be illustrated, using data measured from case-mounted sensors, followed by the application of similar techniques to the combustor component. In each case, the investigation described by this chapter shows how advances in prognostic condition monitoring are being made possible in a principled manner using novelty detection techniques.

1.2 Chapter Overview

Novelty detection theory is introduced in Sect.2, in which data evaluation techniques, modelling methods, and setting of a decision boundary are described. Complementary on-line and off-line analysis of jet engine vibration data are presented in Sect.3, in which different novelty detection methods are investigated for their benefit in providing advance warning of failure, in comparison to conventional techniques. Section 4 presents an analysis of novelty detection techniques applied to a gas-turbine combustor, showing that single-component analysis can provide indication of failure. Finally, conclusions are drawn in Sect.5, providing recommendations for future novelty detection implementations.

2 Novelty Detection for Industrial Systems

This section describes a framework for novelty detection divided into the following stages:

- Existing Methods
- Pre-processing
- Visualisation
- Construction of a model of normality
- Setting novelty thresholds to detect abnormality

2.1 Existing Methods

Novelty detection techniques are typically divided into statistical (or probabilistic) methods, and machine learning (or neural network) methods. The former, being based upon probabilistic foundations, often claim to be the more principled approaches, in which the concept of novelty can be statistically related to the probability of observing abnormal data. The latter are more data-driven approaches, in which models of normality are typically learned from data in such a way as to maximise the chance of making a correct classification decision given previously-unseen examples. Typically, no statistical

assumptions about the data are made, relying instead upon only the observed data to construct a model of normality.

Statistical Methods of Novelty Detection

Fundamental to statistical methods is the assumption that normal data are generated from an underlying data distribution, which may be estimated from example data. The classical statistical approach is the use of density estimation techniques to determine the underlying data distribution [2], which may be thresholded to delimit normal and abnormal areas of data space. These are parametric techniques, in which assumptions are made about the form of the underlying data distribution, and the parameters of the distribution estimated from observed data. Such assumptions may often prove too strong, leading to a poor fit between model and observed data.

More complex forms of data distribution may be assumed by using Gaussian Mixture Models [1,22], or other mixtures of basic distribution types [3,4]. Such techniques can grow the model complexity (increasing the number of distributions in the mixture model) until a good fit with data is deemed to have occurred. Typically, model parameters are estimated from data using the Expectation-Maximisation algorithm [5]. Such methods can suffer from the requirement of large numbers of training data from which to accurately perform parameter estimation [6], commonly termed the curse of dimensionality [31]. Furthermore, the number of components in a mixture model may need to become very large in order to adequately model observed data, which may lead to poor ability to generalise to previously-unseen data (over-fitting).

Non-parametric approaches to statistical novelty detection include so-called boundary- or distance-based methods, such as Parzen window estimators and clustering techniques, and are discussed in Sect. 2.4.

For statistical-based novelty detection of time-series data, Hidden Markov Models (HMMs) have been used, which provide a state-based model of a data set. Transitions between a number of hidden states are governed by a stochastic process [7]. Each state is associated with a set of probability distributions describing the likelihood of generating observable emission events; these distributions may be thresholded to perform novelty detection [8].

HMM parameters are typically learned using the Expectation-Maximisation algorithm. Novelty detection with HMMs may also be performed by constructing an abnormal state, a transition into which implies abnormal system behaviour [9].

A similar state-based approach to novelty detection in time-series data is taken by Factorial Switching Kalman Filters [10]. This is a dynamic extension of the Switched Kalman Filter [12], which models time-series data by assuming a continuous, hidden state is responsible for data generation, the effects of which are observed through a modelled noise process. As with the HMM novelty detection approach, an explicit abnormal mode of behaviour is included within the model, which is used to identify departures from normality.

Also utilising a dynamical model of time-series normal data, the Multi-dimensional Probability Evolution (MDPE) method [13] characterises normal data by using a non-linear state-space model. The regions of state space visited during normal behaviour are modelled, departures from which are deemed abnormal.

Machine Learning Methods of Novelty Detection

Non-statistical methods of novelty detection attempt to construct a model of normality without assuming an underlying data-generating distribution. One of the most well-known examples of this method is the neural network, in which a highly-connected, weighted sum of nodes is trained using observed data. Typically, neural networks are trained for multi-class classification problems, in which example data are classified into one of a pre-defined set of classes [22].

In order to be used in novelty detection (single-class classification), artificial data may be generated around the normal data in order to simulate an abnormal class [11]. Alternative approaches include using the network's instability when presented with previously-unseen abnormal data for the purposes of novelty detection [14].

The Self-Organising Map (SOM), initially proposed for the clustering and visualisation of high-dimensional data [15], provides an unsupervised representation of training data using a neural network. Various applications of the SOM to novelty detection have been proposed [16, 17], while they have been extended into generic density estimators for statistical novelty detection [18].

A more recent successor to the neural network is the Support Vector Machine (SVM), in which a hyperplane is found that best separates data from different classes, after their transformation by a kernel function [19]. In application to novelty detection, two main approaches have been taken. The first finds a hypersphere (in the transformed space) that best surrounds most of the normal data with minimum radius [20]. The second approach separates the normal data from the origin with maximum margin [21]. The latter has also been extended to novelty detection in jet engine vibration data, in which examples of abnormal data can be used to improve the model [37].

2.2 Pre-Processing

When constructing a multi-dimensional novelty detection systems, whose inputs are derived from several different sensors or parameters, normalisation is a primary pre-processing step applied to the data. The aim of normalisation in this case is to equalise the dynamic range of the different features so that all can be regarded as equally important *a priori*. It removes dependence upon absolute amplitudes, whilst preserving information about the relative "normality" of samples.

On each channel of data (or features extracted from them), presented in Sects. 3 and 4, we use the component-wise normalisation function $N(\mathbf{x}_i)$, defined [26] to be a transformation of the d elements within pattern \mathbf{x}_i :

$$N(\mathbf{x}_i) = \frac{\mathbf{x}_i^d - \boldsymbol{\mu}^d}{\boldsymbol{\sigma}^d}, \quad \forall d = 1 \dots D \quad (1)$$

where $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ are vectors of D elements, computed component-wise across all $i = 1 \dots I$ patterns:

$$\boldsymbol{\mu}^d = \frac{1}{I} \sum_{i=1}^I \mathbf{x}_i^d \quad \boldsymbol{\sigma}^d = \left(\frac{1}{I-1} \sum_{i=1}^I (\mathbf{x}_i^d - \boldsymbol{\mu}^d)^2 \right)^{\frac{1}{2}} \quad (2)$$

In practical terms, the patterns are sets of sensor and/or parameter measurements that are most commonly represented as vectors. An example is given later in Equation 16.

2.3 Visualisation

Visualisation is a key method in exploring data sets, both in terms of confirming the results of normalisation, and in deciding which method to use when constructing a model of normality. Typically, patterns are of high dimensionality, which may contain multiple features derived from multiple channels, making the explicit visualisation of such data difficult. This section describes a method that, while allowing exploration of the data set during the construction of the novelty detection system, also provides a convenient method of describing the results of analysis to eventual users of the system.

The “usability” of the system is of great importance in monitoring of industrial systems, in which the eventual users are typically not familiar with pattern recognition techniques. Visualisation can provide a suitable method of showing the model of normality, novelty thresholds, and test patterns in such a way that makes the use of the novelty detection system more intuitive.

Topographic Projection

Unlike variance-preserving techniques such as Principle Component Analysis, topographic projection is a transformation that attempts to best preserve, in the projected space of lower-dimensionality (*latent space*, \mathbb{R}^q), distances between data in their original high-dimensional space (*data space*, \mathbb{R}^d). The *Sammon stress metric* [27] is based upon the distance d_{ij} between pairs of points (x_i, x_j) in \mathbb{R}^d , and the distance d_{ij}^* between the corresponding pair of points (y_i, y_j) in \mathbb{R}^q :

$$E_{\text{sam}} = \sum_{i=1}^N \sum_{j>i}^N (d_{ij} - d_{ij}^*)^2 \quad (3)$$

in which the distance measure is typically Euclidean. *Sammon's mapping* attempts to minimise (3) using gradient descent [28] techniques. However, this method provides a mapping which is only defined for the training set. This limitation is overcome using the *NeuroScale* method which provides a mapping that can be applied to datasets other than the training set, making it well suited for the proposed application of model-based novelty detection when compared with other projection methods which are only defined for the training set.

NeuroScale

With the NeuroScale method [29], a Radial Basis Function (*RBF*) neural network [22] is used to parameterise the mapping from \mathbb{R}^d to \mathbb{R}^q , in which E_{sam} is minimised. This method allows the key advantage that new test patterns may be projected in \mathbb{R}^q without generating a new mapping.

The network architecture (using a single hidden layer between the input and output layers) is typically selected such that the number of hidden nodes is an order of magnitude greater than the dimensionality of the input patterns [30]. Using the same guidelines, the number of available training patterns should be an order of magnitude greater than the number of hidden nodes, in order to adequately populate data space. Each node in the hidden layer corresponds to the centre of a radial basis function in data space, the initial positions of which we set to be those of patterns randomly selected from the training set.

The output weights of the RBF network (i.e., those from hidden layer to output layer) were initialised using Principal Component Analysis (PCA). The $N = 2$ eigenvectors of the data covariance matrix with highest corresponding eigenvalues are found, and these are the N *principal components* of the training set. They represent the projection of the data in N -dimensional space that retains maximum variance information from high-dimensional space. They provide an initial projection of the data, giving targets for the output layer from which initial values of the output weights are found, which is then refined through network training.

Training the RBF network is a two-stage process. In the first stage, the parameters of the radial basis functions are set so that they model the unconditional probability density of the data, $p(x)$ - that is, the distribution of probability mass describing the likelihood of observing normal data x . In the second stage, the output weights are set by optimising an error function using methods from linear algebra [31].

Using Visualisation

Figure 1 shows an example in which 20-dimensional patterns derived from jet engine vibration data (described in more detail later in Sect. 3) are projected into 2-dimensional space. Note that the axes of NeuroScale projections are

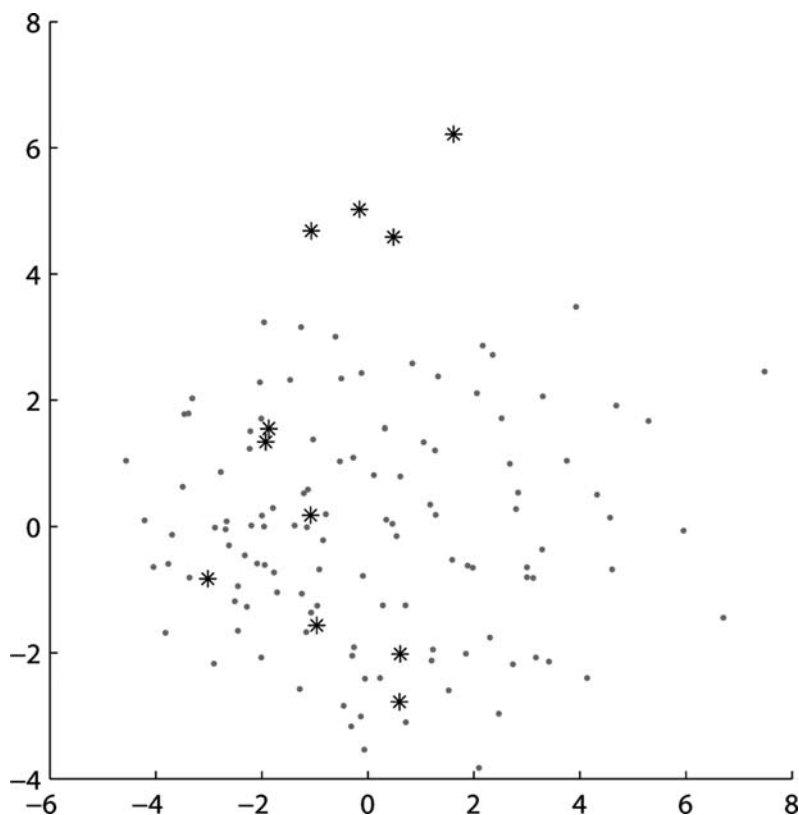


Fig. 1. NeuroScale visualisation of 20-dimensional patterns derived from jet engine vibration data. “Normal” patterns are projected as dots, “abnormal” patterns (as labelled by the data provider) are projected as asterisks. It is clear from the visualisation that seven of the “abnormal” patterns lie within the cluster of points described by the “normal” patterns, revealing that the labels supplied by the data-providers were not accurate and should be modified before selecting patterns from which to train a model of normality

unit-less. Using the component-wise normalisation scheme described above, the visualisation shows that “normal” patterns form a main cluster. 4 of the 11 patterns labelled “abnormal” by the data provider are projected significantly removed from this cluster.

This example illustrates the use of visualisation for the verification of data labels, as it is often the case that, in practice, many patterns labelled “abnormal” by domain experts may actually appear normal after feature extraction, and vice versa. On discussion with the data providers, the seven “abnormal” patterns lying within the cluster formed by normal patterns were found to contain system faults such that no vibration-based consequences were observable in the data [32]. That is, the visualisation technique (based on vibration data) correctly shows that the 7 patterns originally labelled “abnormal” should not

be separate from normal data in this case (and thus may be included as “normal” data in the training set).

Using visualisation in this way allows the exploration of the data set to define which patterns should be used for training a model of normality.

2.4 Constructing a Model of Normality

As previously defined, classifying data as “abnormal” in order to raise an alert about the condition of a system requires a model of normality. This section describes the application of two main types of normal models that have proved successful in the analysis of the gas-turbine data described later in this chapter.

Distance-Based Models of Normality

Distance-based models of normality attempt to characterise the area of data space occupied by normal data, with test data being assigned a novelty score based on their (typically Euclidean or Mahalanobis) distance from that model. A novelty threshold may be set on the novelty score, above which data are deemed to be abnormal; this is further discussed in Sect. 2.5.

Here, we illustrate the process of constructing a distance-based model using two methods:

- Cluster-based models, which have previously been applied to jet engine data [33–35] and ship turbocharger data [36]. This method uses distances in the untransformed feature space, and is illustrated using vibration data in Sect. 3.
- Support Vector Machine models, which have previously been applied to jet-engine data [37] and combustion data [38, 39]. This method uses distances in a transformed kernel space, and is illustrated using combustion data in Sect. 4.

Cluster-Based Models

When modelling a system whose state is represented by a large training set of patterns, it is often desirable to be able to represent the set of patterns by a smaller set of generalised “prototype” patterns, making optimisation of a model computationally-tractable. The k -means clustering algorithm [30] is an iterative method of producing a set of prototype patterns μ_j (for $j = 1 \dots k$) that represent the distribution of a (typically much larger) set of patterns x_i .

The k -means clustering algorithm is used, as described in [40], to construct a model of normality from “normal” patterns in the data sets later described in this chapter. In this method, the distribution of “normal” patterns is defined by \mathbf{C}_k cluster centres in \mathbb{R}^d space, each with an associated *cluster width* σ_k . A novelty score $z(\mathbf{x})$ may be computed for shape vector \mathbf{x} with respect to the K cluster centres:

$$z(\mathbf{x}) = \min_{k=1}^K \frac{d(\mathbf{x}, \mathbf{C}_k)}{\sigma_k} \quad (4)$$

where $d(\mathbf{x}, \mathbf{C}_k)$ is Euclidean distance. We use the definition of width σ_k from [34]

$$\sigma_k = \sqrt{\frac{1}{I_k} \sum_{i=1}^{I_k} d(\mathbf{x}_i, \mathbf{C}_k)^2}. \quad (5)$$

for the I_k points which have closest cluster centre \mathbf{C}_k . This allows an intuitive interpretation of the magnitude of novelty scores: $z(\mathbf{x})$ is the number of standard deviations that pattern \mathbf{x} lies from its closest cluster centre, relative to the distribution of training data about \mathbf{C}_k .

Investigation of the placement of cluster centres is possible using the NeuroScale method. The position of cluster centres with respect to the data set may be determined by projection using the NeuroScale network previously trained using the patterns from the example data set. Selection of a candidate model of normality can be assisted through use of the projections generated by the neural network, where we ensure that the cluster centres accurately represent the distribution of training patterns.

Due to the random initialisation of cluster centres, the positions of those centres at the conclusion of the training process may vary between different models trained upon the same data set. Typically, the algorithm is run several times (each producing a different set of cluster centre locations μ_j), and one of the candidate models is selected based on some metric of fitness in characterising the data x_i .

The optimal number of centres k required to characterise the training set of normal data is often selected using a separate set of data (the validation set), to test the model's ability to generalise to previously-unseen data.

Finally, a threshold H is applied to $z(\mathbf{x})$ such that all patterns $z(\mathbf{x}) \geq H$ are classified "abnormal". This is described in Sect. 2.5.

Support Vector Machine Models

Support vector machines (SVMs) belong to a family of generalized linear classifiers, used for classification and regression. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as *maximum margin classifiers*.

We follow the strategy developed in a key paper for SVM novelty detection [42] that maps the data into the feature space corresponding to the kernel function, and separates them from the origin with maximum margin.

The l training data $\mathbf{x}_1, \dots, \mathbf{x}_l \in \mathbb{R}^d$ can be mapped into another feature space \mathbb{F} through a feature mapping $\Phi: \mathbb{R}^d \rightarrow \mathbb{F}$. The kernel function operates on the dot product of the mapping function

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)). \quad (6)$$

A Gaussian kernel function is used here to suppress the growing distances for larger feature spaces [41]

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2), \quad (7)$$

where σ is the width parameter associated with the kernel function.

For this investigation, we define the following function

$$g(\mathbf{x}) = \rho_0 - \sum_{i=1}^{N_s} \alpha_i k(\mathbf{s}_i, \mathbf{x}) \quad (8)$$

to assign novelty values to data, such that abnormal data (i.e., those outside the single, “normal” training class) take positive values, while normal data take zero values. In this equation, N_s is the number of support vectors, \mathbf{s}_i are support vectors and α_i are non-zero Lagrangian multipliers.

A threshold is set on these novelty scores as described in Sect. 2.5.

Probabilistic Models of Normality

In contrast to distance-based modelling methods, probabilistic approaches to constructing a model of normality estimate the unconditional probability density function $p(\mathbf{x})$ of normal training data. In this investigation, we illustrate the method using Parzen window density estimation [22].

This is a non-parametric kernel-based method, in which the density function is represented by a linear superposition of kernel functions, with one kernel centred on each sample. We choose the Gaussian kernel for Parzen window methods due to its convenient analytical properties. Following [43], we set the width parameter σ for Parzen windows to be the average distance of k nearest neighbours from each sample in the normal training data. Instead of a fixed value of k used in [43], here we specify k to be a fraction of the total number of training data N (e.g., $k = N/10$), so that the value of k is adjusted depending on the number of the training data.

With the unconditional data density $p(\mathbf{x})$ estimated, we may place a probabilistic threshold to separate “normal” from “abnormal” as described in Sect. 2.5.

2.5 Novelty Scores and Thresholds

Though it is desirable for probabilities to be retained as long as possible throughout the classification process, a novelty detection scheme must ultimately decide if a value x is “normal” or “abnormal”, for which a decision boundary H on x must be determined. We term this the *novelty threshold*.

Thresholds for Distance-Based Models of Normality

Typically, for distance-based models of normality, the threshold H on novelty score $z(\mathbf{x})$ is selected to best separate the “normal” and “abnormal” patterns in the data set. However, particularly in the monitoring of high-integrity systems, the number of available “abnormal” examples may be small. Furthermore, this method is heuristic, and requires manual selection of H for each new data set. This is illustrated with clustering models and the vibration data set, described in Sect. 3.

A more principled approach may be obtained by calibrating novelty scores into probabilities, and then setting the novelty threshold as with probabilistic models of normality. This approach is illustrated with the SVM model and combustion data set, described in Sect. 4.

For calibration of novelty scores, [44] proposed a non-parametric form of regression, with the restriction that the mapping from scores into probabilities is isotonic (i.e., non-decreasing); that is, the classifier ranks examples correctly. The Pair-Adjacent Violators (PAV) algorithm [44] is employed to perform the isotonic regression, which finds the stepwise-constant isotonic function $g^*(\mathbf{x})$ that fits the data according to a mean-squared error criterion.

Let \mathbf{x}_i ($i = 1, \dots, l$) be the training examples from normal and abnormal classes $\{\mathcal{C}_0, \mathcal{C}_1\}$, $g(\mathbf{x}_i)$ be the value of the function to be learned for each training samples, and $g^*(\mathbf{x})$ be the function obtained from isotonic regression. The PAV algorithm takes the following steps:

STEP 1: Sort the examples according to their scores, in ascending order.

Initialise $g(\mathbf{x}_i)$ to be 0 if \mathbf{x}_i belongs normal class, and 1 if abnormal class.

STEP 2: If $g(\mathbf{x}_i)$ is isotonic, then return $g^*(\mathbf{x}_i) = g(\mathbf{x}_i)$. If not, go to STEP 3.

STEP 3: Find a subscript i such that $g(\mathbf{x}_i) > g(\mathbf{x}_{i+1})$. The examples \mathbf{x}_i and \mathbf{x}_{i+1} are called pair-adjacent violators. Replace $g(\mathbf{x}_i)$ and $g(\mathbf{x}_{i+1})$ with their average:

$$g^*(\mathbf{x}_i) = g^*(\mathbf{x}_{i+1}) = \{g(\mathbf{x}_i) + g(\mathbf{x}_{i+1})\}/2 \quad (9)$$

STEP 4: Set $g(\mathbf{x}_i)$ as the new $g^*(\mathbf{x}_i)$. Go back to STEP 2.

$g^*(\mathbf{x})$ is a step-wise constant function which consists of horizontal intervals, and may be interpreted as $P(\mathcal{C}_1|\mathbf{x})$, the probability that sample \mathbf{x} is abnormal (i.e., belongs to \mathcal{C}_1 , the abnormal class). For a test example \mathbf{x} , we first find the interval to which its score $z(\mathbf{x})$ belongs. Then we set the value of $g^*(\mathbf{x})$ in this interval to be $P(\mathcal{C}_1|\mathbf{x})$, the probability estimate of \mathcal{C}_1 given \mathbf{x} .

If the scores rank all examples correctly, then all class \mathcal{C}_0 examples will appear before all class \mathcal{C}_1 examples in the sorted data set in STEP 1. The calibrated probability estimate $g^*(\mathbf{x})$ is 0 for class \mathcal{C}_0 and 1 for class \mathcal{C}_1 . Conversely, if the scores do not provide any information, $g^*(\mathbf{x})$ will be a constant function, taking the value of the average score over all examples in class \mathcal{C}_1 .

The PAV algorithm used in isotonic regression may be viewed as a binning algorithm, in which the position and the size of the bins are chosen according

to how well the classifier ranks the samples [44]. The practical use of this algorithm in calibrating novelty scores $z(\mathbf{x})$ into probabilities is illustrated in Sect. 4.

Thresholds for Probabilistic Models of Normality

Novelty threshold H can be defined using the unconditional probability distribution $p(\mathbf{x}) < H$. However, because $p(\mathbf{x})$ is a distribution function, it is necessary to integrate to give the cumulative probability $P(\mathbf{x})$. This is then used to set thresholds in relation to the actual probability of observing sensor noise (e.g., $P(\mathbf{x}) = 10^{-9}$).

A more principled method of setting thresholds for novelty detection uses Extreme Value Theory (EVT) to explicitly model the tails of the distribution of normal data.

EVT is concerned with modelling the distribution of very large or very small values with respect to a generative data distribution. Here, we consider “classical” EVT as previously used for novelty detection [45, 46], in contrast to an alternative method commonly used in financial applications, often termed the *peaks-over-threshold* approach [47]. We do not use the latter for this novelty detection application due to its requirement to adopt an arbitrary threshold above which peaks are measured.

Throughout the investigation described in this chapter, we consider the estimation of the probability of observing abnormally *large* values with respect to a set of normal data. Consideration of abnormally small values requires a simple modification of the theory, not further pursued here.

Consider a “normal” training set of m i.i.d. (independent and identically distributed) data, $\mathbf{X} = \{x_1, x_2, \dots, x_m\}$, distributed according to some function $D(x)$, with maximum $x_{\max} = \max(\mathbf{X})$. We define a distribution function for x_{\max} to be $H(x_{\max} \leq x)$. I.e., our belief in the value of the maximum of the m data drawn from distribution D (over the range of x) is modelled by H .

It can be shown [48] that for H to avoid degeneration as $m \rightarrow \infty$, it must converge according to the transform

$$x_{max} \doteq \sigma_m x + \mu_m \tag{10}$$

for some location and scale parameters, $\mu_m \in \mathbb{R}$ and $\sigma_m \in \mathbb{R}^+$, respectively, and where \doteq is a weak convergence of distributions. Furthermore, for any underlying data distribution D , the limit distribution must take the normalised form

$$H(x_{\max} \leq x) \doteq H\left(\frac{x - \mu_m}{\sigma_m}\right) \doteq H(y_m) \tag{11}$$

where $y_m = (x - \mu_m)/\sigma_m$ is termed the *reduced variate*. According to the Fisher–Tippett theorem [48], H must belong to one of three families of

extreme value distributions (derived from the Generalised Extreme Value distribution [49]). In this investigation, we consider data distributed according to the one-sided standard Gaussian $\mathbf{X} \sim |N(0, 1)|$, which converges to the Gumbel distribution for extreme values [49]. The probability of observing some $x_{\max} \leq x$ given the data is $P(x_{\max} \leq x|\mathbf{X})$, or $P(x|\mathbf{X})$ for simplicity, given by the Gumbel form:

$$\begin{aligned} P(x|\mathbf{X}) &= P(x|\boldsymbol{\theta}) \doteq H(y_m) \\ &\doteq \exp(-\exp(y_m)) \end{aligned} \quad (12)$$

where model parameters $\boldsymbol{\theta} = \{\mu_m, \sigma_m\}$ are the location and scale parameters from the reduced variate $y_m = (x_m - \mu_m)/\sigma_m$, and are derived from \mathbf{X} . The associated probability density function is found by differentiation:

$$\begin{aligned} p(x|\mathbf{X}) &= p(x|\boldsymbol{\theta}) = \sigma_m^{-1} \exp\{-y_m - \exp(y_m)\} \\ &= \sqrt{\lambda_m} \exp\{-y_m - \exp(y_m)\} \end{aligned} \quad (13)$$

which we term the *Extreme Value Distribution* (EVD). Note that, for later convenience, we use the precision $\lambda_m = 1/\sigma_m^2$. Classical EVT assumes that the location and scale parameters are dependent only on m [50], which has been verified via Monte Carlo simulation for $m = 2, \dots, 1,000$ [45]. These take the form

$$\mu_m = \sqrt{2 \ln m} - \frac{\ln \ln m + \ln 2\pi}{2\sqrt{2 \ln m}} \quad \lambda_m = 2 \ln m \quad (14)$$

Thus, using the EVD equation (13), we can directly set novelty thresholds in the Parzen window model, which has placed a set of Gaussian kernels in feature space. This is illustrated in Sect. 3.

3 Gas-Turbine Data Analysis

Here, we use the novelty detection techniques described previously for the analysis of jet-engine vibration data, recorded from a modern civil-aerospace engine. A cluster-based distance model and a Parzen window-based probability model are constructed as described in Sect. 2.4. Novelty thresholds are then set for these models as described in Sect. 2.5.

First, this section provides an overview of jet engine operation, and the vibration data obtained. Then, two monitoring paradigms are described:

- *Off-line Novelty Detection*, in which vibration data from an entire flight are summarised into a single pattern. Novelty detection then takes place on a flight-to-flight basis using these patterns. This scheme is suitable for ground-based monitoring of a fleet of engines.
- *On-line Novelty Detection*, in which data *within* flights are compared to a model of normality. Novelty detection takes place on a sample-by-sample basis. This scheme is suitable for “on-wing” engine monitoring.

3.1 System Description

Jet Engine Operation

Modern aerospace gas-turbine engines divide the task of air compression from atmospheric pressure to that ultimately required within the combustion chamber into several stages. Gas-turbine engines within the civil aerospace market involve up to three consecutive compression stages: the low pressure (*LP*), intermediate pressure (*IP*), and high pressure (*HP*) stages [51]. Air passes through each stage as it travels from the front of the engine to the rear, being further compressed by each, until it reaches the combustion chamber.

Each of the compressor stages is driven by its own turbine assembly, resulting in three corresponding turbine units situated within the exhaust stream at the rear of the engine. Each compressor is linked to its corresponding turbine by a separate shaft, which are mounted concentrically. In three-compressor engines, these are named the LP shaft, the IP shaft, and the HP shaft. The *operational point* of the engine is often defined in terms of the rotational speed of these shafts.

Engine Vibration Measurement

Transducers are mounted on various points of the engine assembly for the measurement of engine vibration. Vibration data used for investigations described in this report were acquired using a system [52] that computes Fast Fourier Transforms (*FFTs*) representative of engine vibration every 0.2 s, for each sensor output. Engine vibration is assumed to be pseudo-stationary over this measurement period such that the generated *FFTs* are assumed to be close approximations of actual engine vibration power spectra.

A *tracked order* is defined [53] to be the amplitude of engine vibration measured within a narrow frequency band centred on the fundamental or a harmonic of the rotational frequency of a shaft.

During normal engine operation, most vibration energy is present within tracked orders centred on the fundamental frequency of each rotating shaft; we define these to be *fundamental tracked orders*. Using the terms *LP*, *IP*, and *HP* to refer to engine shafts, we define fundamental tracked orders associated with those shafts to be *1LP*, *1IP*, and *1HP*, respectively.

Significant vibration energy may also be observed at harmonics of the rotational frequency of each shaft. These *harmonic tracked orders* may be expected to contain less vibration energy than corresponding fundamental tracked orders during normal engine operation. In the example of an LP shaft rotating with frequency 400 Hz, harmonic tracked orders may be observed at frequencies $400n$ Hz, for $n = 0.5, 2, 3, 4, \dots$. We define these harmonic tracked orders of the LP shaft to be *0.5LP*, *2LP*, *3LP*, *4LP*, \dots .

The system used to acquire data for the investigations described within this report automatically identifies peaks in vibration spectra corresponding to fundamental and harmonic tracked orders, using measurements of the

rotational frequency of each shaft. From these peaks, a time series of vibration amplitude and phase for each tracked order is generated.

3.2 Off-Line Novelty Detection

If there is no strict requirement to carry out novelty detection in real-time, then it is possible to compute summary data structures at the end of each flight, and compare the time-averaged behaviour of key features against previous engine runs. This typically suits ground-based monitoring, in which flight data from a fleet of engines is summarised and analysed by engine manufacturers.

An example of this is shown in Fig. 2 where the vibration amplitude (of a fundamental tracked order) has been averaged against speed over the known operating speed-range. This is a *vibration signature* for the flight, and is typically high-dimensional (with 400 speed sub-ranges usually considered).

Increasing dimensionality of data requires exponentially increasing numbers of patterns within the data set used to construct a general model; this is termed the *curse of dimensionality* [22]. In order to avoid this problem, each 400-dimensional vibration signature is summarised by a 20-dimensional pattern \mathbf{x} . This is performed by computing a weighted average of the vibration amplitude values $a(s)$ over $N = 20$ speed sub-ranges [26]. The d^{th} dimension of shape vector \mathbf{x}^n , for $n = 1 \dots N$, is defined to be:

$$\mathbf{x}^d = \int_{s_{\min}}^{s_{\max}} a(s)\omega_d(s)ds \quad (15)$$

in which the vibration amplitude $a(s)$ is integrated over the speed range $s : [s_{\min} \ s_{\max}]$, using weighting functions $\omega_d(s)$, for $d : 1 \dots N$. Thus, each flight of a test engine results in a 20-dimensional pattern.

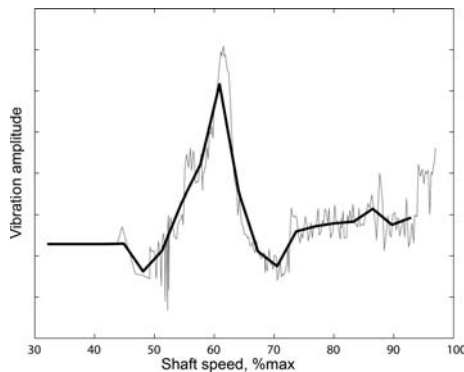


Fig. 2. Constructing a quantised 20-D pattern (*thick line*) from a high-dimensional vibration signature (*thin line*), in which average engine vibration response is plotted against a range of shaft speeds - axes have been anonymised for reasons of data confidentiality

A data set was provided for this investigation consisting of 137 flights. Flights {1...127} were “normal”. A change in engine condition occurred during flights {128...135}, which was retrospectively determined to be an engine component becoming loose. An engine event was observed during flights {136...137}.

Flights {1...80} were used as the training set for both distance-based and probabilistic models. Flights {81...137} were used as a test set. From each flight, a vibration signature was constructed. Those flights which covered less than 60% of the speed range (indicating that the aircraft did not leave the ground) were deemed to be “invalid”, and not considered within the analysis, to ensure that only fully-specified vibration signatures were used.

Component-wise normalisation, as described in Sect. 2.2 was applied to the signatures for each flight, and models of normality constructed using cluster-based and Parzen window methods, as described in Sect. 2.4. Novelty thresholds were then set using the methods described in Sect. 2.5.

An example of resulting models is shown in Fig. 3, in which visualisation has been performed using the NeuroScale method described in Sect. 2.3. Here, models have been trained in 2-dimensional visualisation space for the purposes of explanation. In the actual novelty detection exercise, models were constructed in feature space (here, the 20-dimensional space of the patterns derived from the vibration signatures).

In the example figures, projected normal patterns are plotted as dots (one per flight). The last flights of the engine are plotted as crosses, and are notably separated from the cluster formed by the normal patterns. These flights were known to have been abnormal in the final two flights (in which an engine event occurred), but this retrospective analysis shows that the flights immediately prior to the event are also classified as abnormal.

This is shown in Fig. 4, in which novelty scores for each flight are shown for both distance-based and probability models, with novelty thresholds (determined as described in Sect. 2.5 using heuristic and EVT methods, respectively)

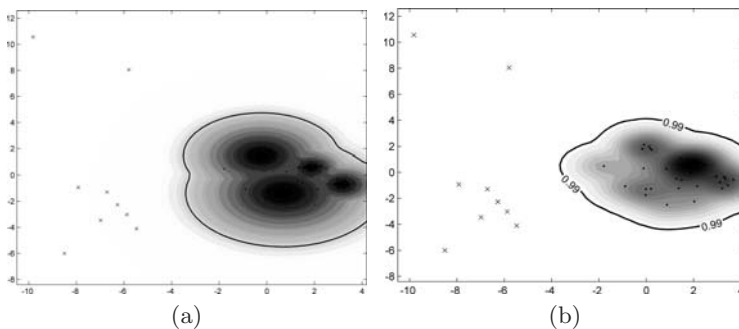


Fig. 3. (a) Distance-based model using $k = 4$ cluster centres and projected data, constructed using cluster-based methods, (b) Probabilistic model and projected data, constructed using Parzen-windows method

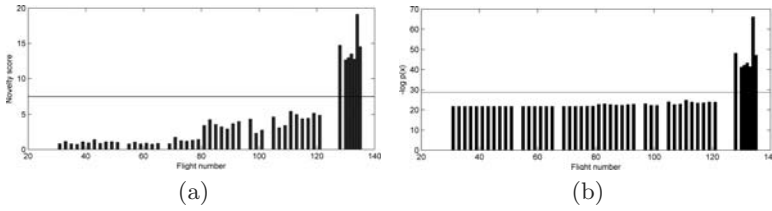


Fig. 4. (a) Novelty scores for the distance-based model, with novelty threshold (*dashed line*), (b) Novelty scores for the probabilistic model, with novelty threshold (*dashed line*)

shown as a horizontal line. It can be seen that both the event flights $\{136, 137\}$ exceed the novelty threshold, but also that the preceding five flights $\{131\dots135\}$ exceed the threshold.

An advantage of the probabilistic-based model is in the setting of the novelty threshold (via EVT), shown as a thick line for the 2-D example model, which occurs at $P(x) = 0.99$. This can be set automatically at run-time, in contrast to the heuristic setting required for the distance-based model.

Both models (distance-based and probability-based) provide early indication of eventual engine events, while the latter does so with automated training and a principled probabilistic approach.

3.3 On-Line Novelty Detection

On-line novelty detection typically takes place “on-wing”. In this investigation, we form a shaft-specific pattern [54] at the sampling rate of the acquisition system, consisting of fundamental and harmonic tracked orders relating to the shaft being modelled:

$$[F, 1H, 1.5H, 2H, 3H, 4H, 5H, RE] \quad (16)$$

where “F” is the amplitude of the fundamental tracked order, “nH” is the amplitude of the nth harmonic tracked order, and “RE” is “residual energy”, defined to be broadband energy unassigned to the fundamental tracked order, or its harmonics (and is thus “residual”).

By considering the specific design of the engine, non-integer multiples of the shaft frequencies that reflect internal gearing configurations may provide specific information about the state of internal bearings in the engine.

As before, each element in the pattern is component-wise normalised to ensure that each varies over a similar dynamic range. The same training and test sets used in off-line monitoring were used to train both distance-based and probabilistic models, as described in Sect. 2.4.

Here, however, the full range of test data is available for training, rather than a summary vibration signature. In order to ensure that the training process is tractable in real-time, the number of data points in the training

set is reduced using the batch k -means algorithm. The training set, consisting of patterns derived from all data in tests $\{1..80\}$, were thus summarised using $k = 500$ cluster centres, which was found to adequately characterise the distribution of normal data in the 8-dimensional feature space defined by (16).

Figure 5 shows novelty scores of in-flight data from three example flights, computed using a distance-based model of normality, constructed as described in Sect. 2.4, with thresholds set heuristically, to include all patterns from the training set as described in Sect. 2.5.

“Normal” flight 75 (Fig. 5a) has low novelty scores throughout the flight. Flight 131 (Fig. 5b) shows that this flight exceeds the novelty threshold for several transient periods. Event flight 137 (Fig. 5c) shows that this flight exceeds the novelty threshold for longer periods.

A probabilistic model was trained using the same training set, and thresholds set using EVT as described in Sects. 2.4 and 2.5.

Figure 6 shows probabilistic model output for the same three example flights, against visualisations using NeuroScale as described in Sect. 2.3. “Normal” flight 75 (Fig. 6a) has low unconditional probabilities $p(x)$ throughout the flight; the visualisation shows that flight data (grey) lies close to the model

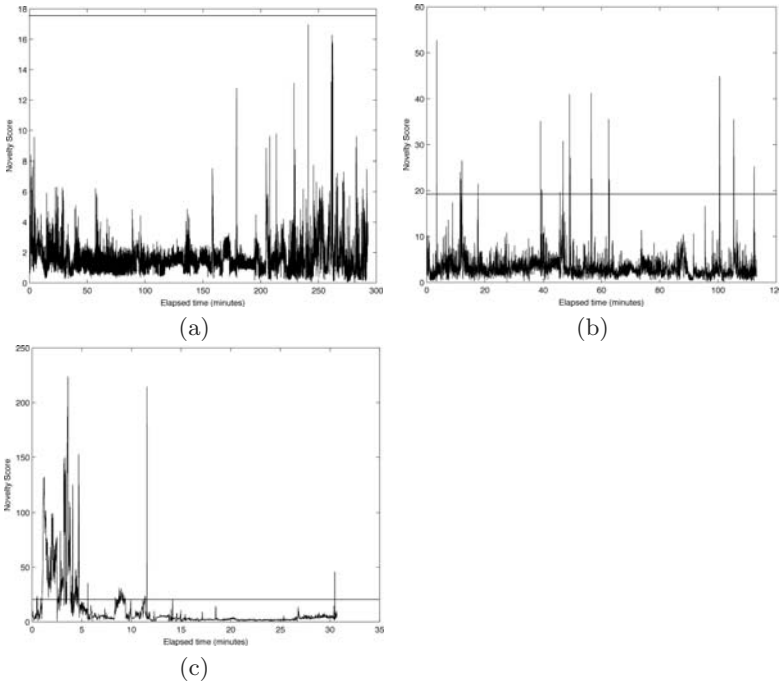


Fig. 5. In-flight novelty scores determined using distance-based model, against novelty threshold (horizontal line - using the same threshold in each case, with varying y-axis). (a) Flight 75, from the training set. (b) Flight 131, during a change in engine condition. (c) Flight 137, an event flight

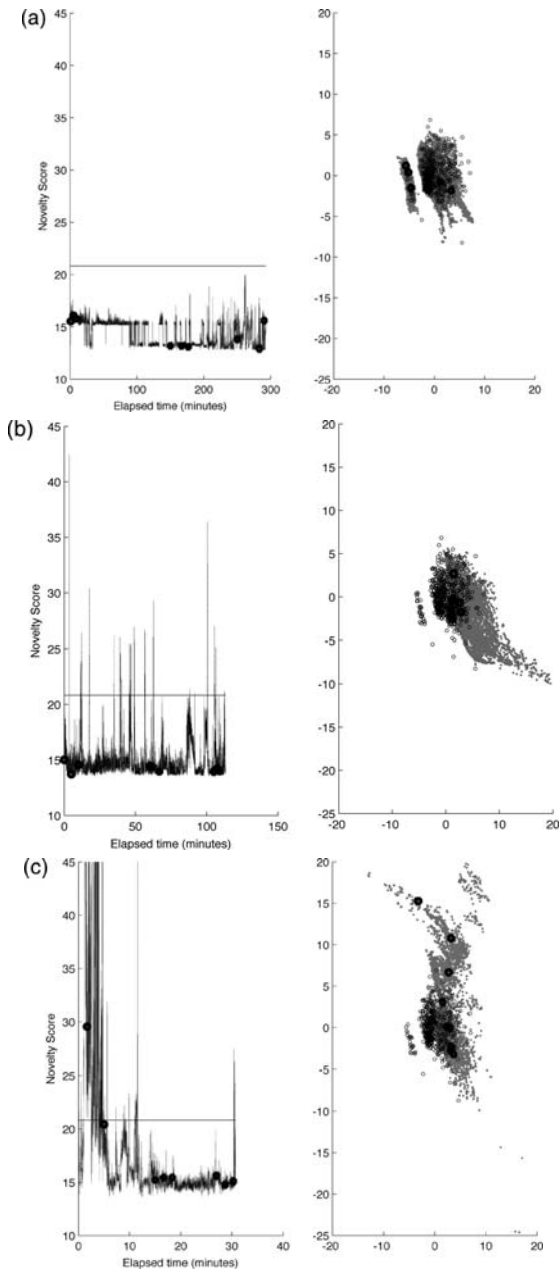


Fig. 6. *Left sub-plots:* in-flight data probabilities $p(x)$ determined using probabilistic model, against novelty threshold. *Right sub-plots:* visualisation of in-flight (grey) against model kernel centres (black). From top to bottom: (a) Flight 75, from the training set. (b) Flight 131, during a change in engine condition. (c) Flight 137, an event flight

kernel centres (black). Flight 131 (Fig. 6b) shows that this flight exceeds the novelty threshold for several transient periods, and that the visualised flight data begin to drift outside the locus of the model kernel centres. Event flight 137 (Fig. 6c) shows that this flight exceeds the novelty threshold for long periods, and that the visualised flight data lie far from the model kernel centres.

Thus, for both models, the event data have been correctly classified as “abnormal” with respect to the models, as has the change in engine condition during flight 131, potentially providing five flights of early warning of eventual failure. Flights {1...130} (not shown here) were below the novelty threshold, indicating that no false alarms would have been generated during “normal” operation.

The visualisation correspondingly shows that flight data drift further from the model kernel centres as abnormality is observed, providing a useful method of communicating the system state to a user.

3.4 Discussion

Off-line and on-line analysis techniques for novelty detection in vibration data of modern jet engines have been presented, indicating that similar techniques can be used in both contexts. In both cases, distance-based and probabilistic models of normality have been shown to provide early warning up to five flights prior to an engine event (where conventional techniques detected only the event itself).

While the distance-based models of normality must be constructed (and novelty threshold set) in a heuristic manner, EVT has shown to be able to set the novelty threshold automatically, such that unsupervised model construction can take place. This is particularly advantageous in the condition monitoring of high-integrity systems, in which heuristic, manual modelling is unattractive (due to the number of systems that must be monitored) and often impractical (because models must be formed during operational use).

Visualisation techniques have been demonstrated to be able to provide meaningful interpretation of system condition from flight-to-flight (for off-line novelty detection) and in-flight (for on-line novelty detection). Such techniques allow system output to be made interpretable by users that might be potentially unfamiliar with pattern recognition methods.

4 Combustion Data Analysis

In this section, we introduce the concept of combustion instabilities, and briefly describe the combustor used for data collection. We describe a method of extracting features from time-series data, and present results of using both SVM distance-based and probabilistic methods of novelty detection. We show that early warning of combustor instability is possible using these techniques, and show the advantage of novelty score calibration.

4.1 System Description

Instabilities in combustion often originate from the resonant coupling between the heat released by the flame and the combustor acoustics, and hence are referred to as “thermo-acoustic” instabilities. The thermo-acoustic instabilities generate increased noise level (high acoustic pressure fluctuations), which in turn lead to excessive mechanical vibrations of the walls of the combustion chamber, and even cause catastrophic failure of the combustor. These instabilities also feature sub-optimal burning, increased emissions, and decreased equipment life. Thus, there is a need to predict the occurrence of combustion instabilities, in order to minimise their effect by imposing appropriate control systems. Readers are directed to [55] and [56] for detailed descriptions of the physical processes involved in combustion instabilities.

The data set used by the investigation described in this section was generated by a Typhoon G30 combustor (a component of gas-turbines), manufactured by Siemens Industrial Turbomachinery Ltd. The system was operated in stable and unstable modes of combustion. Unstable combustion is achieved by increasing fuel flow rates above some threshold, with constant air flow rate. A detailed description of the combustor can be found in [57].

The data set consists of three channels (with sampling frequency of 1 KHz), which are:

1. Gas pressure of the fuel methane (CH_4) in the main burner
2. Luminosity of C_2 radicals in the combustion chamber
3. Global intensity of unfiltered light in the combustion chamber

4.2 Pre-Processing and Feature Extraction

Each channel was normalised using the component-wise method described in Sect. 2.2.

Wavelet analysis [58] was used to extract features from each channel. Wavelet analysis represents a function in terms of basis functions, localised in both location and scale. It is capable of revealing behavioural trends or transient periods within the data. Wavelet decomposition can be regarded as a multi-level or multi-resolution representation of a function $f(t)$, where each level of resolution j (except the initial level) consists of wavelets $\Psi_{j,k}$, with the same scale but differing locations k . Wavelet decomposition of a function $f(t)$ at level J can be written as

$$f(t) = \sum_k \lambda_{J,k} \Phi_{J,k}(t) + \sum_{j=J}^{+\infty} \sum_k \gamma_{j,k} \Psi_{j,k}(t), \quad (17)$$

where $\Phi_{J,k}(t)$ are scaling functions at level J , and $\Psi_{j,k}(t)$ are wavelets functions at different levels j . $\lambda_{J,k}$ are scaling coefficients or approximation coefficients at level J . The set of $\gamma_{j,k}$ are wavelet coefficients or detail coefficients at different levels j .

Mallat [59] developed a filtering algorithm for the *Discrete Wavelet Transform*. Given a signal s , wavelet decomposition of the first level produces two sets of coefficients: approximation coefficients λ_1 and detail γ_1 , by convolving s with a low-pass filter $h(k)$ for approximation, and with a high-pass filter $g(k)$ for detail, followed by dyadic decimation (down-sampling). Wavelet decomposition of the next level splits the approximation coefficients λ_1 in two parts using the same scheme, replacing s by λ_1 , and producing λ_2 and γ_2 , and so on.

The energy e_j of the wavelet detail coefficients $\gamma_{j,k}$ within a window of data at level j reflects the average noise level of the signal within that window [60]. Coefficient energy within the window is defined to be

$$e_j = \frac{\sum_k \gamma_{j,k}^2}{L} \quad (18)$$

for window length L , and wavelet detail coefficients $\gamma_{j,k}$ at level j .

Following [61], we divide the data set into non-overlapping windows of length $L = 64$, decomposed using the Daubauchies-3 wavelet. The mean values of approximation coefficients λ_1 and the energy of the detail coefficients γ_1 (extracted at level $j = 1$) are used as two-dimensional features in the input space, for each channel.

The combustor was operated in stable mode for the duration of 40 windows, followed by a transition into unstable operation for the duration of 48 windows, the first five of which were “transient”, being neither stable nor unstable.

4.3 On-Line Novelty Detection

Both distance-based (using SVM) and probabilistic models of normality were constructed for each channel independently (as described in Sect. 2.4). 80% of the data from stable operation (32 windows) were used to construct models of normality, with all remaining data (56 windows) used as a test set.

The SVM method results in novelty scores computed using distance-based methods in the transformed feature space, which are calibrated into probabilities [0 1] using the method described in Sect. 2.5. An example of calibration for output of the classifier trained using data from channel 2 is shown in Fig. 7. Novelty scores of “normal” data (i.e., windows during which combustion was stable) are calibrated to probabilities $p(x) \approx 0$, while those of “abnormal” data (i.e., windows in which combustion was unstable) have $p(x) \approx 1$. Windows from the period of transient operation between stable and unstable modes are shown as taking calibrated probabilities $p(x) \approx 0.8$ in this case.

Figure 8a shows channel 1 SVM model uncalibrated output as a contour plot. Corresponding calibrated probabilities are shown in Fig. 8b.

It can be seen that data from the training set, windows $\{1..36\}$, are deemed “normal”, with $p(x) \approx 0$, noting here that the probability of abnormality $p(x') = 1 - p(x)$ is plotted (and thus $p(x') = 1$ corresponds to an abnormal

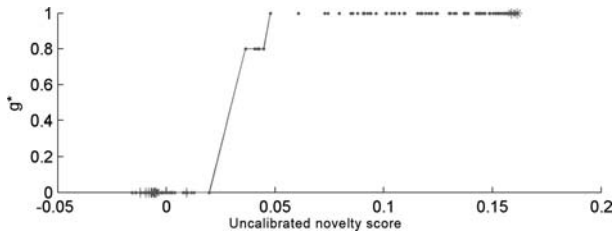


Fig. 7. Calibrating SVM novelty scores into probabilities, using SVM model trained using data from channel 2

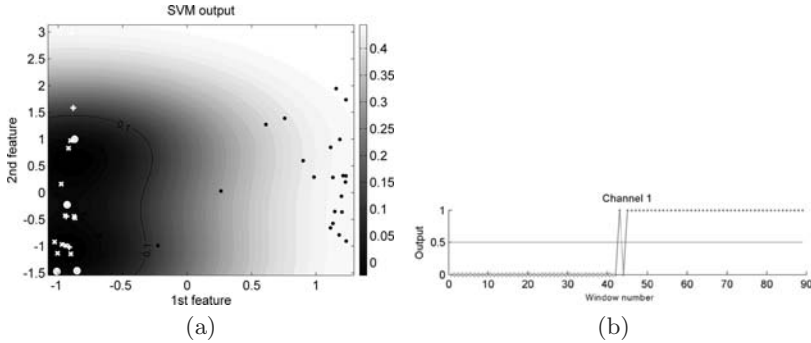


Fig. 8. (a) Visualisation of channel 1 SVM distance-based model, with novelty scores shown as greyscale contours. Training data are white \times , with support vectors shown as white \otimes . Test normal data are shown as white $+$, test abnormal data as black \bullet . (b) Calibrated output probabilities from channel 1 SVM model $p(x')$

window). Windows $\{41...45\}$ are transient, between normal and abnormal, and show corresponding oscillation in the calibrated output probabilities. Window 42 is the first classified as abnormal, which provides three windows early warning prior to the first unstable window (46), after which all windows are abnormal. The visualisation shows this clear separation between normal and abnormal data, again providing a meaningful graphical interpretation of system state. Similar results were obtained for channels 2 and 3 (not shown here).

Figure 9a shows channel 1 probabilistic model output as a contour plot. Corresponding probabilities are shown in Fig. 9b.

Here, the true $p(x)$ of the unconditional data density is shown, in which $p(x) > 0$ for normal data, and $p(x) \approx 0$ for abnormal data. The figure shows that most normal data take non-zero probabilities, but there is a false-positive for window 37 (the first window after the 36 windows in the training set), which is incorrectly classified “abnormal”. The transient data, windows $\{41...45\}$ show decreasing probabilities, but only windows 43 and 45 have $p(x) \approx 0$. All unstable data, windows $\{46...88\}$, are correctly assigned outputs $p(x) \approx 0$.

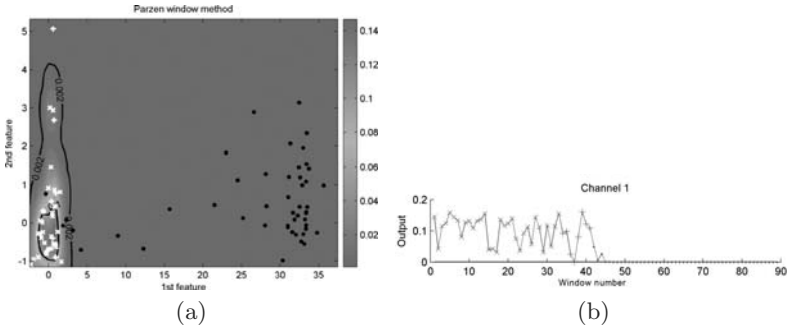


Fig. 9. (a) Visualisation of channel 1 probabilistic model, with novelty scores shown as greyscale contours. Training data are white \times . Test normal data are shown as white $+$, test abnormal data as black \bullet . (b) Output probabilities from channel 1 probabilistic model $p(x)$

This is confirmed by the visualisation, in which some of the test data (shown as black \bullet) fall within the locus of the normal data, though separation to most abnormal data is significant. Similar results (not shown here) were obtained for channels 2 and 3.

4.4 Discussion

On-line analysis of combustion data has shown that early warning of system instability can be provided, by correctly classifying transient periods of operation as “abnormal” with respect to normal data. SVM distance-based models were found to outperform the probabilistic models investigated, by providing early warning of unstable operation without false-positive activations during normal combustor operation. The disadvantages of distance-based methods, in which thresholds are set heuristically, is overcome by calibrating outputs into probabilities, such that a meaningful probabilistic threshold may be set.

5 Conclusion

This chapter has compared the perform of distance-based and probabilistic methods of constructing models of normality using both case-mounted vibration sensors, and single-component monitoring (of the combustor).

Distance-based models have been shown to provide early warning of engine events, but require heuristic setting of thresholds, and manual construction of models. Typically, this can become impractical for condition monitoring of high-integrity systems, in which the number of units being monitored, and the inaccessibility of those units, require unsupervised learning of normal models.

Methods of overcoming this have been investigated, in conjunction with an SVM model, in which classifier output is calibrated into probabilities, resulting in a probabilistic novelty threshold being set automatically.

Probabilistic models have been shown to provide similar early warning, with novelty thresholds set automatically using EVT, which explicitly models the tails of the “normal” data density.

The use of visualisation techniques has been shown to provide guidance during the data evaluation and model construction phases, particularly in the verification of data labels provided by domain experts, which can be unreliable. The same techniques are shown to be able to provide meaningful representations of system state, allowing non-expert users to interpret the output of the novelty detection system.

Further analysis of fusing classifications from multiple channel-specific classifiers is on-going, as is the exploitation of automatic probabilistic threshold-setting techniques for estimates of a multi-variate data density.

Acknowledgements

The authors gratefully acknowledge discussions with Nicholas McGrogan of Oxford BioSignals Ltd.; Dennis King, Steve King, and Paul Anuzis of Rolls-Royce PLC; Hujun Yin and Yang Zhang of the University of Manchester. DAC and PRB acknowledge the support of the HECToR research project (a UK Department of Trade and Industry project), Rolls-Royce PLC., and Oxford BioSignals Ltd.

References

1. Roberts S, Tarassenko L (1994) *Neural Comput* 6:270–284
2. Silverman BW (1986) *Density estimation for statistics and data analysis*. Chapman and Hall, London
3. Mayrose I, Friedman N, Pupko T (2005) A Gamma Mixture Model Better Accounts for Among Site Heterogeneity. *Bioinformatics* 21(2):151–158
4. Agusta Y, Dowe DL (2003) Unsupervised Learning of Gamma Mixture Models Using Minimum Message Length. *Artificial intelligence and applications proceedings* 403
5. Dempster AP, Laird NM, Rubin DB (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *J R Stat Soc Series B* 39:1–38
6. Markou M, Singh S (2003) Novelty Detection: A Review. *Signal Processing* 83:2481–2497
7. Duda RO, Hart PE, Stork DG (2001) *Pattern classification*. Wiley, New York
8. Yeung DY, Ding Y (2002) Host-Based Intrusion Detection Using Dynamic and Static Behavioral Models. *Pattern Recognit* 36:229–243
9. Smyth P (1994) Markov Monitoring with Unknown States. *IEEE J Sel Areas Commun* 12(9):1600–1612
10. Quinn J, Williams CKI (2007) Known Unknowns: Novelty Detection in Condition Monitoring. *Proceedings of 3rd Iberian conference on pattern recognition and image analysis, Lecture Notes in Computer Science*, Springer
11. Markou M, Singh S (2006) A Neural Network-Based Novelty Detector for Image Sequence Analysis. *IEEE Trans Pattern Anal Mach Intell* 28(10):1664–1677

12. Ghahramani Z, Hinton GE (1998) Variational Learning for Switching State-Space Models. *Neural Comput* 12(4):963–996
13. McSharry PE, He T, Smith LA, Tarassenko L (2002) Linear and non-linear methods for automatic seizure detection in scalp electro-encephalogram recordings. *Med Biol Eng Comput* 40:447–461
14. Tax DMJ, Duin RPW (1998) Outlier detection using classifier instability. *Advances in pattern recognition—the joint IAPR international workshops, Sydney, Australia*, 593–601
15. Kohonen T (1982) Self-Organized Formation of Topologically Correct Feature Maps. *Biol Cybern* 43:59–69
16. Ypma A, Duin RPW (1998) Novelty Detection Using Self-Organising Maps. *Prog Connect Based Inf Syst* 2:1322–1325
17. Labib K, Vemuri R (2002) NSOM: A real-time network-based intrusion detection system using self-organizing maps. *Networks security*
18. Yin H, Allinson NM (2001) Self-organizing mixture networks for probability density estimation. *IEEE Trans Neural Netw* 12(2)
19. Vapnik V (2000) *The nature of statistical learning theory*. Second Edition. Springer, Berlin New York Heidelberg
20. Tax DMJ, Duin RPW (1999) Data Domain Description Using Support Vectors. *Proceedings of ESAN99. Brussels*:251–256
21. Scholkopf B, Williamson R, Smola AJ, Shawe-Taylor J, Platt J (2000) Support vector method for novelty detection. *Advances in neural information processing systems 12, (NIPS99)* Solla KMSA, Leen TK (eds.), MIT: 582–588
22. Bishop CM (1995) *Neural networks for pattern recognition*. Oxford University Press, Oxford
23. Jennions IK (2006) Cross-platform challenges in engine health management. *Proceedings of International Conference on Integrated Condition Monitoring, Anaheim, CA*
24. Moya M, Hush D (1996) *Neural Netw* 9(3):463–474
25. Ritter G, Gallegos M (1997) *Pattern Recogn Lett* 18:525–539
26. Clifton DA, Bannister PR, Tarassenko L (2006) Learning shape for jet engine novelty detection. In: Wang J. et al. (eds.): *Advances in neural networks III. Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 3973:828–835
27. Sammon JW (1969) *IEEE Trans Comput* 18(5):401–409
28. DeRidder D, Duin RPW (1997) *Pattern Recogn Lett*
29. Lowe D, Tipping ME (1996) *Neural Comput Appl* 4:83–95
30. Tarassenko L (1998) *A guide to neural computing applications*. Arnold, UK
31. Nabney I (2002) *Netlab: algorithms for pattern recognition*. Springer, Berlin Heidelberg New York
32. Clifton DA, Bannister PR, Tarassenko L (2007) Visualisation of jet engine vibration characteristics for novelty detection. *Proceedings of NCAF, London, UK*
33. Nairac A, Townsend N, Carr R, King S, Cowley P, Tarassenko L (1999) *Integr Comput-Aided Eng* 6(1):53–65
34. Clifton DA, Bannister PR, Tarassenko L (2006) Application of an intuitive novelty metric for jet engine condition monitoring. In: Ali M, Dapoigny R (eds) *Advances in applied artificial intelligence. Lecture Notes in Artificial Intelligence*. Springer, Berlin Heidelberg New York 4031:1149–1158

35. Clifton DA, Bannister PR, Tarassenko L (2007) A framework for novelty detection in jet engine vibration data. In: Garibaldi L, Surace S, Holford K (eds) *Key engineering materials* 347:305–312
36. Clifton DA, Bannister PR, Tarassenko L (2007) Novelty detection in large-vehicle turbochargers. In: Okuno HG, Ali M (eds) *New trends in applied artificial intelligence. Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 4750
37. Hayton P, Scholkopf B, Tarassenko L, Anuzis P (2000) Support vector novelty detection applied to jet engine vibration spectra. *Proceedings of Neural Information Processing Systems*
38. Wang L, Yin H (2004) Wavelet analysis in novelty detection for combustion image data. *Proceedings of 10th CACSC*, Liverpool, UK
39. Clifton LA, Yin H, Zhang Y (2006) Support vector machine in novelty detection for multi-channel combustion data. *Proceedings of 3rd International Symposium on Neural Networks*
40. Nairac A, Corbett-Clark T, Ripley R, Townsend N, Tarassenko L (1997) Choosing an appropriate model for novelty detection. *Proceedings of IEE 5th International Conference on Artificial Neural Networks*
41. Tax D, Duin R (1999) *Pattern Recogn Lett* 20:1191–1199
42. Schölkopf B, Platt J, Shawe-Taylor J, Smola AJ, Williamson RC (2001) *Neural Comput* 13(7):1443–1471
43. Bishop CM (1994) Novelty detection and neural network validation. *Proceedings of IEE Conference on Vision and Image Signal Processing*
44. Zadrozny B, Elkan C (2002) Transforming classifier scores into accurate multiclass probability estimates. *Pro. ACM SIGKDD* 694–699
45. Roberts SJ (1999) *Proc IEE* 146(3)
46. Roberts SJ (2000) *Proc IEE Sci Meas Technol* 147(6)
47. Medova EA, Kriacou MN (2001) Extremes in operational risk management. Technical report, Centre for Financial Research, Cambridge, U.K.
48. Fisher RA, Tippett LHC (1928) *Proc Camb Philos Soc* 24
49. Coles S (2001) An introduction to statistical modelling of extreme values. Springer, Berlin Heidelberg New York
50. Embrechts P, Kluppelberg C, Mikosch T (1997) *Modelling extremal events*. Springer, Berlin Heidelberg New York
51. Rolls-Royce PLC (1996) *The jet engine*. Renault Printing, UK
52. Hayton P, Utete S, Tarassenko L (2003) QUOTE project technical report. University of Oxford, UK
53. Clifton DA (2005) Condition monitoring of gas-turbine engines. Transfer report. Department of Engineering Science, University of Oxford, UK
54. Bannister PR, Clifton DA, Tarassenko L (2007) Visualization of multi-channel sensor data from aero jet engines for condition monitoring and novelty detection. *Proceedings of NCAF*, Liverpool, UK
55. Khanna VK (2001) A study of the dynamics of laminar and turbulent fully and partially premixed flames. Virginia Polytechnic Institute and State University
56. Lieuwen TC (1999) Investigation of combustion instability mechanisms in premixed gas turbines. Georgia Institute of Technology
57. Ng WB, Syed KJ, Zhang Y (2005) Flame dynamics and structures in an industrial-scale gas turbine combustor. *Experimental Thermal and Fluid Science* 29:715–723

58. Daubechies I (1988) Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics* 41:909–996
59. Mallat SG (1989) A theory for multiresolution signal decomposition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 11(7):674–693
60. Guo H, Crossman JA, Murphey YL, Coleman M (2000) *IEEE Trans Vehicular Technol* 49(5):1650–1662
61. Clifton LA, Yin H, Clifton DA, Zhang Y (2007) Combined support vector novelty detection for multi-channel combustion data. *Proceedings of IEEE ICNSC*