Novelty Detection: A Review
Part 2: Neural network based approaches

Markos Markou and Sameer Singh
PANN Research, Department of Computer Science
University of Exeter, Exeter EX4 4PT, UK
{m.markou, s.singh}@ex.ac.uk

Abstract

Novelty detection is the identification of new or unknown data or signal that a machine learning system is not aware of during training. In this paper we focus on neural network based approaches for novelty detection. Statistical approaches are covered in part-I paper.

1. Introduction

Neural networks have been widely used for novelty detection. In this paper we detail a variety of neural network methods for novelty detection. Our emphasis is on the technique rather than the application itself. Compared to statistical methods, some issues for novelty detection are more critical to neural networks such as their ability to generalise, computational expense while training and further expense when they need to be retrained. In this vein, some networks are better suited than others, however, with a lack of enough comparative studies and meta-analysis of their novelty detection performance and its relationship to the type and quality of data used, it is hard to give a subjective view except than simply present a broad review of studies in the area.

The main criteria for evaluating novelty detection is the maximisation of detecting true novel samples while at the same time minimising the false positives. A commonly used method for this is the ROC analysis. Some other metrics of performance have also been suggested. For example, Moya *et al.* (1993) provides three generalisation criteria that can be used to assess the performance of a novelty detector. Most pattern classification algorithms and neural networks fail at automatic detection of novel classes because they are discriminators rather than detectors. They often use open decision boundaries, such as hyper-planes, to separate targets from each other and fail to decide when a feature set does not represent any known class. The performance of such detectors or one-class classifiers can be measured using three generalisation criteria. First, within-class generalisation indicates the network's performance on non-trained known classes. Second, between-class generalisation indicates the performance on near-known class objects from other classes. Finally, out-of-class generalisation indicates the classifiers' performance on unknown classes.

The computational complexity of neural networks has always been an important consideration for practical applications. One important consideration with neural networks is that they cannot be as easily retrained as statistical models. Retraining does not necessarily need to be applied when new class data has to be added to the training set. In some cases, when the training data no longer reflects the environmental conditions. In such cases the network selects the input for retraining from the new environment automatically. Such technique is very useful in applications such as video processing where the same object might gradually change during operation due to different lighting conditions, exposure times and other reasons (Doulamis *et al.,* 2000). Similarly, Zhang and Veenker (1991) introduce a new active learning paradigm enabling a neural network to adapt actively to its environment by self-generating novel training examples during learning using genetic algorithms. New training examples are generated by genetic recombination of two parent examples of the existing training set.

Retraining of networks after novelty detection with an enlarged training data set deserves important consideration. Some networks such as constructive neural networks are capable of on-line adaptation of links and weights as new classes are added. Network types such as cascade correlation are better suited to adaptation compared to multi-layer perceptrons. In the case of multi-layer perceptrons, retraining implies the addition of new output and hidden nodes and often it is not clear how best to train the new configuration.

Some of the important considerations in retraining is that the experimenter often does not wish to retrain a system from scratch and some form of incremental training is attractive. A limited number of approaches have been proposed in this context. Kwok and Yeung (1999) address the very important issue of retraining a neural network using some objective functions after new hidden units have been added. A standard neural network is generally useful only if the user chooses its architecture correctly. A small network will have difficulties in learning while a large network my result in over-generalisation and poor performance. This is particularly true in adaptive learning when new classes might be added to the system and the size of the hidden layer might also need updating. In general, algorithms that automatically determine the correct network architecture are highly desirable in several applications. There are three major approaches to tackle this problem; regularisation algorithms, pruning algorithms, which start with a large network size and remove nodes during training that are not active, and constructive algorithms which start with a small number of hidden units and add units until a satisfactory solution is found. Obviously, the constructive algorithms have several advantages over pruning algorithms. First, it is easier to decide on the initial network with the constructive approach whereas with pruning one does not know how big the initial network should be. Then it is much faster to train smaller networks so the constructive approach will lead to smaller networks capable for learning the problem faster. This paper is concerned with this approach. There are three major problems involved in the design of constructive algorithms. How to connect the new units to the existing network? How to train the new units without loosing the knowledge captured by the rest of the network. When to stop adding units to the network? This paper is concerned with the learning of the new units with computational efficiency in both time and space. A common technique used to improve computational efficiency is to assume that the nodes that already exist in the network are useful in modelling part of the target function. Therefore, these nodes can be frozen and update only the new nodes in the iterative training procedure. Training can be performed using the backpropagation algorithm but a further improvement in terms of computational efficiency can be achieved by proceeding in a layer-by-layer manner. First the weights feeding into the new hidden units are training and then the weights feeding into the output nodes while keeping the input weights constant. This way there is no need to backpropagate the error signals therefore is much faster. During input training, the weights feeding into the new hidden units are trained to optimise an objective function. The authors present four such objective functions and provide proof of their convergence ability. Additionally, these objective functions can be computed in $O(N)$ time where N is the number of training patterns. The proposed objective functions include: (a) projection index that finds "interesting" projections deviating from the Gaussian form; (b) the same error criterion, such as the squared error criterion, is used in output training; (c) the covariance between the residual error and the new hidden unit activation, as in the cascade-correlation network and its variants; and (d) an objective function based on the use of projection matrices although this has a computational and storage requirements of $O(N^2)$. The objective functions were experimentally tested on a set of synthetic data and gave very satisfactory results.

Singh and Markou (2003) present a method of creating new feed-forward networks for new class samples while keeping the earlier trained network on previously known classes. In this manner, as new novel classes are discovered, new networks are created which is computationally efficient. A new test sample is presented to all available networks and its output is thresholded to determine which class it belongs to. Low outputs on all networks signal a novel class sample.

Despite difficulties with neural network retraining and vast amount of parameter settings, neural networks are important novelty detectors. The following sections detail some important types of neural networks that have been used as novelty detectors.

2. Neural Network Approaches

Neural networks have been used extensively in novelty detection. They have the advantage that a very small number of parameters need to be optimised for training networks and no *a priori* assumptions on the properties of data are made. Here we review a number of different architectures and methods that have been used for novelty detection. These include multi-layer perceptrons, self organising maps, radial basis function networks, support vector machines, hopfield networks, oscillatory networks, etc. The number of studies that use a different neural method of novelty detection in addition to the above are fairly limited. Examples of such work include a three layer network trained with Widrow-Hoff LMS associative learning rule (Lewis and Simo, 2001), and a hardware amenable restricted Boltzmann machine (Murray, 2001).

2.1 *MLP approaches*

Multi-layer perceptrons are the best known and most widely used class of neural networks. Since such networks do not generate closed class boundaries, devising methods of novelty detection is a fairly challenging task specially by ensuring that the generalisation property of the network does not interfere with its novelty detection ability (Moya *et. al.*, 1993). A variety of approaches have been proposed as discussed below.

In some studies, parametric statistics has been used for novelty detection by post-processing ordinary neural network output data. Bishop (1994) states that one of the most important sources of errors in neural networks arises from novel input data. A network, which is trained to discriminate between a number of classes coming from a set of distributions, will be completely confused when confronted with data coming from an entirely new distribution. It is necessary for most applications for the system to output along with the classification of a data input, a measure of confidence of this decision or to 'refuse' to make a decision if the data point is found to come from a completely new class. The novelty detection technique is implemented here by estimating the density of the training data thus modelling its distribution and checking whether an input data point comes from this distribution. The goal of network training is to find a good approximation to the regression by minimization of a sum-of-squares error defined over a finite training set. It is expected that this approximation will be most accurate in regions of input space for which the density is high, since only then does the error function penalize the network mapping if it differs from the regression. This is why the unconditional density might give an appropriate quantitative measure of novelty with respect to the training data. If the data point falls in a region with high density then the network is likely to perform well. If the data point falls in a region with low density then it is likely that the data point comes from a class that is not represented by the training data and it is likely that the network will perform poorly. This can be used to assign error bars to the network outputs or by placing a threshold; patterns that fail the threshold may be rejected and classified by other means. The density estimation is done either by using a kernel-based estimator or by using a semi-parametric estimator constructed from a Gaussian Mixture Model. The author states that it is important that density estimation is done on the input data, before any pre-processing techniques take place.

Desforges *et al.* (1998) considers probability density estimation of neural network output as a measure of novelty similar to Bishop (1994). Probability density functions describe the frequency of occurrences of an observation that occur in any point within a domain of interest. The Epanechnikov kernel is used in this work and smoothing parameter *h* is calculated using the least-squares cross-validation method. This study pays special attention to the dimensionality of the data. The most pronounced difficulty in treating high-dimensional data is in the importance that must be accorded to the tails of a distribution. As the dimensionality increases, the relative quantity of data associated with the relatively low-density tails grows. Hence, even very low-density regions must be regarded as important parts of the distribution. This makes the decision, whether a test vector belongs to the distribution or it is novel, a lot more difficult. The number of data points required for an accurate estimate of the density of the data increases as a power of the number of dimensions. For large number of dimensions the amount of training data required might be prohibitive to use for novelty detection. In this work, dimensionality reduction was achieved through data compression using wavelets. A small number of wavelet coefficients were selected to represent the data using Genetic Algorithms. The relative suitability of a subset of coefficients was evaluated on their classification power using radial basis function network. The model showed very good approximations of the underlying distributions. For novelty detection, the application of such technique is very simple. Given a new set of data, the probability of the data corresponding to a set of conditions to a set of conditions for which a density function is available may be evaluated. The returned value represents the scaled probability of the new data corresponding to the original operating conditions.

One of the simplest approaches to novelty detection is based on thresholding output of a neural network. Low confidence indicates novel sample. Ryan *et al.* (1998) present a novelty detection method using neural networks, applied to the detection of illegal use of computer resources. The Neural Network Intrusion Detection (NNID) anomaly detection system is based on identifying a legitimate user based on the distribution of commands she or he executes. After the data is collected, a backpropagation neural network is trained to identify each user based on the training data. An anomaly (novelty) is detected when the neural network places low confidence in its decision. When the maximum activation is below 0.5, a novelty is detected. A similar approach is adopted by Augusteijn and Folkert (1999). For novelty detection it is sufficient to place a threshold on the output values of the network and either take the Euclidean distance between the output pattern and the target pattern and threshold that or threshold the highest output value. This threshold is user set.

LeCun *et al.* (1990) discuss a method of novelty detection on the handwritten character recognition problem using MLP/backpropagation. The rejection criterion was based on three conditions. The activity level of the winning node should be larger than a given threshold T_1, the activity level of the second winning node should be lower than a threshold T_2 and the absolute difference between T_1 and T_2 should be larger than a threshold T_d. All three thresholds are user-defined and in this study optimized using performance measures on the test set.

Vasconcelos (1995) makes an important contribution to novelty detection in MLP by suggesting how to construct closed class boundaries. Such networks tend to classify patterns that do not belong to any of the known classes to one of those classes with a high degree of confidence. The reason for this is that they tend to separate the training classes using hyper-planes forming open boundaries between the classes instead of around the classes (Moya, 1993; Bishop, 1994). One of the first approaches followed to deal with the problem of spurious patterns is to train the classifier with 'negative' examples of random patterns. The objective is to create attractors in the pattern space representing the 'rejection' class so that patterns, which do not belong to one of the known classes, will fall in this 'rejection' class. According to the author, this technique will fail because it is unrealistic to expect that randomly selected patterns will accurately represent the input space where novel patterns will fall. There is, however a similar approach based on bootstrapping that according to the authors works better. When a pattern is rejected by the trained network, based on thresholding its output values, with high degree of confidence then it is assumed that this decision is in fact correct and the pattern is used as a negative example to retrain the network to reinforce its decision. The rejection occurs if the responses of all output neurons are close to 0 or the response of more than one neuron is close to 1. The target output of a rejection pattern is assigned as all zeros.

Vasconcelos et al. (1995) study three feedforward neural networks and compare their ability to deal with the rejection of novel data. These are the standard MLP network, an MLP that employs a Gaussian activation function (GMLP) and the radial basis function (RBF). It is also shown how the MLP can be modified to generate boundaries surrounding the training data to enhance reliability using randomly generated reject class data. The GMLP is an alternative to the MLP in which the sigmoid activation is replaced by Gaussian. The motivation behind this is that in the use of the Gaussian function, the receptive field of each network's unit corresponds to a hyper-hill in the pattern space which "prunes" the unit to respond only to part of the half space causing more confined regions surrounding the training data. This situation can be considered more reliable for rejecting spurious patterns than that obtained with the standard MLP, especially when there is an increasing number of training classes present in the problem. In contrast with both the MLP and GMLP, each hidden unit in the RBF network responds to a localized receptive field in the input space formed by the combination of a Euclidean distance measure as the propagation rule and a Gaussian as the network's activation function. As a result, the network's output reaches its maximum when the input pattern is near a centroid and decreases monotonically when it becomes more distant from the centroid. Since the centroids are randomly selected from the input data and units respond positively only to a local area around the centroids, inputs very dissimilar from the training patterns tend to receive very low output. RBFs places closed decision boundaries around each class making them ideal for novelty detection and much better suited than MLPs or GMLPs for real practical applications.

Cordella *et al*. (1995) define a performance function *P* that takes into account the quality of a classifier in terms of its recognition, misclassification and reject rates. Under this assumption, the optimal reject threshold is the one for which *P* reaches its absolute maximum. After the training phase, the classifier is applied, without reject option to a set *S* of samples whose class is known and is representative of the training set and it is evaluated in terms of correctly classified and misclassified patterns. *S* is used to determine two reject thresholds optimal with respect to the assigned function *P* selecting the values that maximize it. This approach is independent of the network architecture and training method. The first threshold is applied to the winning node rejecting patterns whose activation falls below this value while the second threshold is applied on the difference between the activations of the winning and second winning nodes rejecting patterns that fall below it. The approach was tested on a neural classifier made of a three-level feed-forward fully connected network with sigmoid activations trained using the backpropagation algorithm. The objective was the recognition of unconstrained hand printed and multi-font printed characters. By using the technique proposed, a considerable reduction of the misclassification rate was obtained, at the expense of only a slight decrease of the recognition rate.

Cordella *et al.*,(1998) extend their previous work to other types of neural networks. The authors extend their work to include different types of neural networks such as the MLP, the RBF, the LVQ, the SOM, the ART, the PNN and the Hopfield network. The approach to the reliability problem (rejection of novel patterns) presented in this paper aims to be more general. A neural classifier is considered to be a black box, accepting an input pattern and supplying a vector with numbers as the output. No knowledge of training procedures or networks architectures is necessary. A pattern should be rejected if it is significantly different than the training data and/or it lies in the overlapping region of two or more classes. In the case of the Hopfield network, an approach similar to the autoassociator based novelty detection can be adopted (Bogacz *et al.*, 1999; Crook and Hayes, 2001; Crook *et al.*, 2002; Addison *et al.*, 2002). The rest of the neural networks can be grouped into three categories. The MLP and the RBF can be grouped together because their output indicates a class, the LVQ, SOM and ART together because their output is the distance of the pattern and its nearest prototype and the PNN on its own because its output is a probability. For all groups two criteria are defined namely ψ_a and ψ_b. Different ways of combining these two criteria to decide whether a pattern should be rejected are explored.

Stefano et al. (2000) also extend the technique described by Cordella *et al.*, (1995) to other types of neural network classifiers. In this paper, the method for determining the optimal threshold is generalised and rendered independent of the architecture of the considered classifier making it applicable to any type of classifier. The authors consider the MLP, the Learning Vector Quantisation (LVQ) and the Probabilistic Neural Network (PNN). Similar to Cordella *et al.*, (1995) the rejection is performed on the basis of the output vector given a threshold. The threshold is optimised on the basis of a function P that considers the costs associated with the classifier's correct recognition rate, rejection rate and misclassification rate. The authors use a more complicated and generic way of using a set S similar to the training set to determine the rejection thresholds. The optimised thresholds are used in the case of the MLP the same way as in Cordella *et al.*, (1995). For the LVQ and the PNN a slightly different approach is followed. In the case of the LVQ, the output vector is composed of the values of the distances between each Kohonen neuron (prototype) and the input sample. The final prototypes defined by the net will be the centroids of the regions into which the feature space is partitioned. Samples significantly different from those present in the training data will have a distance from the winning neuron greater than that relative to the samples in the training set. Therefore, a threshold can be placed on the quotient of that distance and the maximum distance in the training set. On the other hand, samples belonging to an overlapping region have a comparable distance from at least two prototypes. A second threshold can be placed on the quotient of the distance of the winning and second winning neurons. In a PNN, for each neuron k the output vector assumes a value proportional to the probability that the input sample belongs to the class associated with the kth neuron. The distances between the input sample x and all the samples belonging to the training set are computed and, on the basis of these values, the probability P_k that x belongs to each class k is evaluated. These probability density functions are generally computed using the Parzen method. Then the PNN assigns the input samples to the class associated to the output neuron with the highest value. Patterns are rejected using the same equations as in the LVQ case.

Wilson *et al*. (1995) demonstrate that an MLP can have as good or better novelty detection performance than competing techniques by making fundamental changes in the network's optimisation strategy. There are three changes necessary. First, regularisation can be used to decrease the volume of the weight space in the optimisation process. This is achieved by adding an error term that is proportional to the sum of the square of the weights. Second, we can change the usual sigmoidal activation function to a sinusoidal function. This creates a significant change in the dynamics of the training since even and odd higher derivatives of the dynamical system are never both small. This improves network training and dynamics and results in better error-reject performance and smaller networks. Third, Boltzmann pruning is used to reduce the weight space dimension and class based error weights are used during training.

Denouex (2000) presented a new adaptive pattern classifier based on the Dempster–Shafer theory of evidence. This method uses reference patterns as items of evidence regarding the class membership of each input pattern under consideration. This evidence is represented by basic belief assignments (BBA's) and pooled using the Dempster's rule of combination. This procedure can be implemented in a multilayer neural network with specific architecture consisting of one input layer, two hidden layers and one output layer. The weight vector, the receptive field and the class membership of each prototype are determined by minimizing the mean squared differences between the classifier outputs and target values. After training, the classifier computes for each input vector a BBA that provides a description of the uncertainty pertaining to the class of the current pattern, given the available evidence. This information may be used to implement various decision rules allowing for ambiguous pattern rejection and novelty detection. The outputs of several classifiers may also be combined in a sensor fusion context, yielding decision procedures that are very robust to sensor failures or changes in the system environment.

Singh and Markou (2003) present a new model for novelty detection using neural networks. They first use the concept developed by Vasconcelos *et al.* (1994, 1995) of using random rejects to close known class boundaries. Their rejection filter is used to discriminate between known and novel samples and only known samples are classified. The novel samples are accumulated and then clustered using fuzzy clustering. These clusters are then compared with known class distributions to check if they could be outliers of known classes or whether they represent truly novel patterns. Truly novel samples are then manually labelled as of a new class and the neural network is incrementally updated to learn this information. Their results are shown on natural scene analysis application where they show how novel objects can be picked up in video analysis.

2.2 *Support Vector Machines based approaches*

Support vector machines are based on the concept of determining optimal hyperplanes for separating data from different classes (Vapnik, 1998). Tax and Duin (1999a, 1999b) seek to solve the problem of novelty detection by distinguishing the class of objects that are represented by the training set and all other possible objects in the object space. A sphere is found that encompasses almost all points in the data set with the minimum radius. Slack variables are also introduced to deal with the problem of outliers in the data set. The radius and the number of slack variables are minimized for a given constant C that gives the trade off between the volume of the sphere and the number of target objects found. A given test point is rejected if its distance from the centre of the sphere is larger than the radius of the sphere. The usage of kernel functions as opposed to inner products solves the problem of non-spherical distributed data. They considered a polynomial and a Gaussian kernel and found that the Gaussian kernel works better for most data sets. A free parameter σ needs to be selected which defines the width of the kernel. The larger the width of the kernel, the less support vectors are selected and the description becomes more spherical like. The authors proposed a leave-one-out method on the training data for optimising σ and for monitoring the generalisation of the system. The advantage of this technique over other techniques for novelty detection such as Tarassenko (1999) is that it does not have to make a probability density estimation of the training data. A drawback of these techniques is, according to the authors, that they often require a large dataset, especially when high dimensionality feature vectors are used. Also, problems may arise when large differences in density exist. Objects in low-density areas will be rejected although they are legitimate objects.

Tax and Duin (2001) suggest creating outliers uniformly in and around the target class. The fraction of the accepted outliers by the classifier is an estimate of the volume of the feature space covered by the classifier and an optimisation of the parameters may be performed. The authors propose using a d-dimensional Gaussian distribution for creating the outlier data. The direction of the object vectors from the origin will not be changed, but they rescale the norm of the object vectors. The authors indicate that the method becomes infeasible in very high dimensional data especially when a hyper-box is defined to surround the target data. In this respect the method described here works better but both methods fail for data with more than 30 features.

Schölkopf *et al.* (2000) offer an alternative the approach used by Tax and Duin. The difference is that instead of trying to find a hyper-sphere with minimal radius to fit the data, here the authors try to separate the surface region containing data from the region containing no data. This is achieved by constructing a hyper-plane which is maximally distant from origin with all data points lying on the opposite side from the origin and such that the margin is positive. The paper proposes an algorithm that computes a binary function. The function returns +1 in “small” regions that contain data and –1 elsewhere. The data is mapped into the feature space corresponding to the kernel and is separated from the origin with maximum margin. Different kernels may be utilized corresponding to a variety of non-linear estimators. To separate the dataset from the origin a quadratic program (QP) needs to be solved. A variable *v* is introduced which takes values between 0 and 1 and controls the effect of outliers in the system or rather how hard or soft the boundary is around the data. The drawback of this method as mentioned by Campbell and Bennett (2001) is that the origin plays a crucial role. This is a disadvantage since the origin effectively acts as a prior for where the class abnormal instances are assumed to lie. The method is tested on both synthetic and real-world data. The experiment was performed on the USPS dataset of handwritten digits. A Gaussian kernel was used for training and the results showed that a number of outliers were in fact identified. Further criticism of this method is available in Manevitz and Yousef (2001).

A simpler method for novelty detection extending the work of Tax and Duin, and Schölkopf *et al.* is proposed by Campbell and Bennett (2001). This system is based on the statistical analysis of data. The data distribution of the data is modelled using a binary-valued function, which is positive in those regions of input space where most of the data lies and negative everywhere else. This is achieved by defining separating hyper-planes in features space that will be positive on the one side and negative on the other. A number of kernels may be used to construct such boundaries. The objective is to find a surface in input space that wraps around the data clusters. Anything outside this surface is considered as novel. This, according to the authors can be easily solved using linear programming. According to the authors, this approach overcomes the problem of the origin (Schölkopf *et al.*, 2000) because rather than repelling the hyper-sphere from an arbitrary point outside the data distribution they attract the hyper-sphere towards the centre of the data distribution. A hyper-plane is pulled over the mapped data points with the restriction that the margin always remains positive or zero. They make the fit of this hyper-plane as tight as possible by minimizing the mean value of the output of the function. Obviously the tighter the hyper-plane, the more sensitive the system becomes to noise and outliers in the data. For this, a soft margin approach is followed that incorporates a user set parameter that controls the size of the boundary. A drawback of this method is the fact that the system performance is very much dependent on the choice of the kernel parameter σ. The only way to set σ is through experimentation and if not enough data is present for validation purposes this can become very difficult. The usage of an ensemble of models with varying σ can be used to lessen the impact of σ. Another drawback of this method is the fact that the system always tries to fit a hyper-sphere around the data points. This, according to Tax and Duin (1998) limits how tight the boundary can be put around the class objects especially when classes are not spherically distributed.

Manevitz and Yousef (2001) investigate the usage of SVM for information retrieval with the aid of novelty detection. The paper first explains the method proposed by Schölkopf *et. al.* (2000) and how this work improves upon that. The authors criticise Schölkopf's technique for being to sensitive to the parameters selected such as the choice of kernel. The difference in performance is very dramatic based on these choices meaning that the method is not robust without a deeper understanding of these representation issues. The basic idea in this research is to work first in the feature space, and assume that not only the origin in the second class, but also that all data points "close enough" to the origin are to be considered as noise or outliers. If a vector has few non-zero entries, then this indicates that the pattern shares very few items with the chosen feature subset of the database. So, intuitively, this item will not serve well as a representative of the class. It is reasonable to treat such a vector as an outlier. By thresholding the number of features with non-zero values, an outlier can be declared. A global threshold can be set for all classes or alternatively each class can have its own threshold. A validation set can be used for setting the thresholds. After the threshold is set, one can continue with the standard two-classes SVM. Linear, sigmoid, polynomial and radial basis kernels were used in this work.

Diehl and Hampshire (2002) discuss novelty detection for image analysis. For image classification and rejection, a set of closed decision regions encompassing the training examples from the various object classes is estimated. First, a large margin partition of the input image feature space is learnt by minimizing an objective function. This function is a generalisation of the standard formulation of support vector learning that is ideally suited for learning partitions of high dimensional spaces from sparse datasets. Once the initial partition is learned, a rejection region R_{reject} is defined by estimating C differential thresholds that yield a given class-conditional probability of detection on a validation set. All images that lie in the rejection region will be rejected. The logistic linear form induces closed decision regions on the surface of the hyper-cube as desired. A logistic linear classifier to partition the class label distribution space is used. As the sequence classifier processes the observed image sequences, the image sequences assigned to a given class are rank ordered based on their likelihood. Given that the likelihood is generally monotonically increasing with increasing differential, they sort the image sequences based on the differential produced by the sequence classifier. This allows the user to quickly focus attention to the examples that cause the greatest degree of confusion for the classifier.

Ratsch *et al.* (2002) show via an equivalence of mathematical programs that a support vector (SV) algorithm can be translated into an equivalent boosting-like algorithm and vice versa. They show this translation procedure for a new algorithm: one-class leveraging, starting from the one-class support vector machine (1-SVM). This is a first step toward unsupervised learning in a boosting framework. Building on so-called barrier methods known from the theory of constrained optimization, it returns a function, written as a convex combination of base hypotheses, that characterizes whether a given test point is likely to have been generated from the distribution underlying the training data. In this manner, novel patterns can be detected.

Davy and Godsill (2002) present a hybrid time-frequency/support vector machine (TFR/SVM) abrupt change detector. The objective of novelty detection is to decide whether a given vector x belongs to the set of training vectors X or it is novel. A solution to estimating the region R consists of fitting a SVM kernel on the support training vectors defining a hyper surface. The most commonly used kernel is the Gaussian kernel. In many situations, the training set may contain a small number of abnormal vectors that may cause the optimal hyperplane to be wrongly placed. The authors suggest the usage of slack variables that allow for some abnormal vectors. The method was successfully applied to audio signal segmentation but no comparison to competing methods was performed.

Diehl and Hampshire (2002) show an interesting application of novelty detection for video sequences using generalised support vector learning. In the first step, all objects in an image are determined through training a classifier and testing it on new video sequences. For a collection of images in a sequence, this then leads to each video frame analysed for the objects it contains and assigning it to a category that best represents that frame. For example, if for a frame that contains mostly a car, it may be labelled as "car". For video sequences whose labels vary greatly, from frame to frame, show a degree of confusion and these sequences are labelled as novel.

2.3 *ART approaches*

Adaptive Resonance Theory has been shown to generate effective classifiers for novelty detection that have been shown to outperform other classifiers such as SOM and LVQ. For example, Moya *et al.* (1993) compared ART2-A, Kohonen's Learning Vector Quantisation (LVQ) and Reilly and Cooper's Restricted Coulomb Energy network (RCE). All these algorithms use hyper-spheres to surround the training classes and produce closed decision boundaries. The difference between these algorithms is the manner in which they determine the number, position and sizes of the hyper-spheres. During training, ART2-A fixes the size of the hyper-spheres, RCE fixes the position and LVQ fixes the number. After training, if a test vector is outside the hyper-spheres it is deemed to be unknown. Synthetic Aperture Radar (SAR) imagery data was used to train and test the networks. The results showed that ART2-A and RCE depend on the value of vigilance, a user set parameter that controls the size of feature space surrounded by the networks and consequently affects the number of hyper-spheres. Large vigilance causes the network to enclose lots of small regions and make the network highly discriminatory at what it calls a target. This allows excellent between-class generalisation but poor within-class generalisation. Small values of vigilance have the opposite effect. This value needs to be optimised. Overall performance is defined as the minimal performance over all three generalisation criteria. After optimisation, LVQ yielded 89% performance, RCE 94% and ART2-A 100%.

A number of ART and fuzzy ART models have been proposed in literature. The fuzzy ARTMAP is a self-organizing neural network. Each input learns to predict an output class *K*. During training the network creates internal recognition categories, with the number of categories determined on-line by predictive success. With fast learning, the j^{th} weight vector records the largest and smallest component values of input vectors placed in the j^{th} category. The weight vector is a hyper-box that encompasses all input vectors that were assigned to that category during training. With winner-take-all strategy, the node that receives the highest activation is selected and remains activated if it satisfies a matching criterion. Otherwise, the network resets the activation field and searches again for the next node that satisfies the matching criterion. If the node makes a wrong class prediction, a match tracking signal raises vigilance just enough to induce search, which continues until either a node becomes active for the first time in which case the class is assigned to that node or a node that has already be allocated that class becomes active. This way, the network learns the number of classes on-line. During testing, a test vector is assigned to the class that is represented by the activated node. A test pattern is classified as novel if a familiarity function is less than a predetermined threshold g. This function considers all of the training objects that belong to the hyper-box defined by the weight vector of the winning node. A test pattern has to lie within this hyper-box to be deemed to belong to that class. Carpenter *et al.* (1997a, 1997b) extended the fuzzy ARTMAP neural network to perform familiarity discrimination (ARTMAP-FD) and test the technique on a simulated radar target recognition task evaluated using ROC curves. Ideally, after training on a set of known objects, the network should not only be able to classify these objects but also abstain from making a meaningless guess when presented with an object belonging to a different, unfamiliar class. The method shows that the threshold is influenced by noise in the system. That is, g will fail with increasing noise levels and new threshold values need to be calculated. This value might be set by first calculating the noise level of the data. As the authors point out, because of noise and varying target patterns encountered during operation, the robustness of the choice of the optimal $gI=G$ is an important factor in the success of applications. The technique presented here for novelty detection, the strategies for setting the familiarity threshold and the results obtained as well as a comparison with another technique are presented in more detail in Granger *et. al.* (1999).

2.4 *RBF Approaches*

Radial basis function network represent an important class of neural networks where the activation of the hidden unit is determined by the distance between an input vector and prototype vector (Bishop, 1995). Fredrickson *et al.* (1994) used an RBF neural network with Gaussian basis functions for novelty detection. The centre positions and covariance matrices are determined by unsupervised clustering using *k*-means followed by a width heuristic or the EM algorithm. Output weights can be computed via supervised learning techniques, such as least mean-square (LMS) gradient descent or matrix pseudo-inversion. The network outputs are estimates of Bayesian *a posteriori* class probabilities. The system is applied to speaker identification. Three novelty assessment techniques are used to evaluate the networks. First, minimum Mahalanobis Distance (MMD) is used. After applying a test pattern, the network with the MMD between the pattern and the kernel with the maximum response is selected indicating very low novelty. Second, a novelty detection method is based on Projective geometry using the RBF hidden layer and pseudo-inversion of the matrix of kernels. Finally, Parzen windows are used to assess the novelty of the test pattern (this serves as a baseline).

Roberts and Penny (1996) present a method for calculating network errors and confidence values relying upon the use of committees of networks. The authors state that the use of a single weight vector is sub-optimal because most problems are complex with several local minima. Although the weights vector is optimised, it still represents the weights set corresponding to one of many minima in the network's energy function. The solution to this problem is to use a committee of networks each initialised with a different weights vector. The output error can be calculated from the committee's error covariance matrix or just take an average of all the error values. The first term in the error equation penalises the variant decisions between committee members and the second penalises the committee as a whole if the error is erroneous. The experiments were performed using a committee of RBF networks each utilising a thin-plate spline functions on the hidden layer. The technique was tested on a regression problem and a real problem of muscle-tremor classification task. The aim of the classification was to distinguish between patient and normal groups. The results showed that this approach outperforms that of a single network without discarding too much of the data as novel.

By adding reverse connections from the output layer to the central layer Albrecht *et al.* (2000) show how a Generalized Radial Basis Function (GRBF) can self-organize to form a Bayesian classifier that is also capable of novelty detection. An RBF is fed with D-dimensional vectors through weighed synaptic connections activating M neurons in the central layer. The induced activities are calculated from non-linear activation functions. The most widely used activation function, and the one used in this research, is the multivariate Gaussian. The authors use a globally normalised alternative to the normal normalisation factor of the Gaussian introducing a small cut-off parameter e in order to confine non-zero activity responses of the central layer to an input pattern from a bounded region within input space. Any activation below e is set to zero. The normalised activation functions used do not exhibit a simple radial decay characteristic and therefore the authors call these functions General RBF (GRBF). If each neuron in the central layer is uniquely associated to one of the training classes then the activity of the j^{th} output neuron is simply the summation of the normalised activities of all those neurons in the central layer that are associated with class j. This makes the GRBF identical to the Bayesian Maximum Likelihood Classifier (BMLC). By adding reverse connections from the output layer to the central layer, the authors enable the GRBF to self-organize and group the neurons of the central layer that belong to the same class together. The cut-off parameter e can be used for novelty detection. A pattern that belongs to a class previously unseen by the network is likely to elicit a very small total activity within the central layer because this total activity is the likelihood that the input pattern has been drawn from the model density of the training set. If the activation is smaller than e then none of the neurons will acquire a non-vanishing activation and the output of the network will also vanish. Thus, a vanishing response of the classifier is an indication of the novelty of the input.

Brotherton *et al.* (1998) have used a Class Dependent-Elliptical Basis Function (CD-EBF) neural network for classification and novelty detection. Unlike the MLP, the EBF and similar networks have nearest neighbour properties that make them well suited for novelty detection. EBFs facilitate novelty detection by the way they are trained. Training is performed in two steps. The first is clustering the training data into hidden-layer elliptical basis units (EBU). The number of basis units required to model a given class is determined and a Linear Vector Quantisation (LVQ) algorithm is used to delineate the basis units for the given class. This is performed for all training classes. The second step is a least mean-square (LMS) weighting of the EBU outputs to form the desired function approximation for classification of each class. Alternatively, to simply select the class of the EBU that has the highest activation. CD-EBF can be used for novelty detection because of its nearest neighbour property. Each of the C sets of EBUs constitutes a model for its associated class.

When novel data is input to the system it is compared with each of the models developed for the C classes and the responses are gauged and measured against a threshold. Each class might have its own threshold calculated by combining the histogram of the class in question and the joint histogram of the rest of the classes. One very interesting property of this system is that the addition of a new class is very easy. A new set of EBUs for the new class only needs to be developed and the information of the rest of the classes is simply carried through to the new system. The LMS step might be required to be performed for all classes though. Jakubek and Strasser (2002) similarly use ellipsoidal functions. Their fault detection scheme works in three steps: First, principal component analysis of training data is used to determine nonsparse areas of the measurement space. Fault detection is accomplished by checking whether a new data record lies in a cluster of training data or not. Therefore, in a second step the distribution function of the available data is estimated using kernel regression techniques. In order to reduce the degrees of freedom and to determine clusters of data efficiently in a third step the distribution function is approximated by a neural network. In order to use as few basis functions as possible a new training algorithm for ellipsoidal basis function networks is presented: New neurons are placed such that they approximate data points in the vicinity of their centers up to the second order. This is accomplished by adapting the spread parameters using Taylor's theorem. Thus, the amount of necessary parameters and the computational effort for online supervision can be reduced dramatically.

Brotherton and Johnson (2001) use an RBF for novelty detection that is constrained so that groups of hidden unit basis functions within the neural net are associated with only a single class. This is achieved by clustering the input data into one of several candidate basis units. LVQ can be used for this clustering or in this paper, however, the easier *k*-means clustering algorithm is used. The basis functions used in this work are Gaussian with the mean and variance of each dimension calculated from the data. Following clustering a least mean square weighting of the basis unit outputs is applied to form the desired function approximation for classification. For novelty detection, the net is able to detect some new event that the network has never encountered before. This comes about because of the nearest-neighbour properties of the RBF. When signal data is input to the system, it is matched against the model developed. If the input signals do not fall in any of the basis units, then anomaly is detected.

In most applications of RBF networks, the output strategy implemented is as with the backpropagation network that of Winner Take All (WTA). However, according to Li *et al.* (2002), this is not the most desirable strategy when one is dealing with unknown classes such as in the case of fault diagnosis. An alternative to WTA is to apply a threshold at the output of the network and if the value of an output neuron exceeds this threshold, then the test vector is assigned to the class represented by that neuron. This approach offers the advantage to classify something as neither 'normal' nor 'existing fault' but rather as something novel. This is particularly useful in fault diagnosis tasks as not all faults are known *a priori* and used for training but also more than one fault might be occurring simultaneously. However, there is a serious problem. There is no theoretical basis for setting this threshold (it is set empirically to 0.5). This paper attempts to give a mathematical and geometrical analysis of how such a threshold might be set.

2.5 *Auto-associator approaches*

The main idea behind the auto-associator approach is to try and recreate the output of a system the same as its input. Song *et al.* (2001) describe a number of different ways for building an auto-associator. The simplest method is to use Principal Component Analysis (PCA). PCA relies on an eigenvector decomposition of the covariance or correlation matrix of the process variables. However, PCA identifies only linear correlation between the variables. Principal curves and surfaces are non-linear generalisation of the first principal component and the principal manifold is a generalisation of the first two principal components. Conceptually the principal curve is a curve that passes through the middle of the data. Auto-associator can also be performed using the multilayer perceptron architecture to implement the mapping functions in a bottleneck and most approaches to novelty detection use feed-forward networks. The whole process can also be used as non-linear Principal Components Analysis where the hidden weights represent a smaller than number of input non-linear components. A novel sample when input to such a network will fail to recreate the same output and hence the error at the output can be thresholded to reject novel samples. The batch self-organising map can also be used as an auto-associator because it can be used for finding discrete approximation of principal curves (or surfaces) by means of a topological map of units. The batch version of the SOM is closely related to the principal curves algorithm. Finally, principal curves can be combined with neural networks. Principal curves can be used to map an *n*-dimensional space to a *k*-dimensional non-linear principal scores (*n*>>*k*) and the neural network can be applied to the *k*-dimensional scores to map them back to the *n*-dimensional corrected data set. The aforementioned methods i.e. PCA, Principal curves and neural networks, Principal curves and splines, and self-supervised MLP were tested on a simple synthetic 3-dimensional mathematical problem. The results show the Principal curves and splines method outperforms the rest of the methods with the self-supervised MLP being the second best.

The auto-encoder neural net has a number of uses, from non-linear principal component analysis, information compression to recovery of missing sensor data (Thompson *et al.*, 2002) and motor fault detection (Petsche et al., 1996). The most striking ability of the auto-encoder is the ability to implicitly learn the underlying characteristics of the input data without any *a priori* knowledge or assumptions. Most of the studies below use a neural network based auto-encoder system. Byungho and Cho (1999) discuss three critical properties of an autoassociator MLP novelty detector that is trained with normal patterns only. First, there exist infinite input vectors for which such a network could produce the same output vector. Second, there exists the 'output-constrained hyperplane' on which all output vectors are projected. As long as the MLP uses the bounded activation functions, the hyperplane is bounded. Finally, minimising the error function leads the hyperplane located in the vicinity of the training pattern. Similarly, a detailed analysis of the probabilistic behaviour of a novelty filter working on the autoassociative principle is available in Ko and Jacyna (2000). Diaz and Hollmen (2002) also studied the properties of autoassociative nets and compared the least squares mapping to kernel regression mapping. They found that kernel regression mapping is better suited and suggest how residuals can be correlated with the prior knowledge for visualisation that can aid fault diagnosis.

Japkowicz *et al.* (1995) deal with the problem of binary classification, that is, classifying a signal as of two classes, normal or abnormal using a novelty detection technique. The threshold setting that determines whether the reconstruction error is small or large is, according to the authors, relatively easy and can be selected during training of the system. The error on training defines the lower bound, which is then relaxed a bit and used for testing. In cases when the separation between the two classes is more difficult, a few samples of the negative class may be used for training for better setting the threshold.

Similarly, Streifel *et al.* (1996) describe the use of auto-associator network for detecting shorted windings in operational turbine-generators (novel samples). They calculate the threshold at output layer based on the training data. The average vector, called prototype, is found and it is subtracted from all the patterns in the training set. This is done to translate the signature signals toward the signal space origin. The simplest detection surface, according to the authors, is the hyper-sphere. The largest Euclidean length of the translated healthy signature signals is used as the threshold. Any signature outside the hyper-sphere is considered to be a fault.

Worden (1997) applied the auto-associator network to a simulated condition monitoring task. A novelty measure *v* is calculated for each input pattern by taking the Euclidean distance between the input and the output pattern. Training stops when *v* is reduced to zero. Novelty is detected if a test pattern returns a non-zero novelty index. Unlike the previous study by Worden (1997), the objective of the study by Surace and Worden (1997) is to detect damage in structures that have at least two normal operating conditions. The three-degrees of freedom system with concentrated masses considered by Worden (1997) was used with two normal conditions. The system was successful in detecting the fault condition in the system. The technique was compared with a naï ve solution that considers the Euclidean distance between training and testing patterns after averaging over the training patterns. However this solution fails in the presence of two normal conditions.

Surace *et al*. (1997) describe novelty detection for crack detection in beams using auto-associator network. The neural network is trained using transmissibility functions of an uncracked beam and it is then tested on patterns from the cracked beam. In this study a comparison was made between using the Euclidean distance to calculate the novelty index, as described in Surace and Worden (1997) and Mahalanobis distance with the covariance matrix being derived from the training data. A novelty index efficiency formula was used to compare the two distances. The cracked cantilevered beam used in this study behaves in a non-linear manner and this study proves that the technique is efficient in both linear and non-linear structures since it requires no *a priori* knowledge of the model. The patterns of the cracked and uncracked beam were simulated using Euler-type finite elements with two degrees of freedom. Using the technique with the Euclidean distance it was possible to positively identify the presence of cracks in all cases in the simulation. The simulation was repeated with the Mahalanobis distance, which showed better performance in the presence of noise.

Surace and Worden (1998) study damage detection in an offshore platform. The method presented here is an extension to previous work of these authors. In the previous paper, the technique proposed is not robust enough to handle more than one normal condition. This paper extends the method to handle a continuum of normal conditions, a situation faced for example in an offshore platform. The difference between this approach and previous suggested by the authors is that the training patterns are contaminated with noise to increase their variability and instead of the Euclidean distance, the Mahalanobis distance is used (with the covariance matrix of the training data) to define the novelty index. The technique was tested on the same problem as in Surace *et al.* (1997) and a Finite Element model of an offshore platform.

Ko *et al.* (2000) present a auto-associator network based hierarchical identification strategy for successive detection of the occurrence, type, location and extent of structural damage in bridges. The first stage of the proposed hierarchical identification strategy is to detect the occurrence of damage or anomaly in the bridges. Just as in the previous studies, the difference of the input vector and the trained network's output vector serves as a novelty index. In this study, five auto-associator networks are used to monitor the condition of each bridge. The second stage of the hierarchical process uses a probabilistic neural network to detect the type and the location on the bridge of the damage detected by the novelty detector. Finally, a backpropagation neural network is used to detect the extent of the damage.

Sohn *et al*. (2001) explore the effect changing environmental conditions on a auto-associator novelty detection system. Novelty detection is in one sense the measurement of deviation of some features of a system from the norm. However, changing environmental conditions such as varying temperature, moisture, lighting conditions and so on affect the normal features and the normal conditions. Moreover, these changes may hide the true novelty within the system (Manson *et al.*, 2000, 2001). The main difference to a conventional auto-associator network used is that the output of the 'bottleneck' layer is fed to another hidden layer with the same number of units and activation functions as the mapping layer. The difference of the reconstructed output and the input to the network is called the residual error that also acts as the novelty index. The system only provides indication of the presence of novelty and not the type or severity of it. When patterns of an unknown condition are presented to the system it is expected that the novelty index will increase. If the index rises above the predefined threshold then the pattern is deemed to be novel.

Manevitz and Yousef (2000) apply the auto-associator network to document classification problem. For acceptance threshold determination, the authors used a sophisticated method based on a combination of variance and calculating the optimal performance. During training, they checked at different levels of error, the performance values of the test set. They ceased training at the point where the performance started a steep decline. Then they perform a secondary analysis to determine an optimal real multiple of the standard deviation of the average error to serve as a threshold. The method was tested and compared with a number of competing approaches and found to outperform them. The competing systems were prototype matching, Nearest Neighbour, Naïve Bayes and Distance based Probability algorithms.

2.6 *Hopfield networks*

The human brain has more capacity for familiarity discrimination rather than recognition of various stimuli. Hopfield networks have been suggested as good quality novelty detectors (Jagota, 1991). Bogacz *et al.* (1999) demonstrate that a neural network has exactly the same characteristic. They implement familiarity discrimination using two models: first, using a single neuron with Hebbian learning, and second, using the energy of a Hopfield network. Their proposed approach differs from existing approaches in that it assumes that the patterns are not correlated. These algorithms compress information and perform discrimination either by discovering the underlying distribution of the familiar patterns and finding outliers, or by constructing prototypes of the various classes. The weights of the neural networks are used to store the information of the uncorrelated patterns. Both models have higher capacity for familiarity discrimination rather than for retrieval. The first model assumes a single neuron with N inputs. The node takes values between –1 and +1 where –1 indicates inactive state whereas +1 indicates active state. All weights are initialised to zero. The Hebbian rule is used to update the weights. The authors have demonstrated that the average value for stored patterns is 1 while for novel patterns it is 0. Therefore by taking as threshold the middle value 0.5 such that $y = \mathrm{sgn}(h - 0.5)$ where h is the output of the network, they can perform novelty detection. Assuming that the noise is small enough then for novel patterns $y = -1$ and for known patterns $y = +1$. The neuron works well if the noise is smaller than the absolute value of the threshold. The larger the amount of stored patterns, the higher the noise. The authors have shown that the model is successful if the number of stored patterns is less than $.046N$ where N is the number of input neurons. The second model used in this paper is based on a Hopfield network trained with the Hebbian learning rule. The value of the energy function is usually lower for stored patterns and higher for other patterns. The authors have shown that for novelty detection and for the case when the noise has zero mean, a known pattern will yield an average energy value $2E = -N$ whereas for novel patterns $2E = 0$ where E is the network's energy and N is the number of input neurons. Therefore an appropriate novelty threshold should be $-N/2$. If a pattern has energy $2E < -N/2$ it is considered to be familiar. As before, errors occur when noise exceeds the novelty threshold. The authors have calculated that the maximum number of stored patterns should be $.023N^2$. The capacity of the second model is exactly half of that of the first model. This is because the Hopfield network has symmetrical weights thus storing each piece of information twice. If the redundant connections are removed, the capacity of both models is the same. The paper offers no experimental results to test these two models in a novelty detection task but it does however present good theoretical answers on the capacity of these models in familiarity discrimination tasks.

A similar system using a Hopfield neural network for familiarity discrimination is discussed by Crook and Hayes (2001). The model stores information about familiar patterns, in the weights of a Hopfield neural network. A Hopfield network can be used to reconstruct the patterns it has learnt at its output space after an iterative process by which each neuron in the network is updated several times until the network relaxes to the recalled pattern. Novelty detection is implemented by calculating the energy of the Hopfield network after a pattern is shown and then threshold this energy. Patterns with low energy are deemed as familiar whereas large energies point towards novel patterns. Estimating the novelty threshold has a theoretical basis and it is calculated as $E < -N/8$ for familiar patterns where N is the number of input neurons in the network. The advantage of using energy to determine novelty in the data and not allow the neurons to settle through iterations and reconstruct the pattern is that energy is more computationally effective and remains constant no matter how many patterns are stored in the network. One of the shortcoming of this technique is the novelty threshold. According to the authors, the more patterns the network learns and obviously the more noise is introduced, the less effective the threshold becomes. The authors compare their technique with that of Marsland *et al.* (2000a) and claim very similar performance but with significantly less learning and novelty detection runs.

A criticism of neural network architectures is their susceptibility to catastrophic interference; the ability to forget previously learned data when presented with new patterns. Addison *et al.* (2002) evaluate two architectures, namely Hopfield and Elman networks and compare them with self-organizing feature maps and time-delayed neural networks in a novelty detection task. The Hopfield network essentially attempts to store a specific set of equilibrium points such that once an initial condition is provided, the network eventually comes to rest at that design point. Elman networks contain an internal feedback loop, which makes it capable of both detecting and generating temporal patterns. Elman networks have the ability to approximate any input/output function with a finite number of discontinuities owing it to their use of a two layer sigmoid/linear architecture. Time delay networks consist of a complete memory temporal encoding stage followed by a feedforward neural network. The results showed that certain architectures are better at recognizing novelty than others. The Hopfield networks were capable of discriminating between normal and extremely obvious novel patterns but had difficulties on other more difficult abnormal patterns. The Elman networks, showed excellent performance in recognising known patterns as well as discriminating the various novel patterns. The Kohonen network also showed good classification performance. The time delayed network was able to discriminate between error and normal patterns but like the Hopfield network it had difficulties recognising novelty in sets that were primarily consisted of normal patterns.

2.7 *Oscillatory neural networks*

Ho and Rouat (1998) proposed a neural network model that allows studying neural information processing in the cortex. The system dynamics and the self-organizing process exhibit robustness against highly noisy input patterns. Along with the neural network model, they present a new paradigm for pattern recognition based on oscillatory neural networks. The relaxation time of oscillatory behaviour is used as a criterion for novelty detection. The neuron model used here is inspired by the integrate-and-fire neuronal model with refractory period and post-synaptic potential decay. This model defines a single two-dimensional sheet of excitatory and inhibitory neurons with recurrent connections. The layer consists of two populations of neurons interspersed within the plane. Each neuron has a set of interconnections chosen according to a square neighbourhood centred at the neuron itself. If the network's action stimulated by an input signal is successful, all connections of firing neurons are reinforced, regardless of whether they participated in creating a successful action or not. If the action is unsuccessful, the connections of firing neurons are weakened. For updating the connection weights, the Hebbian updating rule is applied. For novelty detection, during the learning phase, the network with randomly initialised connection strengths is trained with learning patterns. It reaches an equilibrium stage after learning. In the novelty detection phase, patterns are introduced to the trained network. The network reaches an equilibrium stage after a relatively small number of iterations if these patterns have been learnt before. Otherwise it takes a long time for the network to reach an equilibrium stage. Novelty is defined by this time taken.

Kojima and Ito (1999) proposed an autonomous dynamical pattern recognition and learning system which can learn new patterns without any external observer. For the novelty filter, the network is constructed from Lorenz systems. The learning rule, updates the synaptic weights in a self-organizing manner according to the discrete time Hebbian learning rule. For measuring the output pattern of the network, they calculated Hamming distances. Novelty detection occurs in a manner similar to Ho, and Rouat (1998). When a known pattern is given to the network, the network oscillates periodically and the output pattern of the network oscillates between the relevant embedded pattern and its inverse. On the other hand if a novel pattern is inputted to the network, the network reaches a turbulent state. This turbulent state is considered as confusion and thus the pattern is deemed to be novel. During this state, the Hebbian learning rule was applied to learn the new patterns.

Borisyuk *et al*. (2000) describe a model consisting of a one-layer network of interacting oscillators. The activity of an oscillator represents the average activity of interactive neural populations (local field potential) with the oscillators grouped. In the initial stages, before the oscillatory network stores any information, each group contains oscillators whose natural frequencies are distributed in the whole range of input frequencies. During information storage, these natural frequencies may change. An oscillator reaches and keeps a high level of activity if the signals that are supplied to this oscillator through the first channel arrive in-phase. This implies that the presentation of a stimulus results in a high oscillatory activity at only a small number of randomly chosen locations (groups). The activity in other parts of the network is low. This occurs in both memorization and recall. For novelty detection it is possible to choose the parameters of learning control in such a way that for a new stimulus the number of resonant oscillators at the end of stimulus presentation is small, but becomes large (and exceeds a certain threshold) only if the stimulus has been learnt before. If the stimulus fails to pass the threshold then a novelty is identified. A computer simulation was used to test the model showing that indeed this type of network can be used for novelty detection. The simulation was limited and no clear results or comparison with competing methods were presented.

2.8 *SOM based approaches*

Self-organising Maps proposed by Kohonen (1988, 2001) are an alternative to statistical clustering of data. The approach is unsupervised and therefore no *a priori* information on class labels of samples is necessary. In most SOM based approaches, similar to statistical clustering, some form of cluster membership value is thresholded to determine whether a sample belongs to a cluster or not.

Aeyels (1991) provides proof and clarifies some points regarding the convergence properties of the novelty detector and novelty filter described by Kohonen (1988). These adaptive systems are capable of storing a number of inputs and responding only to 'new' inputs, patterns that the system has not 'seen'. The author tries to elucidate some of the results presented in Kohonen, (1988) but also to indicates some problems. Kohonen (1988) described two types of novelty detectors: novelty detector without forgetting and the novelty detector with forgetting. Regarding the novelty detector without forgetting, in the case when the input to the system is constant, the convergence is easy to derive. However, in the case when the input to the system is a regular bounded function of time, the author proves that in order to have the system reacting to novelty with respect to the stored patterns, all the stimuli should keep coming back. In other words, the system can only memorise patterns when it is frequently exposed to them. Any new stimulus will provoke an initial reaction in the output and will be then added to the memory if it is frequently reiterated. The novelty filter without forgetting consists of a collection of novelty detectors, connected by particular feedback laws. The author shows that the convergence of this system is easy to prove. Finally, the novelty filter with forgetting contains a forgetting term that forces the system to habituate and reduce its response when similar patterns are frequently shown. This is similar to habituation described by Marsland *et al.* (1999, 2000a, 2000b, 2000c).

Harris (1993) presents one of the earliest approaches on using a Kohonen Self-Organising Map (SOM) for novelty detection in an engine health monitoring system. A SOM is trained with examples of normal operation. After training, the map contains the reference vectors of the input data. These reference vectors are optimised to accurately represent both the density and the position of the input data. When testing, the distance between the test vector and these reference vectors is a measure of novelty. A variation of this approach can be used if some examples of the faulty conditions are available. In this case they can also be used during training for better representing the faulty space. The task in this case is reduced to classification and not so much to novelty detection.

Ypma and Duin (1998) employ a Self-Organising Map (SOM) to develop a novelty detection technique used for the detection of faults in a fault monitoring application. The authors comment on the unavailability of samples that describe accurately the faults in the system and agree with other authors that the best solution is to accurately build a representation of the normal operation of the system and measure faults as a deviation of this normality. The usage of a SOM provides a domain description instead of a probability density estimation as used by Barnett and Louis (1994), Bishop (1994), Tarassenko *et al.* (1995) and several others. Additionally, the topology of the input space is preserved as opposed to using some other unsupervised clustering algorithm such as the *k*-means algorithm, giving information about the mapping that could be exploited in defining a more confident "compatibility measure". The novelty detection technique is a very simple one. Once the SOM is trained with samples of normal operation it is tested and patterns from normal operation generate small distance while abnormal patterns generate large distance.

Emamiam et al. (2000) present a very simple novelty detection technique based on a Self-Organizing Map (SOM) to discriminate acoustic emissions of healthy machinery from that of machinery presenting a crack. The technique presented here is similar to that proposed by Harris (1993) and Ypma and Duin (1998) but not as sophisticated as the habituation approach taken by Marsland *et. al.* (2000a,b,c). After the SOM is trained, it is expected that different types of transient input signals will activate different nodes on the Kohonen map. The authors do not threshold the Euclidean distance between the activated neuron and the input data as do most approaches but instead use the index of the activated node to discriminate between normal and fault condition. It is expected that faulty features will excite different nodes than healthy ones.

Labib and Vemuri (2002) recently described an implementation of a network based Intrusion Detection System (IDS) using Self-Organising Maps. NSOM is an anomaly detection system. The SOM implementation used was a Kohonen net with the winning neuron being the one with the shortest distance from the input pattern. The NSOM is first trained with patterns describing normal network traffic and the output response of the NSOM is noted, i.e. the neurons that are activated are stored. Then the network is tested. If the winning neuron is not one of the neurons noted then a novelty is declared. It is expected that the distance of an input pattern and the winning neuron for a novel pattern will be much larger than the corresponding distance of a known pattern. The technique described here is closely related to that of Emamian (2000).

Theofilou et al. (2003) propose a new Long-Term Depression Kohonen Network (LTD-KN) that is very well suited for novelty detection. The network behaves like a normal Kohonen network in every way except for the fact that the change of the weight vectors for winning and neighbouring neurons is determined by an inverse of the classic Kohonen rule. After learning, all patterns used in the training set and all patterns similar to them give decreasing activation values. All patterns dissimilar to the training set (novel patterns) result always in a stable high activation both during and after learning. The differentiation between known and novel patterns comes as a natural consequence of learning. This makes the LTD-KN to function as a novelty detector.

2.9 *Habituation based approaches*

Habituation is the mechanism by which the brain learns to ignore repeated stimuli and it is considered as the most basic form of plasticity within the brain. Marsland *et al.* (1999) attempted to implement this phenomenon for novelty detection for a mobile robot application. A number of subsequent studies by the same authors attempted to improve their original idea. In Marsland *et al.* (1999) they try to implement the original proposal made by Wang and Arbib (1990) on how to construct a neural model with habituation that is capable of novelty detection. Wang and Arbib modelled the tectal relay and anterior thalamus (AT), the areas that process the images taken from the retina. They then extended the model with a neural mechanism for the medial pallium (MP), the region in which habituation is thought to take place. Their design consists of a large number of columns arranged vertically with five layers of cells in each column, each layer consisting of *n* neurons. One neuron in the AT propagates its response strength to all neurons in the next layer simultaneously. The output of these neurons are controlled by a novelty threshold that increases monotonically with each activation and is designed to make cells with higher number of activations harder to fire, so that stronger stimuli have a larger number of neurons firing. Marsland *et al.* describe the procedure in extensive detail. The only concern with the technique is that it is very computationally intensive and impractical in a real-world robot implementation. The authors considered other ways of implementing habituation in series of future papers (Marsland *et al.,* 2000a,b,c).

In Marsland *et al.* (2000a) use a Kohonen self-organizing map for classifying the inputs and habituating synapses for implementing the novelty filter. Both the clustering network and the novelty filter are described in detail in Marsland *et al.* (2000c, 2001). Marsland *et al.* (2000b) use the same technique presented in this paper is the one described in Marsland *et al.* (2000c, 2001). However, in this paper two alternative clustering schemes, Temporal Kohonen map (TKM) and *k*-means clustering are evaluated as opposed to the SOM used in Marsland *et al.* (2000c) and the GWR network presented in Marsland *et al.* (2001). The TKM is based on Kohonen's SOM but uses "leaky integrator" neurons whose activity decays exponentially over time. This is similar to a short-term memory allowing previous inputs to have some effect on the processing of the current input, so that the neurons that have won recently are more likely to win again. In other words some sort of temporal information is retained in the system and that can be very useful in many applications including video analysis. The novelty detection technique used is based on a clustering network that classifies the inputs, and the output is modulated by habituable synapses, so that the more a neuron fires the lower the efficacy of the synapse becomes. If a synapse is fed with zero instead of nothing, then it forgets the inhibition over time. The three clustering schemes were tested on a relatively easy mobile robot application. The overall qualitative results were similar for all three networks, although the SOM took considerably longer to produce consistent output when a new pattern was introduced while the TKM responded to them the quickest. When two additional light patterns were introduced, the TKM and the *k*-means clustering performed much better than the SOM with the TKM responding again the fastest.

Marsland *et al.* (2000c) describe an algorithm suitable for detecting novel stimuli based on habituation and apply it to an autonomous agent. The paper uses the Habituating Self-Organizing Map (HSOM), a neural network that is capable of detecting novel objects. An input vector is presented to a clustering network, which finds a winning neuron using a winner-take-all strategy. Each neuron in the map field is connected to the output neuron via a habituable synapse, so that the more frequently the neuron fires the lower the efficacy of the synapse and hence the lower the strength of the output. The strength of the winning node is taken as the novelty value and the more familiar the object is, the more the value decreases to zero. The clustering network is implemented using a Kohonen network implementing Learning Vector Quantisation (LVQ). The map has the property of self-organizing depending on the input vectors by moving the winning neuron and to a lesser extend its immediate neighbours towards the input. In the HSOM implementation, the synapses of these neighbours are also changed. The neighbourhood size and the learning rate are user defined and are automatically reduced during training. Then the environment was changed or a new environment was used to test the HSOM. The main disadvantage of the method, as pointed out by the authors, is the organising map. The size of the SOM needs to be defined in advance, often without *a priori* knowledge of the number of objects or their complexity that the system is likely to encounter. This can lead to the SOM becoming saturated with previously learnt stimuli being lost and novel stimuli being misclassified as known. The work of Marland and colleagues (2000a,b,c) has been used by Saunders and Gero (2000, 2001) for novelty in design solutions. They use habituated SOMs to estimate the novelty of a doorway design. A reinforcement signal proportional to the novelty of the design situation is produced to reward the controller for finding novel situations.

Marsland *et al.* (2001) improve the techniques proposed in their earlier studies. As an alternative to SOM, a new type of clustering map, called Grow When Required (GWR) network was developed that allows the insertion of new nodes when required. In this network, both the synapses and the nodes have counters that indicate how many times they are fired. Using these counters it is possible to determine whether a given node is still learning the inputs or it is 'confused'. In other words it tries to map inputs from different classes. If this is the case then a new node is added to the network between the input and the winning node that caused the problem. The insertion of nodes is dependent upon two user-defined thresholds. The first is a minimum activity threshold below which the current node is not considered to be a sufficiently good match and second is a maximum habituation threshold above that the current node is not considered to have learnt sufficiently well. These thresholds need to be set experimentally. The experiments were performed with a small SOM, a large SOM and a GWR network and it was found that the small network quickly saturated and was unable to learn all the objects whereas the large SOM had problems learning the novel objects. It was very sensitive to noise in the sensors and kept misclassifying learnt objects as novel. The GWR on the other hand showed very promising results. The network learnt quickly and was successful in recognising the novel objects at the end of the third run.

Crook *et al*. (2002) compare two models of novelty detection: GWR network proposed by Marsland et al. (2001) and Hopfield energy model (Crook and Hayes, 2001). Two different robot experiments were used to compare the two novelty detection methods. Both experiments were very simple so as the comparison is strictly between the filters and not other experimental issues associated with computer vision pre-processing. The GWR novelty filter has shown more robustness against noise. In general both filters show similar results although the GWR is slightly better.

Marsland *et al.* (2002) extend the GWR system. The new system presented in this paper is capable of autonomously selecting which novelty filter to use depending on what environment the system operates under. The problem in most novelty detection systems is that objects that are quite normal in some environments are to be considered novel in others. For example a chair is normal in an office but should be found to be novel in a corridor. In this system, multiple novelty filters are trained in different environments and the correct filter for the current inputs is selected. A vector of familiarity indices keeps track of how novel each novelty filter finds the environment. The novelty indices are updated after each perception has been presented to all novelty filters. The technique was experimentally tested in a similar way to the authors' previous studies. This time three environments were used to train three different novelty filters. The correct filter was chosen at all times.

2.10 *Neural tree*

Martinez (1998) introduce a competitive learning tree as a computationally attractive scheme for adaptive density estimation and novelty detection. When the neural tree is continuously adapted, novelty detection can be performed on-line by comparing, at each time step, the current estimated model with an *a priori* model. The proposed approach combines the unsupervised learning property of competitive neural networks with a binary tree structure. The procedure performs a hierarchical partitioning in the *d*-dimensional feature space by means of hyper-planes perpendicular to coordinate axis. This results in a binary tree structure in which each internal node stores two scalar quantities, an index representing the dimension orthogonal to the hyper-plane and a weight representing the location of the hyper-plane on this axis. The initialisation process can be performed with *N* input data sampled either randomly from the training data or sequentially as data becomes available. The neural tree is built up by splitting nodes one at a time in order to maintain a single count in each partition cell. The cell to be further partitioned is the one in which the input sample falls. Splitting occurs in the middle of the new data point and the one previously stored. For a given tree topology the weights are optimised by maximizing Shannon's entropy. A top-down learning scheme is employed which consists of optimising the parameters level by level within the model starting at the highest level (root node). For on-line learning, all levels are trained in parallel as data arrives. For novelty detection, the tree was applied to a 2-dimensional data to adaptively partition the input space into 64 equi-probable cells. The starting weight configuration was obtained by random initialisation. Each time, the partition revealed smaller cells in high-density regions and larger cells in low-density regions relative to the underlying input distribution. Thus, the learning rule provides an adaptive focusing mechanism capable of tracking time-varying distributions. The constructed tree from the training data serves as a reference tree. Another tree is built for the testing data and a novelty is detected when it "differs" too much from the reference tree. Possible distance measures are the Kullback-Leibler Divergence or the log-likelihood ratio. An appropriate threshold can be used to detect the novelty. Although the paper presents a very interesting approach to density estimation and novelty detection, the authors fail to state how their method copes with very high-dimensional data. The system was only tested on a 2 and 16-dimensional data. The performance obtained for the experiments performed is excellent.

2.11 *Other neural network approaches*

Linares *et al.* (1997) describe a new neural architecture for unsupervised learning of the classification of mixed transient signals. The method is based on neural techniques for blind source separation and subspace methods. The feed-forward network dynamically builds and refreshes an acoustic events classification by detecting novelties, creating and deleting classes. Each output cell of the neural network is associated with an event class, and several output cell activities are considered as simultaneous presence of events of different classes. The unsupervised neural classifier self-organizes in order to adapt itself to environmental evolutions. This self-organizing process is made on-line, by detecting novelties, creating and deleting classes. The first data space modelling reduces input space dimension to a smaller one. A second process computes a de-correlation matrix that achieves prototype rotations in order to minimize the second order moments of the network output. De-correlation operator application on network outputs is equivalent to a class prototype rotation. So, the computed operator is applied directly to the prototype matrix, and the system stabilizes itself in a state of uncorrelated output cell activities. The neural net has two fully inter-connected layers. The input layer receives coefficients of stimuli vectors. The output layer has one cell per class, and another for novelty. Cell activations are computed by the projection of stimuli vectors on prototype space. It is assumed that class prototypes are linearly independent. The system is initially empty; therefore there are no known classes. A new class is created when the novelty cell activation exceeds a fixed vigilance threshold, with the input vector as the new class prototype. The system also permanently scans class inertias. If one of them is lower than a deletion threshold, then the class will be deleted. This class integration and deletion on-line process induces stabilization of subspace dimension, which depends on the thresholds and the input variability. In the first test, the first seven alphabet pattern letters were randomly mixed. All seven classes were effectively found, and original patterns were recovered with low noise level. The second test shows a signal in a real sub-aquatic environment and system responses. Recurrent events were detected well.

Martinelli and Perfetti (1994) described a cellular neural network (CNN) for novelty. Each cell is connected to its neighboring inputs via an adaptive control operator, and interacts with neighboring cells via nonlinear feedback. In the learning mode, the control operator is modified in correspondence to a given set of patterns applied at the input. In the application mode, the CNN behaves like a memory-less system, which detects novelty for those input patterns that cannot be explained as a linear combination of the learned patterns.

4. Conclusions

There are a number of studies in the area of novelty detection but comparative work has been much less. Only a few papers have compared the different models on the same data set, e.g. Zhang et al. (2001), Addison et al. (1999), Singh and Markou (2003). As a result of few comparative studies, there are few guidelines on which techniques will work best on what types of data. We hope that this survey will provide for researchers a detailed account of other approaches available so that more comparative work can be performed and some of the weakness of known approaches can be addressed.

References

1. J.F.D. Addison, S. Wermter and J. MacIntyre, “Effectiveness of feature extraction in neural network architectures for novelty detection”, Proc. 9th ICANN, vol. 2, pp. 976-981, 1999.
2. J.F.D. Addison, S. Wermter, K. McGarry and J. Macintyre, “Methods for integrating memory into neural networks in condition monitoring”, Proc. International Conference on Artificial Intelligence and Soft Computing, Banff, Alberta, Canada, pp. 380-384, 2002.
3. S. Albrecht, J. Busch, M. Kloppenburg, F. Metze and P. Tavan, “Generalised radial basis function networks for classification and novelty detection: self-organisation of optimal Bayesian decision”, Neural Networks, vol. 13, pp. 1075-1093, 2000.
4. M.F. Augusteijn and B.A. Folkert, “Neural network classification and novelty detection”, International Journal of Remote Sensing, 1999.
5. D. Aeyels “On the dynamic behaviour of the novelty detector and the novelty filter”, in *Analysis of Controlled Dynamical Systems*, B. Bonnard, B. Bride, J. Gauthier and I. Kupka (eds.) vol. 8, Progress in Systems and Control theory, pages 1-10, Springer-Verlag, 1991.
6. V. Barnett and T. Lewis, *Outliers in statistical data*, John Wiley, 1994.
7. C. Bishop, *Neural networks for pattern recognition*, Oxford University Press, 1995.
8. C. Bishop, “Novelty detection and neural network validation”, Proc. IEE Conference on Vision and Image Signal Processing, pp. 217-222, 1994.
9. R. Bogacz, M.W. Brown and C. Giraud-Carrier, “High capacity neural networks for familiarity discrimination”, Proc. ICANN'99, Edinburgh, pp. 773-778, 1999.
10. R. Borisyuk, M. Denham, F. Hoppensteadt, Y. Kazanovich and O. Vinogradova. “An oscillatory neural network model of sparse distributed memory and novelty detection”, BioSystems, pp. 265-272, 2000.
11. T. Brotherton, T. Johnson and G. Chadderdon, “Classification and novelty detection using linear models and a class dependent- elliptical basis function neural network”, Proc. IJCNN Conference, Anchorage, May, 1998.
12. T. Brotherton and T. Johnson, “Anomaly detection for advance military aircraft using neural networks”, Proc. 2001 IEEE Aerospace Conference, Big Sky Montana, March 2001.
13. H. Byungho and C. Sungzoon, “Characteristics of auto-associative MLP as a novelty detector”, Proc. IEEE IJCNN Conference, vol. 5, pp. 3086-3091, 1999.
14. C. Campbell and K.P. Bennett, “A linear programming approach to novelty detection”, Advances in NIPS, vol. 14, MIT Press, Cambridge, MA, 2001.

15. G.A. Carpenter, M.A. Rubin and W.W. Streilein, “ARTMAP-FD: familiarity discrimination applied to radar target recognition”, Proc. International Conference on Neural Networks, vol. III, pp. 1459-1464, 1997a.
16. G.A. Carpenter, M.A. Rubin, and W.W. Streilein, “Threshold determination for ARTMAP-FD familiarity discrimination”, In C.H. Dagli et al. (Eds.), Intelligent Engineering Systems Through Artificial Neural Networks 7, pp. 23-28, New York, NY: ASME Press, 1997b.
17. L.P. Cordella, C. De Stefano, F. Tortorella and M. Vento, “A method for improving classification reliability of multilayer perceptrons”, IEEE Transactions on Neural Networks, vol. 6, no. 5, pp. 1140-1147, 1995.
18. L.P. Cordella, C. Sansone, F. Tortorella, M. Vento and C. De Stefano, “Neural network classification reliability: problems and applications”, Image Processing and Pattern Recognition, vol. 5, Neural Network Systems Techniques and Applications, Academic Press, San Diego (California), pp. 161-200, 1998.
19. P. Crook and G. Hayes, “A robot implementation of a biologically inspired method for novelty detection”, Proc. Towards Intelligent Mobile Robots Conference, Manchester, 2001.
20. P.A. Crook, S. Marsland, G. Hayes and U. Nehmzow, “A tale of two filters - online novelty detection”, Proc. 2002 IEEE ICRA Conference, Washington DC, May 2002.
21. M. Davy and S. Godsill, “Detection of abrupt spectral changes using support vector machines: an application to audio signal segmentation”, Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. II-1313-II-1316, 2002.
22. T. Denoeux, “A neural network classifier based on Dempster-Shafer theory”, IEEE Transactions on Systems, Man and Cybernetics- Part A, vol. 30, issue 2, pp. 131-150, 2000.
23. M.J. Desforges, P.J. Jacob and J.E. Cooper, “Applications of probability density estimation to the detection of abnormal conditions in engineering”, Proc. Institute of Mechanical Engineers, vol. 212, pp. 687-703, 1998.
24. C. De Stefano, C. Sansone and M. Vento, “To reject or not to reject: that is the question - an answer in case of neural classifiers”, IEEE Transactions on Systems, Man and Cybernetics-Part C, IEEE Comp. Press, New York, vol. 30, no. 1, pp. 84-94, 2000.
25. Diaz and J. Hollmen, “Residual generation and visualization for understanding novel process conditions”, Proc. IEEE IJCNN Conference, pp. 2070-2075, 2002.
26. C. P. Diehl, J. B. Hampshire II, “Real-time object classification and novelty detection for collaborative video surveillance”, Proc. IEEE IJCNN Conference, 2002.
27. A.D. Doulamis, N.D. Doulamis and S.D. Kollias, “On-line retrainable neural networks: improving the performance of neural networks in image analysis problems”, IEEE Transactions on Neural Networks, vol. 11, no. 1, pp. 137-155, 2000.
28. V. Emamian, M. Kaveh and A. H. Tewfik, “Robust clustering of acoustic emission signals using the Kohonen network”, Proc. IEEE ICASSP Conference, Istanbul, 2000.
29. S. Fredrickson, S. Roberts, N. Townsend and L. Tarassenko, “Speaker identification using networks of radial basis functions”, Proc. of the VII European Signal Processing Conference, Edinburgh, pp. 812-815, 1994.
30. E. Granger, S. Grossberg, M.A. Rubin, and W.W. Streilein, “Familiarity discrimination of radar pulses”, in M.S. Kearns et al. (Eds.), Advances in NIPS 11, pp. 875-881, 1999.
31. T. Harris, “Neural network in machine health monitoring”, Professional Engineering, July/August, 1993.
32. T. Ho, and J. Rouat, “Novelty detection based on relaxation time of a network of integrate-and-fire neurons”, Proc. 2nd IEEE World Congress on Computational Intelligence, WCCI 98, pp. 1524-1529, 1998.
33. A. Jagota, “Novelty detection on a very large number of memories stored in a Hopfield-style network”, Proc. International Joint Conference on Neural Networks IJCNN-91, vol. 2, pp. 905, 1991.
34. N. Japkowicz, C. Myers and M. Gluck, “A novelty detection approach to classification”, Proc. of 14th IJCAI Conference, Montreal, pp. 518-523, 1995.

35. S. Jakubek, T. Strasser "Fault-diagnosis using neural networks with ellipsoidal basis functions", Proceedings of the American Control Conference, vol. 5, pp. 3846 –3851, 2002.
36. J.M. Ko, Y. Q. Ni, J. Y. Wang, Z. G. Sun, and X. T. Zhou, "Studies of vibration-based damage detection of three cable-supported bridges in Hong Kong", Proc. International Conference on Engineering and Technological Sciences, China, pp. 105-112, 2000.
37. H. Ko and G. Jacyna, "Dynamical behavior of autoassociative memory performing novelty filtering", IEEE Transactions on Neural Networks, vol. 11, no. 5, pp. 1152-1161, 2000.
38. T. Kohonen, Self-organisation and associative memory, Springer-Verlag, Berlin, 1988.
39. T. Kohonen, Self organising maps, Springer, 2001.
40. K. Kojima and K. Ito, "Autonomous learning of novel patterns by utilizing chaotic dynamics", IEEE International Conference on Systems, Man, and Cybernetics, IEEE SMC '99, vol. 1, pp. 284 –289, 1999.
41. T. Kwok, D. Yeung, "Objective functions for training new hidden units in constructive neural networks", IEEE Transactions on neural networks, vol. 8, no. 5, pp. 1131-1148, 1999.
42. K. Labib, R. Vemuri, "NSOM: a real-time network-based intrusion detection system using self-organizing maps", Networks and Security, 2002.
43. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network", In Advances in Neural Information Processing Systems, volume 2, pp. 396-404. Morgan Kaufman, 1990.
44. M.A. Lewis and L.S. Simo, "Certain principles of biomorphic robots", Autonomous Robots (in press, 2003).
45. Y. Li, M.J. Pont and N. B. Jones, "Improving the performance of the radial basis function classifiers in condition monitoring and fault diagnosis applications where "unknown" faults may occur", Pattern Recognition Letters (in press, 2002).
46. G. Linares, P. Nocéra and H. Méloni, "Mixed acoustic events classification using ICA and subspace classifier", Proc. IEEE ICASSP'97, Munich, Germany, 1997.
47. L. M. Manevitz and M. Yousef, "Learning from positive data for document classification using neural networks", Proc. 2nd Bar-Ilan Workshop on Knowledge Discovery and Learning, Jerusalem, May 2000.
48. L. M. Manevitz and M. Yousef, "One-class SVMs for document classification", Journal of Machine Learning Research, vol. 2, pp. 139-154, 2001.
49. G. Manson, G. Pierce, K. Worden, T. Monnier, P. Guy, K. Atherton, "Long term stability of normal condition data for novelty detection", Proc. 7th International Symposium on Smart Structures and Materials, California, 2000.
50. G. Manson, G. Pierce and K. Worden, "On the long-term stability of normal condition for damage detection in a composite panel", Proc. 4th International Conference on Damage Assessment of Structures, Cardiff, UK, June 2001.
51. S. Marsland, U. Nehmzow and J. Shapiro, "A model of habituation applied to mobile robots", Proc. TIMR, Towards Intelligent Mobile Robots, Bristol, 1999.
52. S. Marsland, U. Nehmzow and J. Shapiro, "A real-time novelty detector for a mobile robot", Proc. European Advanced Robotics Systems Conference, Salford, 2000a.
53. S. Marsland, U. Nehmzow and J. Shapiro, "Novelty detection for robot neotaxis", Proc. 2nd International ICSC Symposium on Neural Computation, Berlin, pp. 554-559, 2000b.
54. S. Marsland, U. Nehmzow and J. Shapiro, "Detecting novel features of an environment using habituation", Proc. Simulation of Adaptive Behaviour, MIT Press, 2000c.
55. S. Marsland, U. Nehmzow and J. Shapiro, "Novelty detection in large environments", Proc. Towards Intelligent Mobile Robots Conference, Manchester, 2001.
56. S. Marsland, U. Nehmzow and J. Shapiro, "Environment-specific novelty detection", From Animals to Animats, Proc. 7th International Conference on Simulation of Adaptive Behaviour, Edinburgh, 2002.

57. G. Martinelli and R. Perfetti, "Generalized cellular neural network for novelty detection", IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 41, issue 2, pp. 187-190, 1994.
58. D. Martinez, "Neural tree density estimation for novelty detection", IEEE Transactions on Neural Networks, vol. 9, no. 2, pp. 330-338, 1998.
59. M. R. Moya, M. W. Koch, and L. D. Hostetler, "One-class classifier networks for target recognition applications", In Proc. World Congress on Neural Networks, International Neural Network Society (INNS), pages 797-801, 1993
60. A.F. Murray, "Novelty detection using products of simple experts- a potential architecture for embedded systems", Neural Networks, vol. 14, pp. 1257-1264, 2001.
61. T. Petsche, A. Marcantonio, C. Darken, S.J. Hanson, G.M. Kuhn and I. Santoso, "A neural network autoassociator for induction motor failure prediction", Advances in NIPS, vol. 8, pp. 924-930, 1996.
62. G. Ratsch, S. Mika, B. Scholkopf and K. Muller, "Constructing boosting algorithms for SVMs: an application for one-class classification", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 9, pp. 1184-1199, 2002.
63. S.J. Roberts and W. Penny, "Novelty, confidence and errors in connectionist systems", Proc. of IEE Colloquium on Intelligent Sensors and Fault Detection, number 1996/261, Savoy Place, London, 1996.
64. J. Ryan, M.J. Lin, R. Miikkulainen, "Intrusion detection with neural networks", in Advances in Neural Information Processing Systems 10, M. Jordan et al., Eds., Cambridge, MA: MIT Press, pp. 943-949, 1998.
65. R. Saunders and J.S. Gero, "The importance of being emergent", Proc. Artificial Intelligence in Design, 2000.
66. R. Saunders and J.S. Gero, "Designing for interest and novelty, motivating design agents", Proc. 9th International Conference on Computer Aided Architectural Design Futures, pp. 725-738, July 2001.
67. A. Schölkopf, R. Williamson, A. Smola, J.S. Taylor and J. Platt, "Support vector method for novelty detection", In Neural Information Processing Systems, S.A.Solla, T.K. Leen and K.R. Müller (eds.), pp. 582-588, 2000.
68. S. Singh and M. Markou, "An approach to novelty detection applied to the classification of image regions", IEEE Transactions on Knowledge and Data Engineering, (in press, 2003).
69. H. Sohn, K. Worden and C.R. Farrar, "Novelty detection under changing environmental conditions", Proc. 8th Annual SPIE International Symposium on Smart Structures and Materials, Newport Beach, CA, 2001.
70. S.O. Song, D. Shin, E. S. Yoon, "Analysis of novelty detection properties of auto-associators", Proc. COMADEM, pp. 577-584, 2001.
71. R.J. Streifel, R.J. Maks and M.A. El-Sharkawi, "Detection of shorted-turns in the field of turbine-generator rotors using novelty detectors- development and field tests", IEEE Transactions on Energy Conversation, vol. 11, no. 2, pp. 312-317, 1996.
72. C. Surace, K. Worden and G. Tomlinson, "A novelty detection approach to diagnose damage in a cracked beam", Proc. of SPIE, vol. 3089, pp. 947-953, 1997.
73. C. Surace and K. Worden, "A novelty detection method to diagnose damage in structures: an application to an offshore platform", Proc. 8th International Conference of Off-shore and Polar Engineering, vol. 4, pp. 64-70, 1998.
74. D. M. J. Tax and R.P.W. Duin, "Outlier detection using classifier instability", In Advances in Pattern Recognition, the Joint IAPR International Workshops, pp. 593-601, 1998.
75. D.M.J. Tax and R.P.W. Duin, "Data domain description using support vectors", Proc. ESAN99, Brussels, pp. 251-256, 1999a.
76. D.M.J. Tax and R.P.W. Duin, "Support vector domain description", Pattern Recognition Letters, vol. 20, pp. 1191-1199, 1999b.

77. D.M.J. Tax and R.P.W. Duin, "Uniform object generation for optimizing one-class classifiers", Journa l of Machine Learning Research, vol.2, pp. 155-173, 2001.
78. L. Tarassenko, "Novelty detection for the identification of masses in mammograms", Proc. 4th IEE International Conference on Artificial Neural Networks, vol. 4, pp. 442-447, 1995.
79. L. Tarassenko, A. Nairac, N. Townsend and P. Cowley, "Novelty detection in jet engines", IEE Colloquium on Condition Monitoring, Imagery, External Structures and Health, pp. 41-45, 1999.
80. D. Theofilou, V. Steuber and E.D. Schutter, "Novelty detection in Kohonen-like network with a long-term depression learning rule, Neurocomputing, (in press, 2003).
81. B.B. Thompson, R.J. Marks II, J. J. Choi, M.A. El-Sharkawi, M. Huang and C. Bunje, "Implicit learning in auto-encoder novelty assessment", Proc. International Joint Conference on Neural Networks, Honolulu, pp. 2878-2883, May, 2002.
82. V.N. Vapnik, *Statistical learning theory*, Wiley Inter-science, 1998.
83. G.C. Vasconcelos, "A bootstrap-like rejection mechanism for multilayer perceptron networks", II Simposio Brasileiro de Redes Neurais, São Carlos-SP, Brazil, pp. 167-172, 1995.
84. G.C. Vasconcelos, M.C. Fairhurst, and D.L. Bisset, "Recognizing novelty in classification tasks", in Neural Information Processing Systems Workshop (NIPS'94) on Novelty Detection and Adaptive Systems monitoring. Denver - CO, USA, 1994.
85. G.C. Vasconcelos, M.C. Fairhurst, and D.L. Bisset, "Investigating feedforward neural networks with respect to the rejection of spurious patterns", Pattern Recognition Letters, vol. 16, pp. 207-212, 1995.
86. D.L. Wang and M.A. Arbib, "Complex temporal sequence learning based on short-term memory", Proceedings of the IEEE, vol. 78, pp. 1536-1543, 1990.
87. C. L. Wilson, J. L. Blue, O. M. Omidvar, "Improving neural network performance for character and fingerprint classification by altering network dynamics". Proc. of The World Congress on Neural Networks, July 1995.
88. K. Worden, "Structural fault detection using a novelty measure", Journal of Sound and Vibration, vol. 201, issue 1, pp. 85-101, 1997.
89. A. Ypma and R.P.W. Duin, "Novelty detection using self-organising maps", Progress in Connectionist Based Information Systems, vol. 2, pp. 1322-1325, 1998.
90. B.T. Zhang and G. Veenker, "Neural networks that teach themselves through genetic discovery of novel examples". Proc. IEEE International Joint Conference on Neural Networks (IJCNN'91), vol. 1, pp. 690-695, 1991.
91. Z. Zhang, J. Li, C.N. Manikopoulos, J. Jorgenson, J. Ucles, "HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification", Proc. IEEE Workshop on Information Assurance and Security, pp. 85-90, 2001.