A Multi-category Decision Support Framework for the Tennessee Eastman Problem.

G. E. Lee, P. A. Bahri, S. S. Shastri and A. Zaknich

Abstract— **The paper investigates the feasibility of developing a classification framework, based on support vector machines, with the correct properties to act as a decision support system for an industrial process plant, such as the Tennessee Eastman process. The system would provide support to the technicians who monitor plants by signalling the occurrence of abnormal plant measurements marking the onset of a fault condition. To be practical such a system must meet strict standards, in terms of low detection latency, a very low rate of false positive detection and high classification accuracy. Experiments were conducted on examples generated by a simulation of the Tennessee Eastman process and these were preprocessed and classified using a support vector machine. Experiments also considered the efficacy of preprocessing observations using Fisher Discriminant Analysis and a strategy for combining the decisions from a bank of classifiers to improve accuracy when dealing with multiple fault categories.**

I. Introduction

Modern process plants are highly instrumented and give rise to streams of measurements that portray the operational status of the plant at any point in time. This information, collected by networks of supervisory control and data acquisition (SCADA) devices, is normally relayed to a central control room where technicians monitor and adjust the plant. However the rapidly increasing quantity and frequency of plant measurements places a significant burden on the operators who may start to suffer from information overload. In such circumstances, operators may start to suffer from "cognitive tunnel vision", missing the subtle trends in the telemetry that mark the onset of a fault. Given the large scale of modern plants it can be very expensive to rectify faults that are not diagnosed accurately and promptly.

One solution to this problem is to introduce an automated decision support system (DSS) to the control room, which constantly monitors the plant and swiftly brings unusual operating conditions to the operators' attention. The DSS is not part of the control loop of the plant – it only triggers an alarm allowing the operators to determine whether a fault has occured and the appropriate remedial action.

For such a system to be effective a fault detection framework must be constructed that deals with a wide range of fault categories, is able to recognise faults from a very small set of previous examples, is able to detect faults promptly after their onset and is able to achieve a low rate of false positives. We refer to the delay between the onset of a fault and its detection as the *latency*. In industry it may be impractical to generate an extensive set of fault examples, since this would require repeatedly sabotaging a plant and so detection and diagnosis must rely on small sets of "accidental" observations of each fault category. Any DSS which generates too many false alarms will place extra load on the plant operators hence undermining the benefits.

This paper provides a case study of such a DSS constructed for the Tennessee Eastman Process (TEP) simulation [1]. The novel aspect of this paper stems from examining the full set of 20 faults, with low latency, small training sets and testing the sensitivity and specificity as well as the classification accuracy. It also proposes and evaluates a strategy for dealing with multiple fault categories by combining decisions from several classifiers using a secondary vector quantisation classification phase. These characteristics have not been adequately studied in previously reported research.

In [2] a number of multivariate statistical process control methods are evaluated, including methods applying principal component analysis (PCA), to 9 of the 20 faults categories of the TEP model. The plant was monitored using 16 of the 53 possible process variables, but some of the methods used a window size of 300 samples (corresponding to latency of up to 15 hours). No fault category could be detected with statistical reliability of more than 94.5%.

Three TEP faults categories are analysed in [3] using support vector machines and Fisher Discriminant Analysis (FDA) but only fault diagnosis is considered. Fault detection relative to normal operating conditions is not considered. Only 2 of the 53 process variables were needed to differentiate such a small subset of the faults. The smallest misclassification error observed in the tests is 2.5% when a window size of three samples was used (latency of 9 minutes).

Independent component analysis (ICA) is used as a front-end in [4] to try to estimate a small set of state variables that characterise the plant and these are classified using a support vector machine. The authors are mindful of the latency with the best result of 3 minutes and consider 12 fault categories with recognition rates of 100% for certain faults. ICA also forms the basis for [5] which is combined with FDA and classified using a nearest neighbour approach. ICA was applied to 51 process variables and reduced their dimension to 38 independent uncorrelated intermediate variables that were analysed by FDA. All fault categories are considered and performance is compared with other front-ends such as FDA and FDA/PCA hybrids from [6]. Misclassification rates reported vary widely from fault to fault, but no agregate classification results or latency measures are provided.

This work was supported by the Australian Research Council.

The authors are with the School of Electrical, Energy and Process Engineering, Murdoch University, WA 6150, Australia. Electronic mail: gareth.lee@murdoch.edu.au

ISBN 978-3-9524173-8-6

A moving window of variable width is used in [7] with process variables subjected to PCA and subsequently classified using a Hidden Markov Model for each of three fault categories. No agregrate classification results are presented but tests result in latencies between 1 and 20 minutes, due to the variable width window.

The DSS proposed here is based on the support vector machine (SVM) pattern classifier [8]. SVMs have very effective mechanisms for capacity control; they are able to construct the most general recognition strategy that befits the number of fault examples available. Thus they do not over-specialise when presented with small training sets, unlike previous artificial neural networks. In this respect they are a major step forward from previous pattern recognition approaches.

The following section introduces the theory relating to the support vector machine whereas Section III describes Fisher discriminant analysis. Section IV discusses the Tennessee Eastman process model which generates the training and test sets used in the experimental tests. Section V describes a strategy for dealing with multiple fault categories by post-processing the results using vector quantisation. Section VI describes the common aspects of the experimental tests that were undertaken and Section VII analyses the results in the context of decision support. Finally, section VIII summarises the results and draws conclusions regarding the practicality of constructing a decision support system for this process.

II. Support Vector Classification

The support vector machine [9] is a pattern classifier which is trained from a set of labelled examples, denoted $\{\mathbf{x}_i, y_i\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^P$ is a P-dimensional vector of measurements defining the i-th pattern and $y_i \in \{+1, -1\}$ is the corresponding label which selects between two classes. The support vector classifier (SVC) [8] [10] is a hyperplane classifier, which is able to discriminate between test examples of two classes by placing an affine decision boundary between them. Normally such a classification strategy offers very limited performance, since the clusters corresponding to the two classes may interlock in such a way that no affine boundary is able to separate them. However, the SVC overcomes this by mapping the examples from the P-dimensional input space into a much higher dimensional Hilbert space [8] using a function ϕ defined as $\phi(\mathbf{x}) : \mathbb{R}^P \mapsto \mathbb{H}_k$. The hyperplane classifier operates in this more voluminous Hilbert space $\mathbb{H}_k$ where examples are easier to classify.

Consequently the operation of the SVC may be expressed in terms of a hyperplane decision boundary whose function is $\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b = 0$, where $\langle ., . \rangle$ denotes an inner product of two vectors, $\mathbf{w}$ is normal to the hyperplane and b is the bias or nearest distance of the hyperplane from the origin. This implies that, for all the examples to be correctly classified, values of $\mathbf{w}$ and b must be found such that the set of constraints,

$$y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \geq 1, \quad i = 1, \ldots, N, \tag{1}$$

are satisfied. Since there may be many values of $\mathbf{w}$ and b which satisfy these constraints the SVC chooses the values that minimise $\|\mathbf{w}\|$, since this maximises the distance between the hyperplane and the nearest training examples. This is a form of regularisation which is theoretically justified by the Vapnik Chervonenkis Theory [9], a Statistical Learning Theory [11] on which support vector machines are based. It is based on the assumption that the finite set of training examples is drawn from an underlying distribution and that the most effective classifier, on unseen test examples, will be the one that maximises the margin (or distance) between the hyperplane and the known examples.

In practice there may be outliers from the two classes that the SVC cannot classify correctly, so it is also necessary to minimise the amount by which these examples are misclassified. The misclassification errors are represented in an optimisation problem by a set of slack variables, $\{\xi_i\}_{i=1}^N$ that are zero if no misclassification occurred and measure the extent of the error otherwise. The optimisation problem leading to optimal selection of $\mathbf{w}$ and b is therefore,

$$\min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad J_p(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i, \tag{2}$$

which is called the objective function and is subject to constraints,

$$y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \geq 1 - \xi_i, \quad i = 1, \ldots, N, \tag{3}$$

adapted from (1) to introduce the new slack variables. By including two terms in the objective function (2) it is possible to establish a trade-off between the complexity of the classifier and the magnitude of classification errors it makes. A simpler classifier may make greater errors on the training examples, but offer better generalisation on independent test examples. The trade off between the two terms is mediated by a regularisation constant C.

Equations (2) and (3) describe a a quadratic programming problem. It can be solved by creating an equivalent Lagrangian function L which combines the objective function and constraints, using a set of Lagrange multipliers denoted by $\{\alpha_i\}_{i=1}^N$ and constrained such that $\alpha_i \geq 0$:

$$L(\mathbf{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}) = J_p(\mathbf{w}, \boldsymbol{\xi}) - \sum_{i=1}^{N} \alpha_i \left(y_i (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) - 1 + \xi_i \right). \tag{4}$$

A solution can be found for this function which minimises the primal variables whilst maximising the Lagrange multipliers [12]. Determining the relationships between the primal variables of L at the optimum point leads to a Wolfe dual problem of the form,

$$\max_{\boldsymbol{\alpha}} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N} \alpha_i y_i \sum_{j=1}^{N} \alpha_j y_j \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \right\rangle, \tag{5}$$

subject to the constraints $0 \leq \alpha_i \leq C$ for $i = 1, \ldots, N$, and $\sum_{i=1}^{N} \alpha_i y_i = 0$. The dual problem is a more attractive quadratic program to solve as it depends on fewer variables and has simpler constraints.

One consequence of mapping examples into certain Hilbert spaces $\mathbb{H}_k$, is that there exists a related kernel function k such that $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. This is often called the "kernel trick" and allows the use of certain functions referred to as Mercer kernels [8], in place of the mapping function ϕ. A range of functions related to dot products or radial basis functions can be employed as kernels [10]. In the experiments reported here a Gaussian kernel was used of the form $k(\mathbf{x}_i, \mathbf{x}_j; \sigma) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\| \ / \ 2\sigma^2)$. This can be precomputed, for a specific training set, as a Gram matrix and substituted into (5).

Solving the dual problem is equivalent to solving the primal [12]. At the optimal value of (4) the gradient is zero with respect to all the parameters, so $\partial L / \partial \mathbf{w} = 0$, which implies that $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i)$. Consequently, once the optimal values of the Lagrange multipliers have been determined, they may be used in a classification function of the form,

$$y_p = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b, \qquad (6)$$

using the kernel trick described previously. If only considering two classes a definitive classification can be made by just considering the sign of y_p. In this work the magnitude is also considered and this will be examined further in Section V. The classification function (6) may be evaluated for any vector $\mathbf{x}$ to determine the predicted classification y_p. Comparison with the actual labels of a set of test examples allows the accuracy of the classifier to be estimated. Solving the dual quadratic program (5) does not give a value for b, but since this only acts as a bias or offset in (6) there are other simple techniques for determining the optimal value [10].

A useful side effect of minimising $\|\mathbf{w}\|$ is that α_i values for training examples $\mathbf{x}_i$ that are correctly classified and distant from the decision hyperplane are zero and so the right hand term of (6) becomes sparse, offering computational benefits when testing independent examples of $\mathbf{x}$. The small set of non-zero α_i values correspond to $\mathbf{x}_i$ examples referred to as the support vectors. These are the minimum set of training examples needed to anchor the discriminant hyperplane in the correct orientation in the Hilbert space. In the experiments described in Section VII, typically 10–20% of the training examples were selected as support vectors.

Numerical optimisation techniques exist for solving the dual quadratic program in (5) given a specific training set and prespecified values for C and, in the case of a Gaussian kernel, the kernel bandwidth parameter σ. In these tests we use the SVMLIB toolkit [13] and search across a range to determine good values for C and σ.

III. Fisher Discriminant Analysis

Fisher discriminant analysis (FDA) [6] [3] is a method that uses the covariance matrices $\mathbf{S}$ derived from example set $\{\mathbf{x}_i\}_{i=1}^{N}$ to determine a set of basis vectors that constitute a linear transform on $\mathbf{x}$. FDA assumes that each of the examples $\mathbf{x}_i$ is labelled by the corresponding $y_i \in C$ as belonging to one of two classes. $C = \{+1, -1\}$ is the set of possible class labels.

FDA determines a set of basis functions that project the examples into a lower dimension subspace such that the two labelled clusters of examples are separable. The basis functions are provided in order of decreasing efficacy and so it is possible to pick a subset comprised of the most effective basis functions, thereby mapping the examples into a lower dimensional space while losing minimal information that might compromise discrimination of the two classes. FDA is therefore used to preprocess high dimensional measurements prior to pattern classification and it has been used in some of the experiments described in Section VII.

The basis vectors $\mathbf{v} \in \mathbb{R}^m$ are determined as the solution to an optimisation problem,

$$\max_{\mathbf{v} \neq 0} \ \frac{\mathbf{v}^T \mathbf{S}_b \mathbf{v}}{\mathbf{v}^T \mathbf{S}_w \mathbf{v}}, \qquad (7)$$

where $\mathbf{S}_b$ is defined as the inter-class scatter matrix and $\mathbf{S}_w$ is the intra-class scatter matrix. The problem can be solved using the generalised eigenvalue method [14].

The intra-class scatter matrix $\mathbf{S}_w$ is defined as the sum of the covariance matrices of the two classes, $\mathbf{S}_w = \sum_{j \in C} \mathbf{S}_j$, where $\mathbf{S}_j$ is the covariance matrix for the set of examples which are labelled as belonging to class j:

$$\mathbf{S}_j = \sum_{\{\mathbf{x}_i \,:\, y_i = j\}} (\mathbf{x}_i - \bar{\mathbf{x}}_j)(\mathbf{x}_i - \bar{\mathbf{x}}_j)^T, \qquad (8)$$

assuming $\bar{\mathbf{x}}_j$ is the mean of the set of examples belonging to class j.

Conversely the inter-class scatter matrix is defined as the variance of the set of mean values of each of the classes:

$$\mathbf{S}_b = \sum_{j \in C} (\bar{\mathbf{x}}_j - \bar{\mathbf{x}})(\bar{\mathbf{x}}_j - \bar{\mathbf{x}})^T, \qquad (9)$$

where $\bar{\mathbf{x}}$ is the mean of the entire set of examples.

Assuming that the observations comprising each class form a compact cluster in a Euclidean space, maximising the objective function (7), involves finding the projection from the space that maximises the mean distance between the cluster centres whilst simultaneously minimising the covariance (or spread) of each of the clusters. This leads to the projection that best separates the classes.

IV. The Tennessee Eastman Plant

The Tennessee Eastman process (TEP) simulator models a complex exothermic process created by the Eastman Chemical Company [1]. The process, shown in Fig. 1, consists of five major units: a reactor, a product condenser, a vapour-liquid separator, a product stripper and a recycle compressor. The process was originally posed as a challenge to control system designers, but has been widely used in previous publications for comparing process monitoring and diagnosis techniques [2] [3] [4] [5] [7].

The plant is open-loop unstable and therefore requires the addition of a control framework, such as the tiered single-input single-output approach of Lyman and Georgakis [15].

An implementation of [15] provided by Chiang et. al. [6] was used to generate training and test observations for the experiments reported here. The process has 12 manipulated variables, used to control the plant, and 41 measured variables of which 22 are continuously measured plant parameters and 19 are periodic measurements of reactant concentrations in the input, recycle and output streams. The simulation generates samples of the 22 continuously measured variables every 3 minutes and the remaining 19 periodic measurements are updated every 6 or 15 minutes (depending on which stream is being analysed).

The TEP simulation provides 20 disturbance variables that activate fault conditions during operation. The 20 faults fall into 5 distinct categories:

- Seven faults (1-7) are provided which correspond to step changes in the feed reactant composition, feed rates and temperatures.
- Another five faults (8-12) are implemented that are similar to the previous categories but result from random variations in feed compositions and temperatures.
- One fault (13) models a slow drift in reactor kinetics.
- Two further faults (14-15) correspond to sticking valves within the cooling water systems.
- The final five fault categories (16-20) are undocumented.

The TEP simulation was run 260 times for each fault from different pseudo-random number generator seed values. The random number generator controls a pseudo-Gaussian noise source which is added to the measurements made at each of the sensors. This is then fed-back through the control and reactant flow loops to influence the future state of the plant. Each of the TEP simulations was run for a period of 10 simulated hours under normal operating conditions prior to activating the fault. At a specified elapsed time E after the fault was activated, a window of W successive observations was excised from the simulation outputs. Each observation vector consisted of the 41 measured variables concatenated with the 12 manipulated variables to give a $P = 53$ dimensional observation vector. This arrangement is shown in Fig. 2. When the window size is greater than one, several successive observations are concatenated over time to generate a vector of dimension $53W$. For instance setting $W = 2$ creates an observation vector from two successive sample times (separated by 3 minutes) and therefore provides information, not just about the position of the observation vector in its 53 dimensional space, but about its trends.

All the experiments reported here assumed an elapsed time $E = 1$, meaning that the fault was detected at the earliest possible time that it affected the plant measurements. This is the most difficult case, since certain faults result in exponential trends in the plant measurements away from their normal operating points. Since these perturbations have such a small magnitude shortly after their onset, the new trends can easily be overwhelmed by sensor noise, resulting in unreliable detection. The window size W was varied between 1 and 4 so that the latency varied from 3 in the best case to 15 minutes in the worst case.

The 260 examples of each fault were randomly partitioned into disjoint training and test sets. For initial experiments 50 of the examples were used for training whereas the remaining 210 were used for testing, but in later experiments the training set fraction was progressively reduced from 50 down to 6 examples. This allowed the fault detection performance to be assessed in cases where a fault occurred rarely and therefore only a small set of examples had previously been encountered.

V. Multiple Category Classification

One weakness of support vector classifiers (SVCs) for pattern recognition is that each SVC can only discriminate between two classes. Process plants typically generate faults falling into many categories.

To classify examples comprised of F fault categories, it is possible to create a network of $F(F-1)/2$ SVCs which compare all possible pairs of faults and then combine their classifications using a majority voting scheme (called the "1-versus-1" method) [10]. Alternatively it is possible to train F SVCs so that each discriminates between a specific fault category and examples of all other categories [10] and select the classifier with the strongest response (the "1-versus-ALL" case) [10]. This later approach requires more effort to train each of the classifiers since they must typically learn a more complex classification problem. But since the number of classifiers grows linearly with the number of faults, rather than as a square in the 1-versus-1 case, the second approach scales better when many categories must be considered. Experimentally the 1-versus-1 method was found to offer little benefit for this problem compared with 1-versus-ALL and so the latter technique was adopted in all experiments.

Despite each 1-versus-ALL classifier being trained to detect a single fault, similarity between the faults suggests that a classifier may offer discrimination for faults other than that it was trained to detect. Thus a partial response from multiple classifiers may offer greater confirmation of the occurrence of a fault than a strong response from a single classifier alone. This premise is tested in the experiments by basing classification decisions on the pattern of outputs of the entire bank of 1-versus-ALL classifiers, rather than just picking the classifier with the strongest response.

In this regard the outputs of the bank of SVCs is treated as a new pattern vector and further classified using a vector quantisation approach. The response vector consists of the classifier outputs (6) with respect to a bank of 1-versus-ALL classifiers which have been trained from F different faults categories denoted C_1 to C_F.

$$\mathbf{r} = [y_p^{C_1}, y_p^{C_2}, \ldots, y_p^{C_F}] \in \mathbb{R}^F \tag{10}$$

Since no sign function is applied at (6), as is normally the case for SVC, the vector elements describe the level of confidence displayed by each of classifiers. In each case they are implicitly normalised by their classifier's margin.

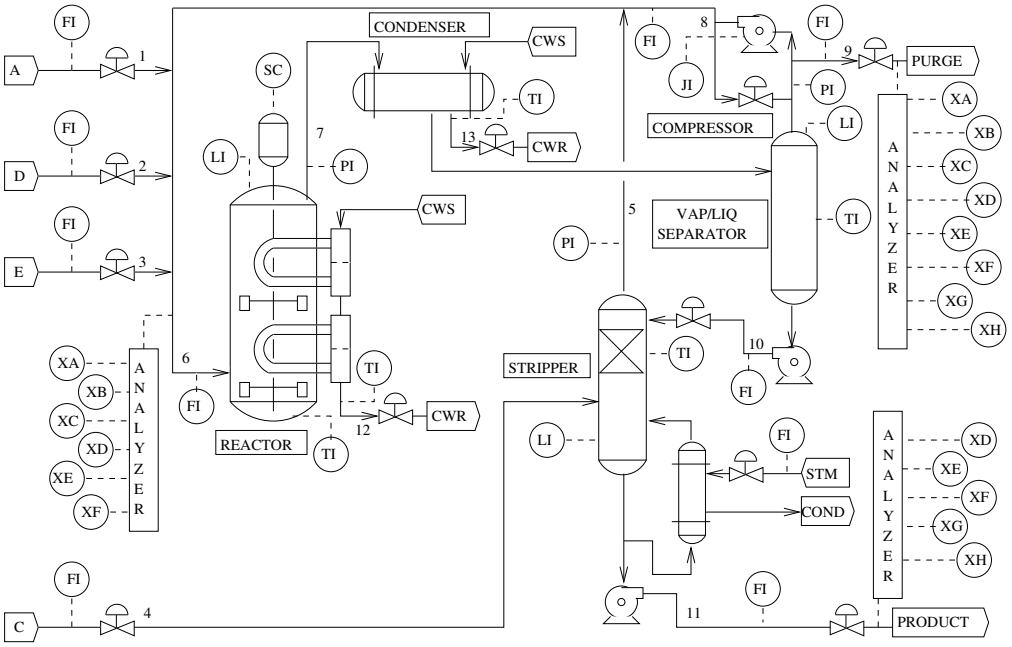

Fig. 1. A schematic of the Tennessee Eastman process.

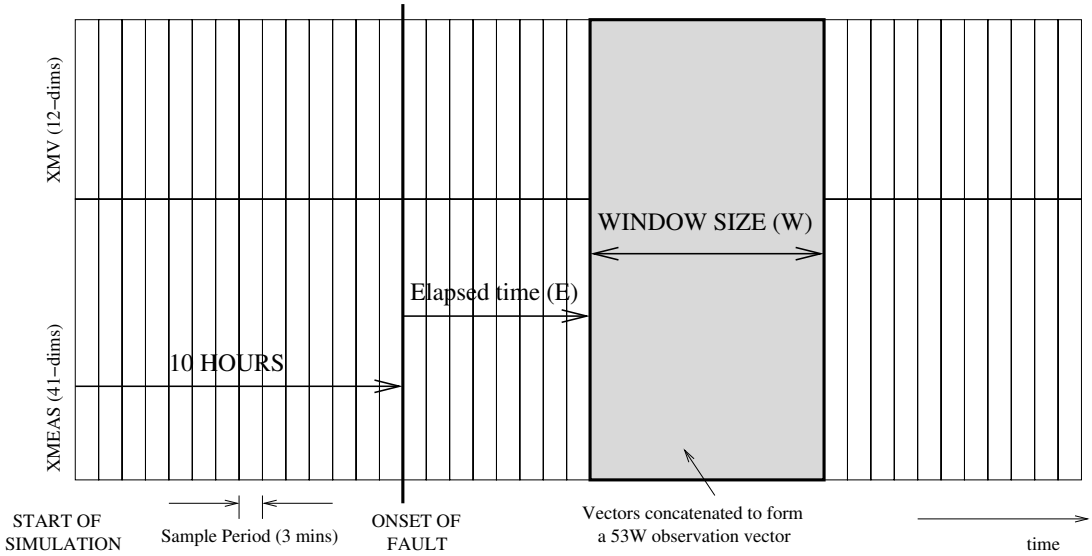

Fig. 2. The window arrangement for collecting data from the TEP simulation.

The approach adopted here is to map the F-dimensional vector $\mathbf{r}$ of output values from the SVC onto the $(F-1)$-dimensional surface of a unit hypersphere as $\mathbf{h}$. This simply involves scaling vector $\mathbf{r}$ to unit magnitude: $\mathbf{h} = \mathbf{r} \;/\; |\mathbf{r}|$. This preserves the ratios of the outputs of the various classifiers. The surface of the hypersphere acts as a Hilbert space $\mathbb{H}_s$ in the sense that distances between vectors can be determined by an inner product operation.

Given the training set labels it is then possible to adopt a vector quantisation approach operating within $\mathbb{H}_s$. A codebook B is constructed from a set of cluster centres and stored to enable subsequent classification.

$$B = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_F\} \tag{11}$$

A centre $\mathbf{c} \in \mathbb{H}_s$ is estimated for the cluster arising from each category of mapped training examples. Thus $\mathbf{c}_i$ is the archetypal pattern of responses from the bank of classifiers when experiencing patterns labelled as class C_i.

Hence, given a new unlabelled example it is first processed with each of the 1-versus-ALL classifiers, leading to an F-dimensional response vector $\mathbf{r}$. This vector is then mapped onto the hypersphere as $\mathbf{h}$ and its inner product is taken to each of the centroids corresponding to known class types. The centroid with the smallest angle to the example is deemed to be the "closest fit", leading to classification

decision C_d where,

$$d = \underset{i=1,\ldots,F}{\operatorname{argmin}} \langle \mathbf{h}, \mathbf{c}_i \rangle . \quad (12)$$

As will be shown in subsequent experiments this approach significantly boosts classification accuracy when dealing with multiple fault categories with minimal additional computational cost. In the experimental results that follow this approach is labelled as hypersphere vector quantisation (HVQ).

VI. Experimental Framework

In each experiment a training set, of appropriate size, was randomly selected from the 260 examples available for each fault category. The results presented are based on 50 independent trials, with each choosing a different training subset. In each case all the examples not selected for the training set were used for testing, thus ensuring that the training and test sets for any trial were disjoint. Consequently the results presented consist of the mean accuracy achieved over the 50 trials and the standard deviation, which measures repeatability. This technique has been used here as an alternative to cross-validation, since it allowed the size of the training set to be scaled to a non-integer fraction of the total examples. It also allowed greater statistical significance to be achieved by running more trials than cross-validation would provide and therefore was deemed more versatile for use in these experiments.

For a classification scheme to be employed in a decision support framework it is essential that the rate of false positives is minimised. A false positive classification occurs when measurements arising from normal operating conditions are misclassified as belonging to one of the fault categories. Past experience has shown that decision support systems that frequently generate false positives lose the confidence of the human operators, who start to distrust their recommendations. In statistics, false positives are measured by the specificity metric, defined as the number of true negative cases divided by the sum of the true negative and false positive cases. If the specificity drops significantly below 100% it is a sign of the prevalence of false positives. Conversely sensitivity is defined as the number of true positives divided by the sum of true positive and false negative cases. If the sensitivity drops below 100% it indicates a tendency to create false negative classifications; the classifier is failing to detect faults when they do occur. In a decision support system this case, whilst still undesirable, is less critical since the operators will often independently detect the fault condition.

The specificity and sensitivity metrics only consider whether a fault has been detected, not whether it was correctly diagnosed. The final metric is accuracy which is defined as the number of correctly diagnosed examples divided by the total number of test examples. When it comes to accuracy, measurements arising from normal operating conditions (NOC) are considered to be a special "fault" category which must be correctly diagnosed just as the other faults. The accuracy results are unbiased since equal numbers of test examples were considered for each of the fault categories (including NOC).

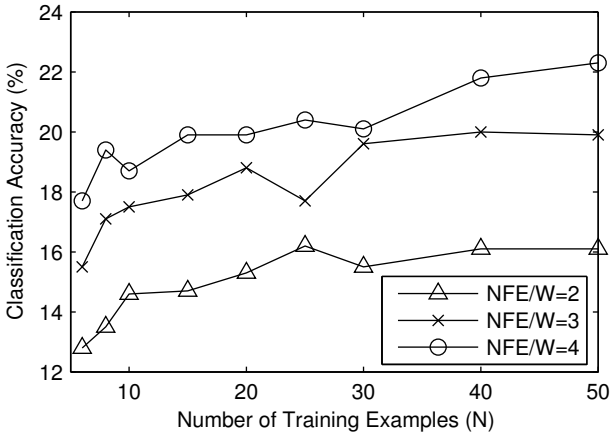

Fig. 3. Classification accuracy achieved with no front end (NFE) as the training set size is reduced from 50 examples to 6. Results are shown for window sizes $W = 2 - 4$; the results with $W = 1$ were little better than random guessing. Also see Fig. 7 for a summary of results.

Prior to classification a normalisation transformation T was applied to measurement vectors, $T : \mathbf{x} \mapsto \mathbf{x}'$, that imposed unit standard deviation and yet preserved the mean:

$$x'_j = ((x_j - \mu_j)/\sigma_j) + \mu_j \quad \text{for } j = 1, \ldots, 53W, \quad (13)$$

where $\mathbf{x} = [x_1, \ldots, x_{53W}]$ is an original example vector, $\mathbf{x}' = [x'_1, \ldots, x'_{53W}]$ is its normalised form. This was found to improve separability and hence classification performance, but in cases where a small number of training examples were available inaccuracies in the estimates of μ_j and σ_j may impose a penalty on performance. This will be the subject of a future investigation.

VII. Analysis of Results

The experiments reported here are concerned with classification of independent test examples of $F = 21$ category data (20 faults + NOC) after training on a set containing N examples of each fault category. Classification was performed in all examples using the 1-versus-ALL scheme.

Initially examples involved normalised (13) examples but no dimension reducing front-end was applied to the training or test examples. In each case the number of training examples was reduced from 50 down to 6 examples of each fault category. The results of this experiment are shown in Fig. 3, and are clearly too inaccurate for use in any form of Decision Support System (DSS). Interestingly the accuracy does not diminish radically as N is curtailed.

For small N the classification task is more difficult, but the SVM has very effective capacity control – it can generalise effectively from a small set of examples – as demonstrated in these results. Moreover the mean and standard deviation estimates used by the normalisation stage (13) will deteriorate as N becomes small but clearly this has a relatively benign impact on the classification accuracy. These two effects apply equally to subsequent experiments.

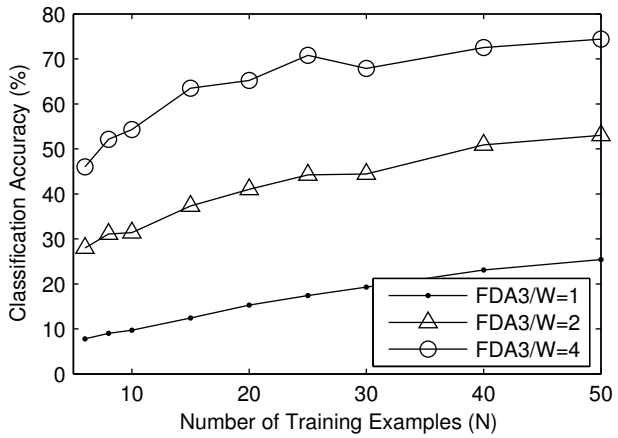

Fig. 4. Classification accuracy achieved with a Fisher Discriminant Analysis front end reducing the feature space dimension to 3 (FDA3) and as the training set size is reduced from 50 examples to 6. Results are shown for window sizes $W = 1 - 4$. Also see Fig. 7 for a summary of results.

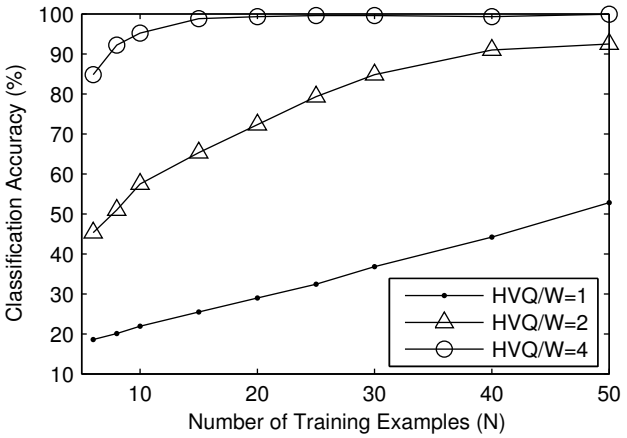

Fig. 5. Classification accuracy achieved when combining hypersphere vector quantisation (HVQ) with the FDA3 front end as the training set size is reduced from 50 examples to 6. Results are shown for window sizes $W = 1 - 4$. Also see Fig. 7 for a summary of results.

In each case a good value was chosen for the SVM regularisation constant C and kernel bandwidth σ by searching values in the geometric series $(\sqrt{10})^n$ for $n = 6, \ldots, 10$.

A further set of tests were used to compare the results with those achieved by including a Fisher Discriminant Analysis (FDA) front-end, after normalisation as described in Section III. FDA created a linear transform that was used to reduce the feature dimension to 3 prior to classification and was denoted "FDA3" in the results. Previous experiments had shown that this is the optimal input space dimension reduction to use for this data set. Fig. 4 illustrates how inclusion of the front end improved recognition accuracy by around 50% compared with the non-front end examples. However with error rates of 30% or more these results fail to meet the exacting standards required by a DSS.

When using the FDA front-end the results not only have to contend with the two effects observed in the previous experiment as N becomes small, but also with the impact on the front-end itself. The FDA front-end is based on an estimate of the covariance matrix of the examples within the training set, which will become imprecise as N becomes small leading to a drift away from the optimal transform. This leads to a proportionately larger drop in classification accuracy as N is curtailed for this experiment compared with the previous one.

The previous experiments were further extended by adding an additional classification phase involving mapping onto a unit hypersphere and vector quantisation as described in Section V. The results are shown in Fig. 5 and demonstrate that by merging the decisions of multiple classifiers the misclassification rate can be reduced substantially. So long as a window size of 4 sample periods (resulting in latency of 12-15 minutes) is allowed and 10 or more examples of each fault category are available, error rates less than 5% can be achieved. Given 20 or more examples of a fault error rates less than 1% can be achieved. At this level the approach becomes feasible as the basis for a DSS.

N	FDA3 Sensitivity	FDA3 Specificity	HVQ Sensitivity	HVQ Specificity
6	97.06	37.31	99.00	75.54
8	95.13	52.36	99.41	91.27
10	97.32	49.00	99.75	93.60
15	97.46	53.08	99.89	97.46
20	97.40	69.63	99.96	99.04
25	97.20	71.36	99.98	99.69
30	96.10	86.36	99.97	99.16
40	97.47	74.50	99.99	99.01
50	98.46	78.32	99.99	99.99

Fig. 6. Sensitivity and Specificity for the FDA3 and HVQ methods as the training set size (N) is reduced from 50 down to 6. All results are percentages and apply to window size $W = 4$.

In light of the accurate results obtained by the previous experiments the sensitivity and specificity achieved by the window size $W = 4$ tests from Figs. 4 and 5 were analysed and presented as Fig. 6. Again each result was the average of 50 trials.

It can be seen how the inclusion of the HVQ stage increases the specificity from 49.0% with FDA3 with $N = 10$ training examples to 93.6% then the additional vector quantisation approach is included. Using HVQ therefore reduces the rate of occurrence of false positive events by a factor of eight. As discussed in Section VI the rate of false positives is critical to the efficacy of any DSS, since this is the case when "phantom" faults are reported when the plant is actually operating normally. Similarly the sensitivity had increased from 97.3% to 99.7% leading to a decrease in false negatives by a factor of 9. This is a sign that the system more reliably detects faults that do occur. A summary of the classification results obtained is provided in Fig. 7.

To put these results in context even a specificity of 93.6% is inadequate for a practical DSS since it would still misclassify about 7% of the measurements corresponding to normal operating conditions. However when 20 or more training

W	Latency	Method	N=6	N=8	N=10	N=15	N=20	N=25	N=30	N=40	N=50
2	6	NFE	12.8	13.5	14.6	14.7	15.3	16.2	15.5	16.1	16.1
2	6	FDA3	28.0	31.1	31.4	37.3	41.0	44.2	44.4	50.9	53.0
2	6	HVQ	45.3	51.0	57.5	65.3	72.3	79.3	84.8	91.0	92.5
4	12	NFE	17.7	19.4	18.7	19.9	19.9	20.4	20.1	21.8	22.3
4	12	FDA3	46.0	52.1	54.3	63.5	65.2	70.8	67.9	72.5	74.4
4	12	HVQ	84.8	92.2	95.2	98.8	99.3	99.6	99.6	99.3	99.9

Fig. 7. Summary of the performance obtained from the three methods discussed in Section VII. All classification accuracies are presented as percentages. Experimental parameters are the number of examples of the fault examples (N) expressing the scarcity of the fault and window size (W) determining the maximum classification latency (in minutes).

examples of a fault are available the specificity increases to greater than 99%. Also in these experiments confidence margins and/or a priori statistics have not been included in the classification model. By including these additional thresholds the rate of false positives can be further reduced, albeit at the cost of increased false negatives. Future work will consider the application of receiver operating characteristic curves to optimise the trade off between sensitivity and specificity.

Moreover, as previous authors have observed [2] a small number of the TEP fault categories overlap with other faults or with normal operating conditions in such a way that it is impossible to achieve zero misclassifications. It would be possible to omit certain fault categories and boost the classification accuracy substantially. Thus the Tennessee Eastman problem may be more exacting than many industrial problems that the DSS approach could be applied to.

VIII. Conclusion

The experiments presented here have investigated whether examples, generated from the Tennessee Eastman Process (TEP) simulation, can be classified with sufficient accuracy to allow a decision support system to be built to assist plant operators in their duties. By investigating a number of different front-end configurations (testing with and without Fisher Discriminant Analysis) and window sizes, resulting in small detection delays (latency from 3–12 mins), it has been shown that such a system is feasible (see Fig. 7).

However, clearly there are still some obstacles that need to be overcome in the development of a final system. The previous experiments have demonstrated a number of useful attributes of the techniques that have been evaluated here, when used in a decision support system:

- The capacity control offered by support vector machines makes them well suited to generalise from small sets oftraining examples;
- A dimension reducing front-end such as Fisher Discriminant Analysis can improve performance by distilling information from high dimensional observation vector, thus overcoming the "curse of dimension";
- Use of a post-processing step such as the Vector Quantisation approach described here can offer substantially improved accuracy and major improvements in specificity and sensitivity.

It is unclear whether the high level of results obtained on the artificial data set arising from the TEP simulation will generalise to other processes, but it appears that the process and procedures examined here will be applicable to a wide range of process plants. This paper has shown that the full benefits of SVM-based fault detection can be unlocked by applying very simple normalisation transformations to the measured data.

References

[1] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[2] M. Kano, K. Nagao, S. Hasebe, I. Hashimoto, H. Ohno, R. Strauss, and B. R. Bakshi, "Comparison of multivariate statistical process monitoring methods and applications to the eastman challenge problem," *Computers & Chemical Engineering*, vol. 26, no. 2, pp. 161–174, Feb. 2002.

[3] L. H. Chiang, M. E. Kotancheck, and A. K. Kordon, "Fault diagnosis based on fisher discriminant analysis and support vector machines," *Computers & Chemical Engineering*, vol. 28, no. 8, pp. 1389–1401, 2004.

[4] M. Guo, L. Xie, S. Wang, and J. Zhang, "Research on an integrated ICA-SVM based framework for fault diagnosis," in *Proceedings of the International Conference on Systems, Man, and Cybernetics*, vol. 3. Washington DC: IEEE, Oct. 2003, pp. 2710–2715.

[5] L. Jiang and S. Wang, "Fault diagnosis based on independent component analysis and fisher discriminant analysis," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 6. Shanghai: IEEE, Aug. 2004, pp. 3638–3643.

[6] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. Springer, 2001.

[7] S. Zhou, L. Xie, and S. Wang, "A variable moving window approach for on-line fault diagnosis in industrial processes," in *Proceedings of the World Congress on Intelligent Control and Automation*, vol. 2. Hangzhou: IEEE, June 2004, pp. 1761–1765.

[8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Knowledge Discovery and Data Mining*, vol. 2, no. 4, pp. 121–167, 1998.

[9] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 2000.

[10] B. Schölkopf and A. Smola, *Learning with Kernels*. The MIT Press, 2002.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[13] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 1992.

[15] P. R. Lyman and C. Georgakis, "Plant-wide control of the tennessee eastman problem," *Computers & Chemical Engineering*, vol. 19, no. 3, pp. 321–331, 1995.