Nikus, Mats; Vermasvuori, Mikko; Vatanski, Nikolai; Jämsä-Jounela, Sirkka-Liisa

Support vector machines for detection of analyzer faults- a case study

Published in:
ALSIS 2006, Finland, 2006

Published: 01/01/2006

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Nikus, M., Vermasvuori, M., Vatanski, N., & Jämsä-Jounela, S.-L. (2006). Support vector machines for detection of analyzer faults- a case study. In L. Leiviskä (Ed.), *ALSIS 2006, Finland, 2006* Suomen Automaatioseura.

Author's accepted manuscript, published in Proceedings of the Applications of Large Scale Industrial Systems (ALSIS´06), 2006, Helsinki, Finland

SUPPORT VECTOR MACHINES FOR DETECTION OF ANALYZER FAULTS – A CASE STUDY

Mats Nikus, Mikko Vermasvuori, Nikolai Vatanski, Sirkka-Liisa Jämsä-Jounela

Aalto University
Laboratory of Process Control and Automation
Kemistintie 1, FI-02150 HUT, Finland

Abstract: The aim of the work presented in this paper is to assess the ability of support vector machines (SVM) for detecting measurement faults. Two different support vector machine approaches for detecting faults are tested and compared to neural networks. The first method is based on a SVM regression model together with an analysis of the residuals whereas the second method is based on a SVM classifier. The methods were applied to a rigorous first principles based dynamic simulator of a dearomatization process. *Copyright © 2006 IFAC*

Keywords: Fault detection, monitoring, support vector machines, classification, regression, dearomatization process

1. INTRODUCTION

Handling of abnormal situations, such as equipment failures and process disturbances, has received increasing attention from industry and academia alike. Due to abnormal situations the petrochemical industries alone lose an estimated 20 billion dollars annually (Venkatasubramanian et al., 2002). The potential benefits, even from modest improvements in abnormal situation handling, are hence enormous. One important group of abnormal situations in process industries are faults in on-line product analyzers. Since these analyzers are increasingly used for closed-loop control it is imperative to detect faults occurring in them as quickly as possible.

Recent developments in kernel methods have made available efficient tools for non-linear classification and regression. One of these powerful techniques is the support vector machines method (Vapnik, 1998).

The aim of this paper is to assess the ability of support vector machines for detecting measurement faults. The paper is organized as follows. In the next section the basic support vector method is discussed and the discussion is followed by a description of two approaches for fault detection using SVMs. Section 3 introduces the studied process while the data processing, model building and validation steps are presented in section 4. The fault detection results are given in section 5. A comparison of the results to those of standard feedforward sigmoidal neural networks is made in section 6 and finally the conclusions are drawn and summarized in section 7.

2. SUPPORT VECTOR MACHINES

The support vector machines (SVM) were created by Vladimir Vapnik in the 1990s and can be used for solving classification and regression tasks. They are based on the principle of structural risk minimization, which enhances model robustness by ensuring that the model complexity is not too high as measured by the so called VC-dimension (Vapnik, 1998). For comparison, neural networks and other traditional black box techniques normally minimize the empirical risk which basically is the average quadratic error over a number of samples, the training set. Within the SVM framework, radial basis networks, single hidden layer sigmoidal neural networks as well as other kinds of models can be set up, depending on the chosen kernel. A nice property of the SVM is that it yields a unique optimal solution of the resulting optimization problem.

When solving the classification problems, a decision surface of the form

$$w^T\phi(x)+b=0 \qquad (1)$$

is sought. The basis function $\phi(x)$ maps the inputs to a high dimensional feature space. Minimizing the so- called structural risk leads to the inclusion of the parameter vector *w* in the cost function

$$\frac{1}{2}w^T w+C\sum_{i=1}^{l}\xi_i \qquad (2)$$

which is minimized with respect to *w* and *b*, subject to the constraints

$$\begin{aligned} &y_i\left(w^T\phi(x_i)+b\right)\geq 1-\xi_i \\ &\xi_i\geq 0, i=1\ldots l \end{aligned} \qquad (3a\text{-}b)$$

where x_i represents an input vector in the data set and y_i (= ±1) the corresponding scalar output. The ξ_i:s are slack variables representing misclassifications. In practice the solution of the SVM optimization problem is solved by introducing a dual problem that arises after the inclusion of Lagrange multipliers. The dual formulation gives rise to a quadratic programming optimization problem that has a unique solution. This yields for the classification case the decision function

$$f_d(x)=\mathrm{sgn}\left(\sum_{i=1}^{l}y_i\alpha_i\phi(x_i)^T\phi(x)+b\right) \qquad (4)$$

where the α_i are the Lagrange multipliers of the dual problem and *l* is the number of support vectors. The support vectors are data vectors selected from the training set to form the basis of the model. When solving the dual problem it turns out that the basis function $\phi(x_i)$ is only present as an inner product in the solution. Hence only the kernel function $K(x_i, x)$ needs to be known and the basis functions $\phi(x_i)$ are not used explicitly (the "kernel trick"). Several kernels have been proposed and the following are just examples sampled from the plethora.

$$\begin{aligned} K(x_i,x)&=\exp\left(-\gamma\|x-x_i\|^2\right) \\ K(x_i,x)&=\tanh\left(\beta_0 x^T x_i+\beta_1\right) \\ K(x_i,x)&=\left(\beta_0 x^T x_i+\beta_1\right)^p \end{aligned} \qquad (5a\text{-}c)$$

The gaussian kernel (equation 5a) gives rise to a radial basis network. With a sigmoidal kernel (equation 5b) perceptron networks much like feedforward neural networks with one hidden layer can be designed within the SVM framework. Also polynomial and linear kernels (5c) can be used.

Regression problems have the primal cost function

$$\frac{1}{2}w^T w+C\sum_{i=1}^{l}\xi_i+C\sum_{i=1}^{l}\xi_i^* \qquad (6)$$

that is minimized with respect to *w*, *b* and ξ, subject to the constraints

$$\begin{aligned} &w^T\phi(x_i)+b-y_i\leq\varepsilon+\xi_i \\ &y_i-w^T\phi(x_i)-b\leq\varepsilon+\xi_i^* \\ &\xi_i\xi_i^*,\geq 0, i=1\ldots l \end{aligned} \qquad (7a\text{-}c)$$

The inclusion of ε in the above constraints facilitates the so-called ε-insensitive cost function. This means that the absolute value of the residuals have to exceed ε before they are included in the cost function.

For the regression case the input-output mapping becomes:

$$f(x)=\sum_{i=1}^{l}\left(\alpha_i^*-\alpha_i\right)K(x_i,x)+b \qquad (8)$$

were the α_i and α_i^* represent the Lagrange multipliers in the dual problem.

The training of the SVMs can be performed with e.g. the LIBSVM software (Chang and Lin, 2001), which uses a numerically powerful decomposition method much like the method by (Platt, 1998). The decomposition method for the SVMs makes it possible to use large data sets with thousands of data points.

There are however some parameters in the SVMs that need to be determined outside of the main optimization. Specifically they are the width parameter of the gaussian *γ*, the weighting factor between model complexity and performance, *C* and for the regression case also *ε*, i.e. the threshold value for the residuals to contribute to the cost function (cf. equations (7a-b)). The best values for these parameters can be determined by performing a grid search on test data. The optimization time required for SVMs is typically an order of magnitude shorter than that of neural networks trained with the efficient Levenberg-Marquardt method. The SVMs however tend to give rather large models with the number of support vectors sometimes as high as one half of the number of patterns.

2.1. Support Vector Machines for Fault detection

Not many results of using SVMs for fault detection have been reported as of yet. Pöyhönen et al., (2005) reported a study related to the fault detection of electrical motors. In the article sets of classifier SVMs are used together to classify abnormal situations into several faults types. In (Kulkarni et al., 2005) prior knowledge is incorporated into a classifier SVM model used for fault detection of the Tennessee Eastman benchmark process. The inclusion of process expertise made it possible to use less complex models for the task. A liquid-liquid extraction process was monitored in (Jemwa and Aldrich, 2005) using a kernel ridge-regression technique similar to support vector machines. Ribeiro (2005) has reported good results using multi-class SVMs for fault detection in a plastic injection moulding process with the SVMs generally outperforming radial basis function networks designed for the same purpose. In (Wang et al., 2006) some promising kernel based methods for fault detection are outlined and applied on the Tennessee Eastman process and a dearomatisation process.

The two main categories of SVMs (classification and regression) can both be used for fault detection using different approaches. The regression approach is based on analytical redundancy and hence we want to model a possibly faulty signal (*y*) using other process variables (*x*)

$$\hat{y} = f(x) \tag{9}$$

The above model should preferably be trained on fault-free data. The difference between the measurement and the estimate gives the residual.

$$\varepsilon = y - \hat{y} \tag{10}$$

By evaluating this residual conclusion can be drawn about the state of the modelled signal. The simplest test for residual is its comparison to a predetermined threshold value. If the residual exceeds this limit a fault has been detected. Also other residual evaluation methods, such as e.g. the CUSUM test (Hinkley, 1971) can be used, but in the present study the former method was applied.

In the classification approach the idea is to model the fault state δ (δ = 1 = fault, δ = 0 = no fault) based on the process variables as well as the analyzer measurement itself.

$$\delta = g(x, y) \tag{11}$$

In the case of a fault classifier, the training data obviously should include faulty data in addition to the healthy data and thus the fault state of the data has to be known.

3. CASE STUDY

This case study deals with the fault detection of on-line analyzers in a simulated dearomatization process. Vermasvuori et al. (2005) studied a number of fault detection methods for this specific process. The purpose of the present case study is to investigate the use of support vector machines for the fault detection task. The study has been limited to bottoms product boiling point analyzer (0% evaporated) while the previous study by Vermasvuori et al. included in total four analyzers. Dearomatisation processes are widely used in the oil refining industry.

In these processes, aromatic compounds in the feedstock are removed by hydrogenation. The process consists of two tricklebed reactors with packed beds of catalyst, a distillation column, a filling plate stripper, several heat exchangers and separation drums, and other unit operations. The process diagram of the LARPO process is presented in Fig. 1.

The cold, liquid feedstock fed to the unit is heated with streams from the two reactors in heat exchangers EA1 and EA2, and then fed to reactor DC1 together with hydrogen and recycle liquid. Exothermic saturation reactions in the first reactor remove most of the aromatic compounds when the catalyst is new, while most of the reactions occur in the second reactor when the catalyst is older and has been partly deactivated. After dearomatisation in reactor DC1, the reaction product is cooled in heat exchanger EA1 and then fed to gas separation drum FA1 where the gaseous and liquid reaction products are separated. Part of the liquid is circulated back to reactor DC1. The rest of the liquid, together with separated gas and fresh hydrogen, are fed to the second reactor DC2, where the aromatics level of the product drops to near zero. After the second reactor, the reaction product is cooled in heat exchangers EA2 and EA3 and fed to the second gas separation drum FA2. Gas separated from the liquid mainly consists of unreacted hydrogen, which is recycled back to the first reactor, and the rest of the gas is removed. The separated liquid is heated with by-product and product streams in heat exchangers EA4 and EA5. Part of the liquid is further heated in heat exchanger EA6 in order to achieve the final temperature before the stream is fed to distillation column DA1. The overhead of the column is cooled in a cooler and then fed to separation drum FA3, where the gaseous part is removed and the liquid is divided into reflux and distillate. The distillate consists of the lightest compounds of the reaction product. Heat exchanger EA6 produces heat for reboiling the bottom stream. From the upper part of the column DA1, a side stream is conducted to a stripper, which is heated with heat exchanger EA7. A by-product stream is drawn off from the bottom of the stripper. The nonaromatic main product is drawn off as the bottom product of the distillation column DA1 and cooled in heat exchanger EA5.

The quality of the cooled product is measured online by flash point and distillation curve analyzers. Laboratory measurements of the product are performed twice a day, and these results are compared to the analyzer output. This way of detecting analyzer faults is tedious and can in the worst of cases lead to delays in the detection of possible faults of many hours. If the fault could be detected earlier the necessary maintenance actions could be made sooner and the quality of the end product could be kept within the production limits, thus improving the plants economical performance.

The analyzer faults are usually caused by one of the following problems; water contamination of the analyzed sample, carbonisation of the flask of the distillation analyzer, or fouling of the gas chromatograph. The first fault type causes the analyzer result to drop abruptly and the others cause the output to drift slowly away from the correct value. These fault types were simulated by adding biases and linear trends to the simulated analyzer output. The data, covering a period of almost 64 hours, used in this study, was created with the PROSimulator software developed by Neste Jacobs Oy. Every hour the values of 1−3 variables were manipulated in order to create variance in the data. These manipulations initiated changes to process measurements similar to changes in real process data caused by normal operation actions. The feed type was not changed during the simulation.

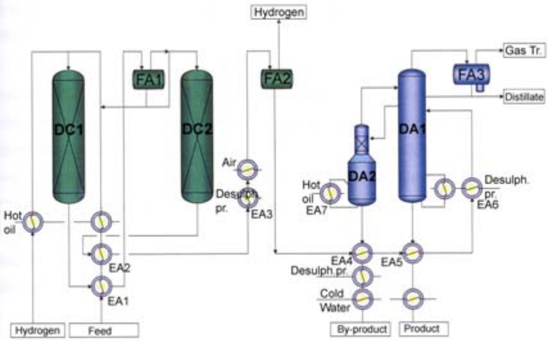

Fig. 1 The dearomatization process

4 DATA PROCESSING AND MODELING

4.1 Selecting the input variables

For any type of modelling a good choice of input variables is a crucial step. If the number of possible input variables is limited to only a few the input variable choice can be done in conjunction with the modelling stage, simply based on the performance of different models with different sets of inputs. If the total number of available variables is large the usual approach for selecting inputs is based on correlation analysis. This method was used in the present study. The correlation analysis was performed only on fault free data in order to get as reliable information as possible. To avoid selecting several variables describing the same phenomena a simple test was made. In order for a candidate variable to be selected two conditions had to be fulfilled:

1) The correlation between the input variable in question and the analyzer had to be at least X%
2) The input variable in question was allowed to correlate at most Y% with any of the previously selected variables

In this study X was 50% and Y was 85%. After the selection of input variables was done, the best delays for these variables were sought by creating high order ARX models with varying time delay for each of the input-output combinations. The time-delay that gave the best fit of the data was chosen. Using this scheme 8 input variables were obtained. The variables along with the obtained correlations and delays are given in Table 1.

Table 1. Selected variables and their correlations with the dependent variable

Variable	delay	correlation
Temperature on tray 38 in distillation column 1	6	0.87
Temperature on level 5 in the side stripper	26	0.83
Reflux flow from column 2 to column 1	0	-0.76
Vapour distillate flow from column 1	3	-0.69
Liquid distillate flow from column 1	0	0.61
Temperature of feed to column 1	14	0.62
Controller output for column feed heating	0	-0.55
Controller output for stripper to column heating	39	-0.55

The third column of the table contains the correlation coefficient between the analyzer measurement and the variable in question.

4.2 Selecting training and testing data

It was crucial to select the training and test data in an efficient way in order for the models to perform well. The training data should be selected to cover as much as possible of the available data for the model to be able to generalize the behaviour of the process well. This is a general requirement for empirical models but it is accentuated for support vector machines using radial basis function kernels, owing to their local nature.

As the system in question is of high dimension, a large amount of data is needed. The data selection was performed by taking random sequences of numbers in the range of the total number of data points. According to these points the data is divided into alternating pieces of training and testing data sequences. Using this technique the data was divided into 50 shorter sequences as illustrated in Fig. 2.

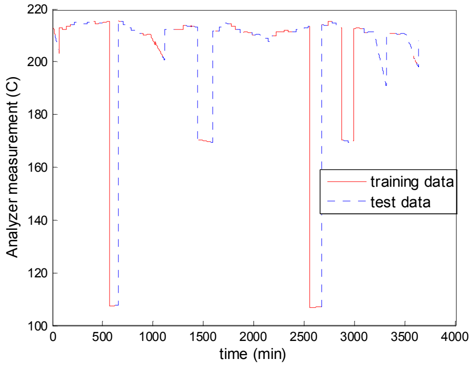

Fig. 2. The principle for dividing the data into training and testing sets

5. RESULTS

The two outlined fault detection methods were tested and evaluated on the data above.

With the regression approach it was possible to make a quite accurate model (95.9% correlation between model output and measurement on the training data) of the fault free behaviour of the analyzer. The results are also good on the fault-free test data (92.7%, correlation). A correlation figure is not given for test data containing also faulty sequences because the models are not trained to simulate the faults. As outlined in section 2.1 a residual analysis is needed for the detection of the analyzer faults. When the estimate and measurement differ by more than a preset limit the analyzer is assumed to be faulty. A limit of 2.0 °C turned out to be optimal, resulting in only 8 cases of misclassification responding to a classification accuracy of 97.5%. Each of the cases were missed alarms. The results with the regression approach are illustrated in Fig. 3 and 4.

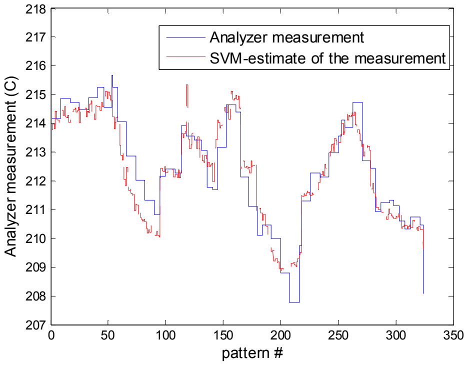

Fig. 3. Regression results on fault free test data

Table 2 illustrates the dependence of the model parameters on the performance of the fault detection. Inside the quite narrow region illustrated in the table the results are not heavily dependent on the parameters with the exception of the fourth line (γ increased to 0.1), for which the results on test data deteriorated substantially when deviating from the optimal value.

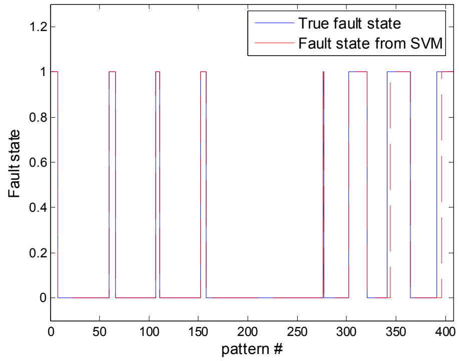

Fig. 4 Classification of fault state by means of model residual (test data)

Table 2. Optimization of SVMs open parameters

ε	γ	C	ρ_{train}	ρ_{test}
0.05	0.05	10	0.961	0.916
0.025	0.05	10	0.961	0.916
0.1	0.05	10	0.961	0.902
0.05	0.1	10	0.970	0.848
0.05	0.025	10	0.953	0.926
0.05	0.04	10	0.958	0.922
0.05	0.03	10	0.955	0.925
0.05	0.035	10	0.956	0.924
0.05	0.03	1	0.933	0.903
0.05	0.03	20	0.958	0.927
0.05	0.03	30	0.961	0.924
0.05	0.03	25	0.960	0.926
0.05	0.03	15	0.957	0.924
0.05	0.03	18	0.958	0.925
0.05	**0.03**	**21**	**0.959**	**0.927**
0.05	0.03	19	0.958	0.926

The classification approach gave an accuracy of 99.7% on the training data and 95.3% on the test data. In terms of alarms it meant that there were 19 missed alarms but no false alarms. Comparing with the results of the regression case, the classification approach has more than double the amount of missed alarms. The results are illustrated in Figure 5.

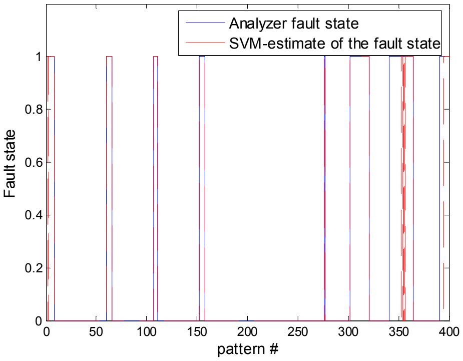

Fig. 5. Classifier results on test data

Evaluating the best model of (Vermasvuori et al., 2005) (a PLS model with 5 latent variables) on the data set in the present study gives 12% lower RMS on the test data. That model, however, uses 26 input variables compared to 8 for the best SVM model. Since a model using many input variables is more susceptible to errors in on-line use it might be better to use a model with fewer inputs at the cost of lower accuracy.

6. COMPARING THE PROPOSED METHOD TO NEURAL NETWORKS

It is interesting to compare the performance of the support vector machines with standard MLP neural networks. In order to perform this test, neural networks of different sizes were trained using the NNDT software (Saxén and Saxén, 1995), which uses the efficient Levenberg-Marquardt method. The results of these neural network models were compared to the best support vector machine. The comparison was done only for the regression case, even though a similar test could have been performed for the classification case. The following table summarizes the results from the comparison.

Table 3 Results with neural networks

N_{hid}	RMS_{tr}	RMS_{te}
0 (linear)	0.2035	0.2795
1	0.188	0.2569
2	0.1431	0.333
3	0.1109	0.2995
SVM_{opt}	0.1523	0.2336

From the study we can conclude the following: On training data the best SVM has a performance roughly on the level of a neural network with 2 hidden nodes, but the SVM clearly outperforms the same network on test data, indicating a better generalization ability of the SVM. Furthermore a model with only one hidden neuron gives the lowest error on the test data of all the NN models tested. The optimal SVM outperforms this network with an RMS that is approximately 10% lower. Another observation to be made is that the improvement achieved by using non-linear methods vs. linear methods is not so big (16% on the RMS).

7. CONCLUSIONS

In this paper the use of support vector machines for detecting analyzer faults is discussed. Two different approaches were introduced and evaluated. It was concluded that special care has to be taken when choosing the training and test data sets for the kernel methods to perform well, probably owing to the local nature of such models. For the studied case the regression approach was clearly better than the classification approach. A comparison to results using traditional MLP neural networks indicated a slight superiority in favour of the proposed method. The SVM models could, if identified on real industrial data, readily be implemented for on-line detection of analyzer faults, provided that the identification results on real data are adequate. The PLS model structure in (Vermasvuori et al., 2005) gave a slightly lower RMS than the best support vector machine, but with the aid of more input variables. A deeper quantitative analysis of differences between the methods evaluated in (Vermasvuori et al., 2005) and SVMs is a topic for future research.

ACKNOWLEDGEMENTS

Engineering Fellow Mauri Sourander at Neste Jacobs is gratefully acknowledged for providing useful comments to this work. Neste Jacobs is also gratefully acknowledged for making available the process simulator used.

REFERENCES

Chang C.-C. and C.-J. Lin (2001), Libsvm: a library for support vector machines, SVM software on the internet, available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Jemwa G.T. and C. Aldrich (2005), Monitoring of an industrial liquid−liquid extraction system with kernel-based methods, *Hydrometallurgy*, **78**, 41−51

Hinkley, D. V. (1971), Inference about the change-point from cumulative sum tests, Biometrika, **58**, 509−523.

Kulkarni A., Jayaraman V.K. and B.D. Kulkarni (2005), Kowledge incorporated support vector machines to detect faults in the Tennessee Eastman process, *Computers & Chemical Engineering*, **29**, 2128−2133.

Platt J.C. (1998), Fast training of support vector machines using sequential minimal optimization, in: *Advances in Kernel Methods − Support Vector Learning*, (B. Schölkopf, C. J. C. Burges, and A. J. Smola (Ed)) 1st Edition, MIT Press, Cambridge, Massachusetts, 185−208.

Pöyhönen, S., A Arkkio, P Jover and H. Hyötyniemi (2005), Coupling pairwise support vector machines for fault classification, *Control Engineering Practice*, **13**, 759−769.

Ribeiro B. (2005), Support vector machines for quality monitoring in a plastic injection moulding process, *IEEE Transactions on systems, man, and cybernetics-Part C: Applications and reviews*, **35**, 401−410

Saxén B. and H. Saxén, NNDT - A neural network development tool - User's guide, Technical Report 94-8, Heat Eng. Lab, Åbo Akademi University, Åbo, Finland, 1994.

Vapnik V. N. (1998), *Statistical learning theory*, John Wiley & Sons, New York, NY

Venkatasubramanian V., R. Rengaswamy, K. Yin, S.N. Kavuri (2003), A review of process fault detection and diagnosis Part I: Quantitative model-based methods, *Computers & Chemical Engineering*, **27**, 293-311

Vermasvuori M.T., N. Vatanski and S-L Jämsä-Jounela (2005), Data-based fault detection of the online analysers in a dearomatisation process, 1st Workshop on Networked Control System and Fault Tolerant Control, Ajaccio, France, October 6−7th, 2005 (www.strep-necst.org/1st workshop/Vermasvuori.pdf)

Wang H.Q., Zhang P and S.X. Ding (2006), Kernel and SVM based theory for industrial process modelling and fault diagnosis. EU project NeCST internal presentation, Paris, March 2006.