DATA-DRIVEN PROCESS MONITORING AND DIAGNOSIS WITH SUPPORT VECTOR DATA DESCRIPTION

by

Esmaeil Tafazzoli Moghaddam

B.Sc. Khaje Nasir Toosi University of Technology, 1999

M.A.Sc. of Mechanical Engineering, Ferdowsi University, 2002

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Applied Science
in the
School of Engineering Science,
Faculty of Applied Science

SIMON FRASER UNIVERSITY
Fall 2011

APPROVAL

Name: Esmaeil Tafazzoli Moghaddam

Degree: Master of Applied Science

Title of Thesis: Data-Driven Process Monitoring and Diagnosis with Support Vector Data Description

Examining Committee: Dr. Rodney Vaughan
Chair

Dr. Mehrdad Saif, P.Eng
Professor
Senior Supervisor

Dr. John Jones, P.Eng
Associate Professor
Supervisor

Dr. Mehrdad Moallem, P.Eng
Associate Professor
Supervisor

Dr. Carlo Menon
Assistant Professor
Internal Examiner

Date Approved: December 19, 2011

Abstract

This thesis targets the problem of fault diagnosis of industrial processes with data-driven approaches. In this context, a class of problems are considered in which the only information about the process is in the form of data and no model is available due to complexity of the process.

Support vector data description is a kernel based method recently proposed in the field of pattern recognition and it is known for its powerful capabilities in nonlinear data classification which can be exploited in fault diagnosis systems.

The purpose of this study is to investigate SVDD applicability as a data-driven method in industrial process fault diagnosis. In this respect, a complete framework for fault diagnosis structure is proposed and studied. The results demonstrate that SVDD is a powerful method in process fault diagnosis.

To my first teachers: My father and my mother

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Mehrdad Saif. I would like to thank him for giving me the opportunity to study at Simon Fraser University under his supervision. I also want to thank him for all the support and help and supervision. I would like to thank the examining committee, Dr. John Jones, Dr. Mehrdad Moallem, and Dr. Carlo Menon, and Dr. Rodney Vaughan as the chair of the committee, for taking their precious time and reviewing my thesis. I truly appreciate their valuable comments on my thesis. I wish to express my gratitude to all my freinds and my colleagues who supported me during my studies, especially, the members of the control and diagnosis lab, Miss. Maryam Soleimani, Miss. Arina Aboonabi, Mr. Kaveh Kianfar, Dr. Amir Niroumand, Dr. Mahdi Alavi, Mr. Foad Samadi, Mr. Jimmy Tsai, Dr. Wu Qing, and Dr. Weitian Chen. Many thanks are given to all my friends at SFU, particularly, Mr. Kourosh Khosraviani, Mrs. Lila Torabi, Mr. Reza Babajani. I am particularly indebted to my mother for her patience, support, and unconditional love.

Table of Acronyms

ANN	Artificial Neural Network
FDA	Fisher's Discriminant Analysis
FDD	Fault Detection and Diagnosis
FDI	Fault Detection and Isolation
IC	Independent Component
ICA	Independent Component Analysis
ISVM	Invariant Support Vector Machine
KDE	kernel Density Estimation
KNN	K-Nearest Neighbor
LDA	Linear discriminant Analysis
MV	Manipulated Variable
NBN	Naive Bayesian Network
ND	Normalized Distance
NOC	Normal operation Data
PC	Principal component
PCA	Principal Component Analysis
PSVM	Proximal Support Vector Machine
QDA	Quadratic discriminant Analysis
SPE	Squared Prediction Error
SV	support Vector
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TAN	Tree Augmented Network
TEP	Tennessee Eastman Process

Nomenclature

A	mixing matrix
A_{cross}	tank cross section area
B	whitened mixing matrix
C	controlling parameter of SVDD
C_i	i^{th} data class
C_{ij}	confusion matrix element
D	distance
E	residual matrix or error matrix
F_α	Fisher's F-distribution value
G	contrast function
H	entropy
H_{max}	highest water level in three tank system
I^2	ICA statistic
J	negentropy
K	kernel function
L	fast SVDD subset size
ND	normalized distance
P	partial eigenvector matrix in PCA
Q	squared prediction error
Q_α	squared prediction error threshold
Q_{max}	highest pump flow rate
R	radius
S	matrix of independent components
SPE	ICA squared prediction error statistics
T	PCA scores matrix
T^2	Hotelling's measure
T^α	Hotelling's measure control limit
V	matrix of eigenvectors
W	de-mixing matrix
X	training data matrix
$\hat{X}$	reconstructed data matrix
Z	whitened data matrix

a_1	constant
az	three tank pipe cross section area
$\mathbf{a}$	Hyper-sphere center
c_α	normal deviate value
d	number of variables
e	ICA residual
f	fault parameter
h	smoothing parameter
h_i	threshold parameter
k	number of nearest neighbors of KNN
n	number of observations
m	number of variables
r_i	PCA residuals
s_i	independent components variables
$\hat{s}_i$	estimated independent component variables
u	ICA contrast function variable
$\mathbf{v}$	random noise vector
$\mathbf{w}$	uniformly distributed random vector
$\mathbf{x}$	observed data point vector
$\mathbf{x}_{new}$	new observation point
$\mathbf{x}_s$	support vector training data point
y	random variable
$y_{Gaussian}$	gaussian random variable

α	confidence value
α_i	Lagrange multipliers
γ_i	Lagrange multipliers
ζ	slack variable
θ_i	threshold parameter
Λ	eigenvalue matrix
λ_i	eigenvalues
ϕ	mapping
σ	kernel parameter
Σ	data covariance matrix

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the ever increasing demand for higher product quality, safety, and efficiency in industrial processes, the need for better process control and monitoring systems has been essential to meet production requirements and standards. In the past three decades, many different approaches have been developed and implemented to enhance process productivity. Various control system configurations have been designed to improve process operations with increasing number of variables and measurements in modern industrial processes. In this area, modern control systems such as model predictive control and supervisory control system have had great influence in industrial process control.

The goal of the process control system is to maintain the process in the desired operating condition by compensating disturbances and process changes with the designed controllers. However, there are changes in the process that are caused by faults and can not be compensated by the control system. These changes are considered as abnormalities or deviation of the process from normal conditions and must be detected and identified to prevent unwanted results in the form of low quality products, components failure, operation shutdown, and extra costs. Detecting abnormal conditions and diagnosing the process is the main purpose of the process monitoring systems that aim at identifying faults and the cause of abnormalities [1]-[3].

Fault detection and diagnosis, FDD, is the heart of any process monitoring system. As a wide area of research for more than three decades, FDD has been attractive to engineers in different disciplines and fields of engineering such as process monitoring, control, manufacturing, automotive industry, chemical process industry, etc. Several approaches have been proposed for fault detection and diagnosis depending on system properties and information availability. Generally, these methods are categorized into three major groups. The first category is known as model-based approaches that highly depend on mathematical model of the system for fault detection and diagnosis. The more precise a model is constructed, the more reliable results is achieved. However, in many systems such as industrial processes, it is not possible to obtain a precise model due to process high dimensionality and complexity of the system which results in large modeling costs and time required to obtain an accurate model. Therefore, as an alternative, data-driven approaches are utilized which heavily depend on data captured from system measurements to detect faults. Data-driven methods find structures or patterns in data and monitor the system behavior by processing available data. Developments in data acquisition equipment allows for more data collection and storage, which leads to availability of large amount of process historical data, making data-driven approaches a suitable choice for monitoring. However, the performance depends on the quality of data as well as data quantity. The third category is known as knowledge-based methods in which a priori knowledge of the process is used for extracting rules for monitoring. Causal analysis and expert system are example methods in this category. Knowledge based expert systems are flexible and can be implemented very fast and the results are easy to interpret. However, it is difficult to apply knowledge based methods to large scale processes and requires considerable amount of process experience[1].

In this work we focus on data-driven methods for process fault detection and diagnosis. Data-driven methods include multivariate statistical analysis such as principal component analysis (PCA), and independent component analysis (ICA), Neural Networks, classification approaches such as Support vector machines(SVM) and Bayesian networks , and density estimation methods such as kernel density estimate (KDE) [2], [4]-[8].

Process monitoring can be divided into two major steps. The first step is fault detection in which the presence of fault in the system is examined. In other word, the result of this step would be a positive or negative answer to the question of occurrence of any fault in the system. When a fault detected, the second step is fault diagnosis.

Fault diagnosis is a complementary procedure in process monitoring which is accomplished by employing classification methods. In this context, fault information is analyzed by classifiers in order to determine the class the fault belongs to, so that further decisions can be made to cure the faulty situation. Many different classification methods have been proposed in the literature. They can be categorized as probabilistic methods such as Bayesian networks, and kernel based approaches such as SVM. In recent years, kernel based methods such as support vector machines have gained attention for their capability in classifying data[7], [9], [10], [11].

Classifiers can be separated based on their structure and functionality as one-class classifier, binary classifier, and multi-class classifiers. Multi-class classification refers to the case when there are more than two classes and the goal is to assign each data or object to one of the classes. One-class classification, also known as anomaly detection, novelty detection, or outlier detection, is a process in which one class of objects (target data) are separated from other objects (outliers for example) by learning a classifier with training data of the target class. Two-class or binary classification is almost the same as one-class type except that data for the second class is also available.

One-class classification methods are divided into density estimation, reconstruction, and boundary methods. In density estimation methods, the distribution of the target class is estimated and a threshold is defined as a probability. Reconstruction methods construct a model to fit training data and use reconstruction error as a separating criteria for in-class data. The main goal in boundary methods is to obtain a descriptive boundary around target class that contains most of the data[17].

As a one-class classifier, support vector data description (SVDD) has found application in a wide range of applications. Recently, SVDD was used in a computer-vision-based automated fabric defect detection system[20]. It was also applied for background modeling in video processing applications[21]. Other applications include target detection in hyperspectral imagery [22], classification for analog circuit fault diagnosis[25], gearbox fault diagnosis [23], pump failure detection [18], face recognition , speaker recognition, image retrieval and medical imaging[19],[24]. However, in the area of process monitoring and fault detection and diagnosis, SVDD is new and there is potential for more comprehensive research. In most of the few cases reported, SVDD has been used as a complementary element for monitoring and not as a basic component in the FDD structure. For example, in [12], ICA is combined with SVDD for fault detection in rotating machinery. Also, in [15],[13], [14], and [15] ICA and PCA were employed as the main part of the fault detection and identification system and SVDD was used for calculating threshold limit. For fault diagnosis, a combination of Linear discriminant analysis, Nearest neighbor rules and SVDD were used for classifying roller bearing faults in [16]. As the main contribution of this work, a complete fault detection and diagnosis scheme based on SVDD is provided and tested on simulated processes, as well as on a real system to investigate SVDD's capabilities for fault detection and diagnosis.

In this respect, SVDD is considered as the core part of the detection and diagnosis system. The proposed approach is implemented and compared with standard multivariate process monitoring methods for fault detection. In addition, for fault diagnosis (fault classification), a multi-classification structure based on SVDD is designed and applied to two industrial processes and compared with standard classifiers. Specifically, KNN, K-nearest neighbor is chosen for comparison which is one of the most well known classification methods.

1.1 Objectives and contributions

The overall objective of this study is to investigate the applicability of support vector data description as a data driven method in fault diagnosis applications, specially in engineering process monitoring, and comparing SVDD with other standard methods, namely, PCA and ICA for detection, and KNN, SVM, and other classification methods for fault diagnosis. The contribution of this work is in providing a complete structure based on SVDD for data driven fault detection and isolation (FDI), and demonstrating the capability of SVDD in detecting and isolating faults in processes as an alternative method for FDI. A complete package is provided which includes fault detection and fault classification as the main parts of an FDI system. The computational complexity and processing time problem has also been addressed and enhanced by embedding a fast SVDD algorithm in the FDI structure.

The following tasks pursue the aforementioned objectives as the contribution of this work:

- Proposing a new framework for process fault detection and developing the proposed method in terms of parameter selection and computational enhancement.
- Implementing support vector data description fault detection scheme on a simple multivariable system as well as a benchmark simulated process, Tennessee Eastman Process (TEP), and a real experimental system (Three Tank System) to demonstrate SVDD's potential strength in process fault detection.
- Developing the proposed fault classification approach based on SVDD as a fault diagnosis method for multi-classification of faults in processes.
- Implementing SVDD fault classification on TEP and Three Tank System.

- Analyzing and discussing detection and diagnosis proficiency of SVDD in comparison to benchmark methods

1.2 Thesis outline

In chapter 2, the theory behind each monitoring method is presented. Specifically, detail information on SVDD, ICA, and PCA and mathematical formulation for each method is provided. Chapter 3 introduces fault detection definitions and pictures the fault detection scheme based on the proposed method and discusses issues in the design procedure. Chapter 4 includes implementation details and the results of SVDD fault detection on three different systems. In chapter 5, fault classification is introduced as a fault diagnosis method and the diagnosis scheme is described. Other classification methods are also introduced for comparison. Chapter 6 details implementation of SVDD for fault classification and presents the results for SVDD and compares with KNN and other classification methods. Chapter 7 summarizes this work and provides some conclusion and recommendations.

Chapter 2

Process monitoring methods

Several process monitoring methods have been developed and widely used in different disciplines, ranging from chemical processes to biological applications. Among linear methods, principal component analysis (PCA) and independent component analysis (ICA) are the two most well known methods which have been recognized as standard process monitoring approaches. Support vector data description (SVDD) is a new one-class classification method which has been developed in pattern recognition society and has received attention in engineering application in recent years.

In this work, we introduce SVDD as the core element of the fault detection and diagnosis (FDD) system and propose an FDD structure which is based on SVDD. In order to assess the applicability and performance of SVDD in process monitoring application, ICA and PCA are selected as standard method to compare with SVDD. In this chapter, theoretical details of each method are presented as a foundation for the following chapters.

2.1 Support Vector Data Description, SVDD

Support vector data description is a newly emerged method that was introduced by Tax and Duin [46] as a one-class classifier in the field of pattern recognition. SVDD can be considered as an extension or modification of SVM since both have similar optimization formulation. Contrary to SVM which searches for the optimum separating hyper-plane in feature space, the main idea in SVDD is to find a spherically shaped boundary around a data set to describe the data in the feature space; Therefore, a tighter boundary can be found for the data and also provides a boundary which can be used as the limiting bound for a one class data. In this method, the target data set is transformed into a feature space in which a hyper-sphere circumscribes the data while its radius is minimized. The nature of the SVDD method makes it suitable to be used in outlier detection problems to detect objects that are different from the data or it can be used as one-class classifiers when the training data for one class is well sampled while other classes are not [46]. As a nonlinear kernel based method, SVDD solves a convex optimization problem and gives global solution which is one of its significant properties. In this work, SVDD is used as fault detector which provides a boundary around normal operating condition (NOC) data.

2.1.1 SVDD Theory

For a set of data points, SVDD defines a nonlinear mapping, $\phi : X \rightarrow F$, to a high dimensional feature space and finds the smallest sphere that contains most of the mapped data points in the feature space[46]. This sphere, when mapped back to the data space, can separate into several components, each enclosing a separate cluster of points [47]. More specifically, suppose we have a set of training data, $\mathbf{x}_i \in X$, $i = 1, \ldots, N$, where $X \subset R^n$ and let ϕ be a mapping from X to a higher dimensional feature space. Then, the problem of finding the hyper-sphere with minimum radius that encloses training data images in the feature space is formulated as follows:

$$min \quad R^2 + C \sum_i \xi_i, \tag{2.1}$$

$$s.t. \qquad \|\phi(\mathbf{x}_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i \quad , \quad \xi_i \geq 0 \quad for \quad i = 1, \ldots, N,$$

where $\mathbf{a}$ is the hyper-sphere center, R is the radius, and ξ_i are slack variables. Including slack variables as shown in equation 2.1 allows some points to be outside the sphere, which results in a softer boundary. Parameter C is a controlling parameter that controls the trade-off between error and volume. The minimization problem can be solved by introducing Lagrangian in 2.2 with α and γ as Lagrange multipliers:

$$L = R^2 + C \sum_i \xi_i - \sum_i (R^2 + \xi_i - \|\phi(\mathbf{x}_i) - \mathbf{a}\|^2)\alpha_i - \sum_i \xi_i \gamma_i. \tag{2.2}$$

L should be minimized with respect to R, $\mathbf{a}$, and ζ_i and maximized with respect to α and γ[46]. Setting partial derivatives equal to zero results in the following constraints:

$$\partial L / \partial R = 0 \rightarrow \quad \sum_i \alpha_i = 1, \tag{2.3}$$

$$\partial L/\partial \mathbf{a} = 0 \rightarrow \quad \mathbf{a} = \sum_i \phi(\mathbf{x}_i)\alpha_i, \tag{2.4}$$

and

$$\partial L/\partial \xi_i = 0 \rightarrow \quad C - \alpha_i - \gamma_i = 0. \tag{2.5}$$

Since $\alpha_i \geq 0$, $\gamma_i \geq 0$, and $C - \alpha_i - \gamma_i = 0$ we can remove γ_i Lagrange multiplier by enforcing $0 \leq \alpha_i \leq C$. Substituting 2.3-2.5 into 2.2 the new objective function becomes:

$$L = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2.6}$$

subject to

$$0 \leq \alpha_i \leq C \quad , \quad \sum_i \alpha_i = 1, \quad i = 1, \ldots, N,$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i).\phi(\mathbf{x}_j)$ is a kernel function. Lagrange multipliers, α_i, are found by maximizing 2.6. As defined in [46], all points located on the boundary satisfy $0 < \alpha_i < C$ and are called Support Vectors; while the points inside boundary have $\alpha_i = 0$. From 2.4, the squared distance of any new data point from the center of the sphere can be calculated as:

$$D^2 = \|\phi(\mathbf{x}_{new}) - \mathbf{a}\|^2 = K(\mathbf{x}_{new}, \mathbf{x}_{new}) - 2\sum_i \alpha_i K(\mathbf{x}_{new}, \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j),$$

where in case of Gaussian kernel becomes:

$$D^2 = \|\phi(\mathbf{x}_{new}) - \mathbf{a}\|^2 = 1 - 2\sum_i \alpha_i K(\mathbf{x}_{new}, \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j),$$

Also, the radius of the sphere is calculated as the distance between the center and the boundary (any of the SVs). The squared radius is:

$$R^2 = K(\mathbf{x}_s, \mathbf{x}_s) - 2\sum_i \alpha_i K(\mathbf{x}_s, \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j),$$

where $\mathbf{x}_s \in \{SVs\}$. The test point is outside the sphere if $D^2 > R^2$ and it is inside or on the boundary if $D^2 \leq R^2$. In the above mentioned formula, a kernel function is embedded in the the equations which simplifies calculations in feature space by replacing the inner product of the vectors. The most used kernel function proposed in the literature is the Gaussian kernel because of its properties which makes it a good choice. The Gaussian kernel is defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)},$$

with σ as kernel width parameter.

2.2 Principal Component Analysis, PCA

Principal component analysis is a standard method that has been well studied for multivariate process monitoring in fault diagnosis systems applied to industrial processes [26],[27]. PCA was first proposed by Pearson in 1901 and later developed by Hotelling in 1947 [4]. The main function of multivariate statistical techniques is to transform a set of process variables into a smaller uncorrelated set. PCA is based on orthogonal decomposition of the covariance matrix of the process variables and finding directions that explain the maximum variation of the data. The main purpose of using PCA is to find factors that have a much lower dimension than the original data set which can properly describe the major trends in the original data set [28]. Analysis of industrial data using principal component analysis, especially for detecting faults in chemical processes, has been intensively studied in [2],[29][30],[31].

PCA formulation

In the following, the basic PCA formulation is presented. For more detail, the reader is referred to [2],[33].

For a set of data X with n observation and m variables, the sample covariance matrix is defined as:

$$\Sigma = \frac{1}{n-1} X^T X.$$

Eigenvalue decomposition of Σ gives:

$$\Sigma = V \Lambda V^T,$$

where V contains principal components as columns and Λ is the eigenvalue matrix whose i^{th} elements represent the variance of data projected along the i^{th} PC. The dimensionality can be reduced by taking only the first a PCs corresponding to the first a largest eigenvalues that capture a specific amount of variance (80% for example). The transformed data points in the new dimensionally reduced space are called PCA scores and are calculated as:

$$T = XP,$$

where P is the matrix of the first a columns of V. In fact, original data points are represented as projected points in a lower dimensional space spanned by PCs also called feature space.

Reconstruction of the data is achieved by:

$$\hat{X} = TP^T,$$

and the residual matrix is found as:

$$E = X - \hat{X}.$$

The separation of the process data space into feature space and residual space results in two criteria or measure for monitoring the process behavior, namely, Hotelling's T^2 distance and squared prediction error (Q). When process is out of control while the process structure is preserved, Hotelling's T^2 exceeds its control limit, showing process model variation outside its Normal Operating Condition (NOC) control limits. On the other hand, when the correlation of the NOC process data is broken then the Q statistic limit is exceeded [31]. Figure 2.1 explains Hotelling's T^2 and Q measures and their interpretation with regard to the process data in the operating space. In this figure, the points representing fault A appear in the Q statistic plot and the points representing fault B will show up in the T^2 statistic plot. The envelope shows the normal operation region [32]. The Hotelling's T^2 statistic can be

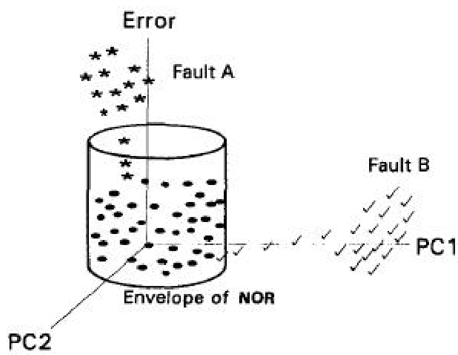

Figure 2.1: Interpretation of T^2 and Q statistics in the space of process data[32].

calculated for any data point by :

$$T^2 = \mathbf{x}_i^T V \Lambda^{-1} V^T \mathbf{x}_i, \tag{2.7}$$

where $\mathbf{x}_i$, a sample data point, which is an $m \times 1$ vector. For PCA scores the T^2 statistic is calculated the same way while retaining a eigenvectors and eigenvalues. The control limit is defined by equation 2.8, assuming that the NOC data conform to a multivariate normal distribution and NOC parameter estimates are sufficiently accurate,

$$T_{\alpha}^{2} = \frac{a(n-1)(n+1)}{n(n-a)} F_{\alpha}(a, n-a), \tag{2.8}$$

where $F_{\alpha}(a, n-a)$ is Fisher F-distribution value with a and $n-a$ degrees of freedom and $1-\alpha$ confidence. Q-statistic (or SPE) is defined as a squared 2-norm of the residuals as described in equation 2.9:

$$Q_i = r_i^T r_i, \quad r_i = (I - PP^T)X_i. \tag{2.9}$$

The threshold for Q-statistic is defined by Q_{α} in equation 2.10 as

$$Q_{\alpha} = \theta_1 [\frac{h_0 c_{\alpha} \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}]^{1/h_0}, \tag{2.10}$$

where

$$\theta_i = \sum_{j=a+1}^{n} \lambda_j^i, \quad i = 1, 2, 3$$

and

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}.$$

In this equation, c_{α} is the normal deviate corresponding to $(1-\alpha)$ percentile [2]. Considering the mentioned statistics, the following conditions are the indicators of occurrence of abnormality or fault in the process:

$$T^2 > T_{\alpha}^2 \quad or \quad Q > Q_{\alpha}.$$

2.3 Independent Component Analysis, ICA

It is known that many process variables that are measured for monitoring are not independent and can be a combination of independent observations. Separating these independent latent variables can reveal valuable information about the structure of the process data that is monitored. ICA is a multivariate data processing method which separates mixed data into its independent source signals [34]. A linear transformation transforms multivariate data by maximizing statistical independence of variables and finds independent latent variables using a function of independence. The latent variables are assumed to be mutually independent and non-Gaussian. This technique was first proposed to solve the blind source separation problem. A classical example of blind source separation is the cocktail party problem [35]. Assume several people are speaking simultaneously in a room, as in a cocktail party. Placing several recording microphones in different locations in the room, the problem is to separate the voices of the different speakers. Applying ICA on recording data, and separating them results in obtaining the latent variables which are the waveforms of the voices.

2.3.1 ICA algorithm

Assume $X_{d\times n}$ as the observation data matrix, with d variables $x_1, x_2, ..., x_d$ having n observations each. These variables can be assumed as a linear combination of k unknown independent variables, $s_1, s_2, ..., s_k$, such that

$$X = AS, \tag{2.11}$$

where $S_{k\times n}$ is the matrix of independent components, and $A_{d\times k}$ is called mixing matrix which is unknown and has to be found [36]. In ICA, the algorithm searches for a de-mixing matrix, W, which transforms X to $\hat{S}$ as $\hat{S} = WX$ such that for all $\hat{s}_i$, latent variables, statistical independence is maximized as much as possible. It is common to perform some preprocessing on data before applying ICA algorithm. Centering and whitening are the most known preprocessing strategies that are useful to do on data. Centering is simply done by subtracting data from its mean. Whitening linearly transforms the observed variables so that the obtained variables are uncorrelated and their variances equal unity, i.e., $E\{\hat{X}\hat{X}^T\} = I$. Whitening can be done by eigenvalue decomposition of the covariance matrix, $E\{XX^T\} = QPQ^T$ as follows:

$$Z = P^{-1/2}Q^T X = P^{-1/2}Q^T AS = BS, \tag{2.12}$$

where Z is the whitened data matrix, P is a diagonal matrix containing eigenvalues of the covariance matrix, Q is a matrix containing the eigenvectors corresponding to eigenvalues in P as its columns, and B is the whitened mixing matrix. Data dimension reduction can be done in whitening step by only including large eigenvalues in P and discarding small eigenvalues.

It can be shown that B is an orthonormal matrix, i.e., $BB^T = I$; Therefore, independent components, s_i, can be obtained using (2.12) as:

$$\hat{S} = B^T Z.$$

The de-mixing matrix B is computed by giving an initial guess for each column of B and iteratively updating it to maximize non-Gaussianity of the corresponding independent component. It is shown that independence is achieved by maximizing non-Gaussianity. The two measures for Non-Gaussianity are kurtosis and negentropy[35].

Since kurtosis is highly affected by outliers it is not recommended to be used. Negentropy is related to entropy (an important concept in information theory) of a random variable. It is defined as

$$J(y) = H(y_{Gaussian}) - H(y). \tag{2.13}$$

In equation (2.13), y is a random variable, $y_{Gaussian}$ is a Gaussian random variable with the same covariance as y, J is negentropy of y, and

$$H(y) = -\int f(y)log(f(y))dy$$

is the differential entropy of a variable with density of $f(y)$ [36]. To find B by maximizing non-Gaussianity using negentropy, J, Hyvarinen [35] proposed an efficient algorithm called FastICA which is based on approximating negentropy with the following procedure[37]:

1- Choose an initial weight vector $\mathbf{b_i}$ with unit norm randomly.

2- Let $\mathbf{b_i} = E\{\mathbf{z}g(\mathbf{b_i}^T\mathbf{z})\} - E\{g'(\mathbf{b_i}^T\mathbf{z})\}\mathbf{b_i}$

3- Normalize $\mathbf{b_i} = \mathbf{b_i}/\|\mathbf{b_i}\|$.

4- If $\mathbf{b_i}$ has not converged, then go to 2.

In the above algorithm, g and g' are the first and second derivatives of G, the contrast function, which is suggested to take one of the following forms:

$$G(u) = -\frac{1}{a_1}logcosh(a_1 u),$$

or

$$G(u) = -exp(-\frac{u^2}{2}),$$

with $1 \leq a_1 \leq 2$ being a constant. When matrix B is obtained, the data can be transformed to get independent components as

$$\hat{S} = B^T Z = B^T P^{-1/2} Q^T X = WX.$$

In process monitoring, ICA algorithm has been implemented to obtain statistic measures for monitoring systematic and nonsystematic variation of the process. The two most known statistics are I^2 and squared prediction error (SPE), that can be computed for the k^{th} observation as:

$$I^2(k) = \hat{\mathbf{s}}(k)^T \hat{\mathbf{s}}(k) = (W\mathbf{x}(k))^T W\mathbf{x}(k),$$

$$SPE = (\mathbf{e}(k)^T \mathbf{e}(k)) = (\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T (\mathbf{x}(k) - \hat{\mathbf{x}}(k)),$$

where $\hat{\mathbf{x}} = A\hat{\mathbf{s}}(k) = WA\mathbf{x}(k)$.

To find the confidence limits for these statistics, kernel density estimation method (KDE) can be used. KDE is a powerful data-driven technique for nonparametric estimation of density functions. Contrary to PCA, in ICA, the Hotelling's T^2 and SPE statistic can not be defined since they are determined based on the assumption that process data distribution is normal, which is not the case in ICA. Here, the kernel estimator is defined as:

$$f(x) = \frac{1}{nh} \sum_{1}^{n} K(\frac{x - x_i}{h}),$$

with x as the point of concern, x_i as observations, h as smoothing parameter, n, as the number of observations, and K as kernel function which is usually Gaussian kernel. More detail information regarding kernel density estimation and confidence limit computation can be found in [39] and [40].

2.4 Chapter Summary

In this chapter, Support Vector Data Description (SVDD) process monitoring method was introduced with detail formulation. In addition, two standard methods that later will be used for comparison, were also presented to form the basic theoretical foundation for the rest of the thesis. Principal Component analysis (PCA) and Independent Component Analysis (ICA) were presented as the two methods which are widely used in data driven process monitoring. The next chapter discusses general fault detection scheme and presents the problem and definitions followed by the proposed framework for fault detection based on SVDD. In this respect, details of the proposed fault detection system are presented and the issues invovled in the design of the system are discussed.

Chapter 3

Fault detection scheme

3.1 Fault detection and diagnosis: Problem and definitions

In almost every FDI systems, the following steps are taken into account. The procedure starts with preprocessing the data, which means removing unnecessary variables (if possible), removing outliers and scaling the data to eliminate the dominance of some variables' values over other ones. Depending on the method used, the data is transformed and/or the dimensionality of the data set is reduced for further steps. Transforming data is performed so that a specific property of the data would appear or can be captured easily, e.g., projecting data points into the direction where they show the most variation. The dimensionality reduction decreases the computational complexity and removes unnecessary variables. However, reduction order is always an important issue and there is a trade off between the number of reduced dimensions and the loss of information. The next step is to define a measure to quantify the information embedded in the data so that it could be compared to a threshold which results in change detection. The analysis of the obtained change is known as fault detection which determines whether a fault exists or not. By locating the fault and determining the size and type of the fault, identification and diagnosis step is accomplished. Many different schemes and techniques have been proposed for each step from data acquisition to fault evaluation in designing FDI systems. The most important issues in designing FDI system are rapidity, robustness and accuracy. A good FDI system must detect process faults as early as possible while being robust, meaning that it is insensitive to noise or other process disturbances. The challenges that engineers may face in designing FDI systems include nonlinearity in process variables, multi co-linearity due to correlation among variables that causes redundancy problems, dimensionality problems with large number of variables, and process dynamics.

3.1.1 Definitions

The terminology in the field of FDD is not consistent. Therefore, in order to avoid further misunderstandings, some definitions frequently used in the literature are given below:

- Condition monitoring: According to [48], condition monitoring is defined as the continuous or periodic measurement and interpretation of data to indicate the condition of an item to determine the need for maintenance. Condition monitoring is needed so that faults can be detected and diagnosed as early as possible.
- Fault: An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable, usual, or standard condition.
- Disturbance: An unknown and uncontrolled input, acting on the system.
- Residual: A fault indicator, based on the deviation between measurements and computed values.
- Fault detection: Determining the presence of faults in a system and the time of detection.
- Fault diagnosis: determining type, size, and location of faults in the system[49].

3.2 Fault detection framework

The proposed fault detection framework is presented in the following paragraphs. As one of the objectives of this work, we implement support vector data description, ICA, and PCA for fault detection in three different systems to investigate the performance of SVDD in industrial process monitoring and fault detection, and study its advantages and disadvantages compared to ICA and PCA. In this work, PCA is considered as the benchmark method which is used for comparison. PCA has been studied very well and has shown its power in process monitoring. More recently developed, compared to PCA, ICA has been the topic of much research work and has shown significant performance in engineering systems monitoring. Originally used for blind source separation in sound signal processing, ICA has found promising applications in monitoring different engineering systems such as chemical process monitoring [50],[51], [34] [37]. Although SVDD has been suggested for fault detection in a few applications, the need for more comprehensive study is prominent. In most of the previous work, SVDD has been used as a complementary component of the fault detection method. In [14] and [15], SVDD is combined with ICA to determine a threshold or boundary for the ICs. Another way to find a threshold is Kernel density estimate method, KDE, which is based on density estimation of the normal data with multiple kernel functions.

Fault detection task can be divided into two stages: Off-line training and Online detection. In off-line training, normal operating condition data are used to train the detection system and find the required limits and parameters for the monitoring system. Online detection involves testing new process data and assessing process health condition by comparing new data features with the results of the off-line stage and declaring faulty condition if any fault occurred. Off-line training can be summarized as:

- Obtaining normal operating condition data, NOC, and preprocessing if required, e.g., scaling, removing outliers, removing noise, etc.
- Transforming NOC data to feature space, i.e., finding scores
- Defining a distance measure in feature space and finding a threshold for monitoring out of limit data

Online detection stage is summarized as:

- Transforming new data point to feature space and computing its score
- Calculating the distance measure of the new point score
- Comparing the distance to the threshold and declaring faulty condition if the distance exceeds the limit

3.2.1 Design issues

Constructing any fault detection system requires considering design issues that highly affect the performance of the system. Some of the most common issues can be categorized as follows:

- Size of the data: The number of variables or measurements and the size of the training data set has a key role in the design process. High dimensional data sets leave the designer with the choice of using methods that can handle high dimensional data or applying dimension reduction techniques.
- If decided to reduce dimensionality, the most important issue becomes the number of dimensions to be reduced and how to determine this number.
- Selecting the most informative variables to be used and discarding those with less information is another important issue.
- Tuning parameters is also an influential issue. If parameters are not adjusted properly, then a powerful method appears ineffective which results in misleading the designer in selecting a suitable method
- Threshold calculation: Depending on process data properties, the right method must be selected to calculate a threshold which is optimal in describing process normal condition region. Failing to find an appropriate threshold in some cases highly affects monitoring and fault detection results
- Finally, the most common issue in all design problems is the trade off between complexity and error: As the error decreases, complexity increases, and the designer job is to find a balancing point between the two.

3.3 SVDD fault detection structure

Given a set of training data, detecting process faults based on SVDD is described in this section. In this framework, SVDD is trained with a set of normal operating condition (NOC) data and a boundary is computed. As mentioned earlier, SVDD constructs a spherical boundary around the data in the feature space with a minimized radius. Therefore, the boundary of the NOC region is characterized by the radius of the trained hyper -sphere and its center. When process is operating in its normal condition, data points remain inside the sphere and when any fault occurs, data points leave the sphere, otherwise, the fault is not detectable. Figures 3.1 and 3.2 show examples of SVDD boundary area for a two dimensional feature space of three tank system data which illustrates the behavior of the system under normal and faulty condition in relation to SVDD boundary.

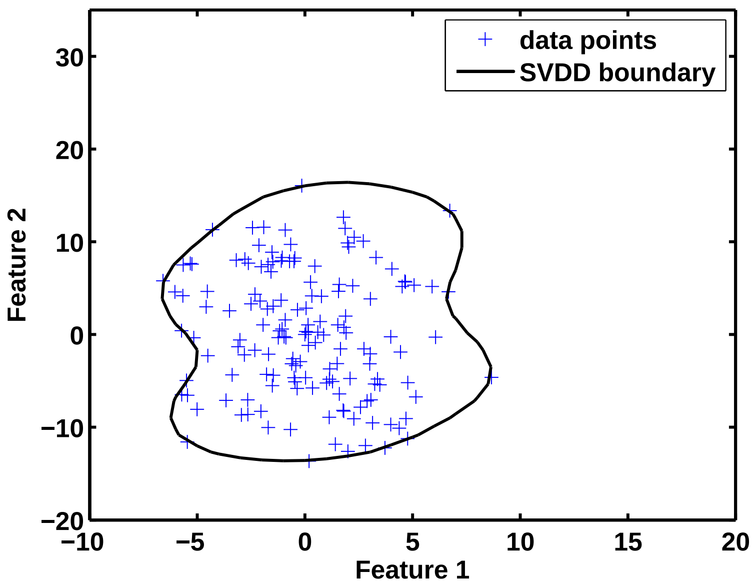

Figure 3.1: Example NOC region in 2D feature space of 3TS data enclosed by SVDD boundary.

We define the threshold or control limit for fault detection as the radius of the sphere. Therefore, by computing the distance of the test points from the center of the sphere, $D = \|\mathbf{a} - \mathbf{x}\|$ ($\mathbf{a}$ is the center of the sphere), and comparing to radius, the condition of the process can be monitored and if any fault occurs, it can be captured. Detection is achieved by checking inequalities in 3.1.

$$\begin{cases} D < R, & normal \\ D = R, & \mathbf{x} \text{ on the boundary} \\ D > R, & faulty \end{cases}, \quad (3.1)$$

where $\mathbf{x}$ is a test data point and R is the radius of the sphere. In this work, SVDD was implemented using ddtools and PRtool Matlab toolboxes provided by [52] and [53] (can be downloaded from:`http://homepage.tudelft.nl/n9d04/dd_tools.html` and `http://www.prtools.org` respectively).

3.3.1 SVDD Computational time problem

The SVDD computation time drastically increases as the number of training data and its dimensionality increases. To solve this problem we modify SVDD by using split and combine

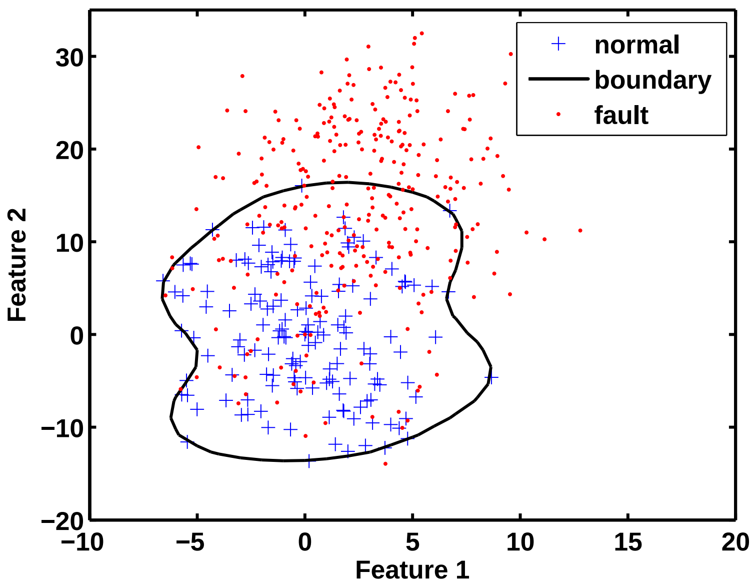

Figure 3.2: Example NOC region in 2D feature space of 3TS data enclosed by SVDD boundary and the effect of fault on the data causing to leave the NOC region.

method proposed in [54]. In this method, a small subset of the training data is randomly selected and SVDD is applied to the subset and its support vectors are retained while the rest of the points are removed. Another subset is taken from the remaining data set and the second set of SVs are added to the first SV sets and SVDD is applied to the combined set of SVs and new set of SVs are found. The procedure continues until all data points in the original training set are used. The only parameters to be tuned for this algorithm are the size of the subsets, L and the kernel width, σ. The split and combine algorithm for fast SVDD is presented here [54]:
Suppose we have an $n \times m$ training data set X.

1. Randomly form a set, L_1, with l points from X and apply SVDD on the set and find the solution as W_1 and the set of support vectors as SV_1
2. Subtract L_1 from X and form set L_2 the same as L_1 with the same size
3. If number of points in $X < l$ then the solution is W_1 otherwise proceed to next step
4. Find solution for L_2 the same way as L_1 and let SV_2 be the set of support vectors for L_2

5. Subtract L_2 from X and update L_2

6. Let $L_3 = SV_1 \bigcup SV_2$ and find SVDD solution for L_3 as W_3. Let SV_3 be the support vectors of L_3

7. If number of points in $X < l$ then the solution is W_3 otherwise proceed

8. Update L_2 and go to step 4

The computation time of the fast SVDD algorithm is compared with the original SVDD for data sets with three different sizes to show the effectiveness of the improved algorithm. A benchmark sample data set named *Banana set* is used for comparison. It is shown in Table 3.1 that computational time is reduced significantly with improved fast SVDD while, both algorithms give almost the same SVDD boundary solution as shown Figure 3.3. In this table, computation time for a set of 100 points is very close for SVDD and fast SVDD. However, when the number of data points increases to 200, the computation time for fast SVDD doubles, while it increases more than 6 times for SVDD. This condition becomes worse for a set of 400 points.

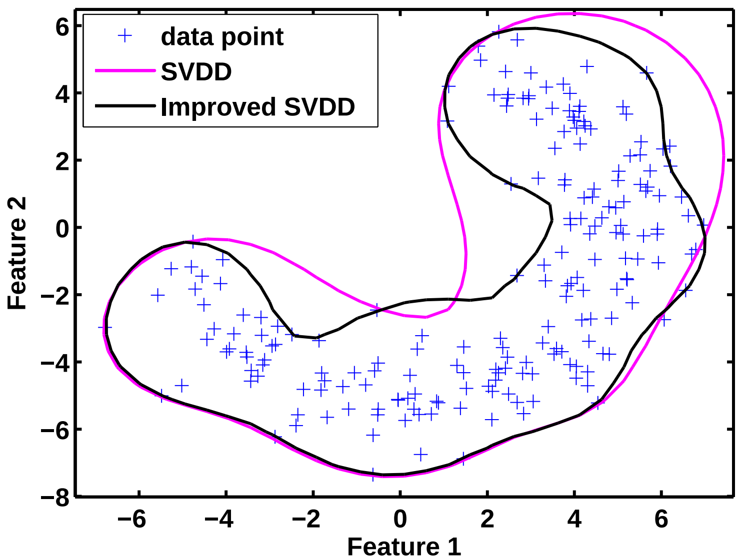

Figure 3.3: SVDD and improved fast SVDD results on Banana data set.

Table 3.1: Computation time comparison of SVDD and fast SVDD

banana set	computation time (sec)	
number of data	SVDD	fast SVDD
100	.462	.439
200	3.186	.958
400	116.021	2.405

We also examine fast SVDD sensitivity to variation of its subset size parameter, L. The results are presented in Figure 3.4. In this figure, it can be seen that the training time increases when L is greater than 50% of the original data size. Therefore, selecting L as $20-50\%$ of the original training set would be reasonable. Note that too small values for L means very few samples in the subsets which results in having too many subsets and in turn, increases the computation time.It should be noted that from here on, we refer to fast SVDD by SVDD unless specified.

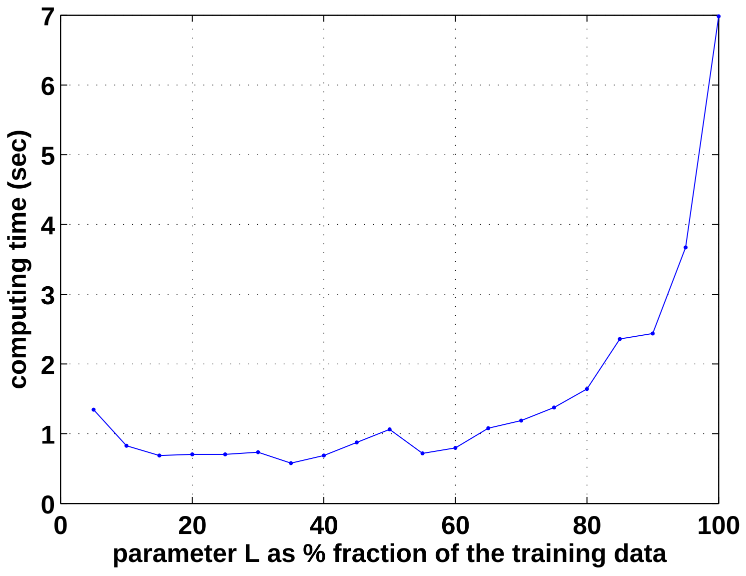

Figure 3.4: Fast SVDD computing time with respect to parameter L for a sample set of training data with 200 points.

Fault detection results of applying fast SVDD to three systems (a simple nonlinear system, Tennessee Eastman process, and three tank real system) are presented in the following chapters.

3.4 Chapter Summary

This chapter describes the structure of the fault detection system based on SVDD. The training and testing method, and threshold definition are presented and discussed. Also, dimensionality and computational time problem is addressed and the solution is provided in this chapter. A faster algorithm used in the fault detection structure is presented in detail. The results show considerable enhancement in terms of computational complexity and time. In the next chapter, the proposed SVDD fault detection method is implemented on three benchmark systems to examine SVDD's performance in comparison with standard methods. Fault detection system's parameter adjustment methods are presented and performance criteria are introduced in the next chapter. Finally, the results of different experiments are presented and discussed in detail.

Chapter 4

Fault detection application

In this chapter, the SVDD based fault detection method is investigated and the experimental results on three different systems are presented and discussed in detail. The systems are: a simple multi-variable system, Tennessee Eastman process as a benchmark simulated process, and a real three tank system. The results of the SVDD fault detection are compared with ICA and PCA. Also, a combination of SVDD with ICA and PCA is studied and compared with simple SVDD.

4.1 Detection performance criteria

The performance of any fault detection method should be measured based on predefined criteria. The most common criteria used in the literature are false alarm rate, missed alarm rate and detection delay. False alarm rate is computed as the number of observations in a set of normal data that are detected as faulty. In fact, false alarm indicates presence of fault in the process while there is no fault in the process. On the other hand, missed alarm rate is defined as the number of faulty observations that are not detected in a set of data. False alarm and missed alarm rates are affected by threshold limit values. Too low threshold limits result in high false alarm rates while high threshold values increases missed alarm rate. The detection delay is considered as the time between the occurrence of fault in the system and the time it is detected by fault detection system[3]. In this work, we compute the three mentioned measures for each experimental case study to assess the performance of the proposed method. By convention, every three consecutive samples exceeding the threshold is counted as an alarm.

4.2 Implementation and experiment

4.2.1 Simple multivariable process

We first examine the performance of SVDD on a simple multivariate process. The process contains five variables which was proposed by Ku [38] and modified by Lee et al. [34]. The system is defined as follows:

$$\mathbf{z}(k) = \begin{pmatrix} .118 & -.191 & .287 \\ .847 & .264 & .943 \\ -.333 & .514 & -.217 \end{pmatrix} \mathbf{z}(k-1) + \begin{pmatrix} 1 & 2 \\ 3 & -4 \\ -2 & 1 \end{pmatrix} \mathbf{u}(k-1) \tag{4.1}$$

$$\mathbf{y}(k) = \mathbf{z}(k) + \mathbf{v}(k) \tag{4.2}$$

where input $\mathbf{u}$ is defined as:

$$\mathbf{u}(k) = \begin{pmatrix} .811 & -.226 \\ .477 & .415 \end{pmatrix} \mathbf{u}(k-1) + \begin{pmatrix} .193 & .689 \\ -.320 & -.749 \end{pmatrix} \mathbf{w}(k-1) \tag{4.3}$$

In the above equations, $\mathbf{v}$ is a random noise vector with its elements having zero mean and variance of 0.1. Vector $\mathbf{w}$ is a random vector with elements uniformly distributed in (-2,2) interval. Each point in the data set is a vector defined as $\mathbf{x}(k) = [\mathbf{y}^T(k) \quad \mathbf{u}^T(k)]^T$ with five variables, y_1, y_2, y_3, u_1, u_2. For monitoring and fault detection, two types of fault are introduced[34].

- Exerting step change in w_1 by four different values (2, 3, 5, and 7 units) at sample 50
- Increasing w_1 linearly from sample 50 by adding $f(k-50)$ to the value of w_1 at each sample where, k is the sample number and *fault parameter,* $f \in \{.05, .1, .15\}$

Also, in some cases, fault was applied from sample 50 to 150. For analysis, 300 samples are generated in each case. Figure 4.1 shows process variables having gradual fault with $f = .05$.

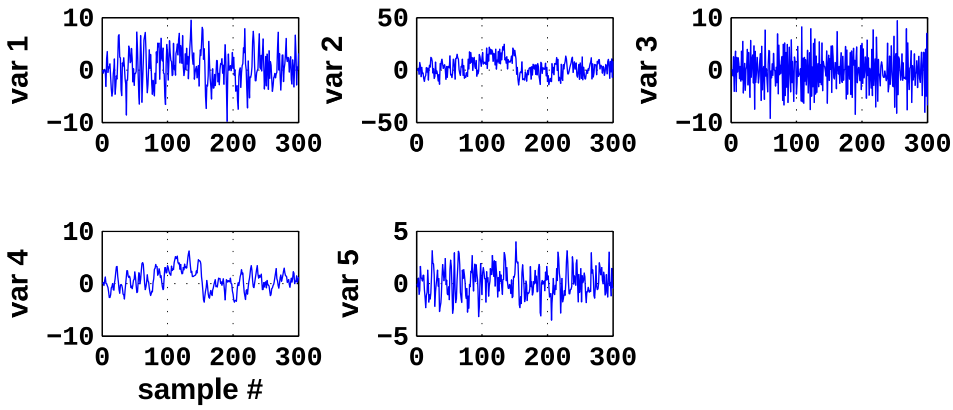

Figure 4.1: Example system states when gradual fault occurs in the system

SVDD parameter adjustments

A SVDD structure is trained with normal operating condition, NOC, data. The structure has only two parameters that need to be selected. One is the size of the split in SVDD algorithm, L, and the second is the kernel width parameter, σ, which is an influential parameter and highly affects the results. To select these two parameters, cross validation approach is suggested. The parameters are obtained by running 5-fold cross validation for each set of (L, σ) on the training data set over the following range for each parameter and finding training error. The range of values considered here is $10 - 100$ for L and $.1 - 10$ for σ.

In k-fold cross validation, the training data set is randomly divided into k subsets. A subset is retained for validation and the rest of the $k-1$ subsets are used for training. The process is repeated for each of the k subsets and the best solution is achieved by averaging the results from each run.

As a result of the cross validation process, a graph of the training error with respect to L and σ can be plotted which gives insight in selecting appropriate parameter values. The number of support vectors in SVDD structure can also be plotted with respect to L and σ to assist selecting parameters. For the simple multivariable system mentioned above, the graph is shown in Figure 4.2. In this Figure, variation of L, compared to σ, has small effect on error. It can be seen that for small values of σ, the training error is very large and it gradually decreases by increasing σ. Although error is decreasing, selecting large values for

σ results in losing sensitivity to faults. In other words, large σ enhances false alarm rate but missed alarm rate is worsened. If training data for fault conditions are available, then the graph of false alarm rate, missed detection rate and detection delay over a range of different σ values can be used to assist in selecting σ. Figure 4.3 shows that false alarm curve intersects with missed detection curve at $\sigma = 7.5$, suggesting that this value would be the optimum. However, it is assumed that training faulty data are not available, which is the case in many real world processes. Therefore, $\sigma = 5$ is selected which is a conservative decision to stay on the safe side. Here, false alarm rate is sacrificed to get better missed detection rates.

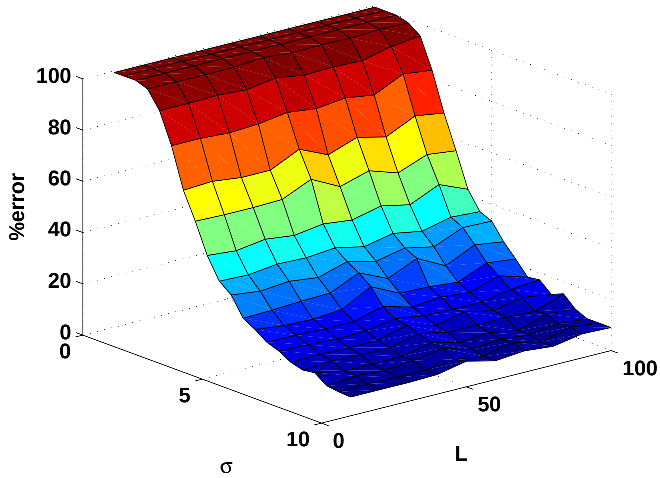

Figure 4.2: Training error surface graph showing variation of error with respect to SVDD parameters

Based on the above findings the two parameter values are selected as: $L = 40$ and $\sigma = 5$ for the SVDD structure which is used for fault detection. The next step would be the testing of different faulty conditions by the obtained SVDD. As an example, the output of SVDD is presented in Figure 4.4 for a ramp fault.

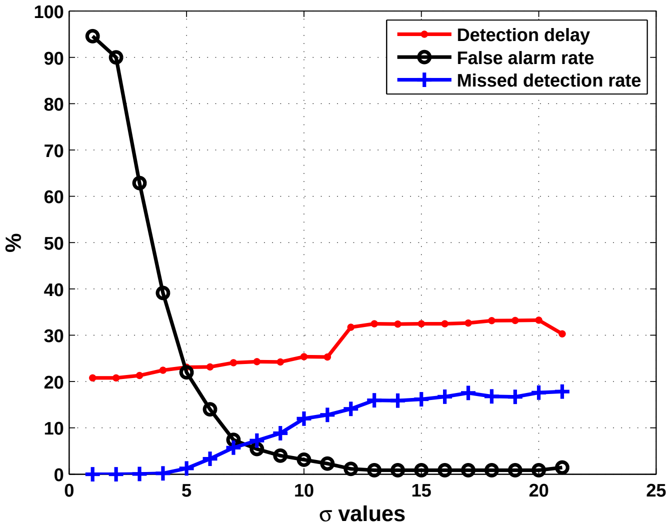

Figure 4.3: False alarm, missed detection, and delay curves for different values of σ.

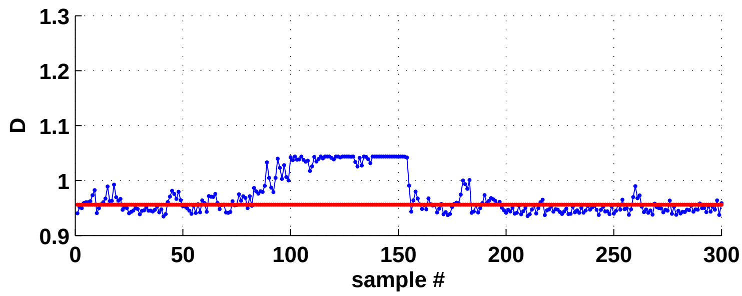

Figure 4.4: Example SVDD monitoring graph for detecting system faults with fault parameter=.1.

PCA parameter adjustment

The problem of selecting the number of principal components has been studied in several research work and many different methods have been suggested. Although many techniques exist for determining the number of PCs (ICs) in the literature, it seems that there is no dominant method. Percent variance test, scree test, parallel analysis and prediction sum of squares are some of the methods suggested in the literatue[60] [62]. The most common way is to select the number of PCs by inspecting the plot of the variance contributions of each principal component and finding the smallest number of PCs required to explain specific percentage of total variance. Since the variance associated with each PC is equal to the corresponding eigenvalue of the covariance matrix, the number of PCs can be determined by sorting eigenvalues of covariance matrix and selecting the first few eigenvalues. Another method is to find the location of the eigenvalue in which the covariance profile has a break and becomes linear. This method is known as scree test and assumes that the linear part of the variance corresponds to random noise. The eigenvalue plot and the graph of cumulative variance explained by PCs are given in Figure 4.5. It is seen that four components explain almost 100 % of the variance of the NOC data. Therefore, selecting the first four principal components for PCA model would be reasonable. Hotelling's T^2 and Q statistic are calcu-

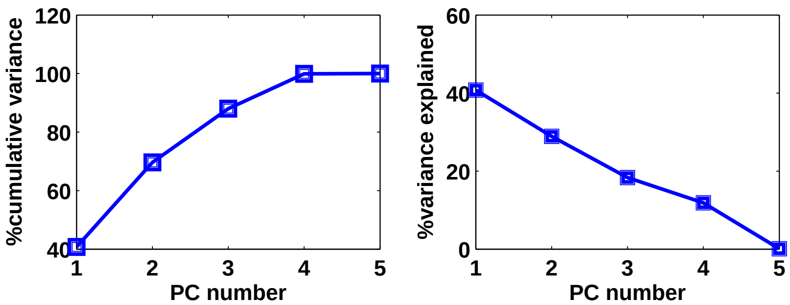

Figure 4.5: Scree plot of the principal components for the system.

lated through equations 2.7 and 2.9 respectively. As noted earlier, the threshold for T^2 is determined by equation 4.4, considering 95% confidence limit as:

$$T^2_\alpha = \frac{a(n-1)(n+1)}{n(n-a)} F_\alpha(a, n-a). \tag{4.4}$$

The threshold for Q-statistic is calculated by Q_α, formulated as

$$Q_\alpha = \theta_1 [\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}]^{1/h_0},$$

where

$$\theta_i = \sum_{j=a+1}^{n} \lambda_j^i, \quad i = 1, 2, 3,$$

and

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}.$$

PCA parameter and threshold values are given in Table 4.1.

Table 4.1: PCA parameters for the simple multi-variable system

parameter	value
# of PCs	4
$F_\alpha(a, n-a)$	2.0402
c_α	1.644

ICA parameter adjustment

One parameter that needs to be selected by designer for ICA is the number of ICs which in turn, determines the number of dimensions after dimensionality reduction process. One way of selecting the number of ICs is suggested by Hyvrinen in [35], [36] in which the number is determined by looking at the eigenvalues, λ_i, of NOC data covariance matrix and discarding eigenvalues that are small, as performed in principal component analysis technique. The eigenvalues of the NOC process data are given in Figure 4.6. We can see that the last eigenvalue is negligible compared to other four, therefore we can retain the first four eigenvalues and discard the last one which reduces data dimension by one. Dimension reduction has the effect of reducing noise and sometimes prevents over-learning. Another way of selecting the number of ICs is proposed by Lee et al. [50], which is based on the $L_2 - norm$ of the rows of de-mixing matrix, W. It is assumed that the rows with highest norm values have the greatest effect on the variation of the corresponding element of the independent component vector.

Fault detection results for simple multi-variable system

For fault detection purpose, 10 cases of fault scenarios are designed and tested with SVDD, PCA, and ICA. There is a step fault, a gradually increasing fault (we call it ramp), and an

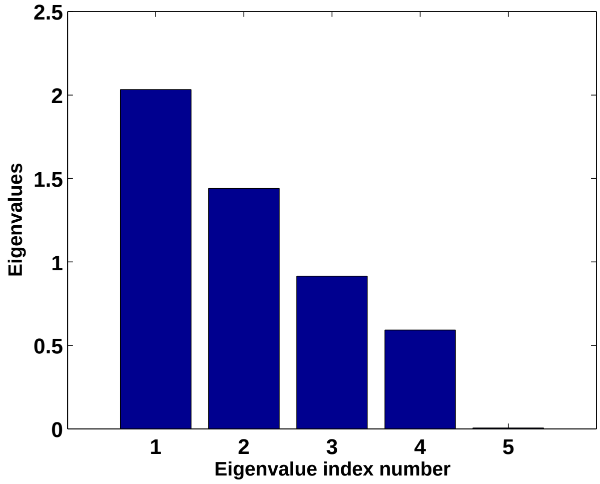

Figure 4.6: Eigenvalues of the system used for selecting the order of data reduction.

increasing fault that occurs in the system for a short time and then disappears. Different intensities (at least two cases) are considered for each type of fault. In each experiment, the process is run for 300 samples and a fault is triggered at sample 50. Table 4.2 summarizes detailed information of faults in each case. We applied SVDD, ICA, and PCA to test data

Table 4.2: Faults specification for the simple multi-variable system. Each experiment contains 300 sample points and fault appears at sample #50.

fault#	type	parameter	fault ends at sample
1	ramp	.05	150
2	ramp	.1	150
3	ramp	.15	150
4	ramp	.05	300
5	ramp	.1	300
6	ramp	.15	300
7	step	2	300
8	step	3	300
9	step	5	300
10	step	7	300

in each case and obtained false alarm rate (for normal test data), missed detection rate (for faulty data) and detection delay time. Tables 4.3, 4.4, and 4.5 contain the results of fault detection for the system.

Table 4.3: Missed detection rates for SVDD, ICA, and PCA methods applied to simple multi-variable system. In this table, D^2 is the SVDD distance measure, I^2 is ICA statistic, SPE is ICA squared prediction error, T^2 is PCA Hoteling's statistic, and Q is PCA prediction error

	SVDD	ICA		PCA	
fault#	D	I^2	SPE	T^2	Q
1	8.4	11.2	9.2	10.0	7.2
2	3.2	7.6	3.2	7.6	7.2
3	3.6	4.0	4.4	4.4	2.4
4	9.6	10.8	9.6	10.8	7.6
5	2.8	4.8	6.4	4.8	4.0
6	0.8	4.0	5.2	4.0	4.0
7	18.0	11.2	22.0	10.0	6.8
8	8.8	1.2	2.8	1.2	0.8
9	0.0	0.0	0.4	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0

Table 4.4: False alarm rate in testing the simple multi-variable system.

SVDD	ICA		PCA	
D	I^2	SPE	T^2	Q
18.0	5.8	1.4	6.0	3.0

Table 4.5: Detection delays for each fault detection method applied to the simple multi-variable system.

	SVDD	ICA		PCA	
fault#	D	I^2	SPE	T^2	Q
1	39	38	63	26	26
2	27	27	32	27	22
3	17	17	19	17	17
4	39	32	48	32	30
5	18	25	26	25	25
6	6	17	17	17	17
7	8	11	7	11	16
8	17	10	11	10	7
9	7	4	5	4	4
10	5	4	4	4	4

A more descriptive comparison of missed detection rates and detection delays between SVDD, ICA, and PCA can be deduced from Figures 4.7 and 4.8. It can be seen that SVDD performance is the same as ICA for most cases except fault 5, and 6, where SVDD shows lower missed detection rate than ICA, and faults 8, and 9, where SVDD shows higher missed detection rates. Fault 8 and 9 are step faults with low intensity. Comparing SVDD with PCA reveals same result as SVDD vs. ICA comparison, except that in addition, SVDD performs better in detecting fault 2. Inspecting Figure 4.8 shows that SVDD detection delay times are close to ICA measures for most cases. The only significant difference is in fault 6 where SVDD detects the fault earlier. On the other hand, PCA gives better detection times than SVDD for faults 1, 4, 8, and 9, while faster detection is achieved by SVDD for fault 5, 6, and 7, and the rest of the faults are detected almost at the same time by the two method. Figures 4.9, 4.10, and 4.11 show the performance of SVDD for normal condition and two other fault cases, i.e., a gradual fault, and a step fault in the system. As shown in Figures, although there are some noise in the graphs for small faults, SVDD is capable of detecting faults in this multi-variable system.

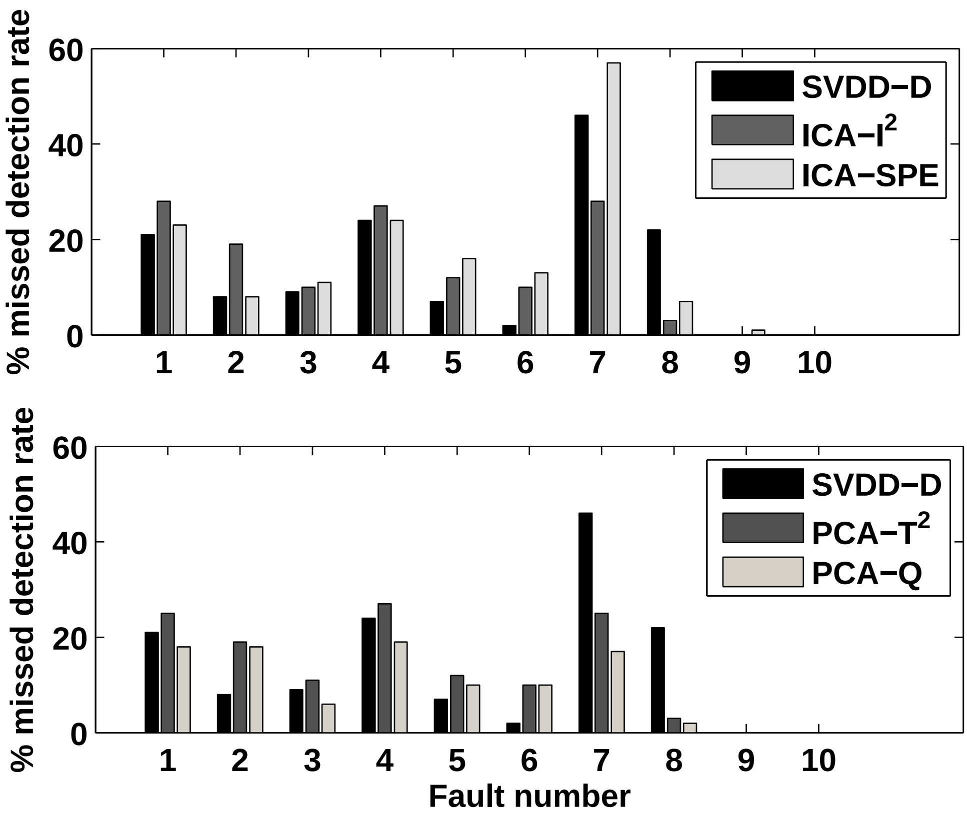

Figure 4.7: Missed detection rates for SVDD, ICA, and PCA methods applied to example multi-variable system.

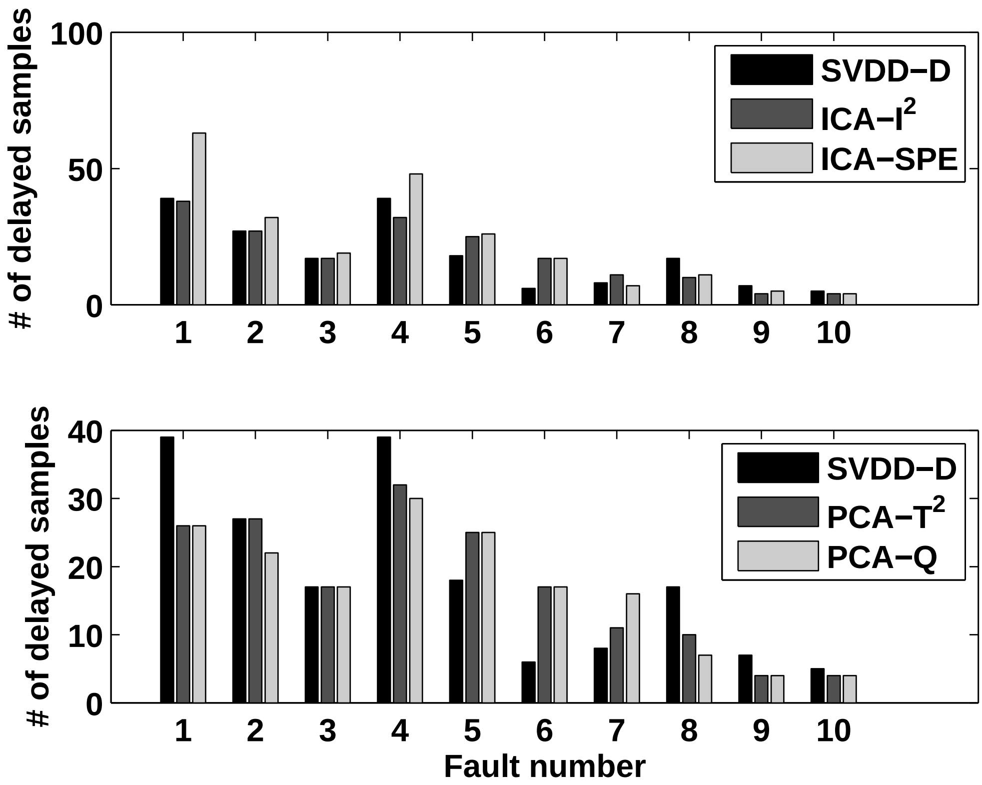

Figure 4.8: Detection delays for SVDD, ICA, and PCA methods applied to example multivariable system.

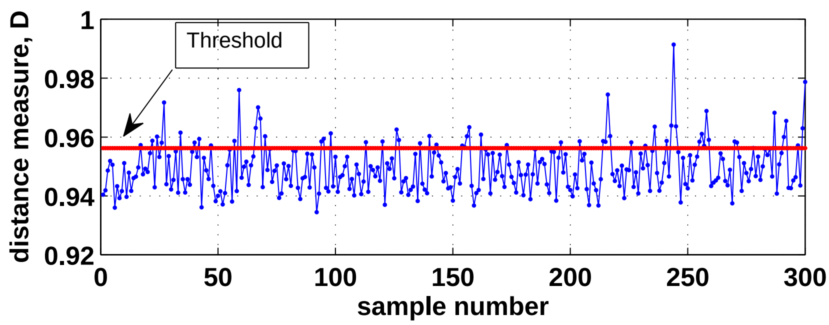

Figure 4.9: SVDD monitoring results in normal condition

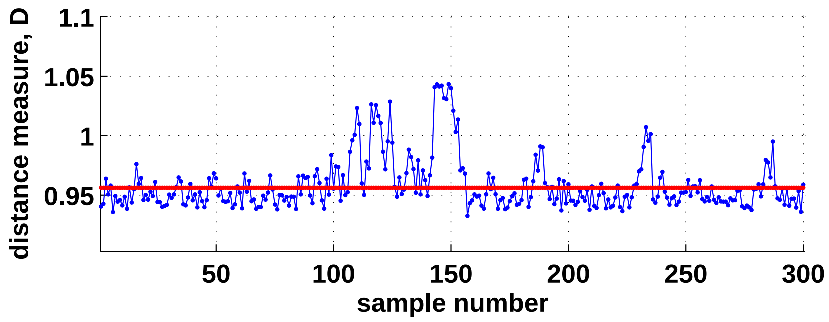

Figure 4.10: SVDD monitoring results for system with gradual fault, $f = .05$.

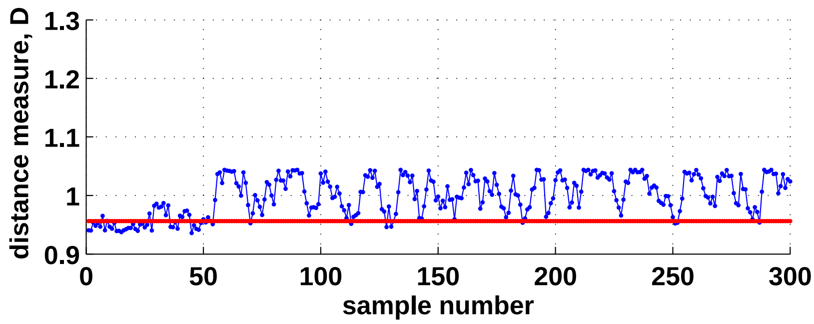

Figure 4.11: SVDD monitoring results for system with 5 unit step fault.

4.2.2 Three tank system (3TS)

As a benchmark control problem, the three tank system (3TS) is used in many different research works. The basic structure of the system contains three tanks which are connected to each other by pipes. Two tanks are filled with two pumps while the third one is filled only through the pipes connected to the other two. Our experimental setup is an AMIRA DTS200 in which the water level is measured with three piezo-resistive difference pressure sensors [56]. DTS200 contains 6 valves which are used to emulate clogging and leakage in the system. Figure 4.12 shows the system flow sheet. The system has the following

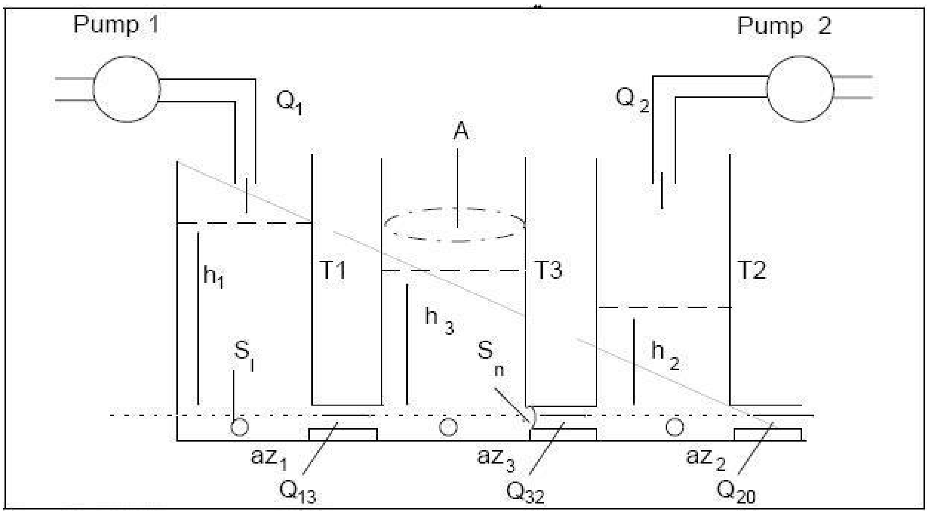

Figure 4.12: Three Tank system structure[56]

specifications:

- Tank cross section area, $A_{cross} = .0154\ m^2$
- Connecting pipes cross section area, $az = 5 \times 10^{-5} m^2$
- Highest liquid level, $H_{max} = 62cm$
- Maximum pump flow rate, $Q_{max} = 100\ mltr/sec$

Our three tank system is equipped with a disturbance module which allows simulating different types of faults for fault detection research purposes including sensor faults, actuator faults, leakage in each tank, clog in connecting pipes, and clog in the outflow. Training and testing data with five variables including water levels and flow rates have been collected for experiment. In this work, we define 12 different fault scenarios which are described in Table 4.6. Faults are instigated at sample 166 in each case and continue until 1000^{th} sample. We assume that only one fault occurs at a time and there are no simultaneous faults. Since

Table 4.6: Faults specification for Three tank system. Each experiment contains 1000 sample points and fault appears at sample #166.

fault#	type	severity
1	clog in outflow	-
2	clog between tank 2 and 3	-
3	leak in tank 1	high
4	leak in tank 1	low
5	leak in tank 3	high
6	fault in pump 1 flow rate	gradually increasing
7	fault in pump 2 flow rate	10%
8	fault in pump 2 flow rate	5%
9	step in sensor 1	5%
10	step in sensor 1 fault	10%
11	step in sensor 3	10%
12	step in sensor 3	20%

exerting leakage with specific severity is not possible with our three tank system, leakage is indicated as low or high which is set by approximately opening leakage valves. Sensor and pump faults are induced in the system by disturbance module that allows for a scaling of the processed sensor signals and the control signals in a range of 0% to 100%. For each fault case, training and testing data have been prepared by conducting experiments for each scenario. An example of the system variables when fault occurs in pump2 is shown in Figure 4.13.

Parameter adjustment

The parameters selected for SVDD, PCA, and ICA structures are selected based on the methods mentioned earlier. The values are given in Table 4.7. The number of ICs and PCs to be retained is selected from eigenvalues and variance cumulative sum plot shown in Figure 4.14 as described before. The value of σ for SVDD structure is determined from Figure 4.15 which presents training error for different values of σ and L. Again, L shows less sensitivity to error, so that we select $L = 50$. The σ values greater than 1.5 give lower error,

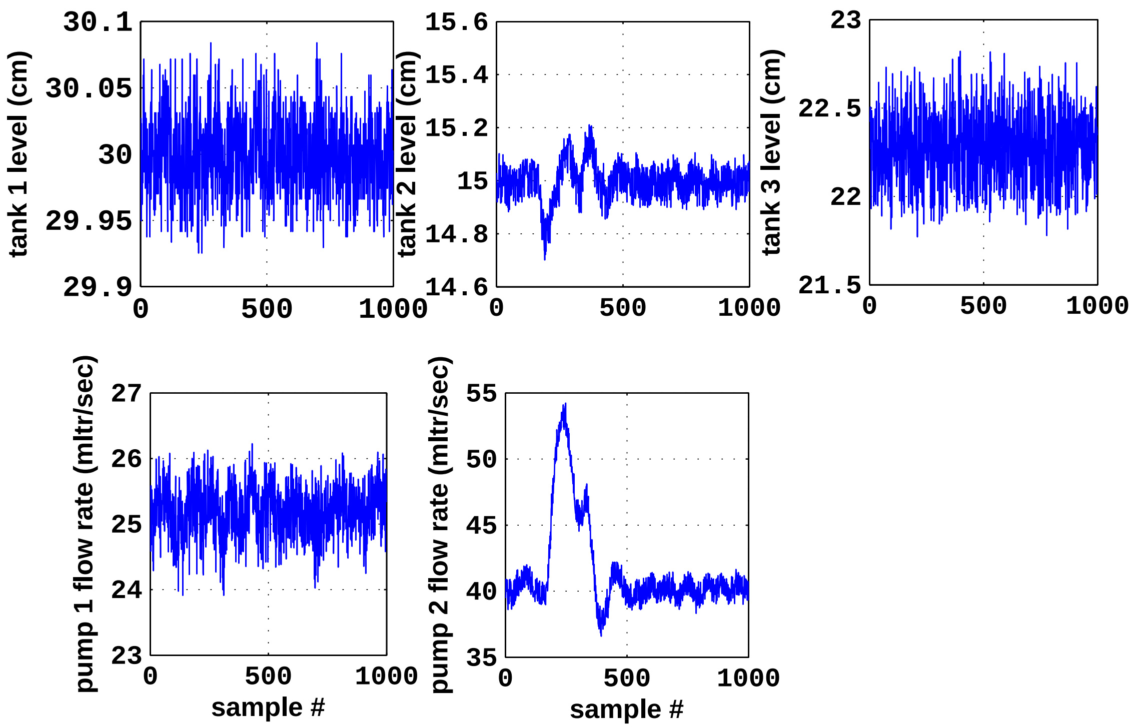

Figure 4.13: Example of the three tank system variables in faulty condition.

compared to smaller values. However, there is a risk that sensitivity to faults will be lost. Therefore, two values for σ are selected; one from the region between the two peaks of the surface, and one from the flat part; specifically, $\sigma_1 = .7$ and $\sigma_2 = 1.5$. We train SVDD with each value and test them on faulty data. Comparing SVDD detection performance with the two values shows that the one with $\sigma = .7$ is more capable of detecting faults in the system. Therefore, $\sigma = .7$ is selected. Figure 4.16 illustrates SVDD detection results when trained with the two values. It can be seen that missed detection rates are considerably lower in case of $\sigma = .7$, especially for the last four faults (step faults). Although, false alarm rate is higher (around 5%, compared to 1%) when using $\sigma = .7$, the detection performance totaly outperforms the other case and justifies the use of this value for SVDD. Therefore, $\sigma = .7$ is selected.

Table 4.7: Parameter values for SVDD, ICA, and PCA methods applied to Three tank system for fault detection.

method	parameter	value
SVDD	L	50
	σ	.7
ICA	# of ICs	3
PCA	# of PCs	3
	$F_\alpha(a, n-a)$	2.627
	c_α	1.644

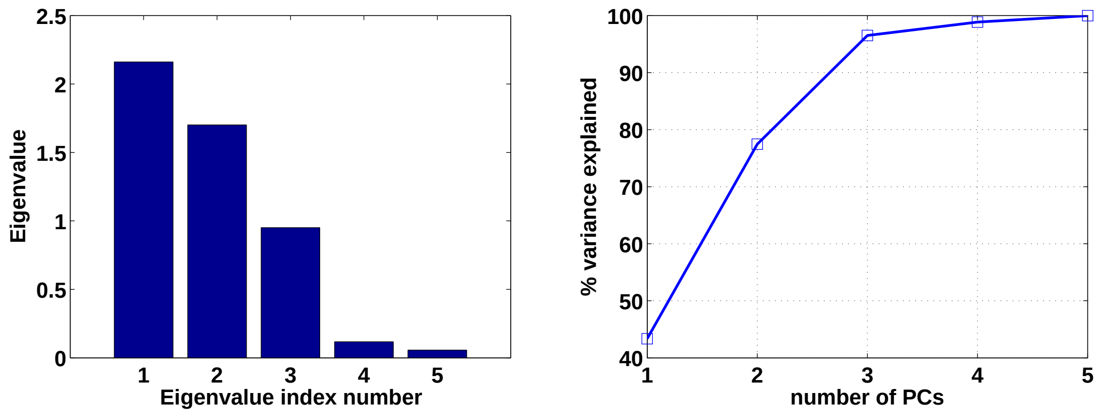

Figure 4.14: Left) NOC data covariance matrix eigenvalues Right) Cumulative sum of variance explained by PCs.

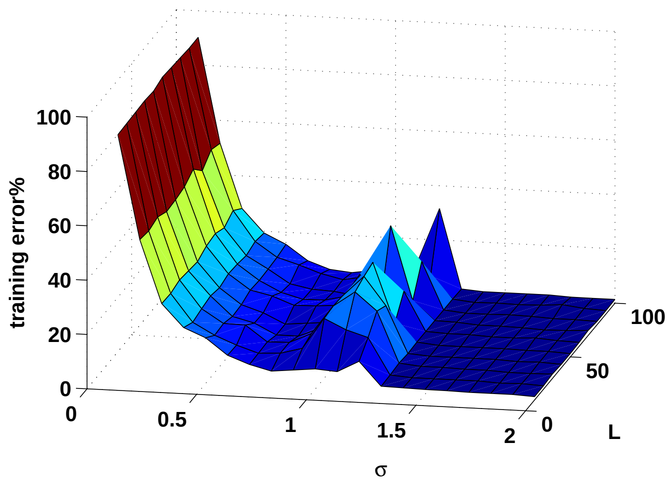

Figure 4.15: Error surface of SVDD for different values of σ and L.

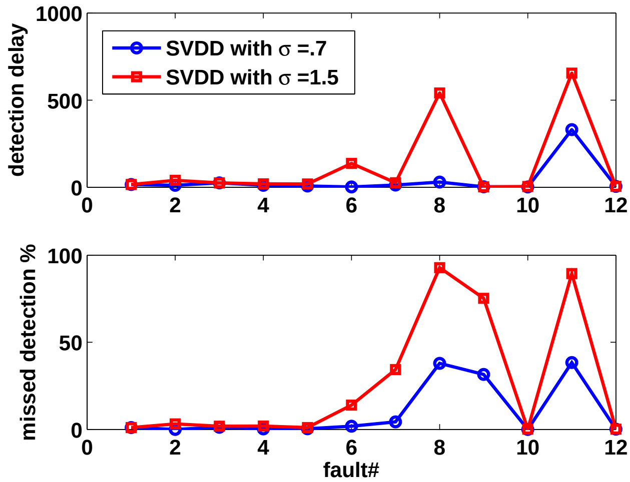

Figure 4.16: SVDD performance for two different values of σ.

Fault detection results for 3TS

Fault detection task is carried out with SVDD, ICA, and PCA for each fault scenario introduced in Table 4.6. As an example, the outputs of SVDD, ICA, and PCA, detecting sensor fault in tank 1, are presented in Figure 4.17.

We apply fault detection methods to each fault data set and find false alarm rates, missed detection rates and detection delays for each method. The results are shown in Tables 4.8 and 4.9. The overall fault detection performance of each method is found by averaging performance criteria for all faults as presented in the form of bar plot in Figure 4.18. In this Figure, it is seen that SVDD false alarm rate is slightly greater than ICA-I^2 and PCA-T^2 measures while it is much smaller than ICA-SPE and PCA-Q statistics values, meaning that SVDD outperforms ICA and PCA, since the maximum of the two criteria is considered, i.e., false alarm rate of ICA is the maximum of I^2 and SPE, and the same for T^2 and Q in PCA. Also, missed detection rates for ICA and PCA are higher than SVDD indicating that SVDD performs better than ICA and PCA in detecting different faults. The bar plot on the right shows detection delay in which SVDD is slightly slower than ICA and PCA in detecting faults.

Although, overall, SVDD outperformed ICA and PCA, we investigate its performance in detail for each fault case. Analyzing resultant data in Figure 4.19 shows that for faults 1-5 (leaks and clogs), SVDD, ICA, and PCA missed detection rates are close to each other and the difference remains below 5%. SVDD rate is lower than ICA and PCA in detecting fault 6, 8, 9, and 11. Fault 6 is a gradual fault in pump1. As introduced earlier, fault 7 and 8 are the faults in the second pump with 10% and 5% severity. It is seen that SVDD missed detection rate is lower in detecting faults with less severity. The same pattern is seen for fault 9 and 10 (5% and 10% fault in sensor 1) and fault 11 and 12 (10% and 20% fault in sensor3) , where SVDD has lower missed detection rate than ICA and PCA for faults with less severity. For larger faults, no difference is seen in detection performance of the three methods. ***This shows that in 3TS process, SVDD is more sensitive to smaller faults than ICA and PCA and it can detect faults before their magnitude increases***.

Considering detection delays, it is seen that all three methods have close detection delays except for fault 8 and 11, where SVDD detection delay is lower for fault 8, but much higher than ICA and PCA for fault 11. To summarize, SVDD (if tuned properly) has

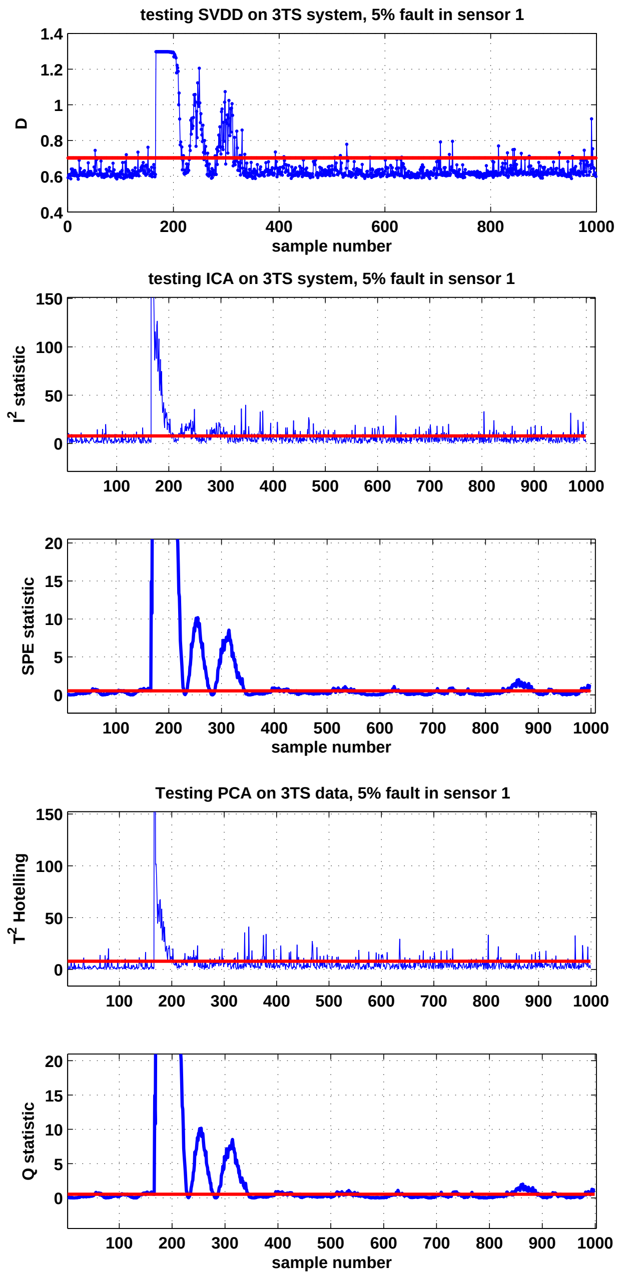

Figure 4.17: Example results of SVDD,ICA, and PCA, detecting sensor fault in Three tank system.

Table 4.8: Missed detection rates and detection delays for SVDD, ICA, and PCA methods applied to Three tank system system. In this table, D^2 is the SVDD distance measure, I^2 is ICA statistic, SPE is ICA squared prediction error, T^2 is PCA Hoteling's statistic, and Q is PCA prediction error

	missed detection					detection delay				
	SVDD	ICA		PCA		SVDD	ICA		PCA	
fault#	D	I^2	SPE	T^2	Q	D	I^2	SPE	T^2	Q
1	1.07	1.07	0	1.79	0	16	19	2	19	2
2	0	0.23	0	0.47	0	11	26	2	26	2
3	1.19	1.67	0.71	1.91	0.71	25	33	10	37	10
4	0.23	0.47	2.39	0.47	2.39	11	7	24	7	24
5	0.35	0.47	2.39	0.95	2.51	7	18	15	18	15
6	1.79	3.83	6.95	5.99	6.95	2	80	2	80	2
7	4.31	19.66	1.07	33.45	1.07	13	14	17	14	17
8	37.88	43.28	79.37	51.67	79.37	30	61	281	61	281
9	31.53	40.04	54.55	54.43	54.55	3	3	3	3	3
10	0	0	0	0.23	0	2	2	6	3	6
11	38.36	52.75	58.63	63.30	58.63	331	444	46	607	46
12	0.11	0.11	71.22	0.11	71.22	5	5	6	5	6

Table 4.9: False alarm rate in testing 3TS.

SVDD	ICA		PCA	
D	I^2	SPE	T^2	Q
5.57	2.86	25.05	1.40	25.05

shown promising results in three tank system fault detection with considerable capabilities, compared to PCA and ICA methods.

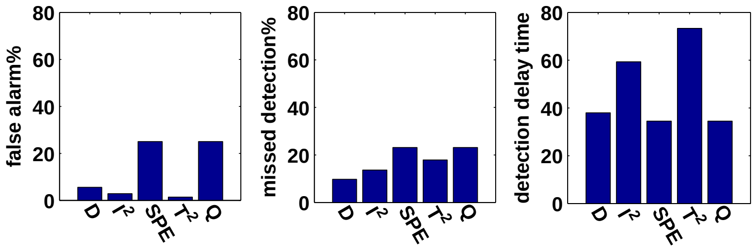

Figure 4.18: Overall performance of 3TS evaluated by average false alarm rate, missed detection rate and detection delay.

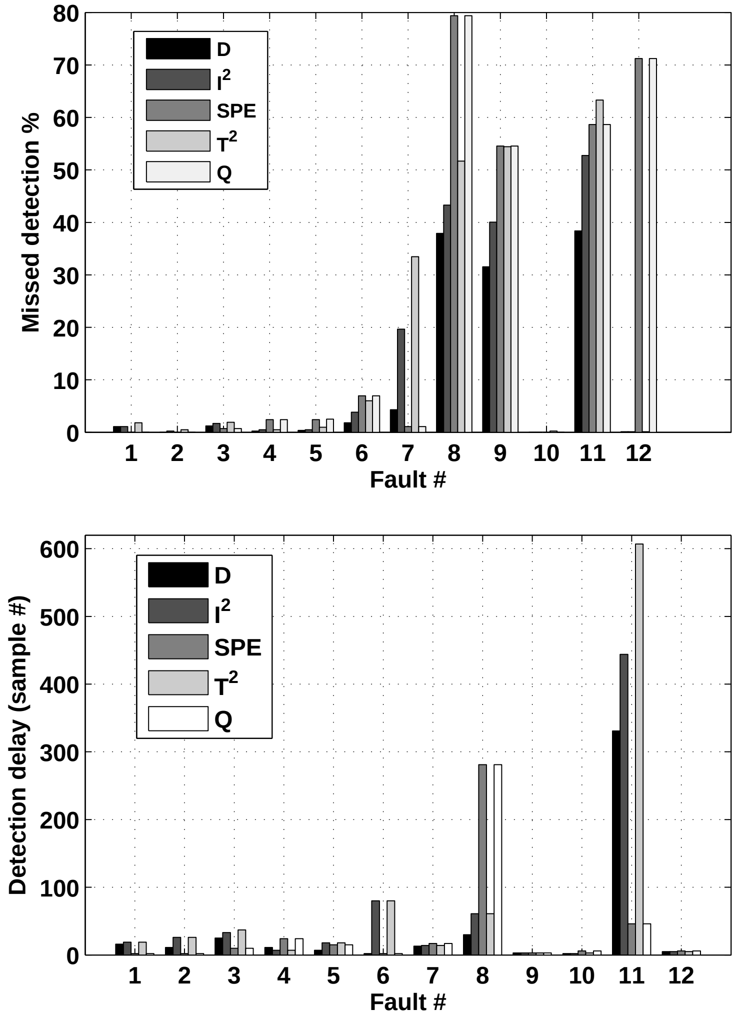

Figure 4.19: Fault detection criteria (missed detection rate and detection delay) for different faults in 3TS.

4.2.3 Tennessee Eastman process (TEP)

Tennessee Eastman process, TEP, is a chemical plant that was proposed and modeled by Downs and Vogel [42] as a plant-wide control challenge problem. The process involves four exothermic gas reactions. These reactions are irreversible and exothermic with rates that depend on temperature and on the reactor gas phase concentration of the reactants. The process has five major units: reactor, product condenser, vapor-liquid separator, recycle compressor and product stripper. Figure 4.20 shows the flow sheet of the TE process presented by Downs and Vogel [42].

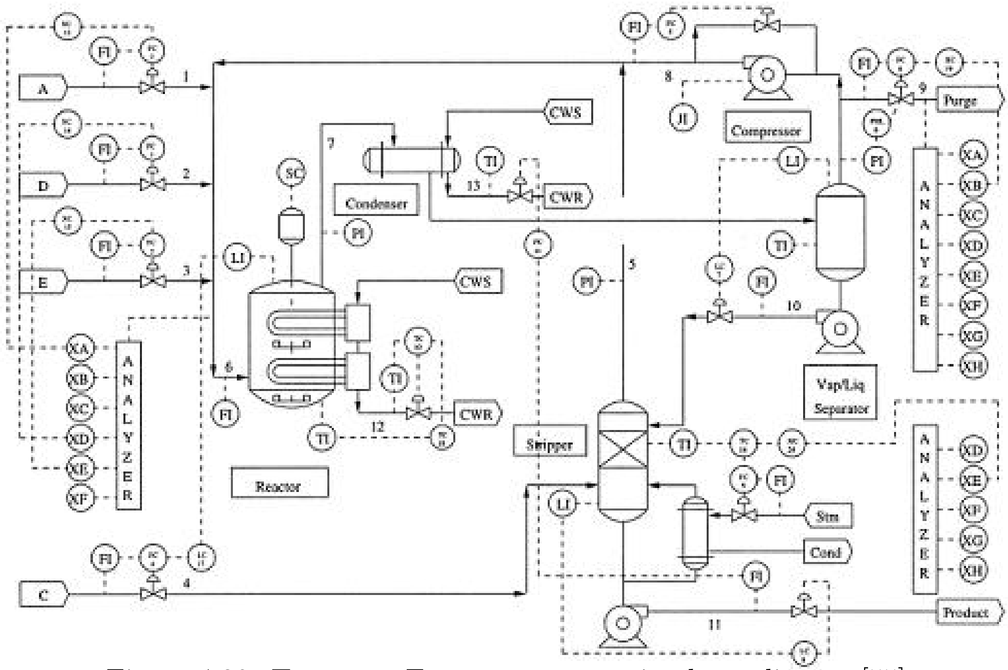

Figure 4.20: Tennessee Eastman process simulator diagram[55]

Four reactants are the inputs of the process that produce two products and two byproducts named alphabetically from A to H. The heat of the reactions is removed by cooling water in a heat exchanger. The products and unconverted reactants leave the reactor as vapor which is partly converted to liquid in the condenser. Table 4.10 presents six operating modes for the TE process with steady-state material and heat balance. The first mode is the base case as indicated in the table and will be considered in this work.

Table 4.10: Base case steady-state material and heat balance[42]

Mode	G/H mass ratio	Production rate (stream 11)
1	50/50	7038 kgh^{-1}G and 7038 kgh^{-1}H (base case)
2	10/90	1408 kgh^{-1}G and 12,669 kgh^{-1}H
3	90/10	10,000kgh^{-1}G and 1111 kgh^{-1}H
4	50/50	maximum production rate
5	10/90	maximum production rate
6	90/10	maximum production rate

The process contains 41 measured and 12 manipulated variables [41]. The TE process simulation code is available in Fortran and Matlab, and detailed description of the process and simulation can be found in [42], [41], and [58]. The data and codes can also be downloaded from [59]. Table 4.11 presents manipulated and measured variables of the TE process.

Table 4.11: Manipulated and Measured Variables of the TE Proces[3]

Var.	description	Var.	Description
1	A feed Stream 1 (MEAS)	27	Reactor feed component E (MEAS)
2	D feed Stream 2 (MEAS)	28	Reactor feed component F (MEAS)
3	E feed Stream 3 (MEAS)	29	Purge component A (MEAS)
4	Total feed Stream 4 (MEAS)	30	Purge component B (MEAS)
5	Recycle flow (MEAS)	31	Purge component C (MEAS)
6	Reactor feed rate (MEAS)	32	Purge component D (MEAS)
7	Reactor pressure (MEAS)	33	Purge component E (MEAS)
8	Reactor level (MEAS)	34	Purge component F (MEAS)
9	Reactor temperature (MEAS)	35	Purge component G (MEAS)
10	Purge rate (MEAS)	36	Purge component H (MEAS)
11	Separator temperature (MEAS)	37	Product component D (MEAS)
12	Separator level (MEAS)	38	Product component E (MEAS)
13	Separator pressure (MEAS)	39	Product component F (MEAS)
14	Separator underflow (MEAS)	40	Product component G (MEAS)
15	Stripper level (MEAS)	41	Product component H (MEAS)
16	Stripper pressure (MEAS)	42	D feed flow Stream 2 (MV)
17	Stripper underflow (MEAS)	43	E feed flow Stream 3 (MV)
18	Stripper temperature (MEAS)	44	A feed flow Stream 1 (MV)
19	Stripper steam flow (MEAS)	45	Total feed flow Stream 4 (MV)
20	Compressor work (MEAS)	46	Compressor recycle valve (MV)
21	Reactor cooling water outlet temp. (MEAS)	47	Purge valve (MV)
22	Separator cooling water outlet temp. (MEAS)	48	Separator product liquid flow (MV)
23	Reactor feed component A (MEAS)	49	Stripper product liquid flow (MV)
24	Reactor feed component B (MEAS)	50	Stripper steam valve (MV)
25	Reactor feed component C (MEAS)	51	Reactor cooling water flow (MV)
26	Reactor feed component D (MEAS)	52	Condenser cooling water flow (MV)

Training and testing data contain 480×52 and 960×52 samples respectively, observed every three minutes of simulation and faults occur after 1 hour and 8 hour of simulation respectively. For this process, 21 disturbances caused by different faults have been defined, among which five unknown and 16 known faults. Table 4.12 introduces different faults defined for the TE process[41].

Table 4.12: Faults Defined in the TE Process[57]

fault ID	description	type
DV1	A/C feed ratio, B composition constant (stream 4)	step
DV2	B composition, A/C ratio constant (stream 4)	step
DV3	D feed temp (stream 2)	step
DV4	reactor cooling water inlet temp	step
DV5	condenser cooling water inlet temp	step
DV6	A feed loss (stream 1)	step
DV7	C header pressure loss-reduced availability (stream 4)	step
DV8	A, B, C feed composition (stream 4)	random variation
DV9	D feed temp (stream 2)	random variation
DV10	C feed temp (stream 4)	random variation
DV11	reactor cooling water inlet temp	random variation
DV12	condenser cooling water inlet temp	random variation
DV13	reaction kinetics	slow drift
DV14	reactor cooling water valve	sticking
DV15	condenser cooling water valve	sticking
DV16-DV20	unknown	
DV21	valve for stream 4 fixed at the steady-state position	constant position

Fault detection scheme for TEP

So far we presented and discussed the application of SVDD on two multivariate processes (simple multi-variable system and 3TS) that are considered as low dimensional processes. Assuming that TEP is a multivariate process with high dimension, in this section we investigate SVDD fault detection on TEP when applied individually and when combined with dimensionality reduction methods, i.e., ICA and PCA. The outcome summarizes our comprehensive study of SVDD method for fault detection purposes.

First, a SVDD structure is designed, using NOC process data without any reduction in the dimensionality of the data. ICA and PCA models are also prepared. Secondly, we combine SVDD with ICA (PCA) by using ICA (PCA) scores (dimensionally reduced features) in normal operating condition as data dimension reduction step and then train SVDD with the low dimension data and finally test combined SVDD-ICA (SVDD-PCA) fault detection capability by TEP testing data for different faults and compare with SVDD.

Parameter adjustment

Parameter values for SVDD, combined SVDD (SVDD-ICA and SVDD-PCA), PCA, and ICA structures are given in Table 4.13. The number of ICs to be retained can be selected from eigenvalue plot of the NOC data shown in Figure 4.21 as described before. We select the same number of PCs selected in [50]. Other values have been cited in the literature for order reduction of Tennessee Eastman process as well. Himes et al. [61] suggested 11 PCs while, in [3], they used 30 PCs in their case study. In [63], they proposed a

Table 4.13: Parameter values for SVDD, ICA, and PCA and combined SVDD methods applied to Tennessee Eastman process for fault detection.

method	parameter	value
SVDD	L	70
	σ	50
SVDD-ICA	L	40
	σ	4
SVDD-PCA	L	40
	σ	6
ICA	# of ICs	7
PCA	# of PCs	9
	$F_\alpha(a, n-a)$	1.8989
	c_α	1.6449

method based on sensitivity of PCA model to faults and suggested different PC numbers for different faults which results in multiple PCA models. SVDD kernel width and split parameter, L, are selected from the graphs of the error surface with respect to parameters variation. The graphs are shown in 4.22 for SVDD and combined SVDD with ICA and PCA. To see if the parameters are selected properly, we examined SVDD fault detection criteria over a range of different values of σ and obtain false alarm rate, missed detection and detection delay curves. As depicted in Figure 4.23, the values between intersection points of false alarm rate curve with missed detection rate and delay curves can be considered as optimum region for parameter selection. Specifically, at $\sigma = 40$ and $\sigma = 60$ are the two

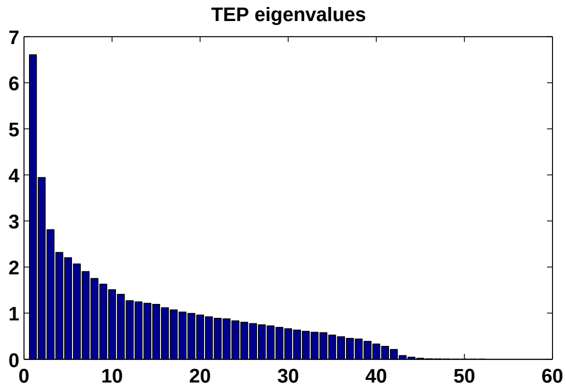

Figure 4.21: NOC data covariance matrix eigenvalues.

point in which lowest error rates of false alarm-missed detection, and false alarm-delay can be achieved, respectively. We selected $\sigma = 50$ in this work. It should be noted that this Figure can only be obtained when training data for different faults are available. In many real world problems, fault data are not available or there is not enough data to represent faults. Therefore, SVDD parameter is selected heuristically by analyzing the training error graph.

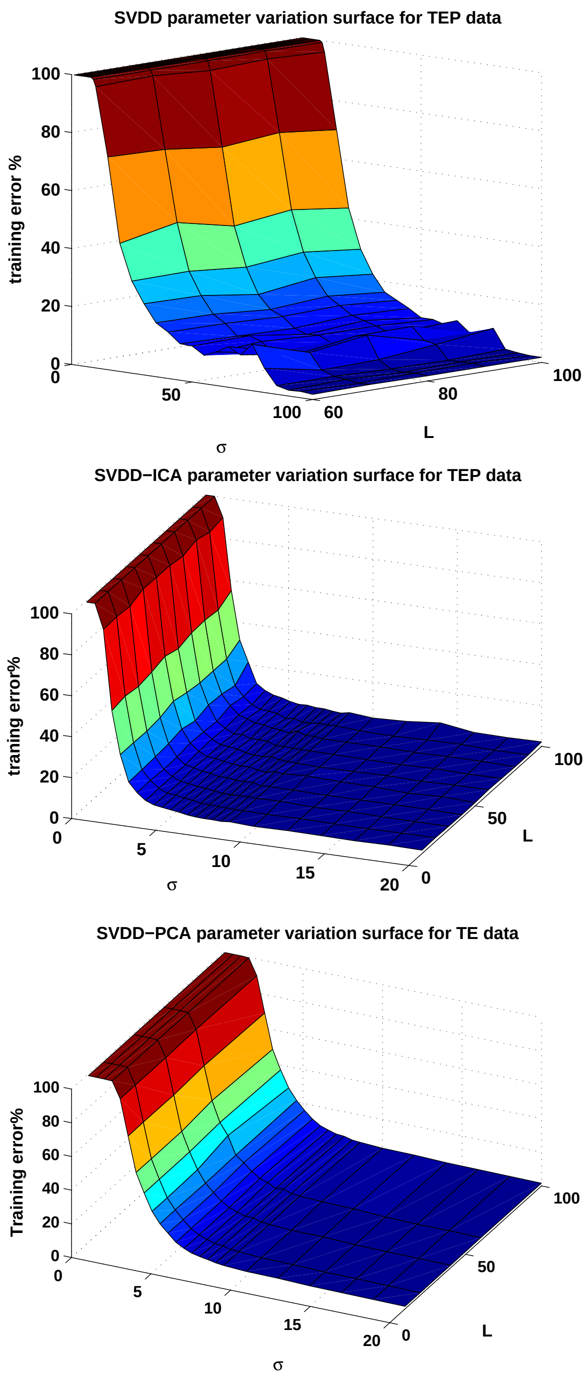

Figure 4.22: Error surface graphs of SVDD and combined SVDD for different values of σ and L, training TEP.

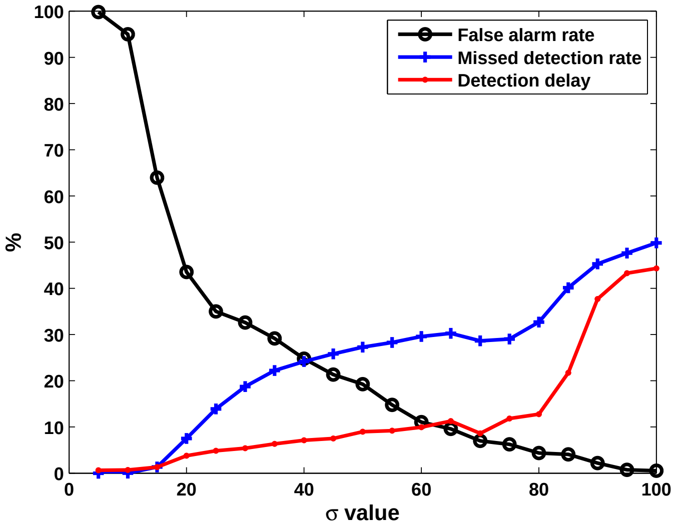

Figure 4.23: Average fault detection performance of SVDD with different values of its parameter, σ, tested on TEP training fault data.

Fault detection results for TEP

In this section, SVDD fault detection algorithm is applied to Tennessee Eastman process data and the results are discussed. The performance of the method is examined by comparing the results with ICA and PCA methods and the combination of SVDD with ICA and PCA. As an example of SVDD monitoring, two graphs are shown in Figure 4.24. The graph on the top is a step fault occurring in condenser cooling water inlet temp and as shown, the process returns to steady state after a while due to control adjustments. The bottom graph is a step fault in component A feed loss (stream 1) and as seen, the process remains above the limit. The threshold is the radius of the sphere in SVDD feature space and the points represent the distance of each data point from the center of the sphere. In all cases, fault inception is at sample number 160 and continues until the end of the experiment. We compare SVDD with ICA and PCA by calculating fault detection performance criteria of each method for all process faults. Figure 4.25 illustrates missed detection rates of the three methods for 21 TEP faults. From this Figure, ICA and SVDD have similar rates for eight faults while SVDD has better rates for faults 3, 5, 9, 10, 15, 21, and ICA performs better for faults 4, 7, 11, 14, 17, 19, and 20. In more details, score based (I^2) comparison of ICA with SVDD shows that SVDD outperforms ICA-I^2, having lower missed detection rates for 11 faults, similar rates for 6 faults and higher rates for only 4 faults. On the other hand, residual based (SPE) comparison reveals that ICA-SPE has better rates for 5 faults while SVDD's rates are lower for 3 fault and the 11 remaining fault rates are close for the two methods(less than 5% difference). From Figure 4.25, it can be seen that PCA performs the same as ICA with small difference in detection rates.

Assessment of detection delays in Figure 4.26 shows that most of the faults are detected earlier by SVDD and ICA-SPE, but ICA-I^2 has higher detection delays. For example, fault 4 is detected with SVDD and SPE after 4 samples, while I^2 detects fault at 62^{th} sample. The situation for fault 15 and 21 are the worst, where ICA-I^2 almost fails to detect these faults. The worst case for SVDD is on fault 14 which is detected after 150 samples. Overall, SVDD detection delay is lower than ICA or PCA on 7 faults; it is higher on 6 faults and similar for the remaining 8 faults.

To investigate the effect of dimension reduction, we combined SVDD with ICA and PCA prior to applying SVDD on the data. In this framework for fault detection, ICA or PCA is used as a data preprocessing stage to reduce data dimension and SVDD is applied on the

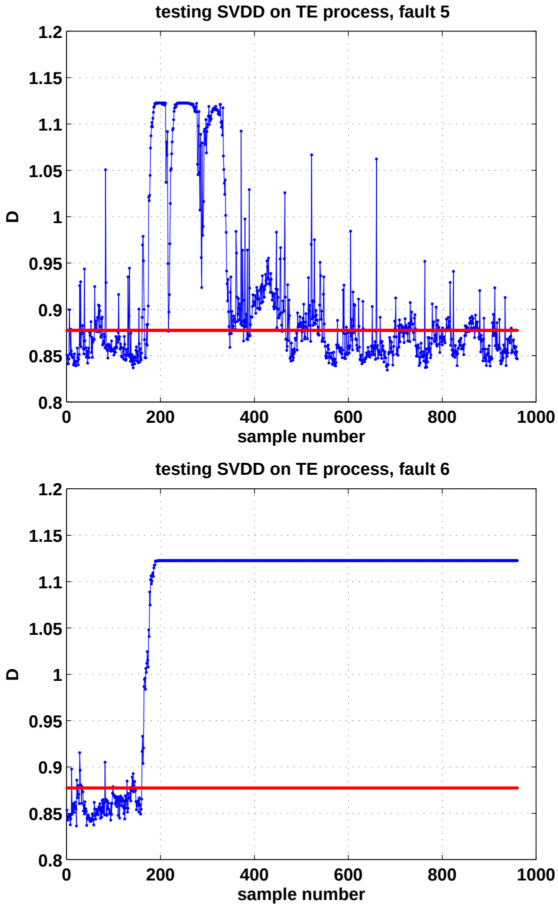

Figure 4.24: Fault detection result for faults 5 (top) and fault 6 (bottom): A step fault in condenser cooling water inlet temperature and step fault in component A feed loss (stream1)

reduced data. Missed detection rates are included in Figure 4.27. Compared to combined SVDD, it is shown that better rates achieved with SVDD for 10 fault, similar rates for 7 faults, and higher rates for only 4 faults. Overall, SVDD performs better than combined SVDD, indicating that dimension reduction deteriorates SVDD performance.

Figure 4.28 shows that the difference between SVDD and combined SVDD detection delay is not significant. As shown in this Figure, combined SVDD is faster than SVDD in detecting faults 14, 16, 17, and 18, but slower in detecting faults 15, 19, 20, and 21, and both have the same detection delays for the rest of the faults.

To obtain an overall description of SVDD performance on TEP data, an averaged missed

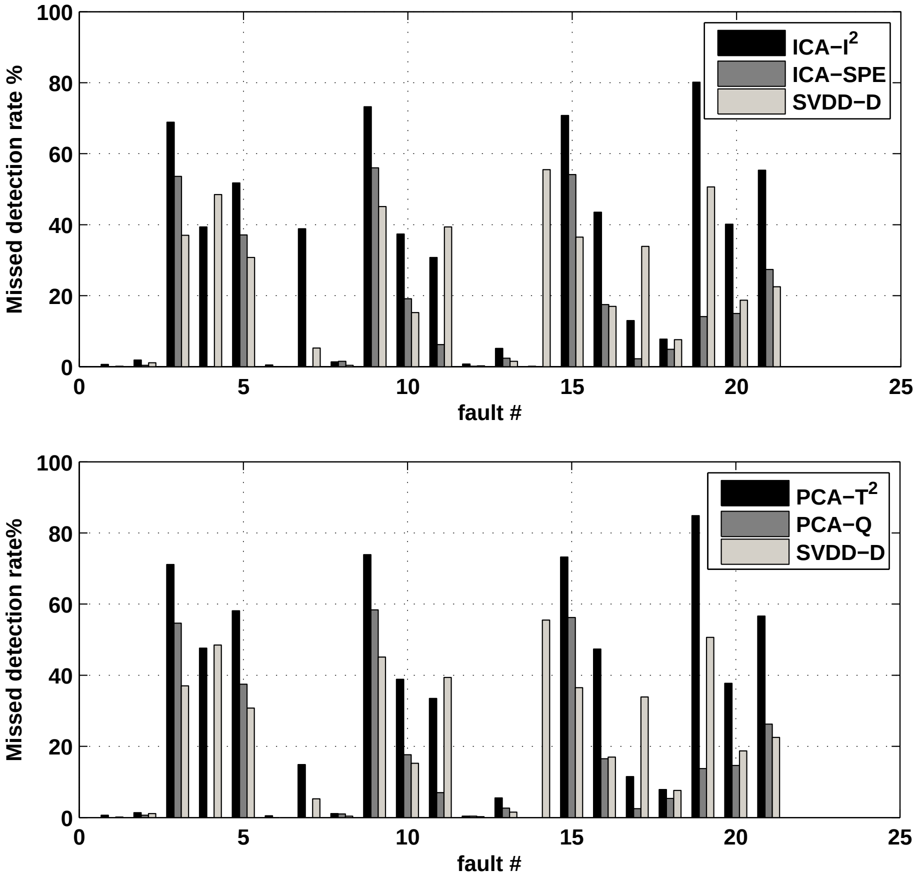

Figure 4.25: Missed detection rates for TEP data, top) SVDD vs. ICA, bottom) SVDD vs. PCA

detection rate and detection delay of all faults is calculated and presented in Figures 4.29-4.30 in bar-plots. For example, each bar in missed detection plot indicates the average percentage for one performance criterion, e.g., T^2 or SPE, etc. Overall, SVDD average detection rates and delays are close to SPE and Q statistics, while SVDD-PCA combination being close to SVDD. The only case in which SVDD has the worst performance is the rate of false alarm as shown in the Figure, where, SVDD has the highest rate (about 10%). False alarm rate can be reduced by re-tuning SVDD parameter (σ), however, there would be a risk of increasing missed detection rate.

In order to investigate the effect of SVDD parameter on performance, we examine its fault detection criteria with different values of σ on TEP testing data. Figure 4.32 illustrates

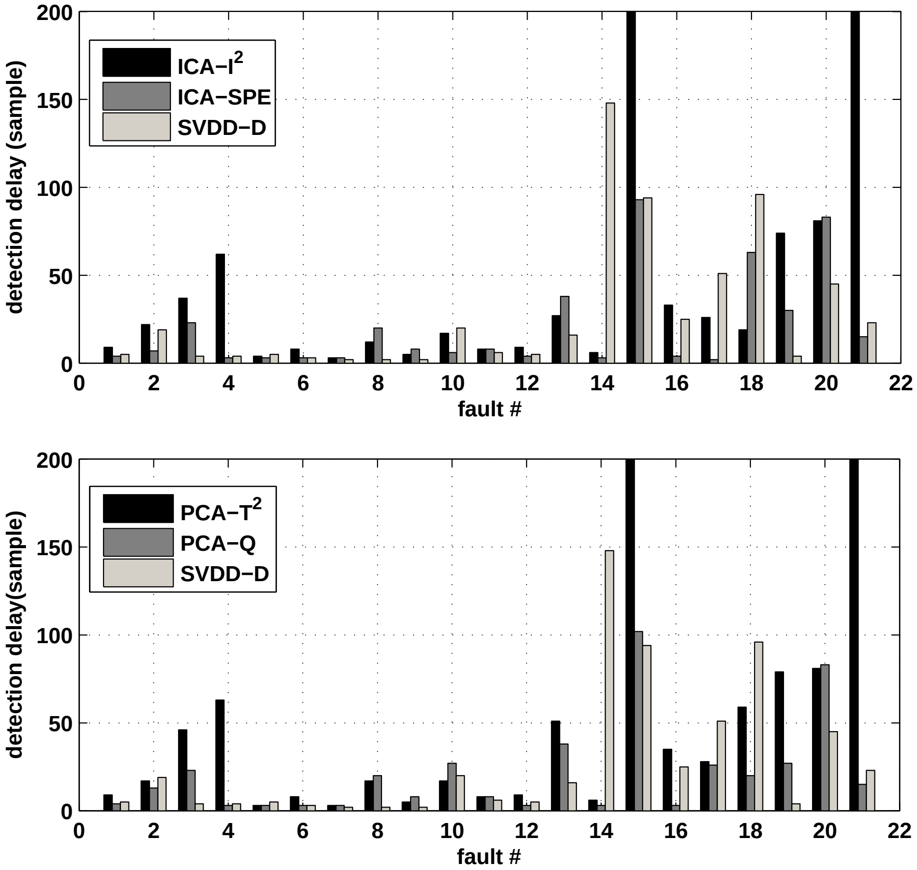

Figure 4.26: Detection delays for TEP data; top) SVDD vs. ICA, bottom) SVDD vs. PCA

SVDD average missed detection rate, false alarm rate, and detection delays for different σ values. The optimum performance appears to be at $\sigma = 35$, which is less than the selected value of 50. Comparing the two points, i.e., $\sigma = 35$ and $\sigma = 50$, false alarm and missed detection rate difference between the two points are less than 5%. Selecting $\sigma = 35$ results in about 5% higher false alarm rate but 5% lower missed detection rates. The delay curve intersects with false alarm curve at $\sigma = 25$. However, selecting either 50 or 25, does not change delay detection rate. Therefore, the parameter selection approach used in tuning SVDD demonstrates effectiveness in determining appropriate value for SVDD parameter.

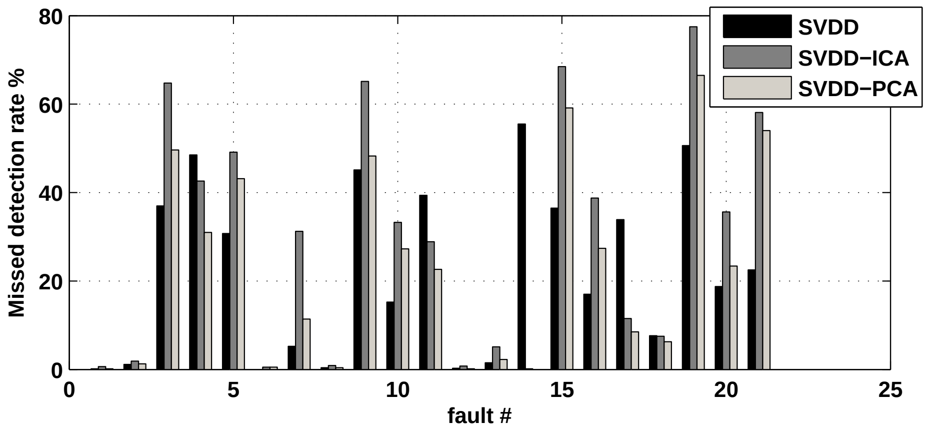

Figure 4.27: Missed detection rates for TEP data, comparing SVDD and its combination with ICA and PCA

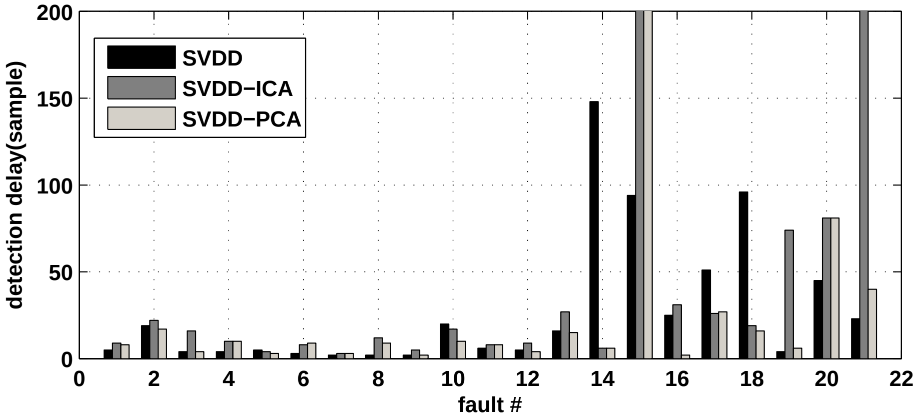

Figure 4.28: Detection delays for TEP data, comparing SVDD and its combination with ICA and PCA

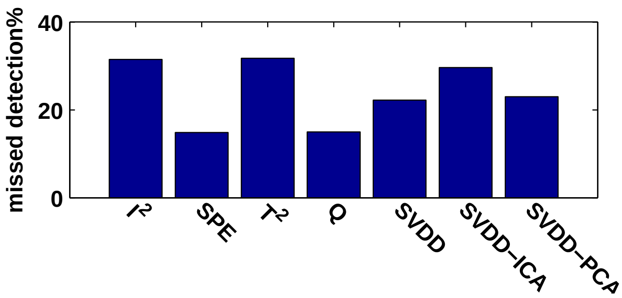

Figure 4.29: Overall average missed detection rates for different methods applied to TEP data.

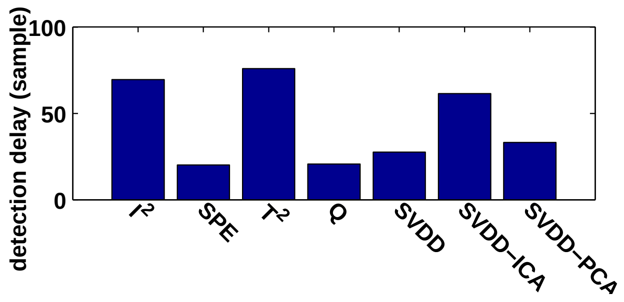

Figure 4.30: Overall average detection delay for different methods applied to TEP data.

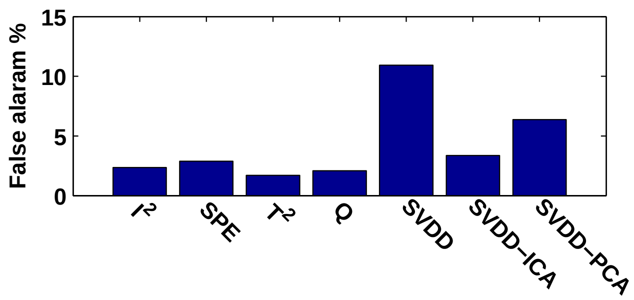

Figure 4.31: False alarm rate for different methods applied to TEP data.

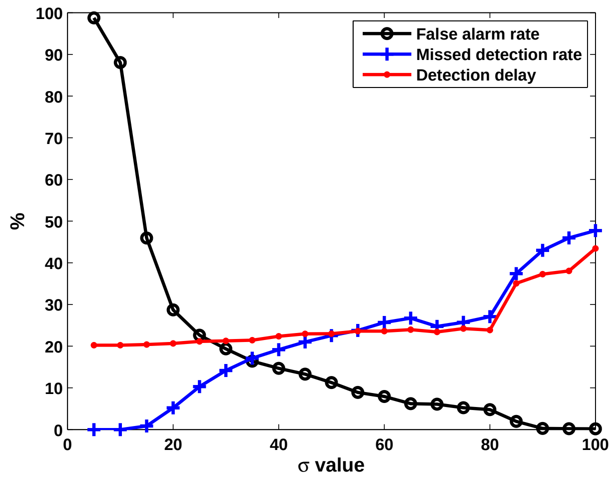

Figure 4.32: Average fault detection criteria for different values of SVDD parameter.

4.2.4 Concluding remark on the results

In conclusion, SVDD has proven considerable capabilities in detecting faults in simple multivariable system, in Three Tank System, and in Tennessee Eastman Process data compared to other standard methods. In all three case studies, SVDD performance was mostly better compared to ICA and PCA in terms of missed detection rates, false alarm rates, and detection delays. However, SVDD performance can be improved if a more accurate method was used for tuning its parameters. The parameters were selected by observing the trend of error in training data and searching the parameter space and using cross validation.

4.3 Chapter Summary

In this chapter we presented the results of three case studies that examined SVDD fault detection system in comparison to PCA and ICA based fault detection as data driven process fault detection methods. The three criteria (Missed detection rate, false alarm rate, and detection delay) mentioned earlier for fault detection were considered in experiments and were used as the tools for evaluating the proposed fault detection system. System parameter adjustment was mainly discussed and a detailed method for selecting appropriate parameter values was presented in this chapter. In 3TS case study, results showed that SVDD is more sensitive to small faults than ICA and PCA which is promising. Up to this point, the problem of fault detection has been discussed comprehensively and has been the center of attention in this work. Now we turn the focus of the research to the next step in process fault diagnosis which is the problem of isolating faults in the process. In the next two chapters, fault classification will be studied and discussed as the proposed approach for fault isolation. Chapter 6 introduces classification methodology and provides definitions, measuring criteria along with classification scheme based on SVDD for fault isolation. As a comparing method, K-Nearest Neighbor (KNN) method is also presented in this chapter.

Chapter 5

Fault classification methodology

Fault detection task is to only indicate the presence of fault in the process but not to identify fault type and location. When the presence of fault is confirmed, fault diagnosis phase is activated to help operators find the root cause of fault and decide for further accommodation. There are different definitions for fault diagnosis that assign several actions to be included as diagnosis. In this work, we consider fault diagnosis as a classification approach that classifies test data and determines the fault class to which the data belongs. In other words, each faulty data is assigned to a class representing the fault. Depending on the classification method, more than one class may be assigned to the data, i.e., providing different membership values of the data point to each class.

We propose a multi-SVDD approach for fault classification by constructing a SVDD classifier for each class of fault. This method is tested in two case studies including an industrial process simulation data and a real experimental system. The results are compared with standard classification methods.

5.1 Introduction

Classification is a very well known term in the field of pattern recognition. Extensive research has been carried out on different classification methods in a wide range of disciplines including medical science [70], [71], image processing and computer vision [74], [73], marketing and business [76], [77], internet and network technology [78], food industry [75], face detection [79], speech recognition[80], [81], and chemical processes and manufacturing[82], [83]. Several algorithms have been proposed for classification such as Neural Networks, K-nearest neighbor classification, Bayesian classifiers, decision trees, linear discriminant analysis methods, support vector machines , etc. [77], [84]. Based on their structure, classifiers are categorized in three groups: one-class classifier, binary classifiers and multi-class classifiers. As a one-class classification method, support vector data description has shown considerable capabilities. SVDD constructs an optimal boundary around data belonging to one class by solving an optimization problem which incorporates training data. As a result, outliers or objects that are outside of the class can be found easily. This method can be modified for classifying multi-class cases by constructing SVDD classifiers for each class of data.

5.1.1 Classification performance criteria

The behavior of a classifier is usually analyzed by confusion matrix which provides a quantitative representation of classifier performance. Each element, C_{ij}, in confusion matrix is defined as the total number of objects in class C_i that have been classified as members of class C_j [68]. Therefore, diagonal elements represent the number of correctly classified objects in each class.
The three most known measures used for classifier performance evaluation are *overall classification rate* (also called *accuracy*), *precision*, and *recall* (or *sensitivity*) that can be calculated from confusion matrix. *Overall classification rate* (or alternatively its equivalent term, error rate or misclassification rate which is equal to one minus the overall classification rate) is the most commonly used measure of classifier performance. *Accuracy*, is defined as the ratio of the number of cases that are correctly classified over the total number of cases[86]. *Precision* is known as the ratio of predicted true positive examples to the total number of actual positive examples where positive example is referred to the examples inside the class and negative example is an example outside the class boundary. *Recall* is defined as the ratio of predicted true positives to the total number of examples predicted as positive[85]. In [69], different classification measures have been introduced and discussed which readers are referred to, for further information.

5.2 SVDD classification scheme

Support vector data description classifier is known as a one-class classifier used for novelty detection or anomaly detection by enclosing training data (also called normal or target data) in a hyper-spherical boundary. Therefore, any new data point that lies outside the boundary is classified as outlier or anomaly. When there are several fault classes, we can construct multiple SVDD classifiers, each trained with specific fault class data set. A new data point is classified by finding the distance of the point to the center of each fault class hyper-sphere and comparing it with their radii. The new data point belongs to class which results in minimum distance to the center and it is smaller than the sphere radius. The mathematical formulation of the classification process is illustrated as follows. Suppose we have k sets of m dimensional faulty data, each representing a specific fault denoted by $F_i = \{\mathbf{x}_j | \mathbf{x}_j \in R^m,\ j = 1, ..., N_i\}$, and $i = 1, ...k$. For each set, a SVDD classifier is trained with a hyper-sphere radius of R_i and its center as $\mathbf{a}_i$. The squared distance of a test point to the center of the i^{th} classifier is defined as:

$$D_i^2 = \|\mathbf{x}_{test} - \mathbf{a}_i\|^2.$$

Having all distances, one may assign the test point to the class with the minimum distance as:

$$C_{test} \ \ s.t. \ \ test = arg(\min_i D_i).$$

In order to eliminate the effect of dominant radii on the results we normalize each class distance with its radius which leads to a new measure that we call Normalized Distance or ND. The measure for the i^{th} class is defined as: $ND_i = D_i/R_i$. Suppose there are two spheres, and the test data point lies outside spheres at an equal distance from the boundary of each. Then, using absolute distance instead of normalized distance leads to the conclusion that the test point is closer to the sphere with the smaller radius while it is in the same distance from the boundaries. With the normalized distance used in classification, the following cases may occur:

$$\begin{cases} if\ ND_i > 1, & test\ point\ is\ outside\ i^{th}\ sphere \\ if\ ND_i = 1, & test\ point\ is\ on\ the\ boundary \\ if\ ND_i < 1, & test\ point\ is\ inside\ i^{th}\ sphere \end{cases},$$

In this work, it is assumed that for all possible faults in the system, training data is available and no novel or unknown fault occurs. Therefore, fault class of a test point is determined as:

$$Fault\ class = arg \min_{i=1,...,k} ND_i.$$

However, the proposed method can be easily modified to consider novel faults as well as known faults.

5.2.1 Design issues

As with other classification approaches, applying SVDD for fault classification requires availability of descriptive training data pertaining to each fault case. Therefore, obtaining more training data which represent a specific fault would enhance the classification performance. Another issue is the decision making approach in determining the class of test points. There might be test points belonging to more than one class as a result of intersecting classes and needing decision on their class. In addition, the kernel parameter for each SVDD class sphere needs to be determined by designer. In other words, if there are m classes, then m parameters should be adjusted while in KNN, there is only one parameter which is k, the number of nearest neighbor points, to be considered in classification.

5.3 K-Nearest Neighbor method

K-Nearest Neighbor (KNN) is one of the most efficient and simple classification algorithms in machine learning. Nearest neighbor classification method was first introduced by Cover and Hart [87], [91], in which the class of each sample point is determined by its k neighboring points in the training set. The point is assigned to the class with the majority of votes amongst the k neighbor points. Setting the classifier only requires determining parameter k and the distance measure. A set of training data with their labels is also required each time a new point is classified. Therefore, computational complexity increases when the size of training data is large. Several modifications of KNN algorithm have been suggested and applied to different data sets in the field of data mining and machine learning. Many papers can be found on KNN or combination of KNN with other methods for improving data classification. For more information on KNN algorithm and its application the following references would be helpful [88], [89], [90], [93] [92].

K-nearest neighbor method has been studied for fault diagnosis purposes in different applications either individually or in combination with other methods. Mostly, KNN is considered as a benchmark method used for comparison of other suggested methods. In [66], weighted KNN approach was used for fault classification of roller bearings which enabled identifying fault class. Choi et al. in [67] compared and combined KNN classification results with other classifiers to obtain better rates for fault diagnosis in automotive systems. The performance of KNN classifier is also compared with SVM and random forest classifiers with respect to their fault diagnosis capabilities in [65] when applied to induction motor.

To the best of our knowledge, very few papers have considered comparing SVDD and KNN specifically for fault diagnosis purposes. Only in [16], extended SVDD classifier is compared with KNN, SVM and ANN (Artificial neural network) classifiers for roller bearing fault detection and classification. However, there has been research work on SVDD which studied its performance and compared it with KNN in areas other than fault diagnosis [64], [46].

5.3.1 K-Nearest Neighbor Classification

In K-nearest neighbor classification, the class of each sample point is determined by its k neighboring points in the training set. The point is assigned to the class with the majority of votes for class label amongst the K-neighbor points. The classifier is defined by its parameters. Setting parameter k depends on the data and affects the performance of the classifier. Parameter k must be large enough to reduce misclassification rate and must be small enough so that the sample test point would be close to the neighboring points which results in better estimation of the point's class [87]. A common approach for finding parameter k is cross validation over a range of predefined values for the parameter.

5.4 SVDD classification pros and cons

The advantage of using SVDD is that a membership value can be obtained for a test data point based on its distance to each class sphere and it can be used for further analysis. In addition, SVDD has the flexibility to be used as a supervised or semi-supervised classification method. In other words, any novel fault can be classified depending on the designer intent on forcing the classification method to distinguish between known fault classes and a new emerging fault. The designer can decide whether to classify a test point as one of the predetermined faults or, if the point lies outside all known fault class spheres, then a novel fault be declared. This flexibility can not be achieved by KNN classifier which is a supervised method in which any new data is classified as one of the predefined fault classes. Another advantage of using SVDD classifier is that it only requires training fault class spheres once and the information of each class sphere can be stored and used for classification. Therefore, the class of any new test point is determined by calculating the distance to the center of the spheres. In KNN method, although the algorithm is simple, the size of the training data highly affects calculation time because it needs calculation of the distances between new test point and all training data points each time a new data is tested. On the other hand, implementing KNN algorithm only requires determining one parameter, k, while for multi-class SVDD, the number of parameters to be adjusted is equal to number of classes defined for classification. In the next chapter we examine and compare SVDD and KNN and discuss their performance and computation time.

5.5 Chapter Summary

In this chapter, fault classification based on SVDD is addressed and a fault isolation scheme is proposed. Confusion matrix is introduced as a standard tool for analyzing classifier performance. Three criteria for assessing the performance of classifier are provided. K-Nearest Neighbor classifier is also presented as a method to be used in comparison with SVDD. Finally, design issues, advantage and disadvantages of SVDD classifier comparing to KNN classifier are discussed. The implementation and experiment results of the SVDD classifier is presented in the next chapter. Two case studies are considered and different faults are applied to the systems. Then, the isolation ability of the SVDD classifier in comparison to other classifiers is examined.

Chapter 6

Implementing SVDD classifier for fault diagnosis

Fault diagnosis includes identification and isolation of faults in the process. However, it can be viewed as a classification task which separates faults from normal condition or other faults. With this view in mind, SVDD one-class classifier is adapted for process fault diagnosis and it is implemented as a multi-class classifier in two case studies: Implementation on Tennessee Eastman process, TEP, and three tank real laboratory system, 3TS. KNN classifier is also applied to the mentioned systems for comparison. In this chapter, classification results are presented and compared for the two methods based on their performance measures introduced earlier in previous chapter. In TEP case study, the results are also compared with several methods studied in other research work, including support vector machines, Fisher's discriminant analysis, naive Bayesian networks, etc. This chapter can be viewed as two main parts that present the two case studies.

6.1 First case study: Fault classification of TEP

In this section the performance of SVDD in fault classification is investigated and compared to other classifiers by applying it to TE process data. SVDD capability in classifying multiple classes with overlap is studied on three classes of faulty data generated from TEP simulator. These faults correspond to faults 4, 9, and 11 as described in table 6.1. Faults 4 and 11 are associated with reactor cooling water inlet temperature. Fault 4 is a step change fault while fault 11 is a random variation in the process variable. Fault 9 is associated with random variation in D feed temperature. These three faults are selected because they are good representation of overlapping data and were studied in different research work which allows consistent comparison of the methods [94], [96].

As a case study, this classification problem has been studied in [55] to compare Fisher's discriminant analysis, which is a linear method, with support vector machines, which is a nonlinear classifier. Specifically, they compare FDA, SVM, and Proximal SVM (PSVM), which is a modified case of SVM, and show that PSVM improves classification rates. Kulkarni et al. in [94] proposed another modification of SVM for TEP fault classification by incorporating knowledge into the algorithm by taking advantage of some data translation. They obtained the same rate as found in [55]. Later, Verron et al. [5], [96] proposed Bayesian Networks and discriminant analysis to the problem while modifying their method by applying mutual information theory to enhance the results.

In the following, we compare SVDD classification performance with KNN, and also with methods proposed and tested in other works on TEP data as mentioned above. The same set of data used in other work is selected here in order to establish a fair comparison. Training

Table 6.1: TEP fault data sets for multiple classification[55].

Fault #	fault spec.	# of training data	# of testing data
Fault 4	step change in the reactor cooling water inlet temperature	480	800
Fault 9	random variation in D feed temperature	480	800
Fault 11	random variation in the reactor cooling water inlet temperature	480	800

and testing data sets were obtained for each fault case. The training data were used to train three SVDD classifiers and testing data were kept for validation. In each case, 480 samples (observations) for training and 800 samples for testing were obtained with each observation containing 52 variables. It is known that only variable 9 (reactor temperature) and variable 51 (reactor cooling water valve position) are the effective variables and the rest of the 50 variables do not indicate any fault in the system [55]. Therefore, only these two variables

are considered for classification. To be consistent with other research work, the data sets were downloaded from `http://www.brahms.scs.uiuc.edu`.
Fault diagnosis initiates by training a classifier for each fault case and obtaining SVDD structure as described in previous chapters, using cross validation for parameter adjustment. When all parameter values for each classifiers are obtained, the information is stored for diagnosis. Each new observation sample is tested by calculating its normalized distance to all SVDD classifiers and the minimum distance defines the class of the sample point. The results of classification are provided in confusion matrix from which *precision*, *recall* and *accuracy* measures are calculated.

Table 6.2 includes confusion matrix with calculated measures explained in the following sentences for SVDD classifier. The last row includes *precisions* with respect to each fault class. *precision* is calculated as:

$$precision(j) = \frac{C_{jj}}{\sum_{i=1}^{n} C_{ij}}, \quad j = 1, ...n.$$

The last column contains *recall* measures for different fault classes which is calculated as:

$$recall(i) = \frac{C_{ii}}{\sum_{j=1}^{n} C_{ij}}, \quad i = 1, ...n,$$

where C_{ij} is the ij^{th} element of the confusion matrix and n is the number of fault classes. The overall classification rate or *accuracy* of the classifier is calculated as the sum of all correctly classified samples (diagonal elements), over the total number of samples as:

$$accuracy = \frac{\sum_{i=1}^{n} C_{ii}}{\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij}} \times 100.$$

To better explain table 6.2 , the following example is discussed. Consider the first fault class predictions which correspond to the first row of the confusion matrix (corresponding row of table 6.2). It is seen that SVDD classifier has successfully classified 790 out of 800 samples of the first fault (fault 4) which results in high value of .987 for *recall* measure (the closer to 1 the better). On the other hand, if we look at the first column we can see that 32 samples belong to the class of fault 11, that have been classified as fault 4 incorrectly, which results in .961 for *precision* measure corresponding to the first fault. The overall classification rate is 94.250%, resulting in 5.75% misclassification rate. The lowest values for *recall* and *precision* are found for fault 11 as a result of overlap between this fault and the other two classes. From confusion matrix, it can be seen that fault 4 and 9 are completely separated since none of the samples in each class are classified as the other and all the confusion is between fault 11 and the other two. K-nearest neighbor classifier has been implemented on TEP.

Table 6.2: Classification results when applying SVDD classifiers to TEP for faults 4, 9, and 11. Each fault case contains 800 observations. Rows represent true fault class and columns represent predicted fault class.

		Predicted fault class			
	fault#	1	2	3	*recall*
True	4	790	0	10	0.987
fault	9	0	781	19	0.910
	11	32	77	691	0.959
	precision	0.9611	0.9103	0.9597	*accuracy*=94.250

Using MATLAB KNN classification function, the confusion matrix is obtained as presented in table6.3. From this table, the classification *accuracy* is computed as 93.125% resulting in 6.875% misclassification rate.

Table 6.3: Classification results when applying KNN classifiers to TEP for faults 4, 9, and 11. Each fault case contains 800 observations. Rows represent true fault class and columns represent predicted fault class.

		Predicted fault class			
	fault#	1	2	3	*recall*
True	4	791	0	9	0.988
fault	9	0	785	15	0.981
	11	46	15	659	0.823
	precision	0.9450	0.8920	0.9649	*accuracy*=93.125

6.1.1 Discussion on TEP fault classification

In order to assess the performance of SVDD, classification criteria are compared with those of KNN classifier. From Figure 6.1, it can be seen that SVDD and KNN *recall* and *precision* values are close to each other with SVDD having slightly better *recall* value for fault 11. The overall classification rate of SVDD outperforms KNN being more than 1% better than KNN. Another important parameter in assessing classifiers performance is the computation time. Since SVDD classifier is trained off-line, we only consider testing time for comparison.

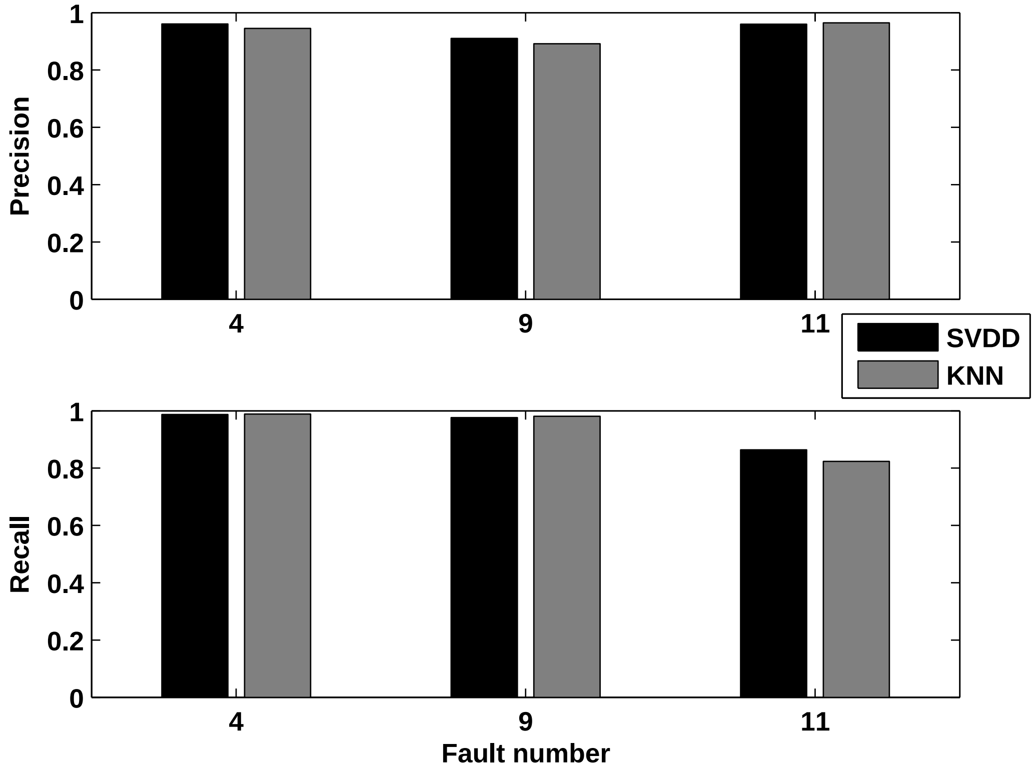

Figure 6.1: Comparing *recall* and *precision* values of SVDD and KNN methods.

Table 6.4 includes classification testing times taken by each method for the set of faults in TEP. From this table, it can be seen that KNN computation time is more than 3 times longer than SVDD. This can be related to the number of support vectors in SVDD structure. The number of support vectors is related to the number of Lagrange multipliers in training SVDD. Appropriately selecting SVDD parameters reduces the number of support vectors which in turn affects SVDD complexity and computation time. However, KNN performs training and testing all at once. On the other hand, the amount of time required for adjusting SVDD parameters depends on the number of classes and the search space for cross validation procedure that must be taken into account when selecting a classifier. If the training time is important then selecting SVDD might not be reasonable or may require modifying methods to accelerate its training time.

As mentioned earlier, we compare the classification results of SVDD and KNN with other

Table 6.4: Computation time for SVDD and KNN

method	time (sec)
SVDD testing	0.251
KNN	0.915

research work proposed in the literature. Table 6.5 summarizes the results of several different methods proposed in literature as well as the results obtained in this work to provide a more comprehensive view on this classification problem and the suggested solutions. In this table, misclassification rates for FDA, SVM, and PSVM are cited from [55], ISVM rate from [94], Naive Bayesian Network (NBN), and Tree Augmented Network (TAN) rates from [5], and linear and quadratic discriminant analysis rates (LDA and QDA) from [96].

Table 6.5: Misclassification rates of different classification methods for TEP test data. Misclassification is defined as $(100 - accuracy)$.

Method	%Misclassification rate
FDA	17
SVM	6.5
PSVM	6.0
ISVM	6.0
NBN	7.8
TAN	5.9
LDA	31.58
QDA	5.87
SVDD	**5.75**
KNN	6.87

As shown in table 6.5, ***SVDD has the best performance amongst the*** 10 ***different classifiers with smallest misclassification rate*** for the TEP data. To the best of our knowledge, no better result has been reported in the literature.

6.2 Second case study: Fault classification of a real system (Three tank system)

We define eight fault case scenarios including sensor fault, actuator fault (fault in pumps), clog, and leakage in the three tank system. For each fault case a set of training and testing experimental data is obtained by running experiment while fault was induced in the system. Each set contains 400 samples with dimension of five (stored in a 400×5 matrix). The results of the classification methods are provided in table 6.6, presenting the confusion matrix with calculated measures explained above. From table 6.6, consider the first fault

Table 6.6: Classification results when applying SVDD classifiers on 3TS system for 8 different faults in the system. Each fault case contains 400 samples. Rows represents true fault class and columns represents predicted fault class.

		Predicted fault class								
	fault#	1	2	3	4	5	6	7	8	*recall*
True lass	1	400	0	0	0	0	0	0	0	1.000
	2	0	400	0	0	0	0	0	0	1.000
	3	0	0	400	0	0	0	0	0	1.000
	4	0	0	0	400	0	0	0	0	1.000
	5	311	0	0	0	10	0	79	0	0.025
	6	0	0	0	0	7	392	1	0	0.980
	7	0	0	0	0	6	1	393	0	0.982
	8	0	0	0	4	0	0	0	396	0.990
	precision	0.562	1.000	1.000	0.990	0.434	0.997	0.830	1.000	*accuracy* =87.218

class predictions which correspond to the first row of the confusion matrix. It is seen that SVDD classifier has successfully classified all 400 samples of the first fault which results in highest value for *recall* measure (*recall* = 1.000). On the other hand, if we look at the first column we can see that there are 311 samples belonging to class 5 that have been classified as class 1, incorrectly, which results in a low value of .562 for *precision* measure corresponding to fault 1 and .434 to fault 5. From table 6.6 we can see that most faults have been classified with high *precision* and *recall* values except fault number 5 with .434 value for *precision* and .025 for *recall*, which stands out amongst other faults which indicates that SVDD classifier confuses fault 1 and 5 and fault 5 degrades the performance of the classifier and highly affects the outcome of the diagnosis process.

Classification results of KNN method are presented in table 6.7. The same as SVDD, KNN

Table 6.7: Classification results when applying KNN classifiers on 3TS system for 8 different faults in the system. Each fault case contains 400 samples. Rows represents true fault class and columns represents predicted fault class.

		Predicted fault class								
	fault#	1	2	3	4	5	6	7	8	*recall*
True class	1	400	0	0	0	0	0	0	0	1.000
	2	0	400	0	0	0	0	0	0	1.000
	3	0	0	383	0	17	0	0	0	0.957
	4	0	0	0	400	0	0	0	0	1.000
	5	0	0	0	0	35	0	365	0	0.087
	6	0	0	0	0	0	400	0	0	1.000
	7	0	0	0	0	1	0	399	0	0.997
	8	0	0	0	1	0	0	0	399	0.997
	precision	1.000	1.000	1.000	0.997	0.660	1.000	0.522	1.000	*accuracy* =88.000

has also problem with classifying samples of class 5, which reduces *precision* values to .660 and .522 corresponding to fault 5 and 7 with *recall* value of .087. From table 6.7, 365 out of 400 samples of fault 5 are classified as class 7 incorrectly and only 35 cases were classified as fault 5 correctly. The *accuracy* of KNN method is close to SVDD (88 compared to 87.218 for SVDD) for the three tank system.

6.2.1 Discussion on 3TS fault classification

Classification criteria of SVDD are compared with KNN as shown in Figure 6.2. It can be seen that SVDD and KNN *recall* values are close to each other and both have the lowest *recall* value for fault number 5. On the other hand, comparing *precision* values shows that KNN performs better than SVDD for faults 1 and 5, while, SVDD *precision* is better for fault 7. For the rest of the five faults, both methods provide the same result.

Table 6.8: Computation time for SVDD and KNN

method	time (sec)
SVDD testing	1.0832
KNN	3.2699

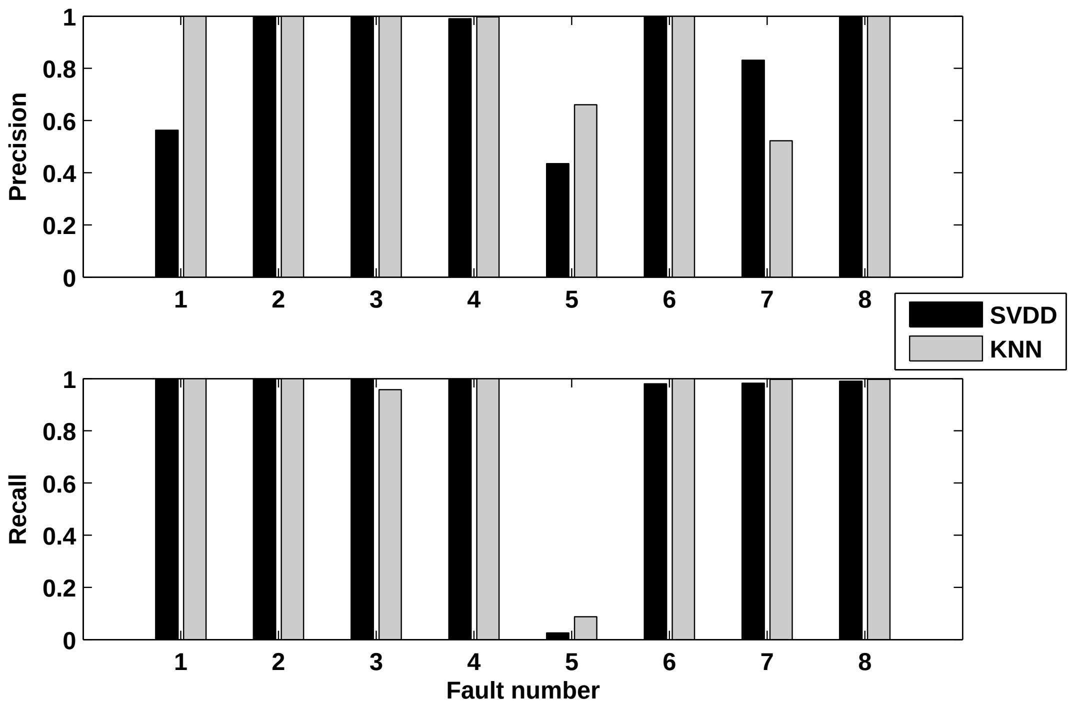

Figure 6.2: Comparing *recall* and *precision* values of SVDD and KNN methods.

Same as TEP, computation time was obtained for the two methods. Table 6.8 includes SVDD and KNN classification times for a set of eight faults of the three tank system. Test data set contains 400 samples for each fault. From this table, it can be seen that KNN computation time is 3 times larger than that of SVDD.

6.3 Chapter Summary

The implementation of the proposed SVDD based classifier for fault isolation was accomplished and it was shown that SVDD has the potential to be used for fault classification as a powerful method in comparison to other classification approaches. To validate the proposed method, two case studies were carried out on Three Tank system and Tennessee Eastman process. The SVDD classifier has been implemented on the two systems to isolate different process faults. For TEP data, there are several research studies with different methods which were considered for comparison in this work. The results showed promising performance for SVDD compared to these methods. The experimental results on a real 3TS setup confirmed the capability of SVDD classifier in fault classification as well. In the next chapter, a comprehensive conclusion of this thesis is provided and future directions for research are highlighted.

Chapter 7

Conclusion

The primary goal of this thesis has been the application and implementation of support vector data description as the main element for fault detection and diagnosis in industrial processes. A complete fault detection and diagnosis framework based on SVDD has been established and tested on different systems and the proficiency of the proposed method has been compared with other standard approaches. The main contribution of this work is that it provides a complete structure based on SVDD for data driven fault detection and isolation (FDI), and demonstrates the capability of SVDD in detecting and isolating faults in processes. A complete package is provided which includes detection and classification. The computational complexity and time problem has also been addressed and enhanced by embedding a fast SVDD algorithm in the FDI structure.

In terms of the objectives of this work the following conclusions can be drawn:

The first objective was the development of a fault detection approach mainly based on SVDD. A complete fault detection scheme was presented in chapter 3 and a fault detection system was designed. Parameter adjustment method was proposed and the fault detection algorithm was modified to increase its computation speed by embedding a fast SVDD into the algorithm. It was shown that the computation time reduces significantly when using modified SVDD.

The second objective was the implementation of the proposed method on three different processes. In chapter 4, SVDD fault detection system was applied to a simple multi-variable process as well as Tennessee Eastman process and a real Three Tank System setup. Its performance was compared to ICA and PCA methods by considering false alarm rate, missed detection rate, and detection delay as comparison criteria. SVDD performance on simple multi-variable system showed comparable results with ICA and PCA. It performed better in detecting some faults and worse for few others. In case of three tank system, based on false alarm rates and missed detection rates, SVDD outperformed ICA and PCA while it showed higher detection delays. Also, the results showed that SVDD is more sensitive to faults with less severity than ICA and PCA for three tank system. To summarize, SVDD (if tuned properly) has shown promising results for Three Tank System fault detection with considerable capabilities, compared to PCA and ICA methods which is one of the contributions of this work. Applying SVDD to TE process also revealed considerable results. It was shown that SVDD, ICA and PCA detected most of the faults similarly and only differed in detecting few faults. Combination of SVDD with ICA and PCA was also tested for fault detection. It appeared that SVDD performs better than the combined system and data reduction deteriorated the results. However, in high dimensional processes, data reduction is inevitable since high dimensionality increases computation complexity.

The third objective of this work was accomplished by proposing a fault classification structure completely based on SVDD classifiers as presented in chapter 5. In this framework, multiple SVDD one-class classifiers were used as classifying units for each fault. Fault classification was achieved by tuning a SVDD classifier for each fault class data and finding a boundary for each fault class. A measure called normalized distance was introduced and used for decision making. As an advantage, employing SVDD in fault classification provides the flexibility in decision making procedure and gives the designer the choice of having a supervised or semi-supervised fault classifier depending on the availability of training data.

As the next objective, the proposed fault diagnosis system was applied to 3TS and to TEP and was compared with KNN as a standard method. In case of TEP, the results were compared with nine other classification methods provided in the literature including linear and nonlinear methods. SVDD outperformed all nine methods, giving the lowest misclassification rates. Comparing SVDD with KNN showed better performance for SVDD with close misclassification rates and much better computation time. In case of 3TS, SVDD and KNN fault classification results were also close to each other, with KNN having slightly better misclassification rate. However, SVDD classification time was three times less than KNN which significantly outperforms KNN.

7.1 General concluding remarks and future work

With increasing complexity in processes and data structures, the demand for more advanced monitoring methods increases as well. In this respect, linear approaches have matured and no longer can satisfy the needs. As a result, nonlinear methods have come into the center of attention in process monitoring and diagnosis. As a nonlinear method, support vector data description has shown potential capabilities in this area which is recommended as a data-driven approach for process fault detection and diagnosis. The focus of this work has been investigating these monitoring and diagnosis capabilities by examining SVDD on different processes and it has shown comparable performance.

It has been noticed that the proficiency of the method depends on properly selecting the parameters involved in the diagnosis process which opens an interesting direction for research. Selecting the kernel parameter has high influence on monitoring results. Cross validation has been used here but still the main burden is on the designer to select appropriate parameter values. Finding an automated method to obtain the optimum parameter values would be a significant change in this area.

Most data-driven methods for process monitoring are trained with normal condition data relevant to a specific operating point. As a result, any change in the operating point might be considered as a fault or unwanted disturbance in the process, even though it is a normal change. Adapting the proposed monitoring method to accommodate normal process changes and process dynamics would be one of the future work which has great potential for research and development.

In this work, the detection of fault in the process and classifying the type of fault has been focused and investigated. For future research, other issues can be taken into account to add more features to the diagnosis system. Along with fault type, finding a way to quantify fault severity and obtaining effective variables related to faults would be very useful as a complementary part of the diagnosis system.

Bibliography

[1] Y. Zhou, “Data driven monitoring based on neural networks and classification trees”, Ph.D. dissertation, Dept. Chem. Eng., Texas A&M Univ., College station, 2004.

[2] L. H. Chiang et al., *Fault detection and diagnosis in industrial systems*, 1st ed, London, Britain, Springer, 2001.

[3] L. Auret, “Process Monitoring and Fault Diagnosis using Random Forests”, PhD dissertation, Dept. Process Eng., Univ. Stellenbosch, South Africa, 2010.

[4] V. Venkatasubramanian et al., “A review of process fault detection and diagnosis Part III: Process history based methods”, *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 327-346, 2003.

[5] S. Verron et al., “Fault Diagnosis with Bayesian Networks: Application to the Tennessee Eastnan Process”, *IEEE Int. Conf. Information Technology*, Mumbai, India, 2007, pp. 98-103.

[6] I. Yelamosa et al., “Performance assessment of a novel fault diagnosis system based on support vector machines”, *Comput. Chem. Eng.*, vol. 33, no. 1, pp. 244-255, Jan., 2009.

[7] S. Mahadevan and S. L. Shah, “Fault detection and diagnosis in process data using one-class support vector machines”, *J. Process Control*, vol. 19, no. 10, pp. 16271639, Dec., 2009.

[8] J. Liang, “Multivariate Statistical Process Monitoring Using Kernel Density Estimation”, *Dev. Chem. Eng. Mineral Process*, vol. 13, no. 1, pp. 185-192, 2005.

[9] Y. Zhao and J. Sun, “Recursive reduced least squares support vector regression”, *Pattern Recognition*, vol. 42, no. 5, pp. 837-842, May, 2009.

[10] R. Fujimaki, “Anomaly Detection Support Vector Machine and Its Application to Fault Diagnosis”, *8th IEEE Int. Conf. Data Mining*, Pisa, Italy, 2008, pp. 797-802.

[11] A. Widodo and B. Yang, “Support vector machine in machine condition monitoring and fault diagnosis”, *Mech. Sys. Sig. Process.*, vol. 21, no. 6, pp. 2560-2574, Aug.,2007.

[12] A. Ypma et al., “Robust machine fault detection with independent component analysis and support vector data description”, *Neural Networks Signal Process. 9th Proc. IEEE Signal. Process. Soc. Workshop*, Madison, WI, 1999, pp. 67-76.

[13] Z. Ge et al., “Sensor fault identification and isolation for multivariate non-Gaussian processes”, *J. Process Control*, vol. 19, no. 10, pp. 17071715, Dec., 2009.

[14] L. Xie and U. Kruger, “Statistical Processes Monitoring Based on Improved ICA and SVDD”, *Int. Conf. Intell. Computing* , Kunming, China, Aug., 2006, pp. 1247-56.

[15] X. Liu et al., “Statistical-Based Monitoring of Multivariate Non-Gaussian Systems”, *AIChE J.*, vol. 54, No. 9, pp. 2379-2391, Sept., 2008.

[16] T. Mu and A. K. Nandi, “Multiclass Classification Based on Extended Support Vector Data Description”, *IEEE Trans. Syst. Man Cybern. Part B: Cybernetics*, vol. 39, No. 5, pp. 1206-1216, Oct, 2009.

[17] H. W. Cho, “Data description and noise filtering based detection with its application and performance comparison”, *Expert Syst. Applicat.*, vol. 36, no. 1, pp.434-441, Jan., 2009.

[18] D.M.J.Tax and R.P.W.Duin, “Support vector data description”, *Mach. Learning*, vol. 54, no. 1, pp.45-66, Jan., 2004.

[19] G. Huang et al., “Two-class support vector data description, *Pattern Recognition*, vol. 44, no. 2, pp. 320329, Feb., 2011.

[20] H.G. Bu et al., “Fabric defect detection based on multiple fractal features and support vector data description”, *Eng. Applicat. Artificial Intell.*, vol. 22, no. 2, pp. 224235, March, 2009.

[21] A. Tavakkoli et al., “Efficient Background Modeling through Incremental Support Vector Data Description”, *19th Int. Conf. Pattern Recognition ICPR*, Tampa, FL, 2008, pp. 1-4.

[22] W. Sakla et al., “An SVDD-Based Algorithm for Target Detection in Hyperspectral Imagery”, *IEEE Trans. Geosci. Remote Sens. lett.*, vol. 8, no. 2, March, 2011.

[23] D. Wang et al., “Support vector data description for fusion of multiple health indicators for enhancing gearbox fault diagnosis and prognosis”, *Meas. Sci. Technol.*, vol. 22, no. 2, pp. 1-13, Feb., 2011.

[24] Y. Zhou et al., “Research on SVDD Applied in Speaker Verification”, *Int. J. Digital Content Technol. and its Applicat.*, vol. 4, no. 5, Aug., 2010, doi: [10.4156/jdcta.vol4.issue5.10].

[25] H. Luo et al., “A SVDD approach of fuzzy classification for analog circuit fault diagnosis with FWT as preprocessor”, *Expert Syst. Applicat.*, vol. 38, no. 8, pp. 10554-10561, , Aug, 2011.

[26] J. F. MacGregor and T. Kourti, “Statistical process control of multivariate processes”, *Control Eng. Practice*, vol. 3, no. 3, pp. 403-414, March, 1995.

[27] J. V. Kresta et al., “Multivariate statistical monitoring of process operating performance”, *Canadian J. Chem. Eng.*, vol. 69, no. 1, pp3547, Feb., 1991.

[28] M. Tamura and S. Tsujita, “A study on the number of principal components and sensitivity of fault detection using PCA”, *Comput. Chem. Eng.*, vol. 31, no. 9, pp. 10351046, Sept., 2007

[29] T. Kourti, “Process analysis and abnormal situation detection: From theory to practice”, *IEEE Control Syst. Mag.*, vol. 22, no. 5, pp. 1025, Oct, 2002.

[30] J. F. MacGregor et al., “Process monitoring and diagnosis by mutiblock PLS methods”, *AIChE J.*, vol. 40, no. 5, pp. 826-838, May, 1994.

[31] R. Dunia and S. J. Qin, “Joint diagnosis of process and sensor faults using principal component analysis”, *Control Eng. Practice*, vol. 6, no. 4, pp. 457469, April, 1998.

[32] A. Riach and A. Cinar, "Diagnosis of process disturbances by statistical distance and angle measures", *Comput. chem. Eng*, vol. 21, No. 6, Feb., 1997.

[33] E. Jackson, *A User's Guide to Principal Components*, Wiley Series in Probability and Statistics, 2003

[34] J. M. Lee et al., "Statistical process monitoring with independent component analysis", *J. Process Control*, vol. 14, no. 5, pp. 467485, Aug., 2004.

[35] Aapo Hyvrinen, "Survey on Independent Component Analysis", *Neural Computing Surveys*, vol. 2, pp. 94-128, 1999.

[36] Aapo Hyvrinen and Erkki Oja, "Independent Component Analysis:Algorithms and Applications", *Neural Networks*, vol. 13, no. 4-5, pp. 411-430, May-June, 2000.

[37] C. C. Hsu et al., "A novel process monitoring approach with dynamic independent component analysis", *Control Eng. Practice*, vol. 18, no. 3, pp. 242253, March, 2010.

[38] W. Ku et al., "Disturbance detection and isolation by dynamic principal component analysis", *Chemom. Intell. Lab. Syst.*, vol. 30, no. 1, pp. 179196, Nov., 1995.

[39] Q. Chen et al., "The application of principal component analysis and kernel density estimation to enhance process monitoring", *Control Eng. Practice*, vol. 8,no. 5, pp. 531-543, May 2000.

[40] E. B. Martin and A. J. Morrist, "Non-parametric confidence bounds for process performance monitoring charts", *J. Process Control*, vol. 6, no. 6, pp. 349-358, Dec., 1996.

[41] L.H. Chiang et al., "Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis", *Chemom. and Intell. Lab. Syst.*, vol. 50, no. 2, pp. 243-252, March, 2000.

[42] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem", *Comput. Chem. Eng.*, vol. 17, no. 3, ,pp. 245-255, March, 1993.

[43] M. Kano et al., "Evolution of multivariate statistical process control: application of independent component analysis and external analysis", *Comput. Chem. Eng.*, vol. 28, no. 6-7, pp. 11571166, June, 2004.

[44] E. Tafazzoli and M. Saif, "Application of combined support vector machines in process fault diagnosis", *Proc. American Control. Conf.*, St. Louis, MO, 2009, pp. 3429-3433.

[45] C.M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed, Singapore, Springer, 2006.

[46] D. M. J. Tax and R. P. W. Duin, "Support Vector Data Description", *Pattern Recognition Lett.*, vol. 20, no. 11-13, pp. 1191-1199, Nov., 1999.

[47] D. Lee and J. Lee, "Domain described support vector classifier for multi-classification problems", *Pattern Recognition*, vol. 40, no. 1, pp. 4151, Jan., 2007.

[48] J.H. Williams et al., *Condition-based Maintenance and Machine Diagnostics*, Chapman & Hall, London, 1994.

[49] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*, 1st ed, Springer-Verlag, Berlin, 2005.

[50] J. Lee et al., "Statistical monitoring of dynamic processes based on dynamic independent component analysis", *Chem. Eng. Sci.*, vol. 59, no. 14, pp. 29953006, July, 2004.

[51] C. K. Yoo et al., "On-line monitoring of batch processes using multiway independent component analysis", *Chemometrics Intell. Lab. Syst.*, vol. 71, no. 2, pp. 151-163, May, 2004.

[52] D.M.J. Tax. (May 7, 2010). Data description toolbox dd tools 1.7.5 A Matlab toolbox for data description, outlier and novelty detection[Online].Available: `http://homepage.tudelft.nl/n9d04/dd_tools.html`.

[53] R.P.W. Duin. et al.(2007). PRTools 4.1. A Matlab Toolbox for Pattern Recognition.[Online].Available:`www.prtools.org`.

[54] J. Luo et al., "A Fast SVDD Algorithm Based on Decomposition and Combination for Fault Detection", *8th IEEE Int. Conf. Control Automation*, Xiamen, China, 2010, pp. 1924-1928.

[55] L.H. Chiang et al., "Fault diagnosis based on Fisher discriminant analysis and support vector machines", *Comput. Chem. Eng.*, vol. 28, no. 8, pp. 1389-1401, July, 2004.

[56] *DTS200 labratory setup Three Tank System manual*, AMIRA, 2002.

[57] G. Lee et al., "Multiple-Fault Diagnosis of the Tennessee Eastman Process Based on System Decomposition and Dynamic PLS", *Ind. Eng. Chem. Res.*, vol. 43, no. 25, pp. 8037-8048, Oct., 2004.

[58] N. L. Ricker.(2007, May 15). *Tennessee Eastman challenge archive.* [Online].Available: `http://depts.washington.edu/control/LARRY/TE/download.html`.

[59] TE process data. (accessed Nov. 2008).[Online]. Available:`http://brahms.scs.uiuc.edu`.

[60] E. Russell et al., *Data-driven methods for fault detection and diagnosis in chemical processes*, 1st ed, London, Springer-Verlog, 2000.

[61] D. M. Himes et al., "Determination of the Number of Principal Components for Disturbance Detection and Isolation", *Proc. American Control Conf.*, Baltimore, Mariland, 1994, vol. 2, pp. 1279-1283.

[62] "Selection of the Number of Principal Components: The Variance of the Reconstruction Error Criterion with a Comparison to Other Methods", *Ind. Eng. Chem. Res.*, vol. 38, no. 11, pp.4389-4401, Sept., 1999.

[63] L. Yuan and T. Xiao-Chu, "Improved Performance of Fault Detection Based on Selection of the Optimal Number of Principal Components", *Acts Automatica Sinica*, vol. 35, no. 12, Dec 2009.

[64] F. Ratle et al., "A comparison of one-class classifiers for novelty detection in forensic case data", *Proc. 8th int. conf. intell. data eng. automated learning*, Birmingham, UK, Dec., 2007.

[65] G. Niu et al., "A Comparison of Classifier Performance for Fault Diagnosis of Induction Motor using Multi-type Signals", *STRUCTURAL HEALTH MONITORING, SAGE*, vol. 6, no. 3, pp. 215-230, 2007.

[66] Y. Lei et al., "A Combination of WKNN to Fault Diagnosis of Rolling Element Bearings", *J. Vib. Acoust.*, vol. 131, no. 6, Dec., 2009.

[67] K. Choi et al., “Novel Classifier Fusion Approaches for Fault Diagnosis in Automotive Systems”, *IEEE Trans. Instrum. Meas.*, vol. 58, no. 3, March, 2009.

[68] C. O. A. Freitas et al., “Confusion Matrix Disagreement for Multiple Classifiers”, *lecture notes comput. sci.*, no. 4756, pp. 387396, 2007.

[69] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks”, *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427437, July, 2009.

[70] J. Levman et al., “Classification of Dynamic Contrast-Enhanced Magnetic Resonance Breast Lesions by Support Vector Machines”, *IEEE Trans. Med. Imag.*, vol. 27, no. 5, May, 2008.

[71] J. Ye et al., “Using Uncorrelated Discriminant Analysis for Tissue Classification with Gene Expression Data”, *IEEE/ACM Trans. computational biology bioinformatic*, vol. 1, no. 4, Oct.-Dec., 2004.

[72] C. K. Loo and M.V.C. Rao, “Accurate and Reliable Diagnosis and Classification Using Probabilistic Ensemble Simplified Fuzzy ARTMAP”, *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, Nov., 2005.

[73] Y. Zhong et al., “A Supervised Artificial Immune Classifier for Remote-Sensing Imagery”, *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, Dec., 2007.

[74] M. E. Farmer and A. K. Jain, “A Wrapper-Based Approach to Image Segmentation and Classification”, *IEEE Trans. Image Process.*, vol. 14, no. 12, Dec., 2005.

[75] K. M. Lee et al., “Effects of Classification Methods on Color-Based Feature DetectionWith Food Processing Applications”, *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, Jan., 2007.

[76] D. Montgomery et al., “Comparison of some AI and statistical classification methods for a marketing case”, *European J. Operational Research*, vol. 103, no. 2, pp. 312-325, Dec., 1997.

[77] M. Y. Kiang, “A comparative assessment of classification methods”, *Decision Support Syst.*, vol. 35, no. 4, pp. 441454, July, 2003.

[78] D. H. Shih et al., "Classification methods in the detection of new malicious emails",*Info. Sci.*, vol. 172, no. 1-2, pp. 241-261, June, 2005.

[79] E. M. kinen and R. Raisamo, "Evaluation of Gender Classification Methods with Automatically Detected and Aligned Faces", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, March, 2008.

[80] A. Ganapathiraju et al., "Applications of Support Vector Machines to Speech Recognition", *IEEE Trans. Signal Process.*, vol. 52, no. 8, Aug., 2004.

[81] C. H. You et al., "GMM-SVM Kernel With a Bhattacharyya-Based Distance for Speaker Recognition", *IEEE Trans. Audio Speech Language Process.*, vol. 18, no. 6, Aug., 2010.

[82] Z. Yi et al., "One-class classifier based on SBT for analog circuit fault diagnosis", *Measurement*, vol. 41, no. 4, pp. 371380, May, 2008.

[83] L. L. Simon. and K. Hungerbuhler, "Industrial batch dryer data mining using intelligent pattern classifiers: Neural network, neuro-fuzzy and Takagi Sugeno fuzzy models", *Chem. Eng. J.*, vol. 157, no. 2-3, pp. 568578, March, 2010.

[84] C. W. Hsu and C. J. Lin, "A Comparison of Methods for Multiclass Support Vector Machines", *IEEE Trans. Neural Netw.*, vol. 13, no. 2, March, 2002.

[85] C. Andorf et al., "Exploring inconsistencies in genome-wide protein function annotations: a machine learning approach", in BMC Bioinformatics, 2007 Andorf et al. licensee BioMed Central Ltd. doi:10.1186/1471-2105-8-284.

[86] W. Yan et al., "Classifier Performance Measures in Multi-Fault Diagnosis for Aircraft Engines", *SPIE Proc., Component Syst. Diagnostics Prognostics Health Manage.*, 2002, vol. 4733, pp.88-97.

[87] Y. Song et al., *IKNN: Informative K-Nearest Neighbor Pattern Classification*, PKDD 2007, Berlin Heidelberg, , Springer-Verlag, 2007.

[88] C. Domeniconi et al., "Locally adaptive metric nearest-neighbor classification", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1281-1285, Sept., 2002.

[89] Q. He and J. Wang,"Principal component based k-nearest-neighbor rule for semiconductor process fault detection", *IEEE Trans. Semicond. Manuf.*, vol. 20, no. 4, pp. 345-354, June, 2008.

[90] Y. Pingpeng et al., "MSVM-kNN: combining SVM and k-NN for multi-class text classification", *IEEE Int. Workshop Semantic Computing Syst.*, pp.133-140, July, 2008.

[91] T. Cover and P.Hart, "Nearest neighbor pattern classification", *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21-27, Jan., 1967.

[92] V. Athitsos and S. Sclaroff, "Boosting nearest neighbor classifiers for multiclass recognition", *Proc. IEEE Comput. Society Conf. Comput. Vision Pattern Recognition*, San Diego, CA, 2005, pp. 45-45.

[93] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607-616, June, 1996.

[94] A. Kulkarni et al., "Knowledge incorporated support vector machines to detect faults in Tennessee Eastman Process", *Comput. Chem. Eng.*, vol. 29, no. 10, pp. 21282133, Sept., 2005.

[95] C. Sumana C. et al., "Dynamic Kernel Scatter-difference-based Discriminant Analysis for Diagnosis of Tennessee Eastman Process", *American Control Conf.*, St. Louis, MO, June, 2009, pp. 3417-3422 .

[96] S. Verron et al., "Fault detection and identification with a new feature selection based on mutual information", *J. Process Control*, vol. 18, no. 5, pp. 479-490, June, 2008.