Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Distribution independent threshold setting based on one-class support vector machine

Chris Louen * **Steven X. Ding** *

* *Institute for Automatic Control and Complex Systems, University of Duisburg-Essen, Bismarckstr. 81 BB, 47057 Duisburg, Germany. (e-mail: chris.louen@uni-due.de).*

Abstract: In this paper, threshold setting issues for data-driven fault detection are addressed. It is state of the art that multivariate analysis based threshold setting schemes are widely applied, which generally require detailed knowledge about the distribution of the process data. The often used Hotelling's T^2, SPE threshold setting is based on the assumption of Gaussian distributed process data. In industrial applications, the distribution of data sets is often unknown or non-Gaussian. Alternatively, the fault detection is formulated as classification or outlier detection (one-class) problem which can be solved e.g. by means of machine learning algorithms. The classifier parameter choice is normally done by expert knowledge or using iterative approaches like cross validation. Such a procedure has considerable influence on the fault detection performance. The availability of training and evaluation data collected under faulty conditions is mostly very limited or time and cost consuming and thus often problematic. This paper presents an iterative threshold setting algorithm, which only uses fault-free data for parameter optimization. For this purpose, a one-class support vector machine which is restricted to convex data sets (including non-Gaussian) is used. The effectiveness of the proposed threshold setting scheme is assessed based on false alarm rate, fault detection rate and randomized algorithm evaluation. Additionally, random uniform distributed uncertainties (scaling and rotation) and offset faults (inside an ellipse) are taken into account. Finally, a comparison study with principal component analysis and Hotelling's T^2, SPE threshold setting schemes is demonstrated.

Keywords: fault detection, one-class, support vector machine, convexity, parameter optimization, randomized evaluation, uncertainties, distribution independent, threshold setting

1. INTRODUCTION

Data-driven fault detection (FD) is an important field in all application areas, where an analytical model is too time and cost consuming e.g. large scale systems (Yin et al., 2012). In the beginning of data-driven FD, multivariate analysis like principal component analysis (PCA) using Hotelling's T^2 and SPE (squared prediction error) threshold setting attracted the main attention in practice (Yin et al., 2012; Ding, 2014). On the other hand, these methods deliver the optimal FD performance only on strict underlying assumptions. In case of PCA with T^2, SPE threshold a Gaussian distribution of the process data is assumed, which does not hold in many applications.

The recent trend of digitalization results in wide application of machine learning algorithms in many engineering areas. FD can be interpreted as a classification problem between a fault-free ($c = 1$) and a faulty ($c = -1$) class, which can be solved by a classifier, e.g. machine learning. The application of a two-class problem requires the availability of training data influenced by faults. In many applications the availability of such data is very limited. Outlier detection (one-class problem) can be applied for FD without training data influenced by faults. However, most classifiers have several parameters which the user have to choose. And the parameter decision has considerable influence on the FD performance. In many cases an iterative approach like cross validation, genetic algorithm and other algorithms are suggested, e.g. for support vector machine (SVM) in Widodo and Yang (2007). They use false alarm rate (FAR) and fault detection rate (FDR) as performance indices. Therefore, a training set with fault-free and faulty data is necessary. As mentioned before, faulty data is not always available. If only fault-free data is available, the parameter optimization criteria is often not clear. Only using FAR can result in a low FDR performance. Alternatively, the parameter choice is based on expert knowledge.

FD based on SVM and one-class SVM (OC-SVM) has been frequently studied in the recent years (Louen et al., 2013; Louen, 2016). OC-SVM trained with normal operation data allows to choose the maximum allowed FAR which is needed in many applications. As mentioned before, the SVM parameters (kernel type and kernel parameters) are normally chosen based on expert knowledge or iterative searching algorithms. If the number of available faulty data for parameter optimization is too low, it is hard to define the optimization criteria.

Nowadays, an increasing attention to design and evaluation of controller (Koenings et al., 2013) and FD (Ding et al., 2019) taking influences of uncertainties into account can be observed. Randomized algorithm (RA) can be used

to describe the uncertainties (Tempo et al., 2012). On the one hand, threshold setting (design) taking uncertainties into account can increase the FD performance (Chen et al., 2018). On the other hand, knowledge about the uncertainties or influences on the data set is necessary. Additionally, the threshold setting can result in a conservative threshold. To evaluate the FD performance for systems with uncertainties, evaluation based on RA can be used (Ding et al., 2019). If the model is unknown, data-driven methods like PCA are available for FD. The uncertainties can be taken into account via the effects on the data set. For PCA like methods random uniform distributed scaling and rotation (360°) of a Gaussian set is suggested in the project DFG DI733-15, which is a worst case approximation. In real applications a rotation uncertainty angle $\boldsymbol{\theta}_{\text{Range}} \ll 360°$ is realistic.

Based on these observations, a novel distribution independent FD threshold setting method is proposed in this work. The main idea is the geometric interpretation, that the smallest sphere which ensures the chosen FAR results in the best possible FDR. OC-SVM is used to learn the optimal separating hypersurface. To achieve an iterative parameter optimization without faulty data, the training data set is assumed to be convex. A suitable mapping method can map non-convex into convex sets. In this paper PCA is used as an example. Based on RA a meaningful performance evaluation taking uncertainties into account is demonstrated. As uncertainties scaling and rotation with limited factor and angle is considered. For random rotation with limited angle range $\theta_{\min} \leq \theta \leq \theta_{\max}$ a solution based on Givens rotation is presented. Since no prior knowledge about the fault is assumed, an offset fault uniformly distributed in an ellipse is applied for evaluation.

The remainder of this paper is organized as follows. In Section II the preliminaries and problem formulation are described. Here, FD is formulated as a classification problem, PCA, OC-SVM, RA and check of convexity are shortly summarized. The novel threshold learning framework based on convex set assumption and OC-SVM is addressed in Section III, including the basic idea, the training algorithm, and usage. Modification of the RA based evaluation is described in Section IV. Section V presents the effectiveness of the proposed threshold setting by using RA based evaluation. Finally, in Section VI the results of this paper are concluded and future work is presented.

2. PRELIMINARIES AND PROBLEM FORMULATION

Let $\mathcal{T}$ be the training set of feature vectors $\boldsymbol{x}_i \in \mathbb{R}^m$ $(i = 1, \ldots, N)$, which can be written as training data matrix

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_N^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{N \times m}. \tag{1}$$

Let J be an evaluation function like

$$J(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \sim \chi^2(m) \tag{2}$$

where $\boldsymbol{\Sigma}$ and χ are the covariance matrix and chi distribution. With a predefined threshold J_{th} a fault can be detected by the decision logic presented in Ding (2014) which can formulated as predicted class $\hat{c}$ by

$$\hat{c} = \begin{cases} -1 & \text{if } J > J_{\text{th}} \Rightarrow \text{faulty} \\ 1 & \text{if } J \leq J_{\text{th}} \Rightarrow \text{fault-free} \end{cases}. \tag{3}$$

An evaluation of FD methods based on FAR and FDR is widely used. FAR and FDR are the probabilities, that a fault is detected if there is no fault (p_{FAR}) and there is a fault (p_{FDR}) in the system. They can be formulated based on a classification problem as

$$p_{\text{FAR}} = P(c = -1 | \boldsymbol{f} = \boldsymbol{0}) \tag{4}$$

$$p_{\text{FDR}} = P(c = -1 | \boldsymbol{f} \neq \boldsymbol{0}) \tag{5}$$

where $P(\cdot)$ and $\boldsymbol{f}$ are the probability and fault vector which is $\boldsymbol{f} = \boldsymbol{0}$ in fault-free case. In many applications a maximum allowed FAR could be accepted and the FDR should be maximized.

2.1 Principal Component Analysis

PCA is a basic multivariate method in FD which can be calculated with the following procedure (Ding et al., 2010). In training, first the covariance matrix is computed

$$\boldsymbol{\Sigma} \approx \frac{\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}}{N-1}. \tag{6}$$

Then principal components and associated singular vectors are calculated using e.g. SVD (singular value decomposition)

$$\begin{aligned} \frac{\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}}{N-1} &= \begin{bmatrix} \boldsymbol{P} & \tilde{\boldsymbol{P}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{\Sigma}} \end{bmatrix} \begin{bmatrix} \boldsymbol{P}^{\mathrm{T}} \\ \tilde{\boldsymbol{P}}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{m \times m} \\ \boldsymbol{\Sigma} &= \operatorname{diag}(\sigma_1, \ldots, \sigma_l) \\ \tilde{\boldsymbol{\Sigma}} &= \operatorname{diag}(\sigma_{l+1}, \ldots, \sigma_m) \\ \sigma_1 \geq \ldots \geq \sigma_l,\ & \sigma_l \gg \sigma_{l+1} \geq \ldots \geq \sigma_m. \end{aligned} \tag{7}$$

For fault detection the Hotelling's T^2 and SPE are calculated. The evaluation functions can be computed as

$$J_{\mathrm{T}^2}(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{P} \boldsymbol{\Sigma}^{-1} \boldsymbol{P}^{\mathrm{T}} \boldsymbol{x} \tag{8}$$

$$J_{\text{SPE}}(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} \tilde{\boldsymbol{P}} \tilde{\boldsymbol{P}}^{\mathrm{T}} \boldsymbol{x}. \tag{9}$$

The thresholds for a significance level α are set by

$$J_{\text{th},\mathrm{T}^2} = \frac{l(N^2-1)}{N(N-l)} F_\alpha(l, N-l) \tag{10}$$

$$J_{\text{th,SPE}} = \theta_1 \left(\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{\frac{1}{h_0}} \tag{11}$$

where $F_\alpha(l, N)$ and c_α are the F-distribution with l, N degrees of freedom and the normal deviate of the upper $1-\alpha$ percentile and

$$\theta_i = \sum_{j=l+1}^{m} \left(\sigma_j^2\right)^i,\ i = 1, 2, 3,\ h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2}. \tag{12}$$

2.2 One-class Support Vector Machine

The basic idea of SVM is to separate two classes with the decision hyperplane which has the biggest margin to both classes (Boser et al., 1992). The OC-SVM is a modification of it, where all data belongs to one-class and only the origin represent the other class. The aim is to find the hyperplane which separates the class with the biggest distance ρ to the origin (Schölkopf et al., 2001). Introducing an allowed

miss classification rate ν the hyperplane separates at least $(1-\nu)\cdot 100\%$ of $\boldsymbol{x}_i \in \mathcal{T}$ from the origin. For this purpose, a slack variable ξ is added.

For a training set $\mathcal{T}$ the optimization problem is given by

$$\underset{\boldsymbol{w},\boldsymbol{\xi},\rho}{\arg\min} \quad \frac{1}{2}\left\|\boldsymbol{w}\right\|^2 + \frac{1}{\nu N}\sum_{i|\boldsymbol{x}_i\in\mathcal{T}} \xi_i - \rho \tag{13}$$

$$\text{s.th.} \quad \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\boldsymbol{x}_i\right) \geq \rho - \xi_i,\ \forall i \in \{i|\boldsymbol{x}_i \in \mathcal{T}\} \tag{14}$$

$$\xi_i \geq 0 \tag{15}$$

Here, $\boldsymbol{w}$ is the weighting vector and $\boldsymbol{\phi}\left(\cdot\right)$ a transformation. The optimization problem eq. (13),(14),(15) can be formulated as dual problem by

$$\underset{\boldsymbol{\alpha}}{\arg\min} \quad \frac{1}{2}\sum_{i,j|\boldsymbol{x}_i\in\mathcal{T}} \alpha_i\alpha_j \boldsymbol{K}\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right) \tag{16}$$

$$\text{s.th.} \quad \sum_{i|\boldsymbol{x}_i\in\mathcal{T}} \alpha_i = 1,\ \forall i \in \{i|\boldsymbol{x}_i \in \mathcal{T}\} \tag{17}$$

$$0 \leq \alpha_i \leq \frac{1}{\nu N} \tag{18}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{K}\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)$ are the Lagrangian multiplier and kernel function. In this paper the Gaussian kernel is used, which it is an indirect transformation in an infinite dimension space and can describe different surface shapes with the kernel parameter γ. The kernel is given by

$$\boldsymbol{K}\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right) = \exp\left(-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x_j}\right\|}{2\gamma^2}\right). \tag{19}$$

The class of new measurement $\boldsymbol{x}_i$ is computed as

$$d_i = \sum_{j|\boldsymbol{x}_j\in\mathcal{SV}} \alpha_j \boldsymbol{K}\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right) - \rho \tag{20}$$

$$\hat{c}_i = \operatorname{sgn}\left(d_i\right) = \begin{cases} -1 & \text{, if } d_i < 0 \\ 1 & \text{, if } d_i \geq 0 \end{cases} \tag{21}$$

where $\mathcal{SV}$ is the set of all support vectors and d is the distance to the separating hypersurface.

2.3 Probabilistic performance evaluation of FD systems

Theorem 1. (Tempo et al., 2012) Given is the significance level $\epsilon \in (0,1)$ and accuracy $\delta \in (0,1)$, then

$$N \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta} \tag{22}$$

independent and identically distributed (i.i.d.) samples ensure

$$P\left(\left|p_{\mathrm{FDR}}\left(\boldsymbol{f}\right) - \hat{p}_{\mathrm{FDR}}\left(\boldsymbol{f}\right)\right| < \epsilon\right) > 1-\delta. \tag{23}$$

Random generation of Gaussian distributed fault-free data is given by

$$\boldsymbol{X} \sim \mathcal{N}\left(\boldsymbol{0},\boldsymbol{\Sigma_0}\right) \in \mathbb{R}^{N\times m}. \tag{24}$$

A random uniform distributed rotation can be generated, using the QR decomposition given by

$$\boldsymbol{Q}\cdot\boldsymbol{R} = \boldsymbol{A} \tag{25}$$

where $\boldsymbol{A} \sim \mathcal{N}\left(\boldsymbol{0},\boldsymbol{I}\right) \in \mathbb{R}^{m\times m}$ is a random normal distributed squared matrix of size m. The rotation matrix $\boldsymbol{U}$ is then given by

$$\boldsymbol{U} = \boldsymbol{Q}\cdot\operatorname{diag}\left(\operatorname{diag}\left(\operatorname{sgn}\left(\boldsymbol{R}\right)\right)\right). \tag{26}$$

According to Calafiore et al. (2007), a random generated uniform distributed ball with radius $r_{\max}$ is given by

$$\boldsymbol{x}_i^{\mathrm{T}} = \left[r_{\max}\cdot r_{i,0}^{\frac{1}{m}}\ \boldsymbol{0}\right]\boldsymbol{U} \tag{27}$$

where $r_{i,0} \sim \mathcal{U}\left(0,1\right)$ is uniformly distributed.

By adding random generated uncertainties to each test data set, a meaningful result is achieved. A random change in covariance matrix $\boldsymbol{\Delta\Sigma}$ is considered by

$$\boldsymbol{\Delta\Sigma} = \operatorname{diag}\left(\beta_1\sigma_{1,0},\cdots,\beta_m\sigma_{m,0}\right) \tag{28}$$

$$\beta_i \sim \mathcal{U}\left(\beta_{\min},\beta_{\max}\right). \tag{29}$$

The random data set with scaling uncertainties is then given by

$$\boldsymbol{X}_{\mathrm{s}} \sim \mathcal{N}\left(\boldsymbol{0},\boldsymbol{\Sigma_0} + \boldsymbol{\Delta\Sigma}\right). \tag{30}$$

Additionally a rotation of the data set is considered. with both disturbances it is finally constructed as

$$\boldsymbol{X}_{\mathrm{d}} = \boldsymbol{X}_{\mathrm{s}}\cdot\boldsymbol{U} \tag{31}$$

2.4 Convexity

In this work convex set and convex function properties are used for parameter optimization. According to Boyd and Vandenberghe (2004), a convex set and function can be defined as follows.

Definition 1. Given is the set $\mathcal{S} \subset \mathbb{R}^n$. The set $\mathcal{S}$ is convex if for each point pair $\boldsymbol{x},\boldsymbol{y} \in \mathcal{S}$ and variable $\lambda \in [0,1]$ holds

$$\lambda\boldsymbol{x} + (1-\lambda)\,\boldsymbol{y} \in \mathcal{S}. \tag{32}$$

Definition 2. Given is the set $\mathcal{S} \subset \mathbb{R}^n$. A function $f : \mathcal{S} \to \mathbb{R}$ is convex if for each point pair $\boldsymbol{x},\boldsymbol{y} \in \mathcal{S}$ and variable $\lambda \in [0,1]$ holds

$$f\left(\lambda\boldsymbol{x} + (1-\lambda)\,\boldsymbol{y}\right) \leq \lambda f\left(\boldsymbol{x}\right) + (1-\lambda)\,f\left(\boldsymbol{y}\right) \tag{33}$$

3. FRAMEWORK

OC-SVM with Gaussian kernel is used to learn the threshold. The maximum FAR p_{FAR} is equal to the parameter ν. An iterative parameter optimization is used to find the kernel parameter γ. The optimal parameter belongs to the smallest sphere which is convex. The user has to choose the initial step size δ_0, initial kernel parameter γ_0, number of outer loops n_{out}, reduction factor for step size r_δ and number of steps to prove convexity n_{con}. These parameters influence the accuracy in a similar way as the step size influences the gradient descent methods. Such that a conservative choice ensures a high accuracy, but also can result in an increased training time.

As part of the initializations an SVM with the starting parameter γ_0 is trained. Afterwards convexity is proven. If the separating surface is the border of a convex set, there could be a smaller convex set. To evaluate this, the kernel parameter γ_j is reduced. Else, the kernel parameter γ_j should be increased until the separating surface is the border of a convex set. For this purpose, the convexity of outer loop i and search direction are calculated as

$$b_{i,j}^{\mathrm{con}} = \begin{cases} -1 & \text{no convex set} \\ 1 & \text{convex set} \end{cases} \tag{34}$$

$$b_i^{\mathrm{dir}} = b_{i,0}^{\mathrm{con}}. \tag{35}$$

The new kernel parameter γ_{j+1} is then given by

$$\gamma_{j+1} = \gamma_j + b_i^{\mathrm{dir}}\cdot\delta_i \tag{36}$$

where δ_i is the step size.

In each inner loop j eq. (34) and (36) are calculated until the stopping criteria $b_i^{\mathrm{con}}\cdot b_{i,j}^{\mathrm{dir}} > 0$ is reached, which is the change from convex to non-convex or vice versa.

To reduce the computation time, the step size δ_i is large at the beginning. To ensure high accuracy an outer loop reduces the step size and changes the search direction n_{out} times as

$$b_{i+1}^{\text{dir}} = -b_i^{\text{dir}} \tag{37}$$
$$\delta_{i+1} = r_\delta \cdot \delta_i. \tag{38}$$

The scheme is illustrated in fig. 1.

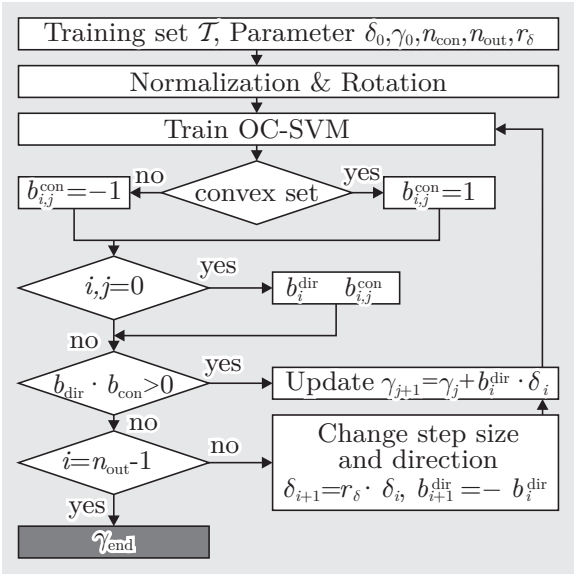

Fig. 1. Threshold setting scheme

To proof the convexity, eq. (32) is approximated with a finite number of λ and only calculated for all possible pairs $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{T}_0$, where $\mathcal{T}_0 \in \left\{\boldsymbol{x}_i | \boldsymbol{x}_i \in \mathcal{T} \wedge \alpha_i < \frac{1}{\nu N}\right\}$ is the set of all training points except outliers. Eq. (32) is formulated as

$$\boldsymbol{x}_i + k\frac{\Delta s}{n_{\text{con}}} \underbrace{(\boldsymbol{x}_i - \boldsymbol{x}_j)}_{\Delta \boldsymbol{s}} \in \mathcal{C}_1,\ k = 1, \ldots, n_{\text{con}} - 1 \tag{39}$$

with $\mathcal{C}_1$ the set of all points belonging to class 1. A further computation time reduction can be achieved by only using non-outliers SVs given by

$$\mathcal{SV}_0 = \left\{\boldsymbol{x}_i | \boldsymbol{x}_i \in \mathcal{T} \wedge 0 < \alpha_i < \frac{1}{\nu N}\right\},\ \mathcal{SV}_0 \subset \mathcal{T}_0 \tag{40}$$

This approximation is only focused on the border of the set.

Alternatively, the distance of OC-SVM can be used. If the separating surface is the border of a convex set, the distance between two SVs is a concave function. Based on eq. (20) and (33) the proof of convexity can be approximated by

$$\begin{aligned} d(\boldsymbol{x_i} + \lambda \Delta \boldsymbol{s}) &\leq d(\boldsymbol{x_i}) + \lambda (d(\boldsymbol{x}_j) - d(\boldsymbol{x}_i)) \\ \lambda &= k\frac{\Delta s}{n_{\text{con}}},\ k = 1, \ldots, n_{\text{con}}. \end{aligned} \tag{41}$$

4. MODIFICATIONS OF EVALUATION WITH RANDOMIZED ALGORITHM

It is known that correlations among the variables may affect the detection performance of FD methods. To address this point, a rotation eq. (25), (26) is added such that random Gaussian distributed and disturbed data is generated as

$$\boldsymbol{X}_{\text{d}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_0 + \Delta\boldsymbol{\Sigma}) \cdot \boldsymbol{U}_0 \cdot \boldsymbol{U}. \tag{42}$$

Uniformly distributed data sets with different shapes are generated, to evaluate the capabilities to non-Gaussian, convex data sets. According to Muller (1959) and Tempo et al. (2012) this can be done by

$$\boldsymbol{x}_i^{\text{T}} = \left[r_{\max} \cdot r_{i,0}^{\frac{1}{m}}\ \boldsymbol{0}\right] \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_p},\ \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_0) \tag{43}$$

where $\|\cdot\|_p$ is the pth norm.

$$\|\boldsymbol{z}\|_p = \left(\sum_{i=1}^{N} |z_i|^p\right)^{\frac{1}{p}}. \tag{44}$$

The generation of a random rotation matrix with limited angles using QR decomposition is problematic, but needed for rotation disturbances. In this paper, Givens rotation is suggested for this purpose. For each feature pair i, k a random uniformly distributed angle $\theta_{i,k} \sim \mathcal{U}\left(\theta_{i,k}^{\min}, \theta_{i,k}^{\max}\right)$ is generated and the related rotation matrix is given as

$$\boldsymbol{G}_{i,k,\theta_{i,k}} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \tag{45}$$

where $c = \cos(\theta_{i,k})$ and $s = \sin(\theta_{i,k})$. The whole rotation matrix is the matrix multiplication of all sub rotations

$$\boldsymbol{U}_{\text{d}} = \prod_{i=1}^{m} \prod_{k=i+1}^{m} \boldsymbol{G}_{i,k,\theta_{i,k}}. \tag{46}$$

In this work, no prior knowledge about the fault is assumed. For evaluation a uniform distributed offset fault is generated as ball scaled by the nominal covariance $\boldsymbol{\Sigma}_0$ and rotated like the disturbed data set $\boldsymbol{U}_{\text{d}}$. Using eq. (27) the additive fault is given by

$$\boldsymbol{f}_{\text{Offset},i}^{\text{T}} = \boldsymbol{x}_i^{\text{T}} \boldsymbol{\Sigma}_0 \boldsymbol{U}_0 \boldsymbol{U}_{\text{d}}. \tag{47}$$

5. RESULTS

PCA is used to compare the standard T^2, SPE and the new OC-SVM threshold. Both thresholds are designed for a FAR of 1% ($\alpha = \nu = 0.01$). In case of a Gaussian distributed data set, principal components build a hyperdimensional ellipse. The optimal data set shape for the OC-SVM threshold is a hyperdimensional ball, which is achieved by a normalization of the principal components. The evaluation is done based on RA algorithm which is used to generate random data sets and uncertainties (Ding et al., 2019). Threshold training is based on a random data set without uncertainties and evaluated with N data sets with random uncertainties. The significance level and accuracy $\epsilon = \delta = 0.01$ are chosen. Based on eq. (22) a sample size of $N = 26,492$ is necessary. Eigenvalues of the covariance matrix $\boldsymbol{\Sigma}_0 = \text{diag}(\sigma_{1,0}, \cdots, \sigma_{m,0}), \sigma_{i,0} \sim \mathcal{U}(0, 2)$ are generated randomly. For scale and rotation

disturbance $\beta_{\min/\max} = \mp 0.5$ and $\theta_{i,k}^{\min/\max} = \mp 2.5°$ are chosen. The initialization parameters in all simulations are kernel parameter $\gamma_0 = 10$, step size $\delta_0 = 1$, update factor $r_\delta = 0.5$, number of outer loops $n_{\text{out}} = 3$ and checks per pair $n_\lambda = 200$. For the fault generation $r_{\max} = 2$ is chosen.

First, Gaussian distributed data with dimension 2 and 5 are evaluated. The FAR and FDR are shown in tab. 1. Here $Q(\cdot)$ is the quantile. Both threshold setting methods

Table 1. FAR/FDR evaluation statistics for Gaussian distributed data sets with uncertainties

	mean	min	$Q(25)$	med	$Q(75)$	max
FAR						
T^2, SPE (2)	1.42	0.00	0.25	0.96	2.52	4.69
OC-SVM (2)	1.42	0.00	0.25	0.97	2.53	4.69
T^2, SPE (5)	3.05	0.04	0.57	2.03	5.45	9.30
OC-SVM (5)	2.29	0.00	0.24	1.34	4.27	7.92
FDR						
T^2, SPE (2)	13.50	5.99	10.01	13.62	17.02	20.62
OC-SVM (2)	13.63	6.14	10.16	13.73	17.14	20.72
T^2, SPE (5)	10.07	1.51	4.68	9.30	15.40	21.05
OC-SVM (5)	9.39	1.22	4.14	8.53	14.55	20.21

find a comparable compromise between FAR and FDR. In both cases the median FAR $p_{\text{FAR,med}} = 0.96\,\% - 2.03\,\%$ is not exact the chosen one, because the training based on the nominal data set and worst case evaluation with uncertainties is carried out. That is why the worst case FAR $p_{\text{FAR}}^{\max} = 9.30\,\%$ is much higher than the chosen one. Depending on the uncertainties taken into account, these values become even greater or smaller.

Beside the accuracy, resource demand like training time and amount of training data is an important factor for the usage. To evaluate this, Gaussian distributed data with different training set size is simulated 100 times. The training PC is equipped with an Intel i5-8600 (3.16 GHz), 16 GB RAM, MATLAB 2018a and Windows 7.The training time results are presented in tab. 2. T^2, SPE threshold setting

Table 2. Average training time [mm:ss] for different training set size

Size	250	1500	3000	7500	15,000	26,492
$\bar{T}$	00:00.1	00:01.2	00:04.3	00:32.5	02:40.4	11:19.2

has a simple calculation and the training time is negligible. Especially for large training sets the computation time of the new method is much higher. Because the training is normally done offline an average training time of ≈ 12 min for the biggest training set ($N = 26492$) is acceptable for most application. In this paper, a standard SVM implementation is used. Optimized implementation e.g. parallelization can reduce the training time. For the online computation, which is often time critical, the difference is much smaller but depending on the number of SVs. The computation time can be reduced by hardware friendly version of SVM like suggested in Anguita et al. (2007). If the training data is generated by experiments, the data set can be reduced for some applications, because OC-SVM only needs the data points in the near of border to get the optimal threshold. However, an additional step is then required to set the maximum allowed FAR.

The results of needed training size is shown in tab 3. It

Table 3. Average FAR/FDR evaluation statistics for Gaussian distributed data sets with uncertainties for different N

Size	250	500	1500	3000	7500	15,000	26,492
FAR							
T^2, SPE	1.62	1.62	1.54	1.50	1.53	1.52	1.53
OC-SVM	3.28	2.04	1.71	1.56	1.59	1.56	1.47
FDR							
T^2, SPE	10.97	9.86	10.53	9.63	9.53	10.15	9.96
OC-SVM	15.82	10.62	10.98	9.80	9.65	10.01	9.77

is observable, that the OC-SVM needs a larger training set to find a good separating surface. This was expected beforehand, because no assumption about the distribution was included.

The threshold setting for a random generated uniformly distributed data sets with different shapes show the capability to non-Gaussian, convex data sets. The shape is an ellipse for $p = 2$ and becomes a rectangle for $p = \infty$. The same parameters as in the Gaussian case are used except scale $\beta_{\min/\max} = \mp 0.02$ and rotation disturbance $\theta_{i,k}^{\min/\max} = \mp 3.6°$. In tab. 4 the FAR and FDR for uniformly distributed data sets of dimension 2 are evaluated. Obviously T^2, SPE threshold show a much

Table 4. FAR/FDR evaluation statistics for uniform distributed data sets with uncertainties

method	mean	min	$Q(25)$	med	$Q(75)$	max
FAR						
T^2, SPE ($p = 2$)	0.00	0.00	0.00	0.00	0.00	0.00
OC-SVM ($p = 2$)	2.25	0.00	1.40	2.25	3.11	5.57
T^2, SPE ($p = 3$)	0.00	0.00	0.00	0.00	0.00	0.00
OC-SVM ($p = 3$)	2.82	0.00	1.83	2.83	3.82	6.77
T^2, SPE ($p = 5$)	0.00	0.00	0.00	0.00	0.00	0.00
OC-SVM ($p = 5$)	1.40	0.17	1.02	1.36	1.75	3.35
FDR						
T^2, SPE ($p = 2$)	8.73	7.63	8.52	8.73	8.95	9.85
OC-SVM ($p = 2$)	41.68	40.02	41.38	41.68	41.98	43.21
T^2, SPE ($p = 3$)	7.21	6.13	6.98	7.20	7.42	8.36
OC-SVM ($p = 3$)	39.67	38.07	39.36	39.67	39.38	41.24
T^2, SPE ($p = 5$)	6.15	5.27	5.95	6.15	6.34	7.14
OC-SVM ($p = 5$)	36.85	35.31	36.53	36.84	37.17	38.56

lower FAR. But because the Gaussian assumption is not fulfilled the FDR is much lower compared to the OC-SVM. The threshold setting cannot find a good compromise between FAR and FDR. The median FAR of OC-SVM $p_{\text{FAR,med}} = 1.36$ to 2.83 is near to the chosen FAR and FDR is much higher than T^2, SPE. In fig. 2 the training result of $p = 5$ is shown. It is visible, that most of the SVs are in the corners. In extreme cases (rectangle), the corners are rounded off incorrectly and a small distance between the threshold and real limit is created in the area between them. The geometric threshold reacts less sensitive to different shapes and distributions. But it is not optimal for the case of a rectangle with sharp corner. As mentioned the rectangle shape is an extreme case of convex data set and in real applications the border is often more smooth.

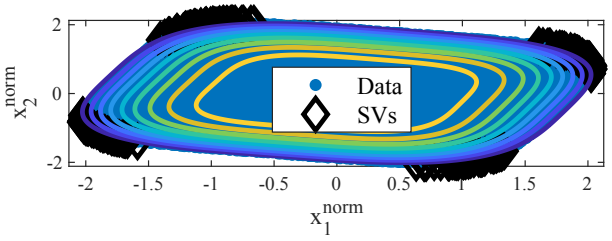

Fig. 2. Training result of $p = 5$

6. CONCLUDING REMARKS AND FUTURE WORK

In this work, we have presented a novel distribution independent threshold setting method for convex data sets. For this purpose, OC-SVM is used which gives the possibility to choose a maximum FAR. As shown, the FD performance of both threshold setting methods for Gaussian distributed data sets is comparable. Because the Gaussian distribution is not assumed in case of OC-SVM, training time is higher. For this reason, traditional threshold setting are preferred for Gaussian distributed data sets. For non-Gaussian distributed but convex data sets (like the uniformly distributed data sets) the performance of the new approach is better. It can be concluded that the OC-SVM threshold setting can be used for all convex data sets independent from the distribution. If the distribution is non-Gaussian or unknown, the proposed method is the better choice.

In addition, an RA based evaluation for data-driven algorithms is presented, which was mainly developed in the DFG project DI733-15 and is further improved in this work. For the simulation of rotation uncertainties with limited angle a solution with Givens rotation was introduced which results in a higher adaptability to real problems.

In this work, PCA and normalization is used as preprocessing which is not always the optimal choice. This was done to get a clear comparison between two threshold setting methods with low influences from preprocessing. In combination with the new threshold setting other mapping methods should be suitable and extend the usage to more problems (e.g. non-convex).

Especially in industrial applications the interpretability is an important factor, such that a nonlinear mapping is not always wanted. For this purpose, a multi-mode method can be used which will be proved in the future.

The most critical disadvantage is the training time for big data sets. It is expected that an adaptive iteration step size and predicted initialization parameter can reduce the computation time significantly. This can further increase the usability.

ACKNOWLEDGEMENTS

The authors thank the colleagues involved in the project DFG DI733-15 who developed the main part of the presented RA based FD evaluation.

REFERENCES

Anguita, D., Ghio, A., Pischiutta, S., and Ridella, S. (2007). A hardware-friendly support vector machine for embedded automotive applications. In *Proceedings of the 2007 IEEE International Joint Conference on Neural Networks*, 1360–1364. Orlando, USA.

Boser, B.E., Guyon, I.M., and Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual workshop on Computational learning theory*, 144–152. Pittsburgh, USA.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Calafiore, G., Dabbene, F., and Tempo, R. (2007). A survey of randomized algorithms for control synthesis and performance verification. *Journal of Complexity*, 23(3), 301 – 316.

Chen, Z., Ding, S.X., Peng, T., Yang, C., and Gui, W. (2018). Fault detection for non-gaussian processes using generalized canonical correlation analysis and randomized algorithms. *IEEE Transactions on Industrial Electronics*, 65(2), 1559–1567.

Ding, S.X. (2014). *Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. Advances in Industrial Control. Springer London.

Ding, S.X., Zhang, P., Ding, E., Yin, S., Naik, A., Deng, P., and Gui, W. (2010). On the application of pca technique to fault diagnosis. *Tsinghua Science and Technology*, 15(2), 138–144.

Ding, S.X., Li, L., and Krüger, M. (2019). Application of randomized algorithms to assessment and design of observer-based fault detection systems. *Automatica*, 107, 175 – 182.

Koenings, T., Esch, J., Kandler, C., Ding, S.X., and Louen, C. (2013). Robust, optimal pi-controller tuning for integrator plus delay plants with varying parameters based on randomised algorithms. In *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, 110–115.

Louen, C., Ding, S.X., and Kandler, C. (2013). A new framework for remaining useful life estimation using support vector machine classifier. In *Proceedings of the 2013 IEEE Conference on Control and Fault-Tolerant Systems*, 228–233. Nice, France.

Louen, C.J. (2016). *Datenbasierte Zustandsüberwachung in Personenkraftfahrzeugen mit Anwendung an einem Drei-Wege-Katalysator*. Springer Vieweg, 1 edition.

Muller, M.E. (1959). A note on a method for generating points uniformly on n-dimensional spheres. *Commun. ACM*, 2(4), 19–20.

Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., and Williamson, R.C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), 1443–1471.

Tempo, R., Calafiore, G., and Dabbene, F. (2012). *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer Science & Business Media.

Widodo, A. and Yang, B.S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6), 2560–2574.

Yin, S., Ding, S.X., Haghani, A., Hao, H., and Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process. *Journal of Process Control*, 22(9), 1567–1581.