Outlier Detection in Condition Monitoring

Master Thesis

Ralph Fehrer

Intelligente Eingebettete Mikrosysteme

2012-2013

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Overview

- Condition Monitoring
- Feature Extraction Methods
- One Class Classification
- Experiments
- Summary
- Outlook

Condition Monitoring

Condition Monitoring

„Condition Monitoring is a process
of monitoring a system
by studying certain selected parameters
in such a way,
that significant changes of those parameters
are related to developing failures“

Artur G. O. Musambara , Harare 2012

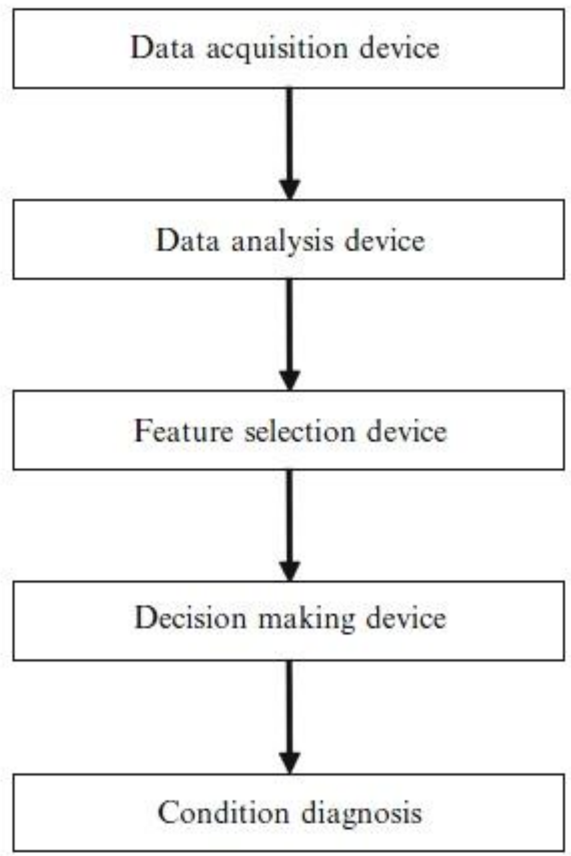
Condition Monitoring

Benefits of (computerized) Condition Monitoring

- Predictive maintenance approach, based on machine conditions
- Prevention of machine failure
- Lifetime prediction
- Overall reduction of operating costs

Condition Monitoring Framework

Marwala, 2012



Sensors atttached to monitored system (accelerometers, thermometers…)

Feature extraction (Fourier, Short Time Fourier, Wavelet Transform…)

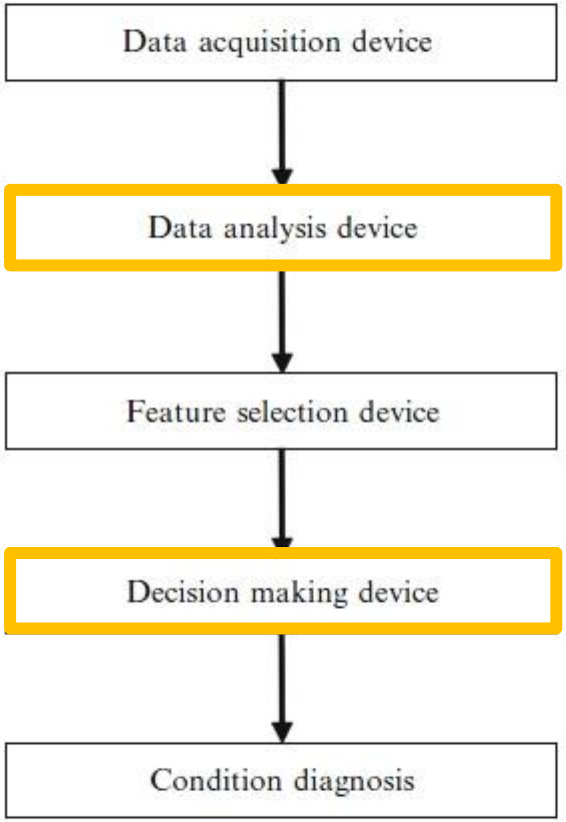
Selection of most significant features (Principal Component Analysis)

Classification / Outlier detection (Machine Learning/Pattern Recognition)

Lifetime prediction, maintenance scheduling…

Condition Monitoring Framework

Marwala, 2012



Focus of this Thesis

Data Analysis
Feature Extraction

Data Analysis / Feature Extraction

- Mel Frequency Cepstral Coefficients (MFCC)
- Higuchi Fractal Dimensions (HFD)
- Kurtosis

Mel Frequency Cepstral Coefficients

Basics

Cepstrum: $C(\tau) = \mathcal{F}^{-1}\left(\log\left(\mathcal{F}(f(t))\right)\right)$

LTI: $y(t) = x(t) * h(t)$

Cepstrum(LTI): $C_y(\tau) = C_x(\tau) + C_H(\tau)$

⇨ Ceptstrum separates signal $C_x(\tau)$ from transmission path characteristics $C_H(\tau)$

Mel Frequency Cepstral Coefficients

MFCC feature extraction method

- Short Time Fourier Transform (STFT) of time signal

  ⇨ transformation into time frequency domain
- Mel transform of frequency spectrum

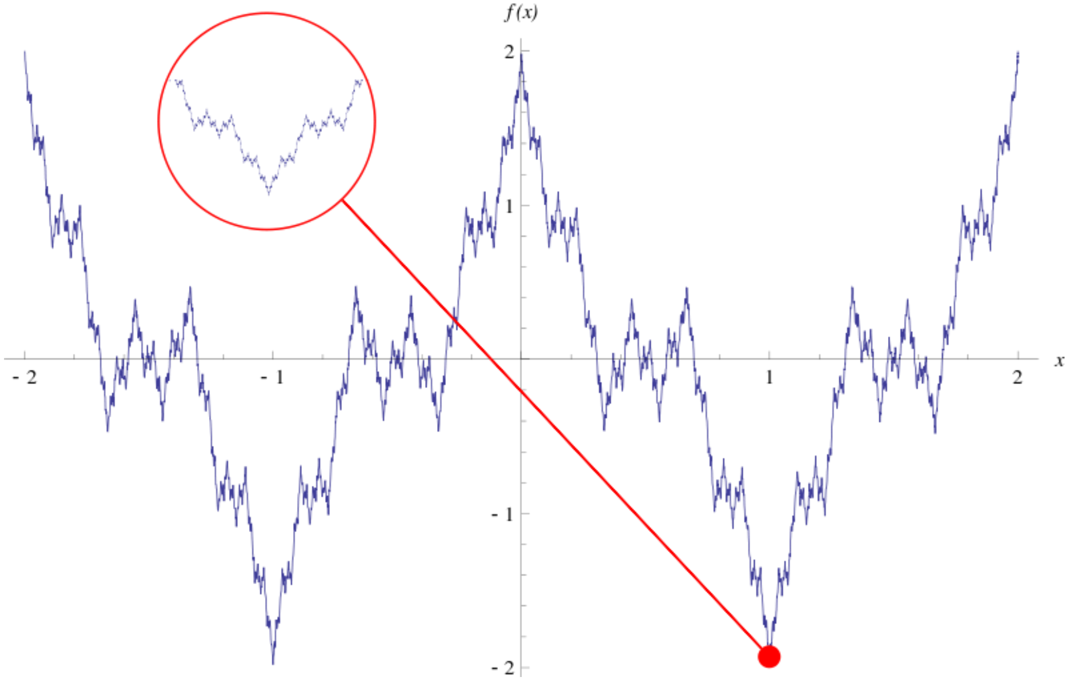
  ⇨ adaption to frequency response of human auditory system
- Transformation of logarithmic frequency spectrum to time domain through Discrete Cosine Transform

Fractal Dimensions

FD in time signal analysis

- Measure for the irregularity of signal shape
- Fractal dimensions of a signal are in the range between 1 and 2
- Several methods exist, usually measuring length at different scales
- Higuchi algorithm implemented for this thesis (Matlab)

Fractal sample



Weierstraß function with known fractal dimension
used to validate Higuchi algorithm

Higuchi Fractal Dimensions

Algorithm

- Creation of k new signals from the time signal at different scales
- Measuring of the length $L(k)$ of the scaled time signals, i.e. of the distance between points
- Calculation of HFD through the relation $L(k) \sim k^{-D}$.

Kurtosis

- Measure of curve „peakedness“
- Normalized 4^{th} order moment:

$$K = \frac{1}{N}\sum_{i=1}^{N}\frac{(x_i - \mu)^4}{\sigma^4}$$

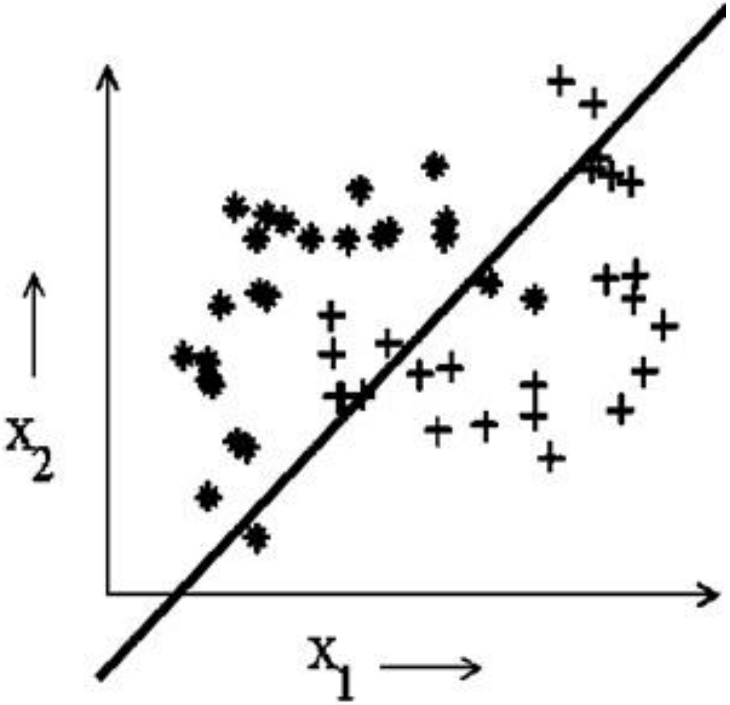
Decision Making Classification

Classification

Supervised, multi-class

- Data samples for all classes
- Classification function $y = f(\boldsymbol{x}, \boldsymbol{w})$
- Training by optimizing $\boldsymbol{w}$ through error function, e.g.

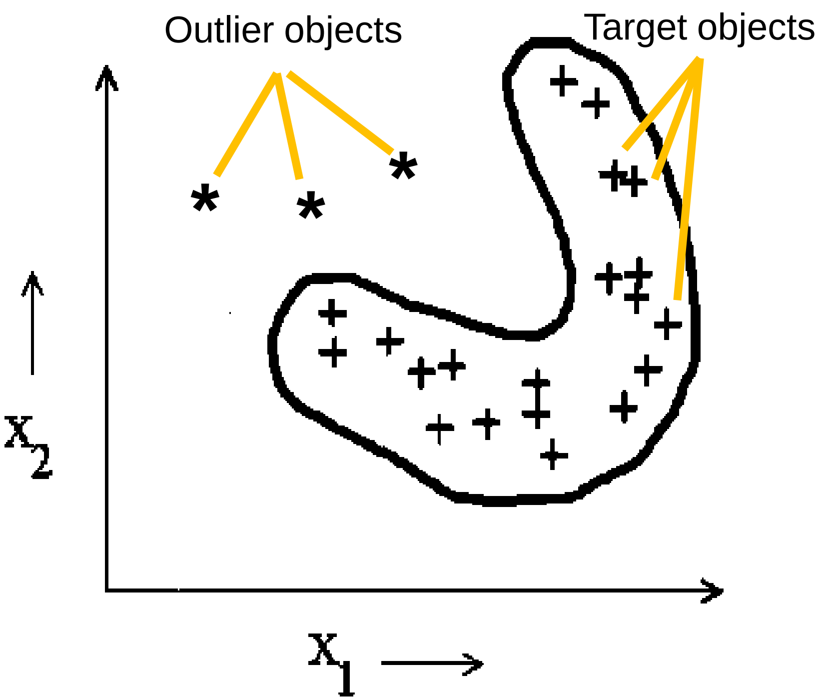
$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\{f(\boldsymbol{x}_n, \boldsymbol{w}) - y_n\}^2$$



One Class Classification

Supervised, one-class

- Typical Condition Monitoring scenario: Only samples for one class (normal conditions)
- Strategy: Precise description of target class (e.g. through closed boundary)
- Also known as: *Data Description, Domain Description, Outlier or Novelty Detection*



One Class Classification

Components

1. **Similarity measure**
   measure for the relation of new data objects $\boldsymbol{x}$ to the known target data description

   - $d(\boldsymbol{x})$ (distance)
   - $p(\boldsymbol{x})$ (probability)

2. **Threshold θ**

One Class Classification

Classifier functions

$$f(\boldsymbol{x}) = I(d(\boldsymbol{x}) < \theta)$$
$$f(\boldsymbol{x}) = I(p(\boldsymbol{x}) > \theta)$$

Where $I(\cdot)$ is the *indicator function* s.t.,

$$f: \mathbb{R}^D \rightarrow \{1,0\}$$

1=target, 0=outlier

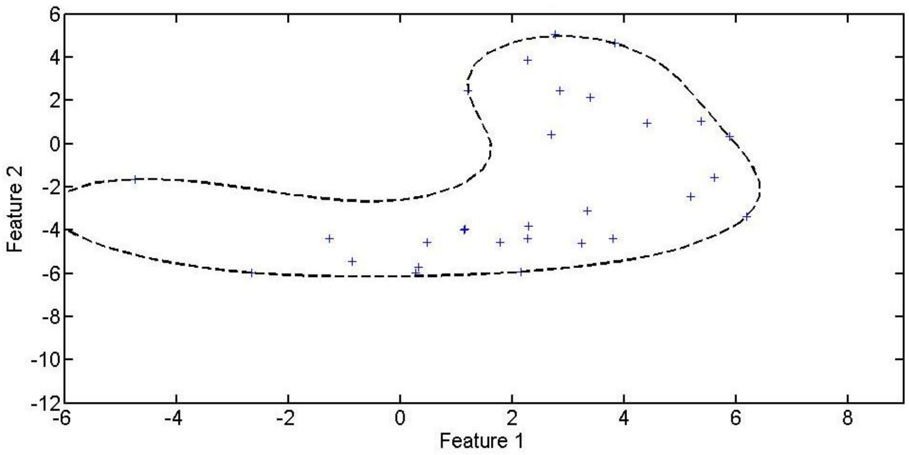
D. M. J Tax,2001

Support Vector Data Description

Principle: Support objects defining a hypersphere with radius *R* and center a

Training: Minimization of *R* such that most traning target objects are inside hypersphere

Classification: new objects are target objects if they are inside the boundary, outliers otherwise

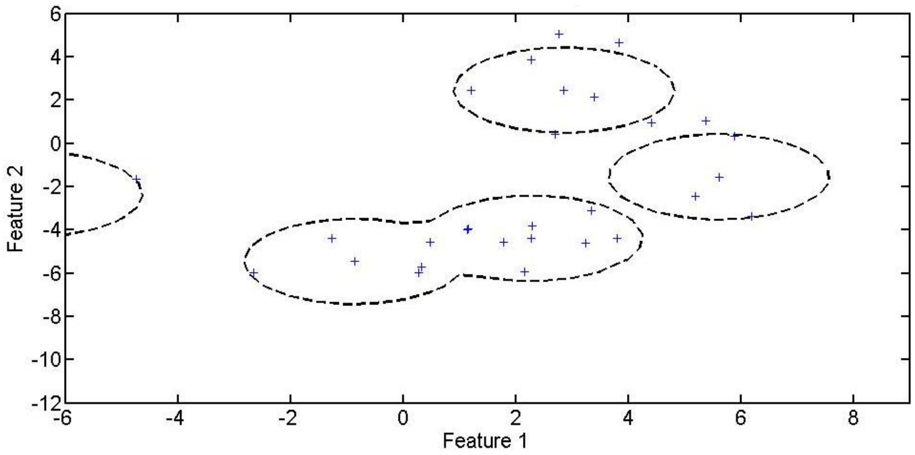


K-Center Data Description

Principle: K support objects surrounded by Receptive Fields of equal radius *R*, data objects belong to the receptive fields of the nearest support object

Training: Selection of support objects such that *R* is minimal and all sample objects are inside receptive fields

Classification: new objects are target objects if their distance to the nearest support object is smaller than the threshold

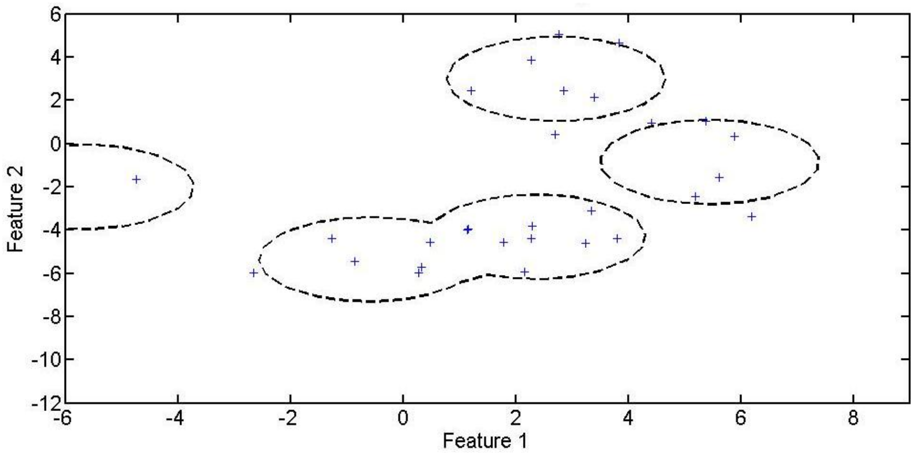


K-Means Data Description

Principle: K clusters, data objects belong to the clusters of the nearest cluster center

Training: recalculation of cluster centers as means of objects in the cluster until convergence

Classification: new objects are target objects if their distance to the nearest cluster center is smaller than the predefined threshold

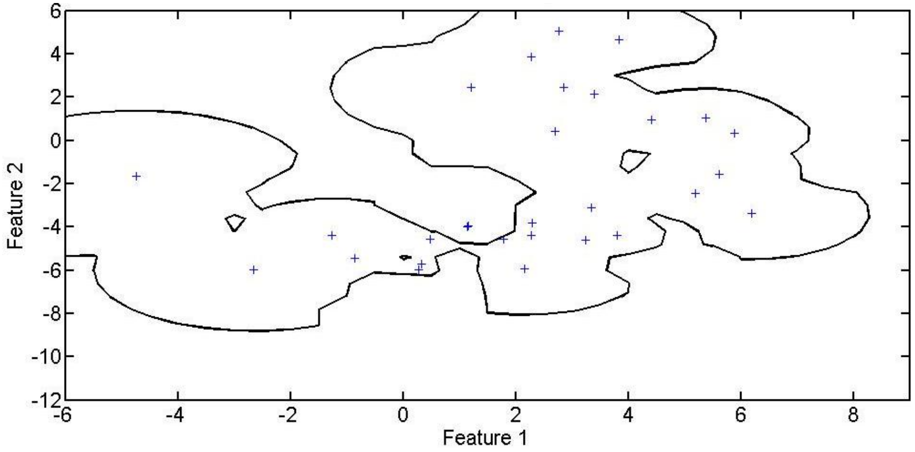


Nearest Neighbor Data Description

Principle: Distances between nearest data objects

Training: No training, but calculation and memorization of distances

Classification: new objects are target objects if their distance to the nearest sample target object is smaller than the distance of the nearest target object to its nearest neighbor

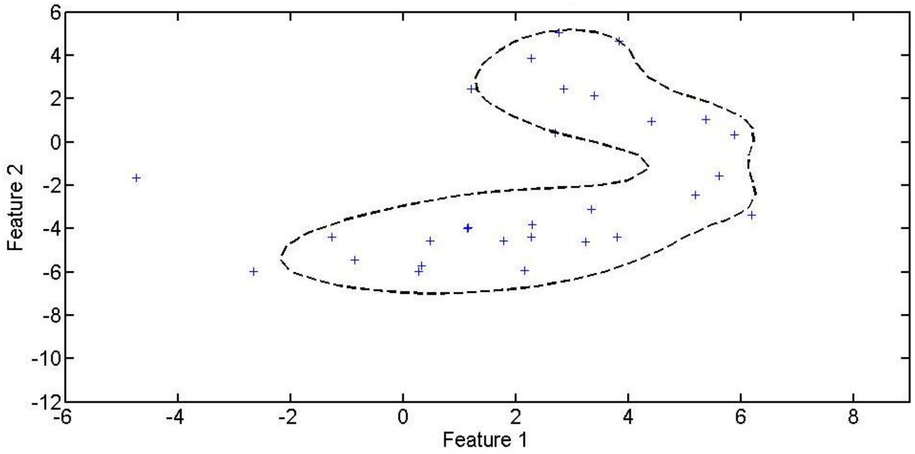


Parzen Window Data Description

Principle: Density measure $p(\boldsymbol{x})$ based on accumulation of distances between a data object and the sample target objects

Training: No training required

Classification: new objects are target objects if their density $p(\boldsymbol{x})$ is above a predefined threshold, outliers otherwise

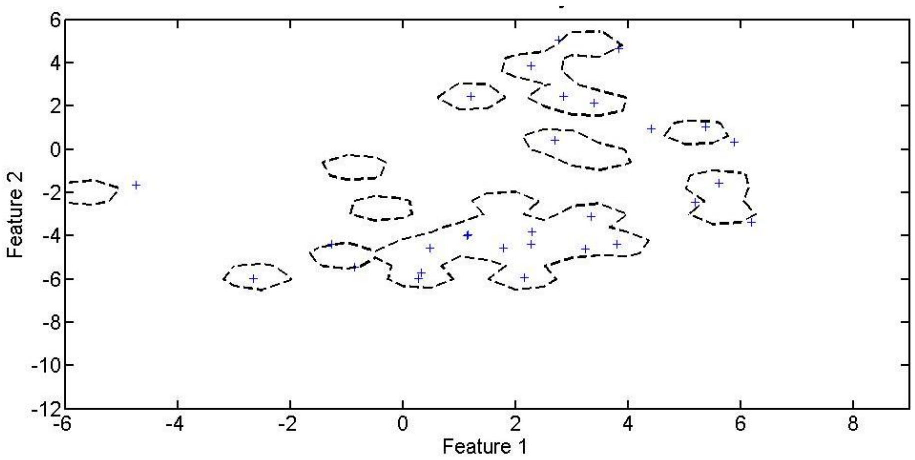


SOM Data Description

Principle: prototypes of sample target objects , associated with nodes in a grid structure

Training: sequential presentation of sample target objects, update of nearest prototype and the surrounding prototypes

Classification: new objects are target objects if their distance to the nearest prototype is below the predefined threshold, outliers otherwise



Random Forest

- Construction of many trees
- Proximity Matrix - $N \times N$ matrix where entries are measures for the proximity of training objects
- Outlier measure – detection of outliers in training set
- Feature importance measure
- Class centers / prototypes
- Unsupervised mode
  - All training samples treated as target class data
  - Construction of a synthetic second class
  - Construction of a 2-class random forest

Random Forest Data Description

Principle:	See above
Training:	Random Forest in supervised mode, calculation of target class center(s) / prototype(s)
Classification:	new objects are target objects if their distance to the (nearest) target class prototype is below a pre-defined threshold, outliers otherwise

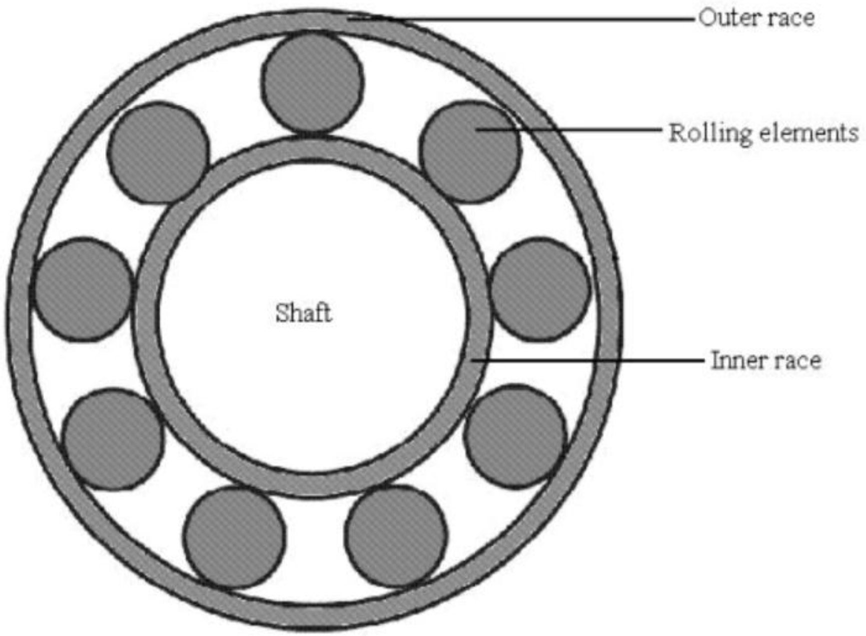
→ Implemented for this thesis in R and Matlab, Proof of concept in experiment part

Experiments

Experiments

- Based on roller bearing benchmark data from Case Western University
- Implemented as Matlab Scripts
- Experiment 1
  - Preprocessing
  - Feature Extraction
  - Classifier Training and classification (repeated with sampled training and test set)
  - Evaluation
- Experiment 2
  - Similar sequence as in experiment 1
  - Additional feature reduction through Random Forest Variable Importance measure

Data Acquisition/Roller Bearing Data



Vibration Data

(accelerometers)

Data Analysis/Preprocessing

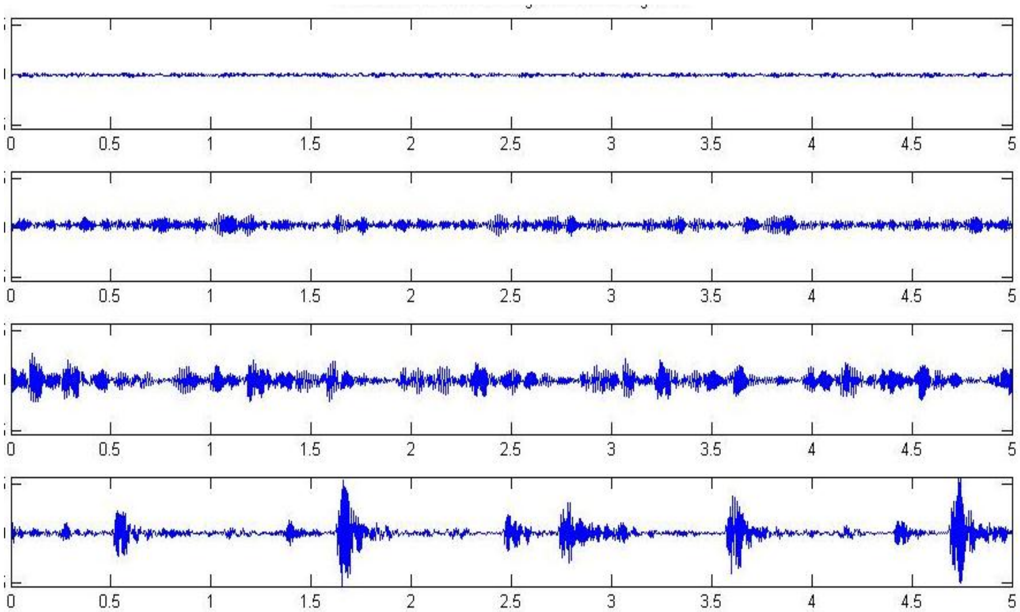
4 vibration signals:

1. normal conditions
2. ball fault conditions
3. inner raceway fault
4. outer raceway fault

Obtained from Case Western University

Split signals into segments corresponding to 5 revolutions of Roller Bearing

Data Analysis/ Data sample



Normal conditions

Ball fault conditions

Inner raceway fault conditions

Outer raceway fault conditions

1 segment corresponding to 5 revolutions of Roller Bearing

Data Analysis/Feature Extraction

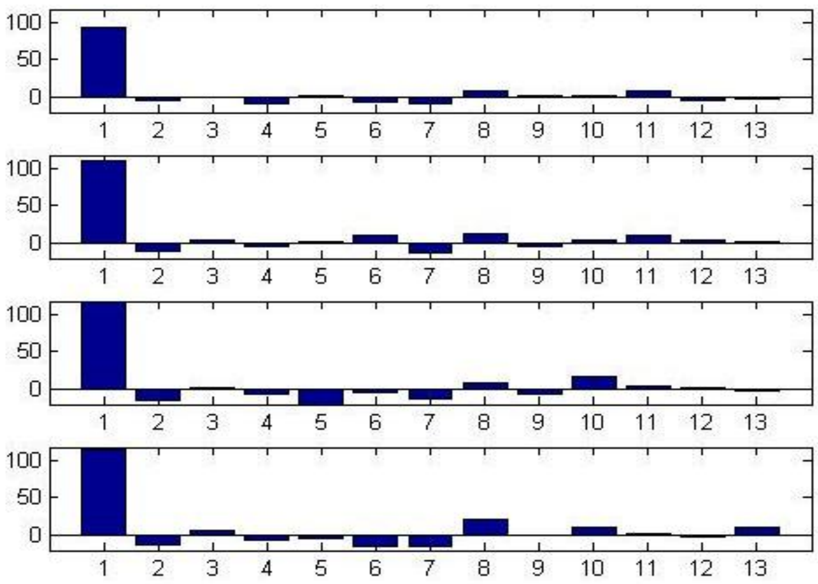
~450 signal segments
per condition signal

For each segment

- Extract n MFCCs
- Extract n HFDs
- Calculate Kurtosis

Data Analysis/Feature Extraction

Mel Frequency Cepstral Coefficients (MFCC) of first segment



Normal condition

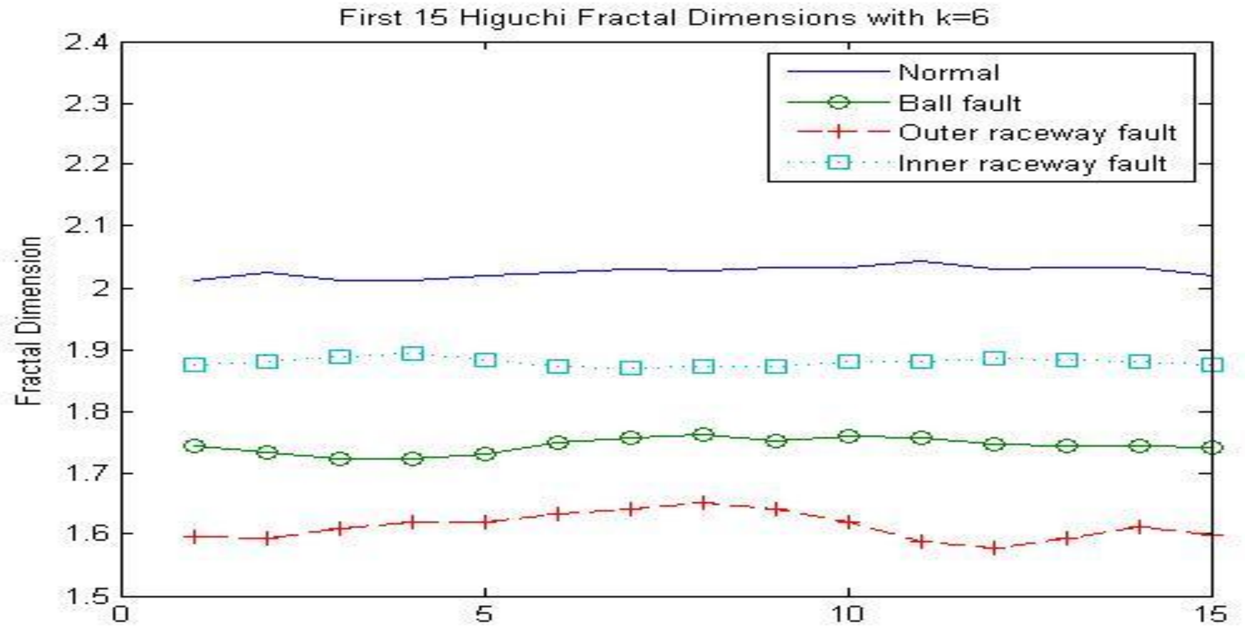
Ball fault

Inner raceway fault

Outer raceway fault

Data Analysis/Feature Extraction

Higuchi Fractal Dimensions (HFD)



Data Analysis/Feature Extraction

Feature data set construction

	$MFCC_1$	…	$MFCC_{13}$	HFD_1	…	HFD_{13}	$kurtosi$
Feature Vector 1	###	###	###	###	###	###	###
…	###	###	###	###	###	###	###
Feature Vector N	###	###	###	###	###	###	###

1 feature data set per condition signal

Decision Making/ OC-training

Training with normal features only

K-Center-dd

K-Means-dd

NN-dd

SV-dd

Parzen-d

SOM-dd

RF-dd

Trained models

Evaluation

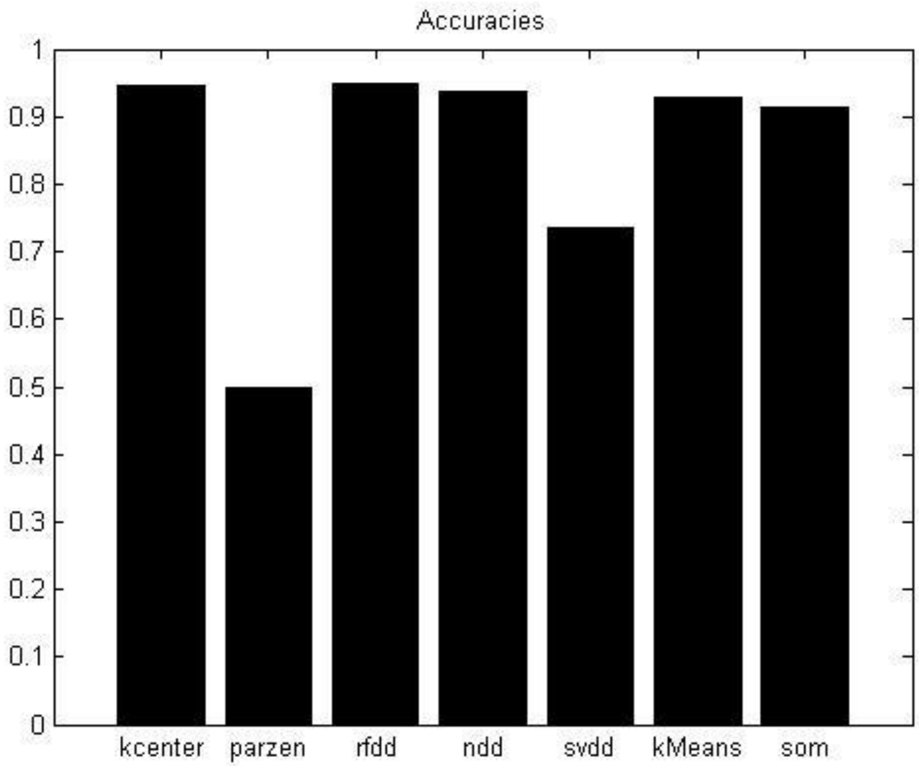
Test set consisting of normal features (not used for training) and features of each fault condition (sampled)

Trained classifiers

Classification of test set and calculation of

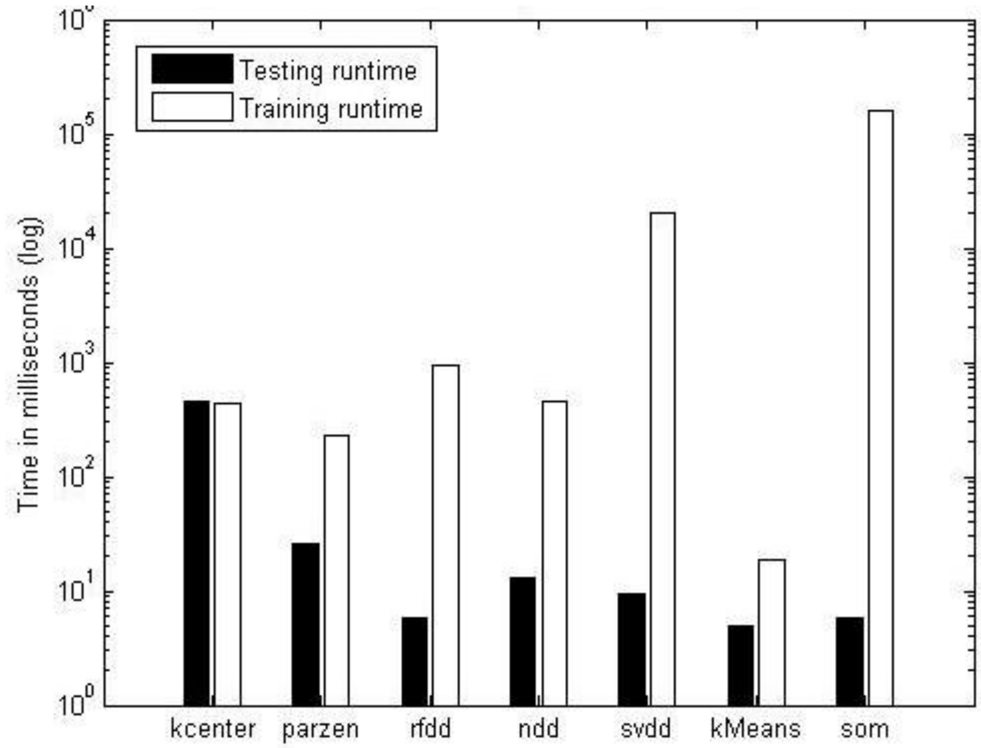
- Fraction of rejected normals (Error I)
- Fraction of accepted outliers (Error II)
- Accuracies

Evaluation



Accuracies
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
kcenter
parzen
rfdd
ndd
svdd
kMeans
som

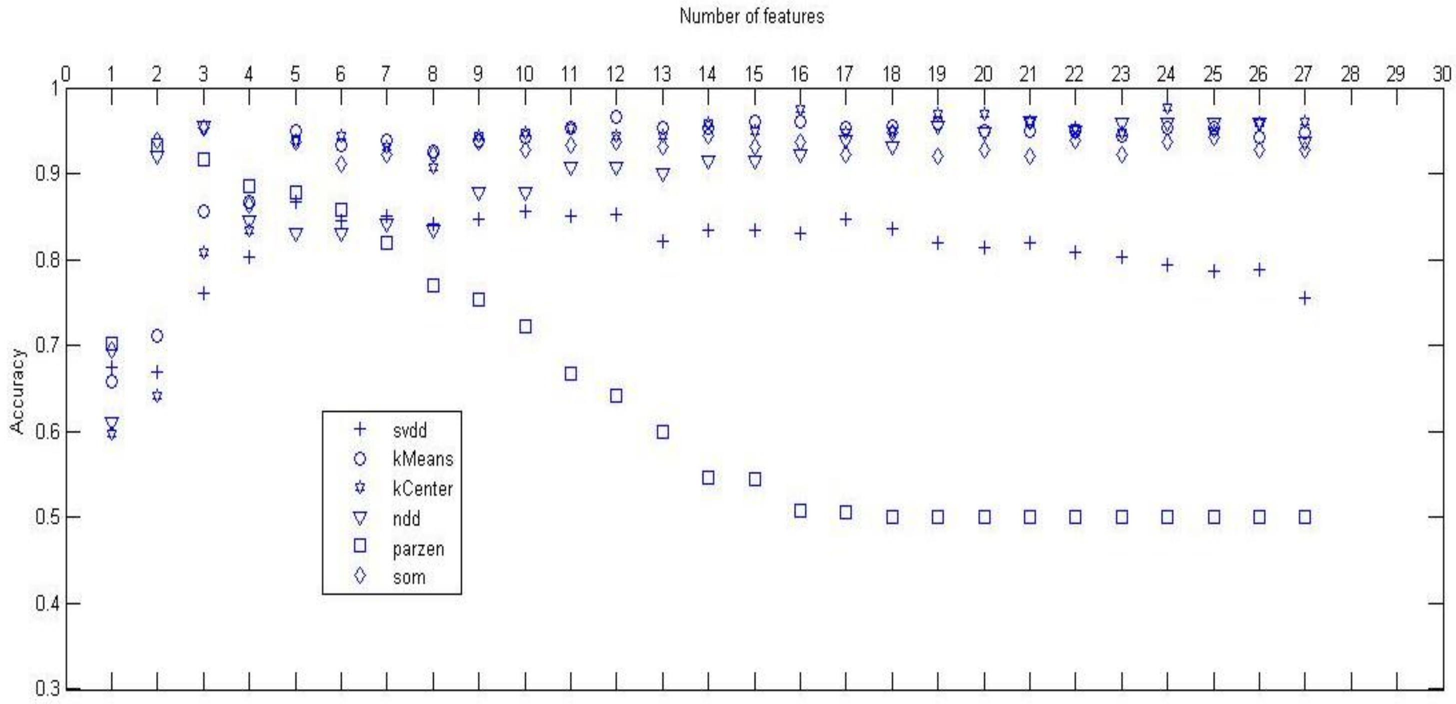
Evaluation



Testing runtime
Training runtime
Time in milliseconds (log)
kcenter
parzen
rfdd
ndd
svdd
kMeans
som

Evaluation

Feature reduction with random forest variable importance measure



Conclusions

- Altogether good results with the combination of feature extraction methods and one-class-classifiers for the roller bearing dataset
- excellent results with the new random forest one class classifier concept
- huge differences in training times (not that important in real world applications)
- underperformers: SVDD (w.r.t. time and accuracy), Parzen Window (worst classifier)

Outlook

- Application of One-Class-Classifiers in Ensemble Methods (Batch and Ada Boost)
- Calculation of several Random Forest class centers/prototypes representing the target class (based on proximity matrix)
- Application of One-Class-Classifiers in semi-supervised setting

End

Thank You!