HERCULES-2 Project

Fuel Flexible, Near Zero Emissions, Adaptive Performance Marine Engine

Deliverable: **D6.1**

Study the result quality of existing subspace-search methods on uncertain data

Revision Final

Nature of the Deliverable:	Report
Due date of the Deliverable:	01/05/2016
Actual Submission Date:	01/03/2016
Dissemination Level:	Public

Contributors:	Georg Steinbuss (Karlsruhe Institute of Technology) Tim Nusch (Karlsruhe Institute of Technology) Prof. Dr. Klemens Böhm (Karlsruhe Institute of Technology) Dr. Florian Plentiger (MAN Diesel & Turbo SE, Augsburg)
Work Package Leader Responsible:	Dr. Mathias Moser (MAN Diesel & Turbo SE, Augsburg)

Start date of Project: 01/05/2015 Duration: 36 months

Grant Agreement No: **634135-HERCULES-2**

HORIZON 2020

The EU Framework Programme for Research and Innovation

TABLE OF CONTENTS

1 Executive Summary

The mechanical engineering industry shows an increasing interest in data-driven approaches which detect faulty system states early. The detection of faulty states in high-dimensional data sets describing the system for predictive maintenance is an open problem. This project investigates the use of subspace search methods in this setting. Subspaces are combinations of data attributes, e. g, one sensor group. The focus of this project are high-contrast subspaces [19], which require the respective data attributes to be correlated. Outlier- as well as change-detection algorithms typically benefit from such high contrast subspaces.

In a first step we have analysed and prepared the available data for machine learning algorithms. The data was compressed lossily. This means that data values were missing. In total only 2 % of the data was directly available. The compression technique used however ensures that no lost observation differs more than a sensor-specific threshold from the others. As our predictive models needed full data access, we had to replace the missing values prior to any further analysis.

Next in order to assess the quality of different prediction frameworks, there has been a need of an event to predict. As machine failure was not available within the provided data, we have chosen sensor-specific alarm thresholds. Unfortunately, violations of these thresholds were too rare as well. Hence we introduced *near alarms* by lowering the alarm threshold just somewhat. Of course machine failure cannot be compared to predicting a threshold violation of a known sensor. Thus, we decided to undertake any further modelling using surrogate data. To increase the use of this, we predict prior to the actual violation. All in all, we arrive at a useful estimate of the result quality for predicting machine failure.

In the next step, we develop and evaluate prediction frameworks in order to quantify the gain in result quality by using subspace search. Out of seven frameworks developed and studied, the one giving way to the best results uses change detection scores within the previous mentioned subspaces of high contrast. We measured the quality of each framework with multiple metrics useful in a predictive maintenance setting. As expected, accuracy on its own is not able to reflect the predictive power of a framework. Since threshold violations are rare, a framework never predicting any alarm would sustain a very high accuracy.

The evaluation indicates the potential of high-contrast subspaces in order to predict machine failure. In the remaining project duration we will try to further improve result quality while increasing the size of the predictive window. Possible next steps could be to investigate different types of classifiers and to evaluate the impact of data quality on result quality.

2 Introduction

2.1 Problem Definition and Objectives

Maintenance activities constitute a substantial portion of the overhead costs in several industries [9]. Wireman [30] has found that the maintenance costs for American industrial companies have increased by about 10-15% per year since 1979. Because of that, industries and scientists nowadays devote much time and effort to the early detection of faulty states. In many scenarios, sensor measurements are the basis for this detection. As the number of sensors grows, the resulting data sets grows in dimensionality. Thus, most datasets of this kind contain up to hundred dimensions. It is an open problem to *detect faulty states for predictive maintenance*, especially in such *high-dimensional data sets*.

One issue when it comes to high dimensional datasets is the *curse of dimensionality*. It arises as with increasing dimensionality, the distance of data points becomes meaningless. This causes problems when the machine learning algorithms used are based upon distances.

Another issue reading high dimensional datasets is that such datasets often contain a vast amount of multivariate relationships between attributes. Patterns, especially *changes*, might be crucial in order to detect faulty states. In general a *subspace* is defined as a set of attributes. *Subspace Search* aims at finding subspaces consisting of related attributes. However it is neither trivial to define interesting relations between attributes nor to find them in a high dimensional dataset. For example a dataset with 5 attributes inherits 31 possible subspaces, a dataset with 10 attributes already inherits 1023 possible subspaces. Hence, algorithms to detect certain subspaces must search the set of possible subspace intelligently, in order to keep runtime at a reasonable level. Subspace Search is a relatively new approach and thus has only been applied to few domains, according to our knowledge. This project investigates their use in a predictive maintenance scenario.

Besides being high dimensional, monitoring data usually is very large. If we monitored a system only every second, we will have gathered about 2.500.000 observations within a month. Apart from a runtime problem, storing and transferring this data can be expensive. Hence, in most applications data is compressed prior to be available for analysis. With compression, data can be lost in order to reduce the data size significantly. This makes available data *uncertain* in the sense that some data values are only estimated and not observed.

Having the issues mentioned above in mind, the project objectives can be summarized as follows:

1. Prepare the datasets for machine learning algorithms
2. Use various Subspace Search methods:
   i. Identify observations which contradict the general structure of the data (outlier)
   ii. Evaluate several models to predict failure
3. Study the relationship between data quality and prediction outcomes
   i. Evaluate quality of compressed or transformed data
   ii. Develop compression techniques which maintain the quality
4. Design and implement prediction framework tailored to the MAN use case

2.2 Related work

Evaluations of approaches for predictive maintenance, change detection and subspace search have been presented in a number of research papers. However, these approaches have not been applied in combination to the problem described in the previous section. In the following, we present related scientific literature on those topics.

There are various approaches to perform predictive maintenance without using any change detection technique. Some of these approaches use methods from digital signal processing, e.g., wavelet filters [23], to detect faulty states on vibration waveform sensor data. Others use Kalman filters [28] to track changes for vibration, frequency and other waveform features. Lall *et al.* [21] analyze the time evolution of the frequency content of signals using Fourier transformation techniques. All these approaches are useful for analyzing the special characteristics of waveform sensor data. But they do not detect changes in a generic manner.

Eklund *et al.* [12] use neural networks and the rank permutation transformation to detect faulty states in aircraft engines. The rank permutation transformation is a simple change detection technique which is robust to outliers and noise. However, the authors apply the rank permutation transformation on each dimension in the full data set individually without reducing the dimensionality beforehand. This approach does not scale well for high-dimensional data sets and is unable to detect multivariate changes in the data.

Chen *et al.* [7] propose a novel algorithm for the detection of contextual time series changes. The behavior of target time series is compared to their context, a group of other dimensions in the data set, the dynamic peer group. The authors show that their approach indicates new types of events compared to traditional change detection approaches. However, the paper focuses on univariate time-series analysis, and the approach is therefore unable to detect multivariate changes in data sets.

Hu *et al.* [17] investigate different univariate and multivariate change detection techniques for engine fault diagnostics. For univariate change detection, the rank permuation techniques is compared to statistical approaches like the likelihood ratio test. The authors also compare several multivariate techniques like auto-associative neural networks, physics-based model approaches and multivariate change detection based on Hotelling's T-squared statistic. The results showed that Hotelling's change detection performs well on the data set and considerably better than univariate approaches. However, the authors computed the T-squared statistic using the full data set without reducing the dimensionality beforehand. This is not meaningful for high-dimensional data sets.

Skubalska [27] presents an approach for multivariate change detection in high-dimensional data streams. The approach is also based on Hotelling's T-squared statistic. As method for dimensionality reduction, random projections of dimensions are used. Data from the resulting random subspaces are then presented to the change detection algorithm. The results from multiple instantiations are later combined into an overall score. This approach is able to work when high-dimensional data is present but suffers from the fact that a large number of subspace samples is needed to obtain robust results.

Keller *et al.* [19] propose a novel approach for selecting meaningful subspaces, or combinations of dimensions, from the full data set. Subspaces are selected using a contrast measure. The underlying assumption is that rare events such as outliers are statistically more likely to appear in high contrast subspaces. The approach presented in the paper proves to select low-dimensional subspaces where outliers are more apparent in comparison to searching in the full data set or in random subspaces, as proposed in [27]. The high contrast subspace approach is intended as a pre-processing step for density-based outlier detection. Other usages for this approach, such as a preprocessing steps for supervised or unsupervised learning techniques have not been investigated, at least to our knowledge.

3 Background

In this entire section we discuss established methods for data analysis in some detail. This detail is needed as in Section 4 we employ these methods in order to assemble our prediction frameworks. Then we evaluate these prediction frameworks in our predictive maintenance use case with a ship engine.

3.1 Tree-based Data Mining Algorithms

Concepts such as decision trees are widely used in data mining for predictive modelling. Tree-based models recursively partition the predictor space into regions. The set of splitting rules is then called decision tree. The notion **C**lassification **a**nd **R**egression **T**ree (**CART**) was first introduced by Breiman *et al.* [6] and is used as an umbrella term for tree models for both categorical and continuous predictions. Tree models where the categorical class membership of objects is predicted are called classification trees. Such models where the target variable is continuous are called regression trees. Decision trees have several advantages such as comprehensibility, scalability and the ability to work with both numerical and non-numerical data. This makes decision tree models a popular choice for predictive modelling.

Classification Trees usually use the Gini impurity as splitting criterion. Assuming a classification Task wit K distinct classes, where each class occurs with probability p_i $i \in \{1, \ldots, K\}$, the Gini impurity (G) is defined as

$$G := \sum_{i \neq j} p_i p_j = \sum_{i=1}^{K} p_i(1 - p_i) = 1 - \sum_{i=1}^{K} p_i^2$$

Note that always $\sum_{i=1}^{K} p_i = 1$. The measure can be interpreted as the total variance across all classes [18]. If most p_i's are close to either 0 or 1, the Gini impurity takes on a small value. Thus, the measure indicates if nodes have predominantly observations from a single class [18]. – Figure 1 illustrates the use of a fully trained tree: It is easy to follow the decisions and to understand predictions made.

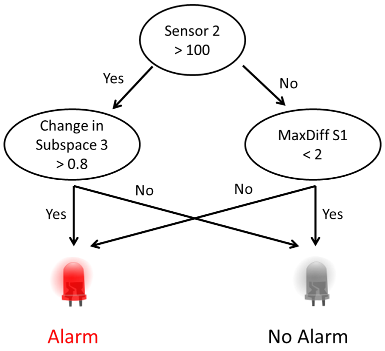


Figure 1: Exemplary decision tree

Classification and Regression Trees have the advantage that they are both understandable without substantial knowledge of data mining tools and useful to generate workable rules. Such rules could be used in technical systems as dynamic thresholds for monitoring purposes or as parameters for system operations. This advantage is especially important for the given use case; black-box models which would be the result of other classification approaches such as neural networks can not be operationalized.

The construction of efficient Classification and Regression Trees is known to be NP-hard [4]. Because of that, most algorithms are based on heuristics. Variations of classic tree-approaches such as the dual information distance tree were proposed to approach this problem [4]. In some situations, decision trees are more likely to overfit to the training data in comparison to other approaches. Mechanisms such as tree pruning help to create decision trees which generalize well from the training data.

3.2 Ensemble Learning

Ensemble learning is an umbrella term for methods that combine multiple prediction models to construct more powerful models. The prediction models can be obtained from any learning algorithm. There are multiple approaches to combine the models, the most popular ones being bagging and boosting.

Bootstrap aggregation or bagging is a procedure to reduce the variance of a statistical method [18]. Especially tree-based methods suffer from high variance. If a training set is split into two parts, decision trees trained on the different halves can be very different. This might cause unwanted effects because the results highly depend on the selected data partition. The idea of bagging is to take repeated samples from the data, train a classifier on each sample and average their predictions for a final result. Bagging has been demonstrated to give improvements in accuracy by combining hundreds or more models into a single procedure [18]. Although we do not use bagging in its original form, the random subspace approach described in 3.3.2 can be seen as an analogon where data attributes instead of observations are bagged.

Like bagging, boosting is a general approach to combine multiple models for classification or regression. Boosting uses a sequential procedure to construct powerful ensembles. Each new model is trained with emphasis on training instances that were misclassified by previous models. Boosting has shown to yield better results than bagging, but the sequential approach tends to overfit the training data.

3.3 Subspace Search

Outlier detection can be seen as a generalization of change detection. While change detection techniques identify abrupt changes in time series, outlier detection techniques search for data points which differ considerably from the remaining data set and are also applicable to data sets without temporal components. They share plenty of characteristics, such as their sensitivity to a large number of data attributes. That is why, although the following sections aim at outlier detection, most issues mentioned and solutions also apply to change detection.

3.3.1 Many Attributes

Most issues with plenty of data attributes can be summarized with the term *curse of dimensionality*. The curse of dimensionality arises when data in high-dimensional spaces is organized or analysed. There are numerous phenomena in different domains, such as in sampling, optimization or in data mining which can be summarized as the curse of dimensionality. The term has first been coined by Bellman in 1957 [3] and is still of high importance.

In many data mining algorithms, for example in many outlier detection algorithms, the similarity of data points has to be calculated in some way. Algorithms which calculate similarity or distance between data points usually suffer from the curse of dimensionality. Under certain general preconditions discussed in Beyer et al. [5], the distance between the farthest and the nearest points in a sequence of random points decreases with the dimensionality and converges to zero. The theorem explains why nearest neighbour approaches, which are often used for outlier detection, fail to produce meaningful results. The increasing dimensionality leads to data points which are equidistantly distributed in the high-dimensional space, as illustrated in Figure 2.

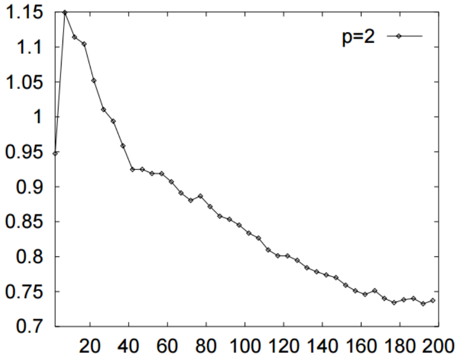


Figure 2: $|D_{max} - D_{min}|$ for increasing dimensions using the L_3 metric [15]

Subspace search refers to the selection of specific subsets of dimensions in which outlier detection techniques try to detect outliers. This does not just reduce the dimensionality but has also the property of allowing us to find outliers specific to a certain subspace. The idea of subspace outliers assumes that some outlier might only behave as an outlier in a very small subset of attributes, rather than all attributes. For example in the scenario of engine monitoring, an anomalous value in only Sensor X and Sensor Y might indicate the failure of a certain part of the engine, while anomalous values in all sensor might only arise from an unusual engine control, e. g. a full stop. Although both kind of outliers might be interesting to investigate, only the first are non trivial to find. For the detection of these kinds of outliers we examine current subspace search methods. There are some outlier detection approaches which use an integrated processing of subspace search. However, the decoupling of subspace search and outlier detection proposed in [19] is necessary for us, as we want to do change detection instead of outlier detection. Two approaches which use a decoupled processing are the random subspace approach [22], which selects several subspace projections randomly, and high contrast subspaces [19], which selects subspaces by measuring the contrast of subspaces.

3.3.2 Random subspace approach

The random subspace approach [22] is a simple technique to select subspace projections. Outliers can then be detected in the resulting lower-dimensional subspaces of the full data set. This feature bagging method is similar to the bagging ensemble technique often used in supervised learning.

The basic idea of the approach is that subspaces are randomly selected from the original data set D. The procedure runs in a series of multiple rounds T. In every round t, an outlier detection algorithm is called on the randomly selected set of features. It is possible to use a different outlier detection algorithm in each iteration. The number of features or the subspace dimensionality is chosen randomly between $(d/2)$ and d for each round. The features are randomly selected from the original feature set without replacement and create a data set D_t in the tth iteration.

In the original paper, the resulting subspaces are presented to the outlier detection algorithms during each round, and the results of each outlier detection step are combined. As mentioned previously, this decoupled and generic two-step approach enables researchers to use the random subspaces for other applications as well. A huge disadvantage of the random subspace approach is that irrelevant subspaces blur the potentially good results of randomly picked highly relevant subspaces.

3.3.3 High contrast subspaces

The idea of high contrast subspaces (HiCS) has been first proposed by Keller et al. in 2012 [19] and aims at selecting meaningful subspaces from the full data set using a contrast measure. As opposed to the random subspace approach [22], where a large number of potentially noisy subspace samples is needed to obtain robust results, the high contrast subspaces approach only selects a small number of high contrast subspaces. The assumption is that rare events are statistically more likely to appear in high contrast subspaces with a non-uniform data distribution.

High contrast subspaces have the characteristic that outliers can be well distinguished from other (regular) objects within the subspace. The approach aims at selecting such subspaces as a pre-processing step for density-based outlier detection. Density-based outliers are outliers that have low densities compared to their local neighbourhoods [19]. Many outlier detection techniques use an integrated processing of subspace search and outlier detection [13, 20]. One contribution in that publication is the decoupling of subspace search as a preprocessing step for outlier ranking. The advantage of this decoupling is that both steps can be treated as independent problems, and algorithms can be combined in a modular fashion [19]. It also allows us to use change detection instead of outlier detection within the subspaces.

The algorithm searches for subspaces based on a novel contrast measure. In short, the contrast of a subspace is based on the deviation between the conditional probability densities and the (marginal) densities for all attributes of the subspace. Hence, highly correlating features will also have a high subspace contrast. This defines multivariate correlation as the relevant relation in between attributes. In order to achieve high statistical precision, a large number of different tests are performed by the algorithm. There is a toy example from the publication displayed in Figure 3, which illustrates the contrast measure.

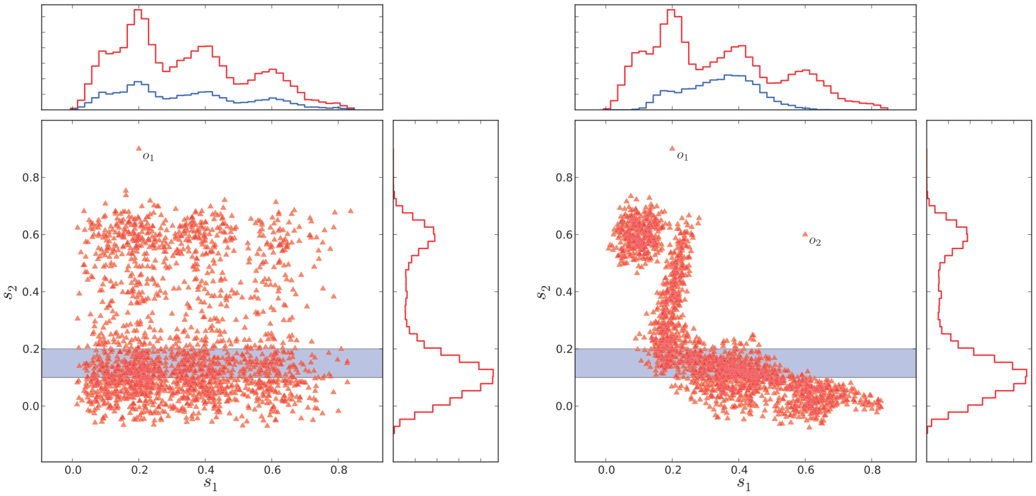


Figure 3: High vs. low contrast subspaces [19]

Both two-dimensional data sets exhibit the same one-dimensional projections. In the first data set, on the left side, both attributes are uncorrelated. The two-dimensional subspace is filled with objects which are consistent to the marginal distribution. The outlier o_1 is a trivial outlier. This means that it could easily be detected by analysing the one-dimensional distribution of s_2. In the second data set, on the right side, attributes are correlated. This reveals empty regions which are undetectable when considering only the one-dimensional distributions. The outlier o_2 is hidden in the one-dimensional distributions of s_1 and s_2 and only visible in the two-dimensional subspace. This example illustrates why subspace search approaches are so important when searching for outliers. Only when the subspace contrast is analysed and this specific subspace is examined, outlier o_2 and other non-trivial outliers are revealed in higher-dimensional projections.

3.4 Change Detection

The identification of abrupt changes is a very important topic in the area of time series analysis. The main assumption of time series analysis is that the properties describing the data are either constant or only slowly time-varying [2]. However, in many practical situations, abrupt changes are quite common and lead to problems when applying traditional time series analysis methods. Following the definition of Basseville et al. [2], an abrupt change is a time instant at which a change in a property occurs very fast or instantaneously in comparison to the sampling period of the measurement.

The basic idea of *change detection* is to detect such abrupt changes in the data. Change detection techniques are broadly used in quality control and condition monitoring, in signal processing and in failure detection. Other changes, such as slowly changing level shifts, can also be of high interest, but are not targeted by change detection.

Many different approaches to detect abrupt changes have been published, elementary ones like the cumulative sum (CUSUM) algorithm [26] or approaches based on the Hotelling statistic. Other approaches like the Bayesian Online Changepoint Detection [1] follow the Bayesian approach or utilize support vector machines such as the Online Kernel Change Detection algorithm [10].

The basic performance metrics for evaluating change detection algorithms are the following ones:

- False alarm rate (Type I error rate)
- Misdetection rate (Type II error rate)
- Detection delay

Every change detection algorithm must make a trade-off between these metrics. Additionally, robustness with respect to noisy data and to modelling errors is another important characteristic when choosing a change detection algorithm. Currently, a popular change detection approach based on Hotelling's statistic is used in this project. This approach can detect abrupt changes in univariate and multivariate data. As discussed earlier, multivariate methods have shown to yield better results than univariate approaches for fault diagnostics [17].

Hotelling's T-square statistic was first proposed by Harold Hotelling [16]. It is used in multivariate hypothesis testing, and it is a multivariate extension of the Student's t statistic. Basically, a test for the plausibility of a multivariate vector for a normal mean vector is performed. The T-square statistic is often used in control charts [24] [29], where the mean vector can be monitored to provide diagnostic information, prevent defects and to improve the quality.

The observation is a multivariate time series vector in the form of

$$X(t) = (X_1(t), X_2(t), ...X_m(t))^T$$

consisting of m variables at time t. The baseline projection is a random sample of a multivariate random variable which follows a multivariate normal distribution. Both the mean vector μ_b and the variance-covariance matrix $\sum$ of the baseline are known. From the observation, only the variance-covariance matrix $\sum$ is known, but the mean vector μ_o is unknown. The assumption is that both the observation and the baseline projection follow normal distributions. The null hypothesis H_0 is that the observation X(t) is not statistically different from the sample population drawn from the data. The alternative hypothesis H_1 is that the observation is not distributed as the baseline. Formally, the null hypothesis $H_0 : \mu_o = \mu_b$ is tested against the alternative hypothesis $H_1 : \mu_o \neq \mu_b$.

The T^2 test statistic is based on the sample mean vector $\overline{X}$ and the sample variance-covariance matrix W, where

$$\overline{X} = (\overline{X}_1, \overline{X}_2, ...\overline{X}_m)^T$$

and

$$W = \frac{1}{(n-1)} \sum_{t=1}^{n} (X(t) - \overline{X})(X(t) - \overline{X})^T$$

are used to estimate μ_b and $\sum$. Note that it is assumed that there are n data samples. The T^2 test statistic for the observation $X(t)$ is calculated by

$$T^2 = (X(t) - \overline{X})^T W^{-1} (X(t) - \overline{X}).$$

A large value of T^2 indicates a high possibility to reject the null hypothesis but also a small possibility to commit a Type I error. Small T^2 values indicate a high possibility of failing to reject the null hypothesis. The T^2 test statistic is closely related to the Mahalanobis distance [8], measuring the distance from a point to a distribution. Here, this distribution is the multivariate normal baseline projection. This relation allows us to visualize the T^2 test statistic.

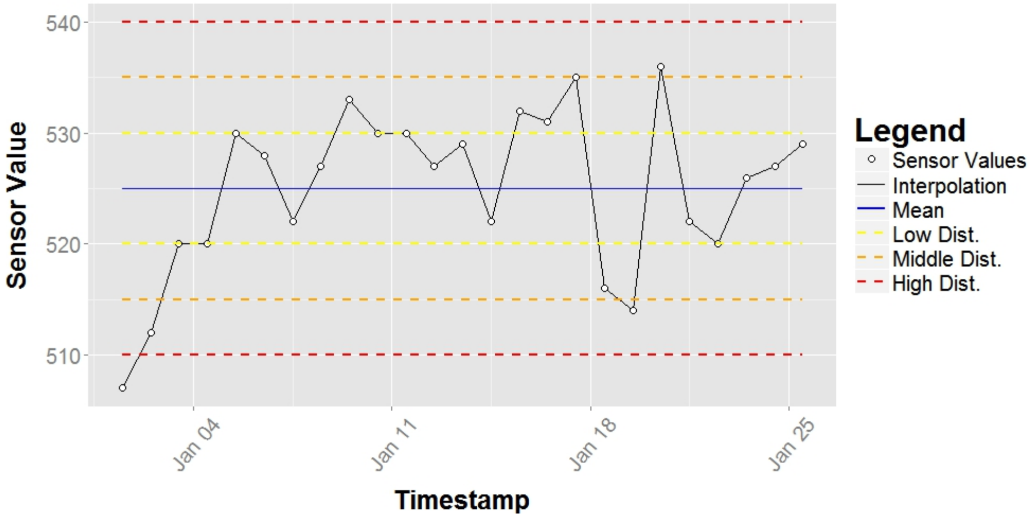


Figure 4: Distance from baseline projection regarding one dimension

Figure 4 and 5 display low, middle and high distances to the baseline projections. While Figure 4 displays the rather simple relation of distance to the mean in single data attributes, Figure 5 reveals the true potential of T^2 test statistics in multivariate subspaces. The ellipsoidal shape of the different Mahalanobis distance levels capture the data structure much better than simple circles around the mean. This circle represents a fixed Euclidean distance around the mean. We could assume a circle shape of the data if all attributes within a subspace were independently distributed following a univariate Gaussian distribution. However, we want to use change detection within high contrast subspaces, which are by definition spanned by correlated

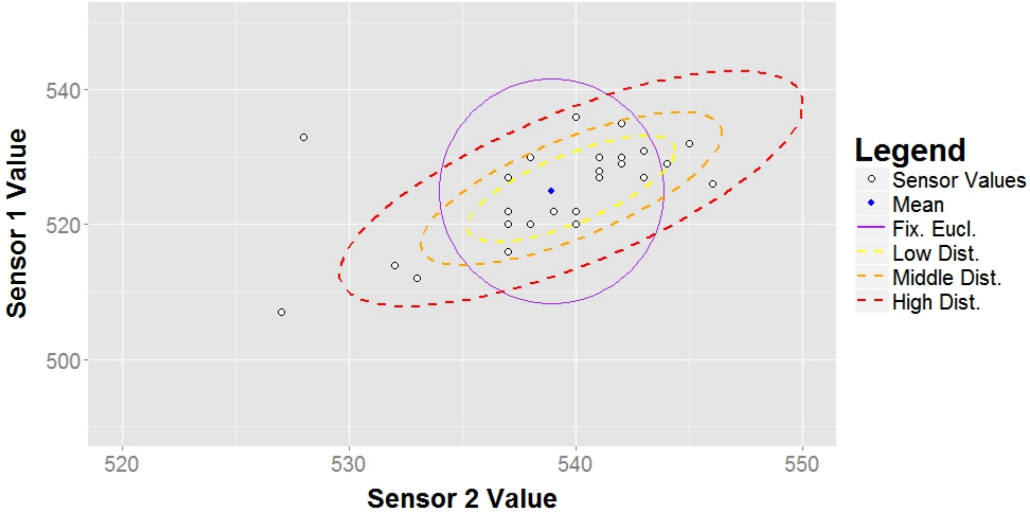


Figure 5: Distance from baseline projection regarding two dimensions

attributes. Summarized *Hotelling's T-square* control charts allow us to identify abrupt changes within both multivariate subspaces as well as univariate attributes. Once all parameters for the baseline projection are fitted, this allows us a very fast data handling, contrary to many other outlier detection methods. Unfortunately, *Hotelling's T-square* statistics requires Gaussian distributed data to be accurate. When violating this assumption, results could be meaningless.

4 Development of the Prediction Model

In this section, the development of different frameworks for the prediction model is described. This section also describes the data set provided and both the preprocessing and the feature engineering steps used.

4.1 Data – introduction

This section gives an introduction to the data which his currently in use for this project. The data sets were provided by MAN Diesel & Turbo SE and consisted of sensor measurements of one specific diesel engine during one year. While the engine operated, both the actual measurement and the measurement time were recorded. A separate measurement file was created for each of the 79 sensors.The physical quantities which were measured by the sensors are temperature in Kelvin, pressure in Pascal, frequency in Hertz, power in Watt and kinematic viscosity in m^2/s. Several sensors measured different physical quantities. Hence it is crucial to scale the data in order to compare attributes. Sensors which measured the same property at different positions could be logically grouped together to sensor groups. Twelve groups of sensors emerged which we could use for feature engineering later. In order to associate the different sensor measurements with each other we used the timestamp as join criterion.

By definition, a supervised learning task is based on data containing ground truth, at least for some training data. The ground truth, or label, holds the information which data point belongs to which class. This ground truth is essential in order to assess the result quality in our predictive maintenance setting. The actual number of instances with an alarm, which is the ground truth here, has been too low to train classifiers. This problem is known as the rare event [11, 14] or class imbalance [25] problem. We have mitigated the ”rare event” problem through the definition of near-alarms with a reduced threshold. The number of instances was not to increase too much, as alarms should mark emergency states which are rare by nature. Having said this, after reducing the alarm threshold, the second most frequent alarm was chosen as ground truth. Advantages of this alarm are that it is rare enough to be compared to an emergency state, and it is relatively uniformly distributed within the data. The prediction forecast was set to 30 seconds, which corresponds to a prediction of an alarm in $t + 30\ seconds$, with attribute values at time t.

4.2 Data preprocessing

In order to be able to find potential relationships between the measurements, we generated one integrated data set from the data of each individual sensor. Because of varying measurement periods and the compression of the measurement files, the integrated data set was very sparse. Only 2% of the measurements in the integrated set were non-empty and directly available for analysis.

A first simple strategy for handling the sparse and incomplete data set is to discard incomplete data points. However, if applying the strategy to the data, 99,9% of the data points are discarded and removed. The remaining data points are not a sufficient sample size for further analysis. Aggregation strategies which condense consecutive measurements have the advantage that the time granularity is fully adjustable. Values can be aggregated by minute or any other time period with aggregation functions such as mean, median or maximum. This aggregation strategy basically leads to a more complete data set. However, interesting data points such as outliers get flattened out and disappear completely.

Interpolation strategies keep all known data points and only impute the missing data points with regard to an interpolation method. Interpolation methods can be simple approaches such as linear interpolation, piecewise constant interpolation or more complex methods such as polynomial or spline interpolation. Due to a linear nature of the compression technique, we have chosen *linear interpolation* as the strategy to deal with the incomplete data set. It is guaranteed that the interpolated values do not differ more than the sensor-specific threshold from the actual value when linear interpolation is used.

Another part of the preprocessing was to remove the attributes which are directly connected to the outcome which shall be predicted. As described earlier, in this case the outcome is the alarm corresponding to one specific sensor. Thus, we removed the sensors which measure this property directly from the data set. This results in surrogate data. As we see it, this is quite important as most machine failures could also be predicted using only surrogate data.

4.3 Feature engineering

The process of feature engineering is a crucial step in data mining. Feature engineering generates one or more features which can be calculated from the raw data in some way. Feature engineering can be hardly automated because the number of possible features is boundless, and it is impossible to know the good representations for the given problem definition a priori. Although additional features increase the data dimension further, they are often more useful for the classifier than raw measurements. As in the context of this project, subspace search allows us to handle very high dimensional data. Thus any helpful additional features should be added without regarding the resulting dimensionality.

Currently, we have extracted static and dynamic time-based features. These features have been useful for the predictive models to different extents. Having said this, all approaches are presented in this section.

4.3.1 Static features

Static features are features which are neither time-based nor generated by change detection algorithms. Additional features change the way the data is presented, which could directly influence the predictive models. Raw data such as the sensor measurements can be transformed to features which better represent the characteristics of the data to the predictive models.

Seven different approaches resulted in 18 additional static features which we have added to the data set. The features are displayed in Table 1.

Table 1: Extracted static features

Name	Category	# of Features	Range
Maximal deviation in sensor group	Domain	13 (1 per sensor group)	$[0 .. \infty]$
Current operating time	Domain	1	$[1 .. \infty]$
Sensor uncertainty	Missing values	1	[0 .. 1]
Hour	Date / Time	1	[0 .. 23]
Weekday	Date / Time	1	[0 .. 6]
Month	Date / Time	1	[0 .. 11]

- **Maximal deviation in sensor group**: This feature describes the maximal (absolute) deviation in a sensor group. Similar values in a sensor group are represented by low values, and high values point to big differences or outliers.
- **Current operating time**: This feature represents the time (in seconds) since the last engine start. An engine start can be identified by the generator power of the engine.
- **Sensor uncertainty**: Due to data compression during the data collection, the gaps between known points have to be interpolated. This feature represents the normalized average gap size for all sensors. A value of zero means that all sensor measurements were known, and the (maximal) value of 1 identifies the data point with the biggest average gap or the most uncertain data points.
- **Hour**: The given timestamp in its native form, the ISO 8601 (i.e. ”2013-06-1990T12:03:62”), contains a lot of information, which leads to difficulties for training data mining models. This feature is the hour of the day. Daytime (e.g. 6-18) and nighttime (e.g. 19-23 and 0-5) can be distinguished with this feature if the time zone is known.

- **Weekday**: This feature is the day of the week. Weekdays (0-4) and weekends (5-6) can also be distinguished with this feature.
- **Month**: This feature is the month of the year. Seasons can also be distinguished with this feature.

4.3.2 Dynamic features

Dynamic features are features which are based on the behavior of the sensor measurement. We have tested two dynamic features:

- **Difference to previous value**: This simple approach is based on the difference between the current value and the previous value. It therefore maps the variances in the measurement directly to a feature. This feature is very sensitive to noise and abrupt changes.
- **Difference to moving average**: This approach is based on the difference between the current value and the average value of a predefined number N of previous values. The average of N previous values is calculated during each step. This approach is less sensitive to noise and abrupt changes.

As discussed before, the results of the change detection algorithms can also be used as dynamic features.

4.4 Proposed Frameworks

Currently, we have designed and developed seven frameworks for predictive maintenance. All frameworks are based on the same preprocessing methodology. The preprocessing steps have been discussed in the previous section and can be summarized as follows:

- Integration of the individual data sets into one data set
- Data cleaning to remove invalid data points
- Interpolation of the integrated data set to remove missing data points
- Removal of attributes which are directly connected to the outcome to be predicted

Additionally, we have noticed that most alarms occurred in the last third (4 million data points) of the data set. Because of that we chose only this part of the data set for the analysis. On top of the preprocessed data, we have added the static features from Table 1. The overall process is displayed in Figure 6. All seven framework alternatives work with identical data and generate a prediction based on a test set. The predictions are then compared with the actual ground truth in a second step in order to analyse which frameworks perform best.

4.4.1 BaselineSimple

The first strategy which serves as a baseline approach was named BaselineSimple. As all other approaches, the strategy works with the identical preprocessed data after the previously described data integration, data interpolation, data cleaning steps and CART decision tree classifier. We used the tree to predict the alarm value by learning decision rules inferred from the preprocessed data directly. In case of simple relationships between the preprocessed features and the target alarm values, this simple strategy will already perform well.

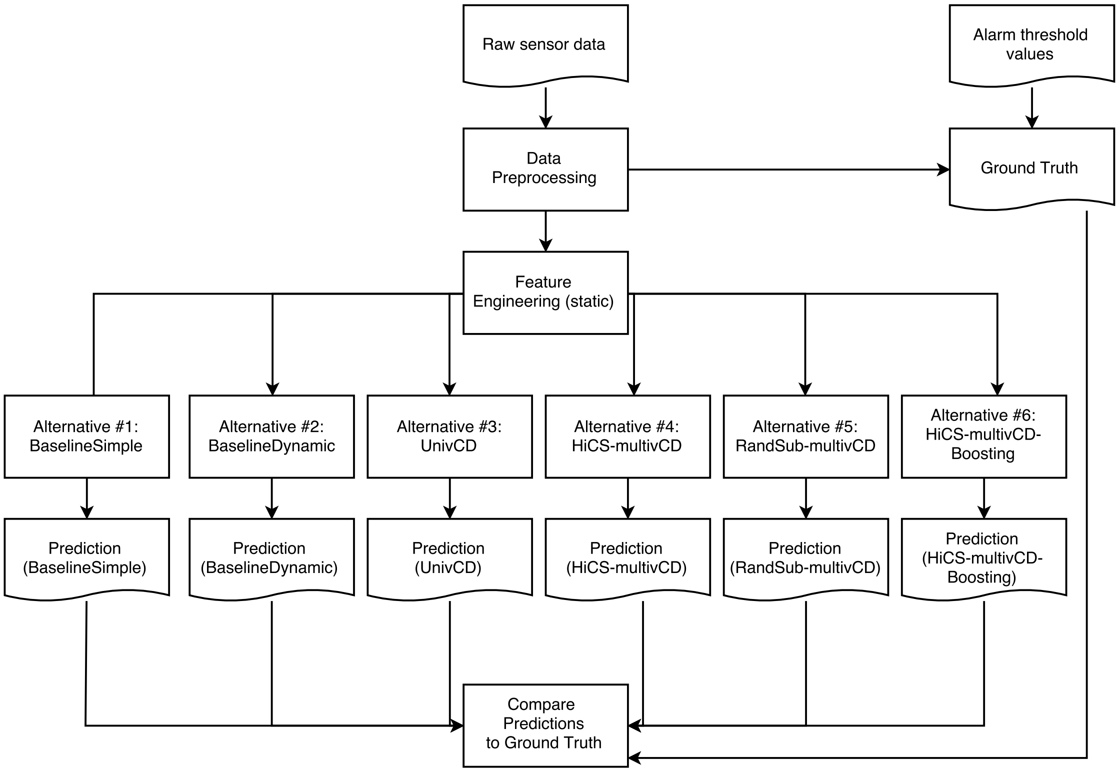

Raw sensor data
Alarm threshold values
Data Preprocessing
Ground Truth
Feature Engineering (static)
Alternative #1: BaselineSimple
Alternative #2: BaselineDynamic
Alternative #3: UnivCD
Alternative #4: HiCS-multivCD
Alternative #5: RandSub-multivCD
Alternative #6: HiCS-multivCD-Boosting
Prediction (BaselineSimple)
Prediction (BaselineDynamic)
Prediction (UnivCD)
Prediction (HiCS-multivCD)
Prediction (RandSub-multivCD)
Prediction (HiCS-multivCD-Boosting)
Compare Predictions to Ground Truth

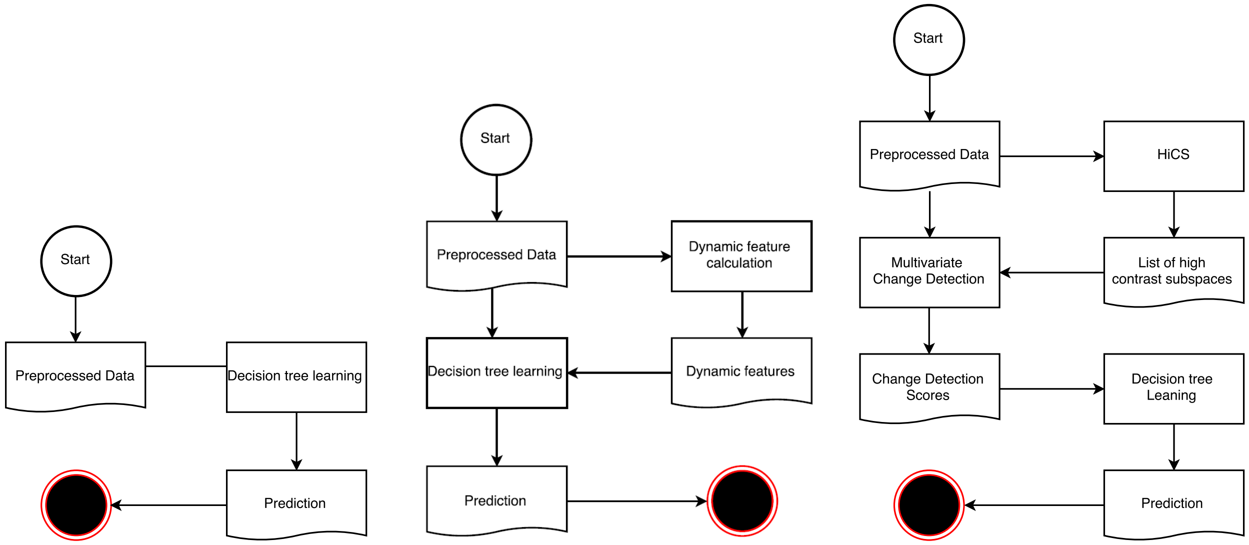
Figure 6: Overall process

4.4.2 BaselineDynamic

The second strategy, named BaselineDynamic, is an extension of the first strategy. It extends the data set of the first approach by dynamic features. The features are the differences to the previous value and the differences to the moving average. Both approaches map the variances in the sensor values to features but are differently sensitive to noise and abrupt changes. We have added dynamic features to the preprocessed data set and have built a CART decision tree with a parametrization identical to one of the first strategy. This strategy performs well in case of relationships between the target alarm values and the values of the dynamic features. A disadvantage of this approach is that the data set gets very wide – the number of features is nearly doubled. This could decrease result quality, as described in Section 3.3.1.

4.4.3 univCD

The univCD approach displayed in Figure 7c is a modification of the BaselineDynamic approach. It uses univariate methods for change detection to generate a change detection score for each feature in the preprocessed data set. As described in Section 3.4, the resulting scores help to detect abrupt changes in the data. A high value in the detection score of a particular attribute implies that an abrupt change with regard to that attribute occurred. Such abrupt changes might arise from faulty system states. Because of that possible relationship, we have added the detection scores to the preprocessed data set. Equally to the other approaches, we have built a CART decision tree on the resulting data set. This strategy has the same disadvantage as the BaselineDynamic approach, namely that the data set gets very high-dimensional (see Section 3.3.1).



(a) BaselineSimple (b) BaselineDynamic (c) univCD

Figure 7: Three frameworks

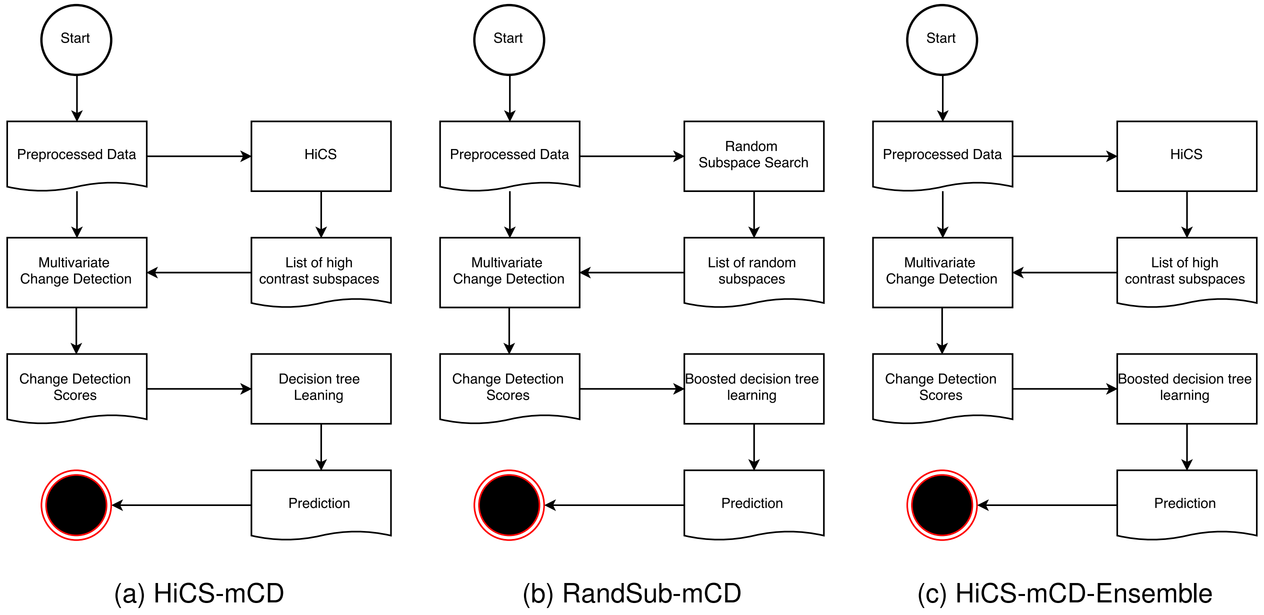
4.4.4 HiCS-mCD

The fourth approach, named HiCS-mCD, uses high contrast subspaces together with multivariate change detection. In a first step, we select high contrast subspaces (HiCS) from the preprocessed data set. As discussed in Section 3.3, the assumption is that rare events are statistically more likely to appear in such subspaces which have a non-uniform data distribution. The result of this step is a list of high contrast subspaces in descending order.

In a second step, we apply multivariate change detection to the data. In each of the top N subspaces, multivariate changes are identified using Hotelling's T-square statistic. We generated a score for each of the N subspaces. The resulting N score vectors form a new data set on which a CART decision tree is trained. One advantage of this approach is that the dimensionality of the final data set can easily be controlled by a parameter which minimizes the negative effects resulting from high-dimensional data sets. The algorithm can be summarized as follows:

```
Data: Time-series data set D; Number of top subspaces N
Apply HiCS to D with parameter N;
The output of HiCS is the list L of high contrast subspaces, sorted by contrast;
for $i \leftarrow 1$ to $N$ do
    Apply change detection algorithm CD by employing the subspace $F_i$ from L;
    The output of the change detection algorithm CD is the change detection score $S_i$;
end
```

Algorithm 1: HiCS-mCD



(a) HiCS-mCD (b) RandSub-mCD (c) HiCS-mCD-Ensemble

Figure 8: Three frameworks

4.4.5 RandSub-mCD

The RandSub-mCD approach is a modification of the HiCS-mCD approach. It is based on the Random subspace approach described in Section 3.3 and on multivariate change detection. Instead of using HiCS to detect high contrast subspaces, we used random subspace projections from the full data set. Here we applied multivariate change detection to the subspaces. Equally to the HiCS-mCD approach, the score vectors form a new data set on which a CART decision tree is trained. In order to get more robust results, we repeated the randomized process multiple times. The approach generates a dataset consisting of the multivariate change detection scores. One advantage of this approach is that this dataset is controllable in size. It is not necessarily high-dimensional.

4.4.6 HiCS-mCD-Features

This approach is an extension of the HiCS-mCD framework. Instead of using just the change detection scores as new data, we add these scores as features to the preprocessed data. Adding the scores as features could be beneficial when there is dependence between actual sensor values and alarms. These dependencies are lost when only using the change detection scores for predicting.

4.4.7 HiCS-mCD-Ensemble

The seventh approach is again an extension of the HiCS-mCD framework. It extends the previously introduced framework by a boosting technique. We trained multiple CART decision trees sequentially on the data set containing the score vectors of the multivariate change detection. The intention of using boosting as an ensemble technique is to improve the predictive power. However, boosted ensemble models tend to overfit when more models are combined. This is because it may reduce the ability to generalize.

5 Evaluation

5.1 Current Limitations

A limitation of this study is the definition of faulty states due to missing machine failure data. The data set provided only contains the sensor measurements and a list of alarm threshold values for some of the sensors. In order to assess the performance of a supervised learning model, data containing failures is required. Here, such data had to be generated using a combination of raw measurements and the corresponding alarm threshold values. This is a transformation of predicting machine failure into predicting alarm threshold violations. Obviously, with data on actual machine failures, better insights in the quality of results could be provided.

From an algorithmic perspective, we used decision trees for classification because of their inherent understandability and applicability of the models. The resulting decision trees represent both the decisions learned and the decision making. Other classification algorithms could be used as well but were not analysed yet. Using only decision trees might have limited the predictive power of the models. As for the detection of abrupt changes, we used Hotelling's T^2 statistic. This is because the method has proven to give good results for the detection of both univariate and multivariate changes. Future research in this area could leverage other algorithms for change detection as well.

With regard to data preprocessing, we used linear interpolation to handle the gaps in the raw data sets. This corresponds to the underlying data compression method which is based on a linear regression. However, other methods to deal with incomplete data sets have not been analysed in depth yet.

5.2 Evaluation Metrics

There are various evaluation metrics which are based on errors and successes. Each classified instance falls into one out of four categories, two types of error and two types of success. Successes are either actual alarms which are correctly classified as such or non-alarms which are correctly classified as such. An error occurs when either a non-alarm has been predicted as alarm or when an alarm has been incorrectly predicted as non-alarm. Table 2 summarizes the evaluation metrics currently used. All these are based on the errors and successes just described.

5.3 Results

Due to the enormous size of the data we split it into 4 relatively equally sized partitions. Up to now we mostly used the last one of these four parts for our evaluation. Within this part, we used a 4-fold cross-validation to split the partition into training and test sets. This exhaustive cross-validation technique ensures that all data points are contained at least once in both training and test set. The average results are displayed in Table 3. We averaged them across all four cross validations within the last partition.

Table 2: Evaluation metrics, their interpretation and domain. Domain presented in the form of *worst* - *best*.

Metric	Interpretation	Domain
Accuracy	How many alarms and non-alarms are correctly classified?	0% - 100%
Specificity	Of those which are truly non-alarms, how many were classified as non-alarms?	0% - 100%
Sensitivity	Of those which are truly alarms, how many were classified as alarms?	0% - 100%
Precision	Of those that are classified as alarms, how many were truly alarms?	0% - 100%
F-Score	Harmonic mean of sensitivity and precision	0% - 100%
AUC	General ability to achieve perfect result. Perfect in the sense of no missed alarms while never predicting false alarms	0.5 - 1

Table 3: Average results for each framework. Best values for a metric are indicated in bold

Framework	Accuracy	Specificity	Sensitivity	Precision	F-score	AUC
BaselineSimple	82.2%	82.4%	21.2%	4.60%	4.20%	0.550
BaselineDynamic	75.4%	75.6%	**30.4%**	6.20%	7.50%	0.548
UnivCD	92.9%	93.2%	23.4%	7.80%	8.90%	0.531
HiCS-mCD	**99.5%**	**99.8%**	23.1%	**36.5%**	**28.2%**	**0.631**
RandSub-mCD	96.0%	96.4%	8.20%	2.30%	3.00%	0.520
HiCS-mCD-Features	77.9%	78.0%	30.5%	9.9%	11.2%	0.537
HiCS-mCD-Ensemble	99.3%	99.6%	14.4%	14.7%	14.6%	0.581

For the final model evaluation it is important to perform well in several metrics. Averaging metrics such as the F-score were developed to reward models which perform well with regard to multiple metrics. Single high values might point to an over-fit in a particular direction or wrong optimization targets.

The model which performs worst overall is the RandomSub-mCD strategy. This model performs worst in 4 metrics and does not reach top performance in any metric. The fair performance on accuracy and specificity can easily be reached. This applies in general to data sets with a high class imbalance. The inherent high portion of non-alarms in the data set can easily be predicted. This has a strong effect on the accuracy and the specificity. The bad overall performance of the RandomSub-mCD strategy can be explained through the trivial random selection of the subspaces. Although some interesting subspaces might be selected in a particular run of the method, irrelevant subspaces blur the overall performance.

Both baseline approaches do not perform well with regard to the evaluation metrics. The static features which are closely related to the raw sensor measurements do not lead to good predictions alone. The added dynamic features do not lead to improvements regarding the overall prediction performance of the model. The Baseline_Dynamic model only performs well for the sensitivity metric. As discussed earlier, sensitivity describes how well actual alarms are predicted as such. High sensitivity in combination with low accuracy values point to models which favor predicting alarms over non-alarms.

Both the UnivCD and the HiCS-mCD-Ensemble model perform partially better than the other previously discussed models. Especially the ensemble model performs considerably well in the metrics accuracy and specificity. This means that the amount of false positives or type I errors is low. However, the amount of type II errors or misses is higher. This can be achieved if predicting non-alarms is chosen over predicting alarms.

The model based on HiCS-mCD performs best for 5 out of 6 metrics. The model which is based on multivariate change detection in high contrast subspaces performs not only better in accuracy and specificity, but especially well in precision. The good performance for precision also leads to the best average value for the F-score.

6 Conclusion and Future Work

This report addresses the problem of the detection of faulty states for predictive maintenance using subspace search methods. Novel frameworks have been proposed to both identify and predict faulty states at early stages. The frameworks use well-known techniques such as Hotelling's T-square change detection to identify abrupt changes and novel techniques such as high contrast subspaces (HiCS) to find potentially interesting subspaces. Each proposed framework uses decision trees with identical parametrization as classification method. The frameworks have been evaluated on a real-world data set obtained by MAN Diesel & Turbo SE.

The currently most promising framework is HiCS-mCD. This model performed best regarding the specified evaluation metrics. This framework is based on multivariate change detection using Hotelling's T-square statistic in such subspaces identified by the HiCS subspace search algorithm. This framework is especially useful because of the inherent scalability; the subspace search step ensures that only a limited number of high contrast subspaces is used for further steps. Additionally, searching for multivariate changes within subspaces or combinations of attributes instead of searching for univariate changes has shown to yield better results. This shows that the system state and potential faulty states can be characterized by the relationships between the variables. This relationship seems to be much more powerful than looking at abrupt changes at individual sensor measurements. Hence, subspace search seems to improve prediction quality.

Further contributions are the concept of using subspace search as preprocessing step before identifying abrupt changes and the idea of treating change detection scores as dimensions or features. Both concepts have been applied in the HiCS-mCD framework, and the superior performance has been shown by the evaluation on the real-world data set. Additionally, another use case for the high contrast subspace approach has been shown. Apart from using the subspace search technique as a dimensionality reduction step for traditional outlier detection, the method can also be used together with change detection. In comparison with the random subspace method, HiCS has shown to yield much better results during the evaluation. This can be explained by the fact that irrelevant subspaces blur the overall result when subspaces are selected randomly.

Future work in this project should improve the predictive power, while increasing the predictive window. First steps to this would be to address the current limitations. This includes evaluating multiple different classifiers as well as increasing the data quality. This project would also benefit from further studies, based on data with machine failures.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 634135.

References

[1] Ryan Prescott Adams and David JC MacKay. “Bayesian online changepoint detection”. In: *arXiv preprint arXiv:0710.3742* (2007).

[2] Michèle Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*. Vol. 104. Prentice Hall Englewood Cliffs, 1993.

[3] R. Bellman and Rand Corporation. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. ISBN: 9780691079516. URL: `https://books.google.it/books?id=wdtoPwAACAAJ`.

[4] Irad Ben-Gal et al. “Efficient construction of decision trees by the dual information distance method”. In: *Quality Technology & Quantitative Management (QTQM)* 11.1 (2014), pp. 133–147.

[5] Kevin Beyer et al. “When is “nearest neighbor” meaningful?” In: *Database Theory—ICDT’99*. Springer, 1999, pp. 217–235.

[6] Leo Breiman et al. *Classification and regression trees*. CRC press, 1984.

[7] Xi C Chen et al. “Contextual Time Series Change Detection.” In: *SDM*. SIAM. 2013, pp. 503–511.

[8] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. “The mahalanobis distance”. In: *Chemometrics and intelligent laboratory systems* 50.1 (2000), pp. 1–18.

[9] Rommert Dekker. “Applications of maintenance optimization models: a review and analysis”. In: *Reliability Engineering & System Safety* 51.3 (1996), pp. 229–240.

[10] Frédéric Desobry, Manuel Davy, and Christian Doncarli. “An online kernel change detection algorithm”. In: *Signal Processing, IEEE Transactions on* 53.8 (2005), pp. 2961–2974.

[11] Charles A Doswell III, Robert Davies-Jones, and David L Keller. “On summary measures of skill in rare event forecasting based on contingency tables”. In: *Weather and Forecasting* 5.4 (1990), pp. 576–585.

[12] Neil H Eklund and Kai F Goebel. “Using neural networks and the rank permutation transformation to detect abnormal conditions in aircraft engines”. In: *Soft Computing in Industrial Applications, 2005. SMCia/05. Proceedings of the 2005 IEEE Mid-Summer Workshop on*. IEEE. 2005, pp. 1–5.

[13] Peter Filzmoser, Ricardo Maronna, and Mark Werner. “Outlier identification in high dimensions”. In: *Computational Statistics & Data Analysis* 52.3 (2008), pp. 1694–1711.

[14] Shuli Han, Bo Yuan, and Wenhuang Liu. “Rare class mining: progress and prospect”. In: *Pattern Recognition, 2009. CCPR 2009. Chinese Conference on*. IEEE. 2009, pp. 1–5.

[15] Alexander Hinneburg, Charu C Aggarwal, and Daniel A Keim. “What is the nearest neighbor in high dimensional spaces?” In: (2000).

[16] Harold Hotelling. *The generalization of Student’s ratio*. Springer, 1992.

[17] Xiao Hu, Hai Qiu, and Naresh Iyer. “Multivariate change detection for time series data in aircraft engine fault diagnostics”. In: *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*. IEEE. 2007, pp. 2484–2489.

[18] Gareth James et al. *An introduction to statistical learning*. Springer, 2013.

[19] Fabian Keller, Emmanuel Müller, and Klemens Böhm. "HiCS: high contrast subspaces for density-based outlier ranking". In: *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE. 2012, pp. 1037–1048.

[20] Hans-Peter Kriegel et al. "Outlier detection in axis-parallel subspaces of high dimensional data". In: *Advances in Knowledge Discovery and Data Mining*. Springer, 2009, pp. 831–838.

[21] Pradeep Lall et al. "Statistical pattern recognition and built-in reliability test for feature extraction and health monitoring of electronics under shock loads". In: *Electronic Components and Technology Conference, 2007. ECTC'07. Proceedings. 57th*. IEEE. 2007, pp. 1161–1178.

[22] Aleksandar Lazarevic and Vipin Kumar. "Feature bagging for outlier detection". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM. 2005, pp. 157–166.

[23] Jay Lee et al. "Intelligent prognostics tools and e-maintenance". In: *Computers in industry* 57.6 (2006), pp. 476–489.

[24] Robert L Mason and John C Young. *Multivariate statistical process control with industrial applications*. Vol. 9. Siam, 2002.

[25] RA Mollineda, R Alejo, and JM Sotoca. "The class imbalance problem in pattern classification and learning". In: *II Congreso Español de Informática (CEDI 2007). ISBN*. Citeseer. 2007, pp. 978–84.

[26] ES Page. "Continuous inspection schemes". In: *Biometrika* (1954), pp. 100–115.

[27] Ewa Skubalska-Rafajłowicz. "Random projections and Hotelling's T2 statistics for change detection in high-dimensional data streams". In: *International Journal of Applied Mathematics and Computer Science* 23.2 (2013), pp. 447–461.

[28] David C Swanson. "A general prognostic tracking algorithm for predictive maintenance". In: *Aerospace Conference, 2001, IEEE Proceedings.* Vol. 6. IEEE. 2001, pp. 2971–2977.

[29] Kaibo Wang and Wei Jiang. "High-dimensional process monitoring and fault isolation via variable selection". In: *Journal of Quality Technology* 41.3 (2009), p. 247.

[30] Terry Wireman. "World class maintenance management." In: *AUTOFACT'89* (1989), p. 1989.