Unsupervised anomaly detection based on clustering methods and sensor data on a marine diesel engine

Erik Vanem[a, b] and Andreas Brandsæter[a, b]

[a]DNV GL, Veritasveien 1, Høvik, Norway; [b]Department of Mathematics, University of Oslo, Oslo, Norway

ARTICLE HISTORY
Compiled June 14, 2019

ABSTRACT
Sensor data from marine engine systems can be used to detect changes in performance in near real-time which may be indicative of an impending failure. Thus, sensor-based condition monitoring can be important for the reliability of ship machinery systems and improve maritime safety. However, there is a need for efficient and robust algorithms to detect such changes in the data streams. In this paper, sensor data from a marine diesel engine on an ocean-going ship are used for anomaly detection. The focus is on unsupervised methods based on clustering and the idea is to identify clusters in sensor data in normal operating conditions and to assess whether new observations belong to any of these clusters. The anomaly detection methods presented in this paper are applied to sensor data with no known faults. Being fully unsupervised, however, they do not rely on the assumption that all measurements are fault-free as long as the amount of faulty data is small. The methods explored in this study include K-means clustering, Mixture of Gaussian models, density based clustering, self-organizing maps and support vector machines. These could be used separately or in combination to provide an efficient initial screening of the data and decide whether more detailed analysis is required. The performance of the various methods is generally found to be good, also in comparison with other methods. Overall, cluster-based methods are found to be promising candidates for online anomaly detection and condition monitoring of ship machinery systems based on sensor data.

KEYWORDS
Ship propulsion system; condition monitoring; maritime safety and reliability; anomaly detection; sensor data; data-driven methods; unsupervised learning

1. Introduction

Sensor data collected from machinery systems on board ships provide real-time information about the condition of the ship. Such sensor-based condition monitoring can be used to detect changes in the performance of the system in near real-time which may be indicative of a system fault or even an impending failure. However, there is a need for efficient and robust algorithms to detect such changes in the data streams. Typically, a data-driven condition monitoring system includes anomaly detection, fault identification and prognostics. The first task is to monitor the data streams to detect deviations from normal system behaviour indicative of a change of the system. This is

CONTACT E. Vanem. Email: Erik.Vanem@dnvgl.com

referred to as anomaly detection, and for this task only nominal data is needed to train the algorithms. The next step is fault isolation or fault identification where a diagnostic tool is applied to estimate what type of deviation the anomaly is, i.e. to distinguish real faults from unexpected but normal behaviour, and to identify the type of fault. In order to train such algorithms, information (data) about both normal and faulty states of the system is needed. Essentially, in a data-driven approach, this is a classification task, where labelled data is needed in order to perform the classification, see e.g. Vanem (2018b) for a review of statistical methods that can be used for this purpose. Finally, the prognostics task try to estimate the future behaviour of the system, conditioned on the current state, and to estimate the remaining useful life (RUL). Typically for this task to be feasible with a data-driven approach, there is a need for run-to-failure data under varying conditions, something which is rarely available. Data-driven methods are alternatives to model-based approaches based on a physical modelling of the system from first principles (see e.g. Maftei et al. (2009); Lamaris and Hountalas (2010); Dimopoulos et al. (2014); Zymaris et al. (2016); Zacharewicz and Kniaziewicz (2017); Cipollini et al. (2018)), which may be more difficult to develop and use.

Sensor data from a marine diesel engine onboard an ocean going ship are analysed in this paper, collecting essential parameters such as power output from the engine, engine speed, bearing temperatures and various other temperatures, speeds and pressures for selected engine components. The idea is to utilize the information in these sensor signals to monitor the condition of the engine. The initial data streams collected from the ship are high-dimensional, with more than 100 data streams, but a subset of the data streams are carefully chosen for this analysis. The signals that are believed to be informative about the condition of the engine is selected based on engineering knowledge. Hence, what remains is a 24-dimensional dataset that will be used for condition monitoring. Further dimension reduction is applied in order to alleviate condition monitoring and anomaly detection.

The focus of this paper is on unsupervised methods for anomaly detection based on clustering. The idea is to identify clusters in the sensor data for normal operating conditions and to assess whether new data belong in any of these clusters. New data that cannot be assigned to any of the identified clusters, may be regarded as anomalies and call for further scrutiny and more detailed analysis of the data in order to diagnose the deviation and possibly flag an alarm. However, there are many ways for the data to fall outside a cluster without there being an actual fault in the system. Hence, the unsupervised techniques that are explored in this paper could be recommended for initial screening of the data and should be used in combination with other methods.

The approaches to anomaly detection presented in this paper is truly unsupervised, and they are applied to sensor data with no known faults. This does not mean, however, that the data are guaranteed to be without faults. Being fully unsupervised, the cluster based approaches does not need to explicitly assume that all observations in the training data are fault-free as long as the faulty data are not forming a separate cluster. This may be an advantage compared to for example the method based on AAKR Hines and Garvey (2006); Garvey et al. (2007), where a single faulty training data point may have a big influence on the signal reconstruction and thereby on the anomaly detection. The unsupervised anomaly detection presented in this paper may also detect anomalies in the training data and there is no need to be completely confident that the training data contains no faults.

Previously, different approaches for anomaly detection have been applied to the same dataset, i.e. the use of dynamical linear models (DLM) and sequential testing (Vanem and Storvik 2017) and the use of auto associative kernel regression (AAKR) (Brandsæter et al. 2016, 2017). Both these approaches are based on fitting a model to normal data and predict or reconstruct new sensor data and then comparing to the predicted or reconstructed signals. Sequential testing are then performed on the residuals to detect anomalies. In both cases, sequential probability ratio tests (SPRT) were applied. Even though these methods generally work well, they did encounter some problems with the marine engine data streams, due to the different operational conditions which give rise to spurious jumps in the data. This time- and operational state dependence in the data makes prediction and re-construction challenging and anomaly detection based on signal reconstruction or predictions are not straightforward. Hence, in this paper, a simple and unsupervised approach to anomaly detection based on clustering is explored.

2. Data description and exploratory analysis

The dataset contains several sensor signals that can be related to the main bearing condition of one of four separate diesel engines on a ship. It is noted that the collected data do not contain any known faults or failures of the system, and the data are not compared to maintenance logs of the system. The list of selected signals are included in table 1. The MG1TE702-stream contains only zero-values and these signals are excluded from the subsequent analysis.

Table 1. Sensor signals in the dataset

MAIN GENERATOR ENGINE 1	
MG019	MGE1 ENGINE SPEED [rpm]
MG1PT201	MGE1 LO PRESS ENGINE INLET [bar]
MG1PT401	MGE1 HT WATER JACKET INLET PRESS [bar]
MG1PT601	MGE1 CHARGE AIR PRESS AT ENGINE INLET [bar]
MG1SE518	MG1 TC A SPEED [rpm]
MG1SE528	MG1 TC B SPEED [rpm]
MG1TE201	MGE1 LO TEMP ENGINE INLET [C]
MG1TE272	MGE1 LO TEMP TC OUTLET A [C]
MG1TE282	MGE1 LO TEMP TC OUTLET B [C]
MG1TE511	MGE1 EXHAUST GAS TEMP TC A INLET [C]
MG1TE517	MGE1 EXHAUST GAS TEMP TC A OUTLET [C]
MG1TE521	MGE1 EXHAUST GAS TEMP TC B INLET [C]
MG1TE527	MGE1 EXHAUST GAS TEMP TC B OUTLET [C]
MG1TE600	MGE1 AIR TEMP TC INLET [C]
MG1TE601	MGE1 CHARGE AIR TEMP AT ENGINE INLET [C]
MG1TE700	MAIN BEARING NO 0 TEMP MGE1 [C]
MG1TE701	MAIN BEARING NO 1 TEMP MGE1 [C]
MG1TE702	MAIN BEARING NO 2 TEMP MGE1 [C]
MG1TE703	MAIN BEARING NO 3 TEMP MGE1 [C]
MG1TE704	MAIN BEARING NO 4 TEMP MGE1 [C]
MG1TE705	MAIN BEARING NO 5 TEMP MGE1 [C]
MG1TE706	MAIN BEARING NO 6 TEMP MGE1 [C]
MG1TE707	MAIN BEARING NO 7 TEMP MGE1 [C]
PM100.07	MG1 POWER [kW]

The sensor signals cover a period of about 10 months starting from December 2014 with a sampling frequency of one minute, but the hourly means are calculated and used in the subsequent analysis. It is observed that many of the signals are highly correlated. For example, the various temperature measurements for the main bearings are all very strongly correlated, see the traceplots in Figure 1. Traceplots of the engine speed is also shown in the figure. Note that the reduced dataset contains hourly averaged values and that this is different from a moving average. Thus, there are no overlap between data points within the different hours. The data for engine speed display two main modes of operation, with some transient states between these. This corresponds to the engine being turned on or off, for example in a load sharing scheme with the other generator engines.

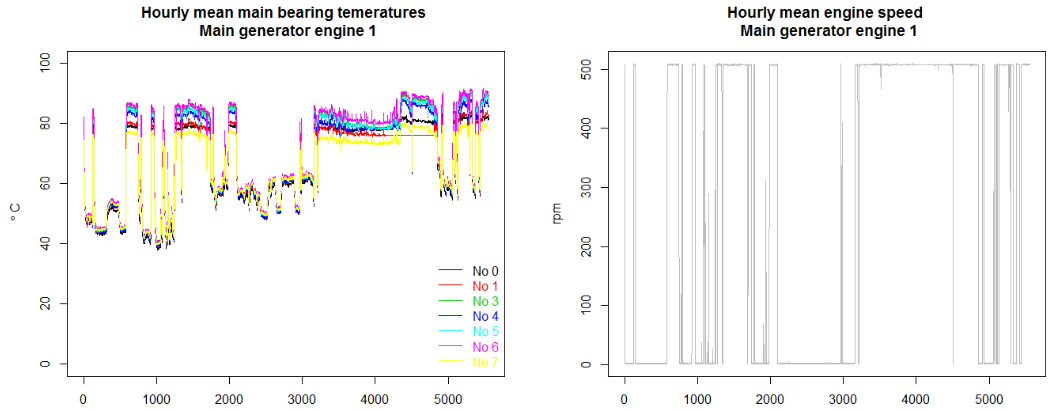

Figure 1. Traceplots illustrating strong correlation between some of the signals (engine 1); the various temperature readings for main bearing temperatures (left) and the engine speed (right); hourly averaged data

One important property of the sensor data is the temporal dependence in the signals. The partial autocorrelation function of the individual data streams gives an indication of the temporal dependence and memory in the data, and these show that there are strong temporal dependence in the temperature signals, but less so for the engine speed. Temperatures display a memory of at least 10 minutes, but with residual serial correlation beyond this. Nevertheless, in the cluster-based anomaly detection this time-dependence will be disregarded, and data-points will be clustered individually without any regard of the sequence they arrive in.

The data are divided into a training and a test dataset. The training data are used to identify clusters in the data, and the test data will then be assigned to one of these clusters. The underlying assumption is that the data naturally tend to cluster in a few clusters and that if new data arrives that are far from these clusters, this deviating behaviour causes suspicion of faults in the system. The time-dependence in the signals are neglected and the training data consist of 75% of the original data randomly selected. The remaining 25% constitute the test data. It is noted that randomly splitting the data into training- and test data is normally not recommended for time series data (Bergmeir et al. 2018), but for the purpose of clustering this can be defended.

2.1. Data preprocessing and dimension reduction

The data for generator engine 1 is 23-dimensional, and although it is possible to perform clustering in this 23-dimensional space, one may hope to get better performance if some form of dimension reduction is performed. Hence, principle component analysis and decomposition is performed on the training data and the same decomposition is subsequently applied to the test data. Plotting the variance and the cumulative proportion of the variance that are explained by the principal components can aid in selecting number of principal components to keep for the subsequent analysis, as shown in Figure 2. 99.5% of the information in the data is kept by the 7 first principal components, and this is the number of principal components kept brought forward for further analysis.

Traceplots of the 7 first principal components are shown in Figure 2, including both the training and test data.

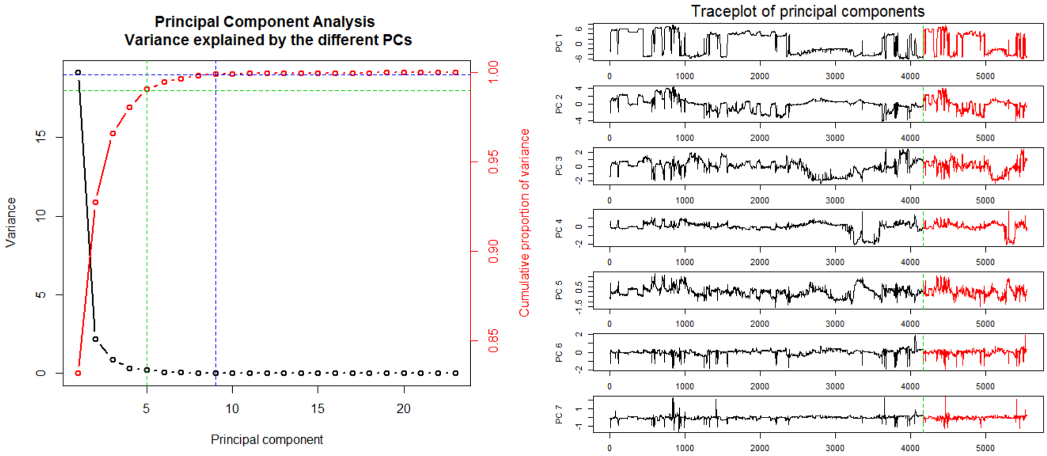

Figure 2. Variance explained by the principal components in the training data (left) and 7 first principal components (right)

3. Clustering methods for unsupervised anomaly detection

This section outlines the cluster analyses on the sensor signals and the subsequent application for anomaly detection. Various methods for clustering have been investigated, and different ways of using the various cluster methods for anomaly detection is explored.

3.1. *K-means clustering*

Before exploring the use of various clustering-methods for anomaly detection, the K-means clustering algorithm is applied in an initial cluster analysis (Hastie et al. 2009). This method divides the data into a specified number, K, of clusters based on the squared Euclidean distance, and requires K to be given. Essentially, the method iteratively identifies K centre-points and clusters the data around these in such a way that the distance between the data and the centre-points within each cluster is minimized. There are no way to unambiguously determine the optimal number of clusters. However, one may look at the ratio of the between-cluster variance and the total variance and indications of reasonable values of K can be found by looking at so-called elbow plots. This is shown in Figure 3. Vertical lines indicate $K = 5, 8$, and 15, and the elbow in the graph appear around $K = 5$. Hence, this is presumably a reasonable value of K for these data.

Scatterplots of the data (first 7 principal components) which indicate cluster membership based on K-means clustering with $K = 5$ are shown in Figure 4. The distribution of points within each cluster is also shown in the figure showing that all clusters are reasonably populated. Similar plots for the test data, where each data point in the test data is assigned to the cluster with the nearest cluster center are shown in the figure. By inspecting these plots, it appears that the test data has been reasonably clustered and that the distribution of observations to each cluster seem to be comparable.

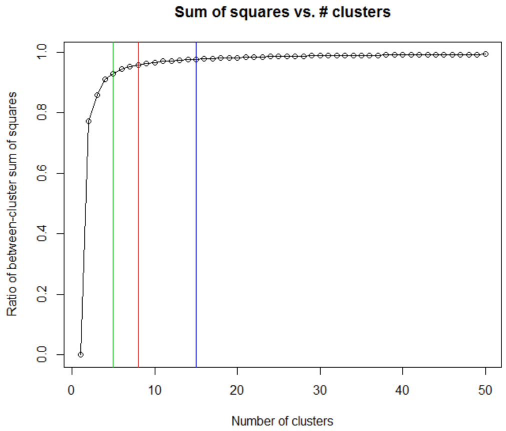

Figure 3. Estimating the number of clusters in the data, K

3.2. Mixture of Gaussian models

Compared to K-means clustering, clustering with mixture of Gaussian models has two main advantages. First, a parametric model is fitted to the data, so it is possible to obtain density estimates and p-values for how likely the data are given the model. Moreover, since K-means clustering is based on the Euclidean distance, the clusters will be defined by hyperspheres around the cluster centres, whereas the mixture of Gaussian models take the correlation into account and can give ellipsoid-shaped clusters of varying shapes and orientations.

The density of a Gaussian mixture model is on the form of a mix of K individual Gaussian densities, and the density function can be written as (see e.g. Hastie et al. (2009))

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \phi(\boldsymbol{x}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}). \tag{1}$$

The π_k's are the mixing proportions determining the contribution from each of the K mixtures, and $\sum_k \pi_k = 1$. The density of each mixture is described by the Gaussian density function, $\phi(\cdot)$ with a mean vector $\boldsymbol{\mu_k}$ and a covariance matrix $\boldsymbol{\Sigma_k}$. Fitting such a model to data means estimating the model parameters, $\hat{\pi}_k$, $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}_k$, and this is done using the maximum likelihood and the Expectation-Maximization (EM) algorithm. Having estimated a mixture model, it may provide an estimate for the probability than an observation, i belongs to a component, l as shown in eq. (2) and clustering may be performed by assigning the observation to the component with the highest probability.

$$\hat{p}_{il} = \frac{\hat{\pi}_l \phi(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_l, \hat{\boldsymbol{\Sigma}}_l)}{\sum_{k=1}^{K} \hat{\pi}_k \phi(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)} \tag{2}$$

One of the tasks of fitting a mixture model to data is to determine the value of K. One way to do this is to calculate the Bayesian Information Criterion (BIC) and choose the model that maximizes this. Alternatively, the integrated complete likelihood (ICL) can be used. ICL can be thought of as similar to BIC, but penalized by the mean entropy (Baudry et al. 2010). Typically, this will suggest a lower number of clusters compared

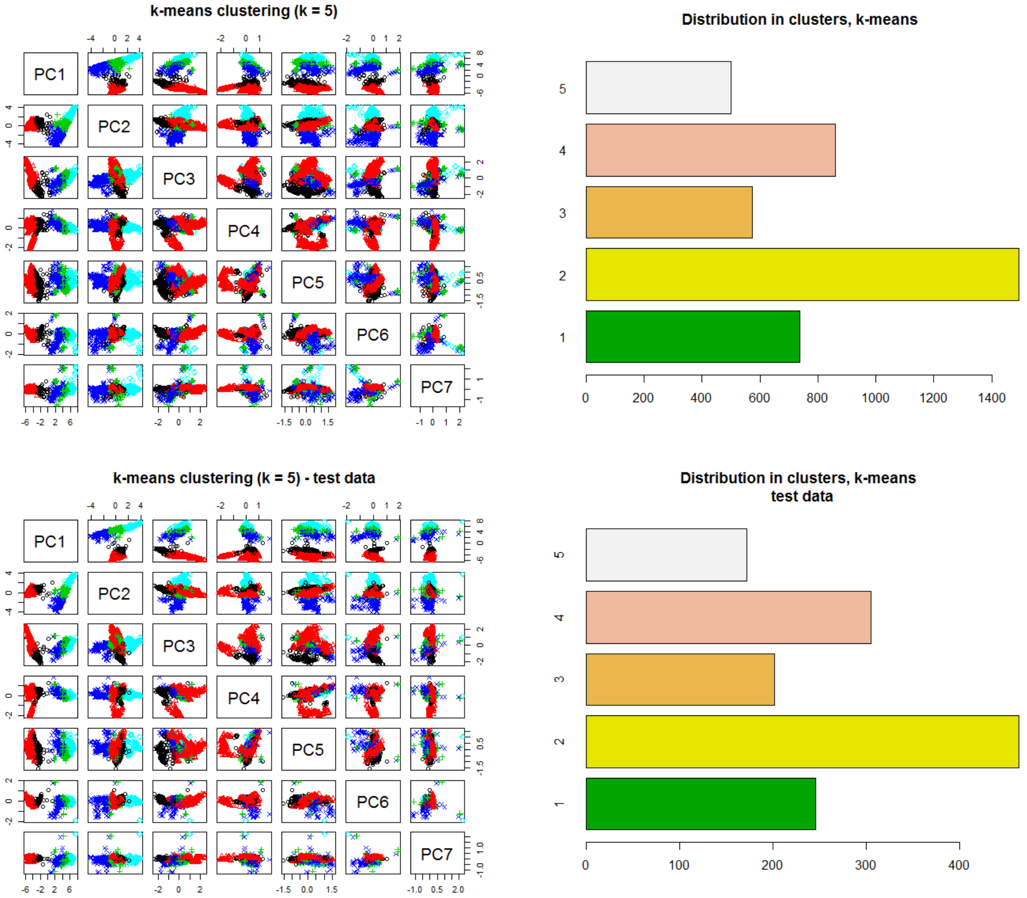

Figure 4. Data clustered in $K = 5$ clusters with K-means (left) and distribution of observations within the clusters (right); Clustering on training data (top) and applied to the test data (bottom)

to the BIC. The BIC and the ICL as a function of number of components in a mixture of Gaussians model are shown in Figure 5 for the training data. No restrictions have been put on the various components, which may have varying orientation, shape and volume. Both the BIC and ICL favour models with a high number of components and according to both criteria, the mixture model with $K = 31$ components is suggested as this corresponds to maximum BIC and ICL, as shown in Figure 5.

Including many components in the mixture model increases the probability of overfitting and it might be reasonable to choose a lower number of components. Hence, in this study, both the suggested value of $K = 31$ as well as $K = 5$ will be tried. The distribution of observations assigned to each cluster for both models are shown in Figure 6, for both the training and test data. One thing that is observed is that for the mixture model with $K = 31$ components one of the clusters did not get any observation assigned to it in the test data. This indicates that the model is overfitted. Apart from this, the same clustering structure is observed in the training as in the test data.

3.2.1. Anomaly detection based on mixture of Gaussian clustering

A fitted Gaussian mixture model can be used directly in condition monitoring and anomaly detection of new observations. The implicit assumption is that new patterns

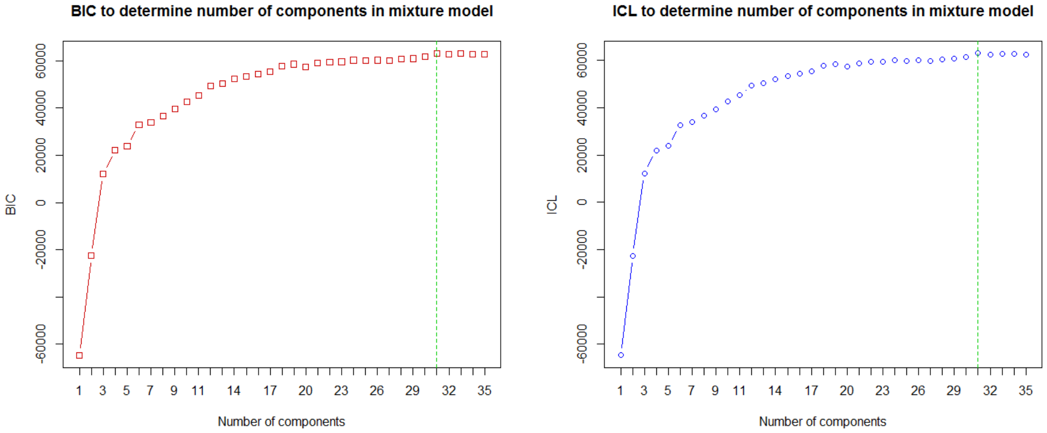

Figure 5. Estimating the number of components of the mixture model by BIC (left) and ICL (right)

in the sensor signals that are extreme according to the established model will be flagged as anomalous. There are many ways of defining extremes within the context of mixture models, and in this study an anomaly will be defined based on some probability of being extreme according to all the model components, and a p-value can be calculated for each Gaussian component separately. Then, the overall p-value will be the maximum p-value among the K components.

In the multivariate setting, there are different definitions of being extreme, see e.g. Serinaldi (2015); Vanem (2018a). Figure 7 illustrates four ways of defining extremes in a bivariate setting, where the shaded areas correspond to the probability of being more extreme than a particular point. In the first example, the probability of being more extreme than an observation is defined as the probability of both marginals being more extreme, P_{AND}. The second example defines the probability of being extreme as the probability of either of the marginals being as extreme, P_{OR}. The third example defines extreme as being outside an exceedance hyperplane, P_e. This definition of multivariate extremes is in line with the concept of environmental contours often applied in structural reliability analysis (Haver and Winterstein 2009; Huseby et al. 2013, 2015) and P_e can be calculated in different ways. Finally, the last example defines extreme as being further away from the central point (mean vector) of the distribution in any direction, P_D, and can be calculated based on the Mahalanobis distance. In the n-variate normal case, the squared Mahalanobis distance will be distributed according to the χ^2-distribution with n degrees of freedom, and one may define a P_D as the probability of having a squared Mahalanobis distance greater than what is observed. I.e. for an observation $\boldsymbol{x}_i$ with distance D_i, $P_D(\boldsymbol{x}_i) = P_{\chi^2_n}(d \geq D_i^2)$.

The p-values for characterising how extreme an observation is will be very different according to the definition that is adopted. For higher dimensions, this difference will grow. The P_D value will typically be larger than the other P-values discussed above. Hence, in the following, the P_D value is calculated for each observation based on every Gaussian component, $k = 1, \ldots, K$, of the mixture model and a p-value can be set as the largest of the K P_D-values, i.e.

$$p = \max_{1 \leq k \leq K} P_{D,k} \tag{3}$$

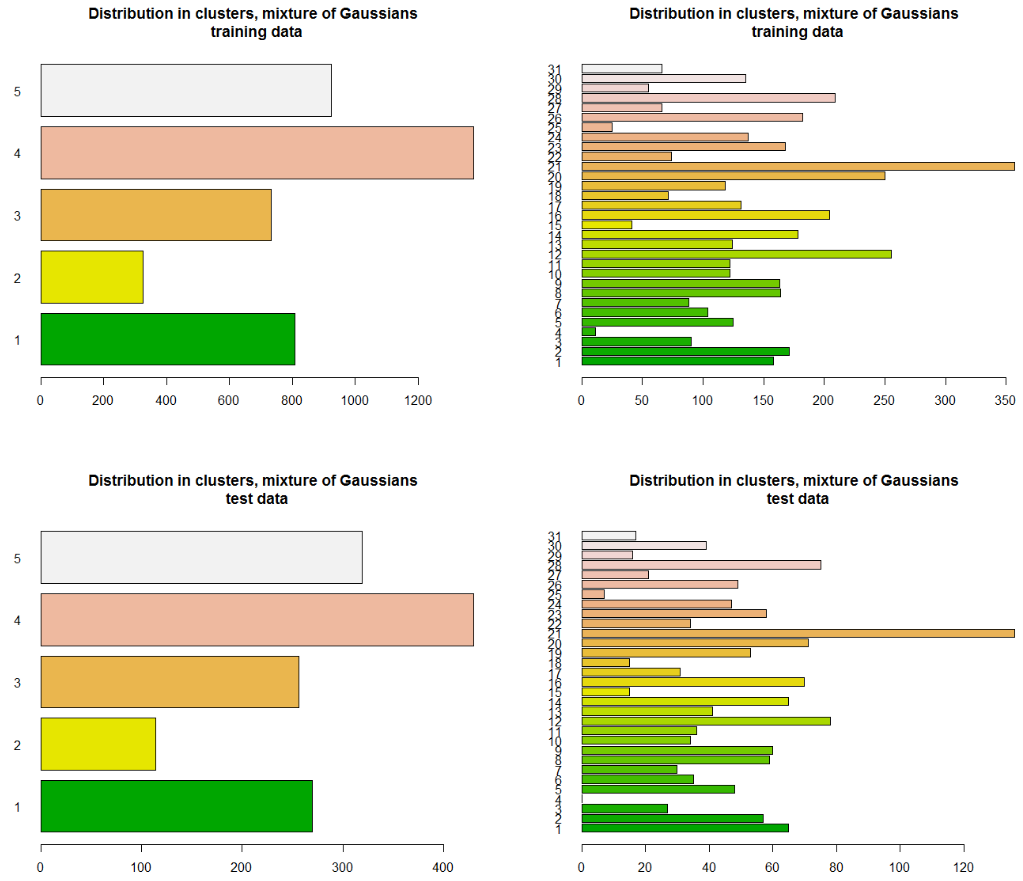

Figure 6. Distribution of observations assigned to each cluster by the mixture of Gaussian models with $K = 5$ (left) and $K = 31$ (right); training data (top) and test data (bottom)

One may use this for anomaly detection and flag any observations with small p-values as possibly anomalous. This corresponds to testing whether the observation belongs to the mixture component for which it is most likely to belong to, and if one can reject the hypothesis that it belongs to this component, one may reject the overall hypothesis that it belongs to the mixture model. What remains is to choose a suitable α-level for the test. In this study, each observation with $p < 0.05$ is initially regarded as an anomaly. Figure 8 shows the time series of the largest p-values for both the training and the test data and reports the number of anomalies in the training and test data, respectively, for the mixture models with $K = 31$ and $K = 5$. The solid horizontal line in the plot corresponds to $p = 0.05$. It is observed that if a mixture model with $K = 5$ is chosen, then approximately 4-5% of the observations are flagged as anomalous. If $K = 31$ this is reduced to 2-3%. This agrees well with the 5% level of the test and could be expected even if the mixture models were entirely correct and in the absence of any anomalies.

This anomaly detection scheme essentially performs one test for each component of the mixture model, and as such it can be construed as multiple testing. This may give rise to false negatives just by chance, and it may be reasonable to correct for this. One common correction is the Bonferroni correction that adjusts the α-level in the test to $\frac{\alpha}{n}$, where n is the number of tests performed. An implicit assumption here is that the tests are independent. In this case, with $n = K$ and $K = 5$ and $K = 31$, this gives the

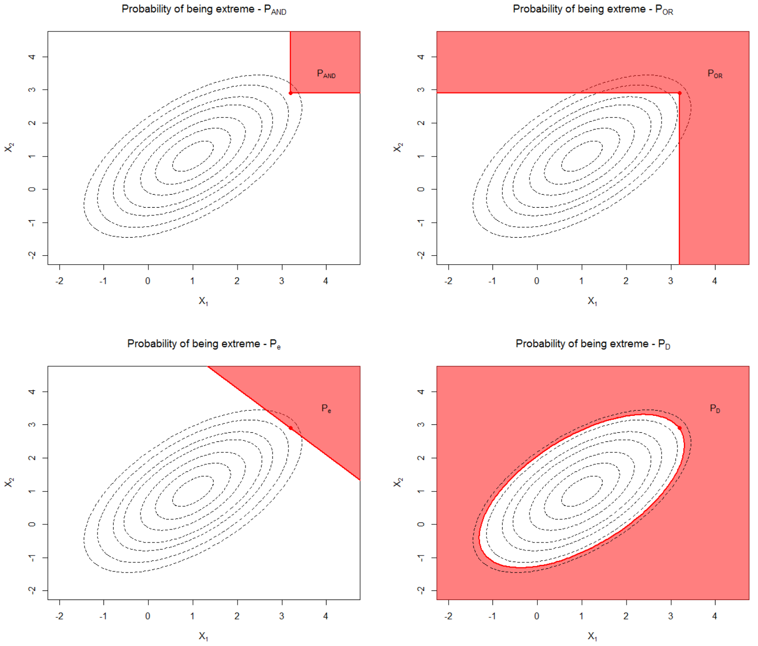

Figure 7. Different definitions of characterizing extremeness in the bivariate case

adjusted α-levels as 0.01 and 0.0016, respectively, and these levels are also indicated in Figure 8. The rate of observations flagged as anomalous with the adjusted levels are approximately 2% for the model with $K = 5$ and less than 0.1% for the model with $K = 31$.

If there are actual faults in the system, this presumably will be reflected in subsequent sensor readings, and one may require more than one anomalous observation to flag an alarm. Hence, sequential tests for anomalies could be established. A very simple approach could be to monitor the P_D-values and flag an alarm whenever a pre-defined number of subsequent values are below the p-value. This would significantly reduce false alarms due to spurious outliers. For example, for the data used in this study, the anomaly ratio for the various set-ups reduce to the numbers in Table2 by only requiring that two subsequent values of P_D are below the specified p-value. Depending on the system being monitored, a larger number of subsequent anomalous readings could be required in order to reduce the sensitivity to individual outliers, if needed. More elaborate sequential tests, based on combining the p-values from subsequent measurements could also be envisaged, but this is out of scope of this study.

3.3. Density based clustering - DBscan

Another approach to clustering is based on the density of observations in the feature space and groups observations with many neighbouring points into clusters. DBscan is an algorithm for such clustering (Martin et al. 1996) where the number of clusters in the data will be determined by the algorithm. However, there is a need to specify two parameters; the size of a neighbourhood (ε) and the minimum number of core points that needs to be contained within a neighbourhood for it to form a cluster, k. In principle, any distance function could be used, but in this application, the Euclidean distance will be assumed.

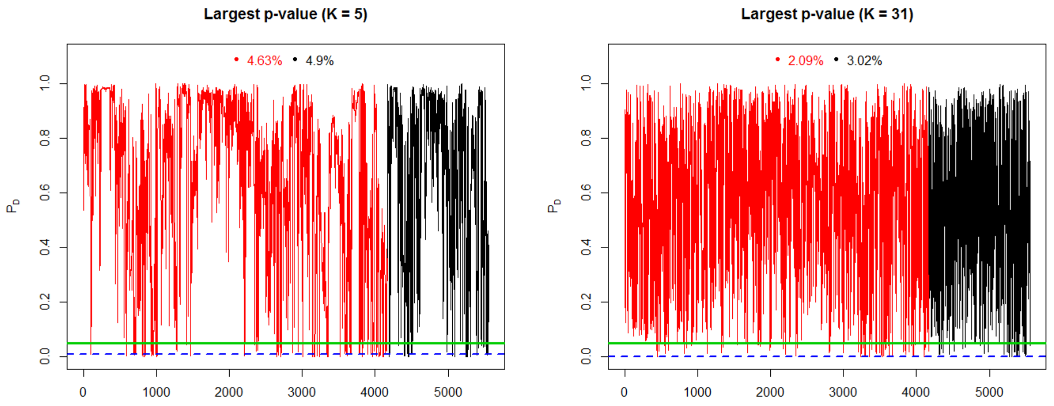

Figure 8. Maximal p-values for anomaly detection; mixture model with $K = 5$ (left) and $K = 31$ (right)

Table 2. Ratio of anomalies in the data, as detected by the mixture of Gaussian model clustering and flagging for 2 subsequent low p-values

	Training data	Test data
	$p = \alpha = 0.05$	
$K = 5$	2.95%	2.59%
$K = 31$	0.43%	0.58%
	$p = \alpha/K = 0.01$	
$K = 5$	1.06%	0.94%
	$p = \alpha/K \approx 0.0016$	
$K = 31$	0%	0%

DBscan estimates the density around each data-point by counting the number of observations within the specified neighbourhood size. It then distinguishes between core points, bordering points and a noise points. A core point has at least k points within the specified distance, ε. Points within the neighbourhood distance from a core point is said to be directly reachable from those core points. A point that is directly reachable from a core point, but with less than k points within the neighbourhood distance is referred to as a border point. A cluster is then all points that are reachable from a core point. Hence, each cluster must contain at least one core point and one or more border points. Points that are not reachable from any other points are regarded as outliers or noise-points. In this way, clusters may take any shape, and they may differ from the spherical- or ellipsoid shapes used for defining clusters with K-means og mixture of Gaussian models.

One attractive feature of density based clustering algorithms is that outliers or noise points are identified directly, which can be exploited for anomaly detection. However, the cluster structure will be highly dependent on the parameters ε and k and it may not be straightforward to determine the optimal values of these. As ε increases, the number of clusters decreases towards 1 and also the number of noise points decreases towards 0. Choosing too large value for ε thus results in all the data forming one cluster with no outliers. On the other hand, too small ε will give a complicated model with many clusters and may tend to overfit. Plots of number of clusters and number of noise points versus ε for various values of k (not shown herein) indicate that reasonable values for ε

should be in the range of 0.25 - 1.

It is recommended to use domain knowledge to determine the value of k in DBscan. In this study, results for three values of k are reported, i.e. $k = 10$, $k = 20$ and $k = 50$. For $k = 10$ ε is set to 0.5, for $k = 20$ $\varepsilon = 0.6$ and for $K = 50$ it is set to 1. These parameter choices yield clustering with 10, 5 and 3 clusters, respectively. The distribution of training data points in each cluster are shown in Figure 9. It is interesting to observe that the number of noise points or outliers (denoted as cluster 0) is very similar for the three pairs of parameter values. With $k = 10$, $k = 20$ and $k = 50$ the anomaly rates are 5.47%, 5.55% and 3.67%, respectively, which slightly higher than the ratios obtained by the mixture of Gaussian clustering.

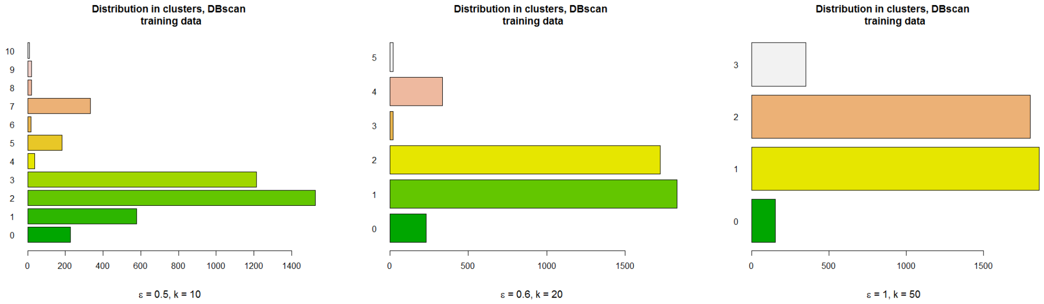

Figure 9. Distribution of data points in each clusters for the training data for different values of ε and k

3.3.1. Anomaly detection with DBscan

Having applied DBscan to the training data and defined a set of clusters, new observations can be assigned to any of the clusters as they are collected. Observations that do not belong to any of the clusters will be regarded as noise points and can be regarded as anomalies. The distribution of observations assigned to the various clusters are shown in Figure 10, and the distributions appear to be very similar to the distribution for the training data.

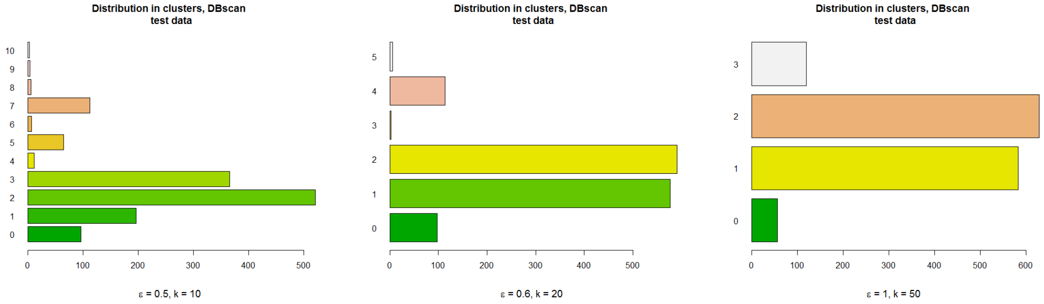

Figure 10. Distribution of number of points in each clusters for the test data for different values of ε and k

The ratio of noise points or anomalies in the test data, as detected by the DBscan clustering method is 6.9% for $k = 10$, 7.1% for $k = 20$ and 4.1% for $k = 50$. This is slightly higher than for the training data, and seems reasonable. It can also be observed that most of the data points detected as anomalies are the same regardless of the parameters. For example, of the 98 anomalies detected with $k = 20$, 80 of the same points were detected with $k = 10$ and 70 with $k = 50$. Hence, even though the number of clusters are different, there is general agreement for most of the detected anomalies.

In a simple sequential manner, one may also regard outliers as possible anomalies only if two or more subsequent observation are regarded as noise points. If only flagging for possible anomalies with at least two subsequent noise point, the anomaly rate in the training data reduces to 4.08% ($k = 10$), 4.32 ($k = 20$) and 2.67 ($k = 50$), respectively. For the test data, the corresponding rates are 4.47% ($k = 10$), 4.76% ($k = 20$) and 2.09% ($k = 50$), respectively.

3.3.2. Hierarchical DBscan - HDBscan

Hierarchical DBscan, HDBscan, is an extension of DBscan (Campello et al. 2013). It allows for varying density clusters and does not require the neighbourhood distance ε to be specified. Instead, it provides a hierarchy of clusters for any value of ε in a three-like structure. This three can then be cut at any place, corresponding to fixing the value of ε at any value to give different number of clusters. The algorithm then finds the optimal cuts in the hierarchy based on a cluster stability score.

Hierarchical DBscan clustering is performed on the training data for the three different values of k that was also used in the DBscan clustering above, i.e. $k = 10$, $k = 20$ and $k = 50$, as well as $k = 100$, and the flat solution corresponds to a solution with 68 clusters for $k = 10$, 33 clusters for $k = 20$, 12 clusters for $k = 50$ and 7 clusters for $k = 100$. Thus, the hierarchical DBscan results in significantly more clusters compared to the ones found by fixing the ε-parameter with DBscan.

The HDBscan algorithm calculates an outlier score for each data point, ranging from 0 to 1, with higher value corresponding to higher degree of outlierness. The score is based on both local and global properties of the hierarchy (Campello et al. 2015), and may identify points that are outliers compared to points in its neighbouring region without necessarily being outliers globally. It is possible to base anomaly detection on this score and regard all data points with an outlier score above a predefined threshold as possible anomalies. Histograms of the outlier score for the various values of k are shown in Figure 11, where the percentage of points having an outlier score above 0.95 is indicated. These percentages are 5.9% for $k = 10$, 4.4% for $k = 20$, 2.6% for $k = 50$ and 0.84% for $k = 100$.

Hierarchical DBscan is a transductive method, and this means that new observations should in principle be allowed to influence the underlying cluster structure and prediction of cluster membership and outlier scores are not straightforward for new observations based on a fixed clustering. Even though there are ways around this, clustering based on HDBscan are not brought forward for use in anomaly detection of new observations in this study.

3.4. Self-Organising Maps (SOM)

Self-organising maps, also sometimes referred to as Kohonen maps is a type of artificial neural networks for unsupervised learning (Kohonen 1982). They contain nodes with a weight vector of the same dimension as the input data which are represented as a location on the map. The weight vectors of a map are set at random and then iteratively updated by feeding input vectors from the training data. For each training data point, the distance (typically Euclidean distance) to all weight vectors is computed and the node with the weight vector that is closes to the input data will be called the best matching unit (BMU). The weight vectors of the nodes in the neighbourhood of the

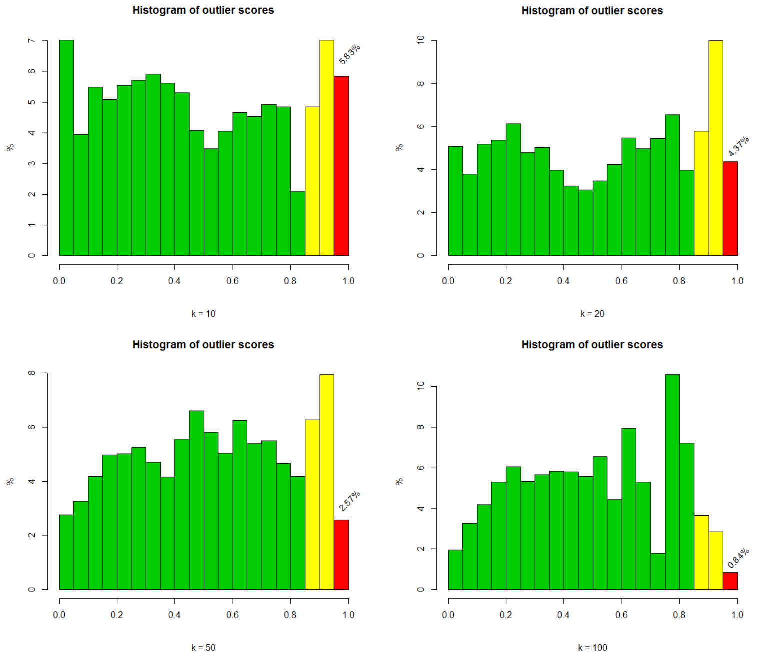

Figure 11. Histograms of outlier scores as calculated from HDBscan clustering for different values of k

BMU will be updated by pulling them closer to the input vector (training data point). This is repeated iteratively for all training data and for a specified number of iterations (cycles). The result is a map which associates output nodes with groups or patterns in the training data set. With a trained map, new observations can be mapped by assigning input vectors to the node with the closest weight vector, the so-called winning node.

One must specify the dimensions of the map and the distance function used to calculate the distances. In this study, all maps are based on the squared Euclidean distances. Moreover, one must specify the number of cycles or number of times the training data should be sent to the network. It must be ensured that a sufficient number of cycles is specified so that the training of the map converges. In order to check whether a reasonable map has been specified, there are some diagnostics plots that can be made, such as node count plots, plot of changes between iterations, and plot of the distribution of parameter values across the map. Such plots are not shown in this paper but self-organizing maps of different sizes have been explored. However, only results for self-organizing maps with 15 × 15 nodes are reported in the following. A previous application of self-organising maps for condition monitoring of marine engines is reported in Raptodimos and Lazakis (2018).

Clustering with self organizing maps can be based on the distances to neighbouring nodes. Initial K-means clustering of the map nodes indicates that around 5 clusters are reasonable. The actual clustering of the map nodes will be performed by hierarchical clustering. and the resulting clustering of the map is illustrated in Figure 12 for number of clusters $k = 4, \ldots, 9$. These plots agrees well with a value of $k = 5$.

Having performed clustering on the self organizing map, one may look at the cluster assignment for the training data and also predict the cluster membership on the test data. The distribution of observations in each cluster for both data sets, based on $k = 5$ and a map with 15 × 15 nodes, is shown in Figure 13.

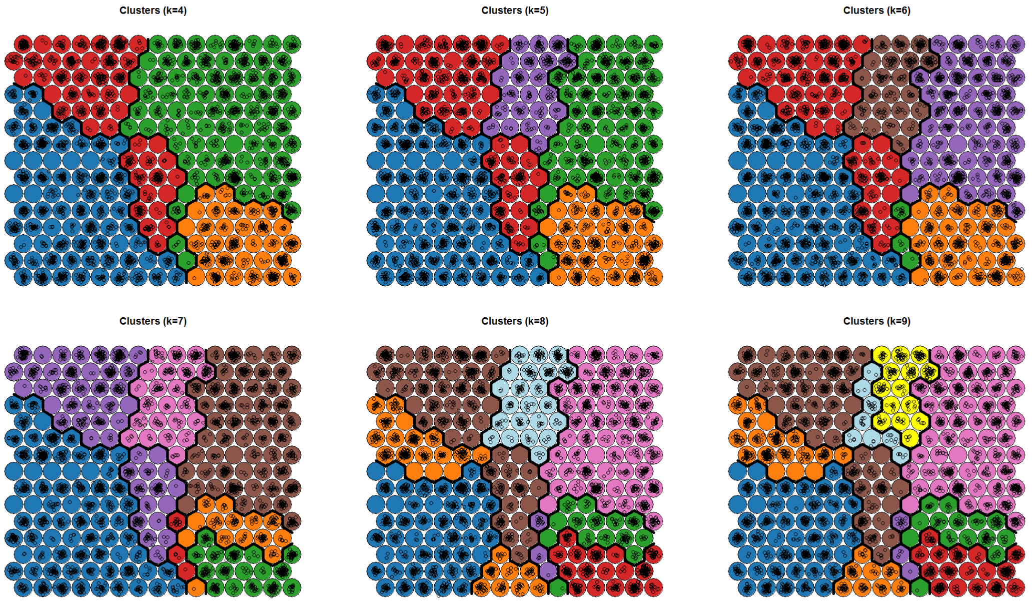

Figure 12. Clustering of the self organizing maps for different number of clusters; hierarchical clustering

3.4.1. Anomaly detection using self-organizing maps

There are different ways one could use self-organizing maps for anomaly detection. For example, one could identify some nodes in the map as outliers and do anomaly detection similar to the scheme based on DBscan above. However, in this study, a somewhat different approach is investigated, based on signal reconstruction and residual analysis. This resemble the anomaly detection approaches based on AAKR or DLM as reported in Brandsæter et al. (2017); Vanem and Storvik (2017). Self-organizing maps is also used for marine engine condition monitoring in e.g. Raptodimos and Lazakis (2018).

Reconstruction of new observations based on a trained map consists of first mapping the new observation to a node in the trained map and then to predict the parameter values corresponding to that node. This will typically be the average of all the training data that belongs to the same node. Assuming that the map has been trained on anomaly-free data, large deviations of the reconstructed signal from the observed signal can be regarded as a possible anomaly. One must then either define a threshold for when a residual is construed as large, or one could apply a sequential test such as the sequential probability ratio test (SPRT) as outlined in e.g. Brandsæter et al. (2017); Vanem and Storvik (2017). In this study a simple threshold approach is taken, and a possible anomaly is flagged whenever the absolute value of the residual is larger than a predetermined threshold. For the purpose of this exercise, this threshold is set to ±0.6 since the prediction error is typically below this for the training data. This is done on each sensor signal. Trace plots of the test data and the predictions based on the self-organizing maps are shown in Figure 14 (top row) for the three first principal components. Also the residuals are shown in the bottom row and the threshold is indicated by a horizontal dashed line.

Applying such an anomaly detection approach on the ship sensor signals, one gets an anomaly rate of 0.58% on the training data and 1.66% on the test data. If one require two subsequent anomalies to trigger an alarm, these rates reduce to 0.14% on the training data and 0.43% on the test data, respectively.

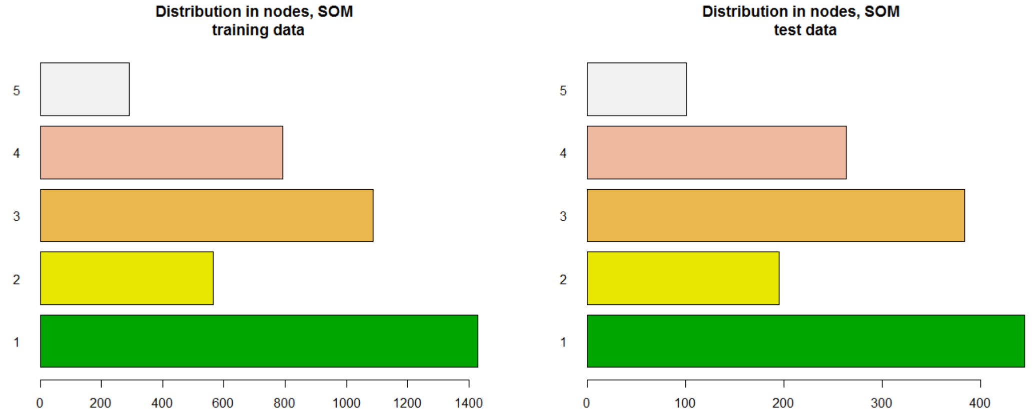

Figure 13. Distribution of observations within each cluster for the training (left) and test (right) data with clustering performed by self organizing maps; 5 clusters based on map with 15 × 15 nodes

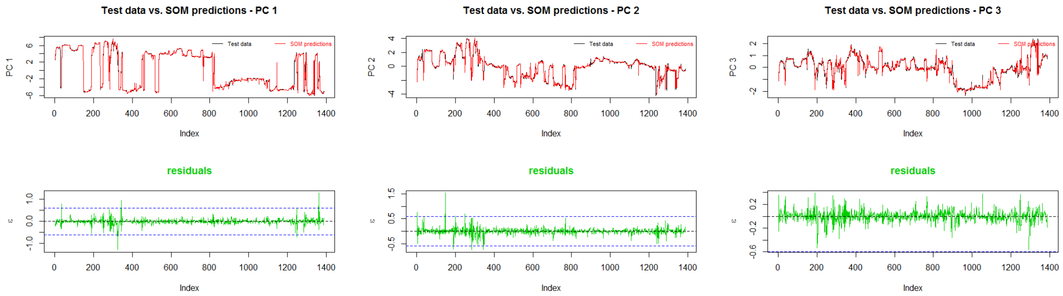

Figure 14. Anomaly detection can be performed by studying the residuals between the observed signals and the ones predicted by the trained map; Traceplots of test data and SOM prdictions (top row) and traceplots of residuals (bottom row)

3.5. Novelty detection with Support Vector Machines (SVM)

Support vector machines (SVM) is a supervised learning technique typically used for classification problems, see e.g. Hastie et al. (2009). However, it can also be used for anomaly detection by formulating this as a one-class problem, sometimes referred to as unary classification. The idea is that all training data are assumed to belong to one class (i.e. no fault) and the task is to detect deviations from this class and regard them as anomalies. This is often referred to as novelty detection.

Various kernels may be defined, but only the Gaussian radial basis function kernel have been employed in this study. The kernel can be interpreted as a similarity measure between data vectors, and the radial basis function kernel on two sample vectors, $\mathbf{X}$ and $\mathbf{X}'$ is

$$K(\mathbf{X}, \mathbf{X}') = e^{-\frac{\|\mathbf{X}-\mathbf{X}'\|^2}{2\sigma^2}} \tag{4}$$

The inverse kernel width σ is a hyperparameter that is estimated from the training data; typically it is a value between the 10- and 90-percentile of the euclidean distance in a fraction of the training data. In addition, one parameter, ν, needs to be specified which sets the upper bound on the training error and the lower bound on the fraction of data-points that may become support vectors. Essentially this determines the degree of ”softness” of the margins of the support vector machine.

One-class support vector machines have been fitted to the training data for various values of the parameter ν and applied to the test data for anomaly detection. The ratio of anomalies for various values of ν is presented in Table 3. By definition, all training data are labelled as ”known”, so there will be no anomalies in the training data with this method.

Table 3. Support vector machines: Anomaly rates in the test data for different values of ν

ν	0.001	0.005	0.01	0.02	0.05	0.1	0.15	0.2	0.25	0.5	0.75
# anomalies	37	43	38	49	77	149	202	260	335	667	1018
Anomaly rate	0.027	0.031	0.027	0.035	0.056	0.11	0.15	0.19	0.24	0.48	0.73

It is generally observed that the number of anomalies increases with the value of ν, but it is not straightforward to determine an optimal value. However, one may assume that the different classes, i.e. normal data and anomalous data, are perfectly separable in some enlarged space, and this suggests that support vector machines with hard margins should be preferred, i.e. small value of ν. When the time-points where the various support vector machines suggests anomalies are investigated it is generally observed that the points flagged as anomalous with smallest ν-parameter are also regarded as anomalies by the other models, but with additional points added for increasing values of ν. Hence, it may be concluded that one of the models with low value of ν should be used for anomaly detection.

4. Discussion

4.1. Anomalies detected by the different methods

One way to compare different methods is to compare the anomaly ratios. However, it can also be of interest to investigate how robust cluster-based anomaly detection is by comparing which observations are regarded as anomalous in the test data by the various methods. Figure 15 plots the flags that would occur from different schemes based on mixture of Gaussian, DBscan and SOM. This illustrates that many of the same data points are flagged as anomalies by several methods. Summing the number of possible anomalies from each method one gets 412 possible anomalies, but there are only 192 unique observations that are detected at least once.

Table 4 summarizes the number of methods that has detected the various possible anomalies in the data, for both the training and test data. Comparing all the methods, it is seen that the overall anomaly rate, as detected by any of the methods are 11.3% for the training data and 13.8% for the test data. This is probably too high, and much higher than the anomaly ratio from any of the individual methods.

Table 4. Anomalies detected by different number of methods

	Number of times detected								
	1	2	3	4	5	6	7	8	$\sum$
Training data	190	123	95	48	9	6	1	0	472
Test data	80	46	35	23	6	1	1	0	192

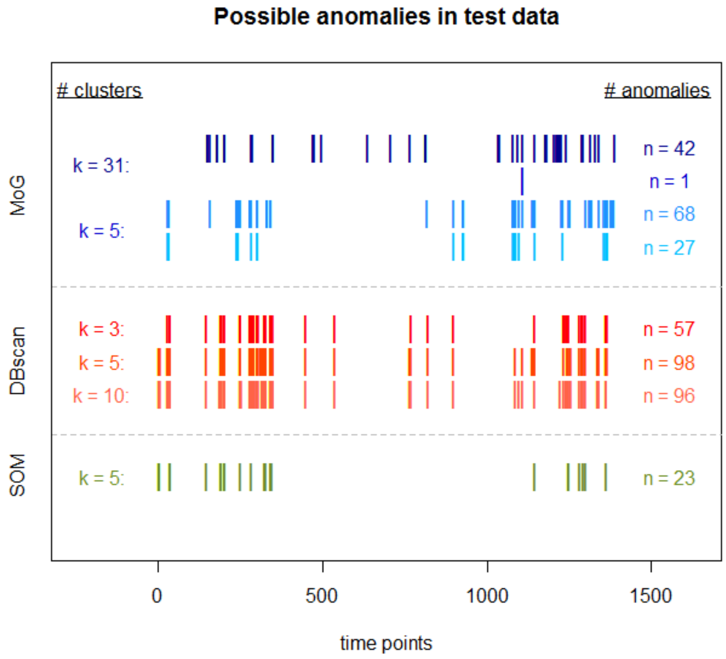

Figure 15. Comparing the possible anomalies detected by the different clustering methods. Vertical lines corresponds to time points where the different methods would flag an alarm.

On way to get more robust anomaly detection is to apply an ensemble of methods and disregard anomalies that are only detected by one method. For example, if applying all the 8 methods above and flagging an alarm only if two or more methods regards an observation as a possible anomaly, one would obtain anomaly ratios of 6.8% for the training data and 8.1% for the test data, respectively. If detection by three or more methods are required, the anomaly rates would reduce even further, to 3.8% for the training data and 4.8% for the test data.

By requiring two subsequent anomalous observations to trigger an alarm, the anomaly rate for each individual method decreased notably. Figure 16 illustrate which observations will be regarded as possible anomalies using this approach for the test data. There are notable overlap and the overall anomaly rate from all the methods with this setup is 7.7% for the training data and 8.4% for the test data, respectively. This is considerably lower than the overall anomaly ratio as detected without requiring 2 subsequent anomalous readings. Moreover, the methods could again be combined to raise a flag only if a minimum number of the methods agree on a possible anomaly. If the requirement is that a possible anomaly is detected by at least two methods the anomaly rate reduce to 4.7% for both the training and the test data. If the requirement is set to at least 3 methods these rates reduce further to 2.4% and 2.2%, respectively, for the training and test data.

There are several ways to combine an ensemble of methods to establish robust anomaly detection methods for ship sensor data. Combinations with other methods, such as for example the AAKR method (Brandsæter et al. 2017, 2019) or DLM (Vanem and Storvik 2017, 2018) could also be investigated, but is out of scope of the current study. For final implementation in an actual condition monitoring system, the performance of the methods and how they are combined should be investigated in more detail.

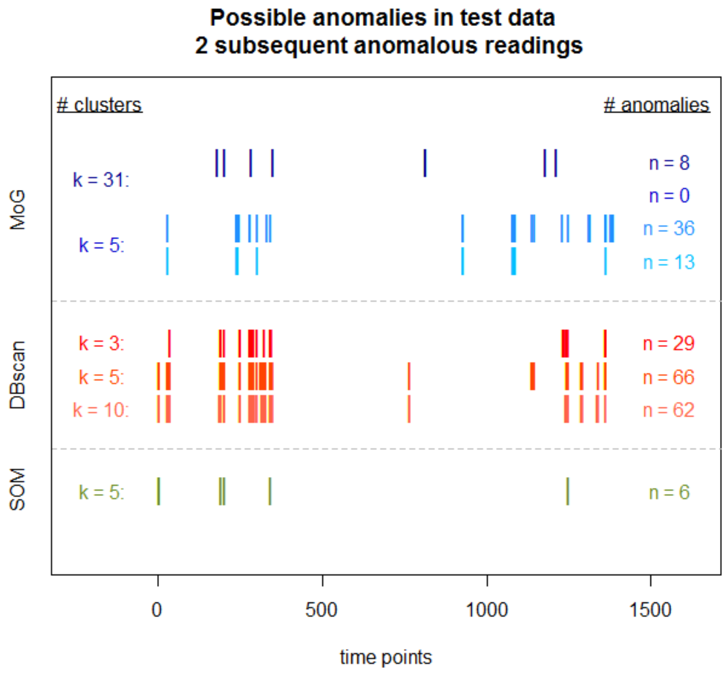

Figure 16. Comparing the possible anomalies detected by the different clustering methods when requiring 2 subsequent anomalous readings to flag an anomaly. Vertical lines corresponds to time points where the different methods would flag an alarm.

4.2. Time-dependencies in the data

The sensor data analysed in this study are essentially time series data, with temporal dependencies on various scales, both across different sensor streams and within the same signal. These temporal cross- and autocorrelations have not been taken into account, and the observations are simply regarded as independent observations of the marine engine system. Presumably, there are information in the temporal dependencies, and it may be that better and more robust detection strategies could have been developed if the time-dependence are taken into account. One approach to deal with these is to apply a suitable time-series model to the data in the preprocessing step to obtain residuals free from auto-correlation and then do the subsequent analysis on the residuals. This route, however, was not taken in this study. In general, time-series data should not be treated as independent observations of a system. For example, it is well known that autocorrelation might influence the cross-correlation between time series (Yule 1926). In this study, principal component analysis is performed and these are, by definition, linearly uncorrelated, so the linear cross-correlation between the transformed sensor signals is zero.

In time series the ordering of the data is meaningful, as opposed to independent data. This has an effect on how the data should be split in different parts, e.g. in a training set and a test set, see e.g. Bergmeir and Benitez (2012); Bergmeir et al. (2018). This is ignored in this study, and the splitting of the data is done completely at random. This splitting of the data into two parts without accounting for the autocorrelation will presumably give more similar training- and test-data than what would be the case if the data had been truly independent. This is reflected in the results, where the clustering on the training- and the test data yields very similar distribution of observations across the clusters. On the other hand, it ensures the representativeness of the training data compared to the test data, which will be discussed further in the following subsection.

4.3. Importance of representative training data

If data-driven methods are trained on a dataset that is not representative of new observations, one cannot expect the methods to perform well on new data. In unsupervised anomaly detection, the implicit assumption is that the training data contain measurements of the system in all normal conditions, and that if new observations exhibit very different characteristics they will be regarded as anomalies, for example due to faults in the system or due to deviation from nominal operation of the system. If measurements of some normal conditions are not included in the training data, future measurements under such conditions may be categorized as an anomaly even though it is perfectly normal. On the other hand, if the training data contain extensive measurements from a faulty or wrongly operated system, this would be regarded as normal and the method would fail to identify similar future measurements as anomalies.

In the study presented herein, anomaly detection is performed on sensor signals collected from a main generator engine onboard a ship in operation. Even though the data are time-series data, the splitting between training data and test data was done completely random, ignoring the temporal ordering of the data. This is not entirely correct for time series data, but it ensures that the training data is a good representation of the test data. To illustrate how important this is the clustering methods presented in this paper are repeated with a different separation of the data into training and test-sets; the training data will be chosen to be the first 75% of the sensor measurements, whereas the last 25% are kept as test data. In this case, the training data will be less similar to the test data. Only the anomaly detection based on mixture of Gaussian modelling is reported, but similar results are found for the other methods. Thus, a mixture of Gaussian model with $k = 5$ is fitted to the training data and the test data are assigned to one of the mixtures as outlined above.

With this setup, the test data are distributed differently to the various clusters compared to the training data and this suggests that the ship has been operated differently during the training phase and the test phase. If one calculates the p-values corresponding to the Gaussian mixtures as outlined above, the anomaly rate in the training data becomes 3.75%, but the anomaly rate in the test data is almost 75%. This is obviously too high, indicating that there is something wrong with the engine in 75% of the time during the test phase. These data contain no known faults, so this is clearly not the case, but is an effect of the training data not being representative for the test data.

The above demonstrates the importance of having a representative training data set for doing data-driven anomaly detection based on sensor data. However, it is not straightforward to obtain such a representative training data. Splitting the data at random is demonstrated to yield two subsets that are representative of one another. However, there is no guarantee that any of these subsets are representative of future observations of the system. This would be the case when one employs anomaly detection in an actual online condition monitoring system. In that case, one would need to have training data that one could reasonably assume to be representative for *all* future measurements of the system, in that

- The training data contain observations corresponding to all possible nominal conditions in order to avoid false alarms
- The training data contain no observations from a faulty or wrongly operated

system in order to avoid missed alarms

Obviously, it is difficult to ensure that these conditions are fully met, but one way to fulfil the first condition is to extend the coverage of the data used for training. In the case of ship monitoring systems, the training data should cover all operational modes and all environmental conditions the ship is believed to be operating in. This means that training data covering several years of operation should be collected to cover normal variations due to different seasons, different trades, different fuel quality, different operations, etc. In order to comply with the second condition, one may need to perform some cleaning of the data to reduce the amount of anomalous observations in the training data.

Notwithstanding these difficulties in obtaining a representative training data, this study demonstrates that various cluster methods can be used in different ways for anomaly detection on sensor data, and that the various methods perform reasonably well if the training data is representative of future measurements.

4.4. Information loss due to dimensionality reduction

Principal component analysis is performed in order to reduce the dimensionality of the problem, i.e. from 23 to 7. This makes the anomaly detection problem more manageable and the algorithms runs much faster. Moreover, it was found that almost all information content in the sensor data would be preserved; 99.5% of the variation in the data will be explained by the first 7 principal components. Typically, the first principal components are assumed to contain the signal in the data, whereas the last principal components contains mostly noise.

However, it may be that the last principal components will be most affected by certain types of anomalies. For example, if faults in the systems affects the noise more than the actual signal. In order to check if this is a problem with the current dataset, the cluster-based anomaly detection methods are carried out on the 7 last principal components. The training data now consist of the 7 last principal components of the training data that was analysed above and the test data is the last principal components of the previous test data.

Applying a mixture of Gaussian models on these data, the BIC criterion suggests a mixture of 4 components, whereas the ICL criterion suggests 2. This indicates that the data structure is less complex in the last principal components. Assuming a model with 4 clusters, the anomaly detection scheme based on the Mahalanobis distance now yields anomaly rates of 0.19% and 0.22%, respectively in the training and test data. If two subsequent anomalous readings are required to trigger an alarm, no alarms will be triggered in the training data, and only one in the test data. Similar results are obtained with the other clustering methods.

Thus it is demonstrated that using the last principal components detects much fewer alarms. However, it also illustrates that some possible anomalies can be detected from analysing the least varying principle components, and these are not necessarily the same time points as the anomalies in the first principal components. Hence, there is a risk of loosing this information when applying dimensionality reduction. It is not entirely clear whether these were false alarms or indeed real anomalies, and data with known faults would be needed in order to assess this. In general it is difficult a priori to know in which principal component a possible fault will be detectable, and it may vary for different types of faults.

One possible compromise between the need for dimensionality reduction to make the problem manageable and the need to minimize the risk of throwing away important information is to run several models in parallel, each monitoring a subset of the principal components. In this way, all principal components would be monitored, and each model would be manageable. Obviously, some information would still be lost, e.g. regarding the inter-dependencies between the subsets of data streams, but since the principle components are linearly uncorrelated the effect of this is presumably small. Notwithstanding, this study has demonstrated that far more possible anomalies are detected by only looking at the most varying principal components. Thus, it is believed to be reasonable to base anomaly detection routines on the first principal components in most cases.

5. Summary and conclusions

This paper has presented a study on the use of cluster-based methods for unsupervised anomaly detection of ship machinery sensor data for condition monitoring. In particular, four very different approaches are explored, based on a mixture of Gaussian models, density based clustering, self-organizing maps and support vector machines, respectively. The methods are simple to use and have been found to perform well on the sensor data from a marine engine system, and were able to detect a reasonable number of anomalies. However, all the algorithms have different parameters that needs to be determined and fine-tuning and validation would need to be carried out before the methods can be employed in actual online condition monitoring systems.

One advantage of the methods presented in this paper, compared to other methods that has recently been proposed, is that they are truly unsupervised. All methods except one are able to account for faulty training data and can work well even if some erroneous measurements are used to train the models. This is deemed to be important, since it is very difficult to ensure that sensor data are completely without faults. In terms of detection rates, the different methods are comparable, and there are great overlap between the times the different methods would flag an alarm. However, it is suggested that more robust detection algorithms can be obtained by combining the different methods in ensembles. However, further investigations on the optimal combinations and detection strategies are recommended for future research. It is demonstrated that representative training data is crucial, something that is of paramount importance for all data-driven methods, and this is generally difficult to guarantee. Notwithstanding, this study has demonstrated the usefulness of cluster-based methods for anomaly detection in condition monitoring systems of ship machinery systems.

Acknowledgement

The study presented in this paper is partly carried out within the centre for research-based innovation, BigInsight.

References

Baudry JP, Raftery AE, Celeux G, Lo K, Gottardo R. 2010. Combining mixture components for clustering. Journal of Computational and Graphical Statistics. 19:332–353.

Bergmeir C, Benitez JM. 2012. On the use of cross-validation for time series predictor evaluation. Information Sciences. 191:192–213.

Bergmeir C, Hyndman RJ, Koo B. 2018. A note on the validity of cross-validation for evaluating autoregressive time series prediction. Computational Statistics and Data Analysis. 120:70–83.

Brandsæter A, Manno G, Vanem E, Glad IK. 2016. An application of sensor based anomaly detection in the maritime industry. In: Proc. IEEE PHM2016; June. IEEE Reliability Society.

Brandsæter A, Vanem E, Glad IK. 2017. Cluster based auto associative kernel regression with applications in the maritime industry. In: Proc. 2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC 2017); August. IEEE Reliability Society.

Brandsæter A, Vanem E, Glad IK. 2019. Efficient on-line anomaly detection for ship systems in operation. Expert Systems With Applications. 121:418–437.

Campello RJ, Moulavi D, Sander J. 2013. Density-based clustering based on hierarchical density estimates. In: Pei J, Tseng V, Cao L, Motoda H, Xu G, editors. Advances in knowledge discovery and data mining. pakdd 2013. Springer; p. 160–172. Lecture Notes in Computer Science, vol 7819.

Campello RJGB, Moulavi D, Zimek A, Sander J. 2015. Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Transactions on Knowledge Discovery from Data. 10:5:1–5:51.

Cipollini F, Oneto L, Coraddu A, Murphy AJ, Anguita D. 2018. Condition-based maintenance of naval propulsion systems with supervised data analysis. Ocean Engineering. 149:268–278.

Dimopoulos GG, Georgopoulou CA, Stefanatos IC, Zymaris AS, Kakalis NM. 2014. A general-purpose process modelling framework for marine energy systems. Energy Conversion and Management. 86:325–339.

Garvey J, Garvey D, Seibert R, Hines JW. 2007. Validation of on-line monitoring techniques to nuclear plant data. Nuclear Engineering and Technology. 39:149–158.

Hastie T, Tibshirani R, Friedman J. 2009. The elements of statistical learning. 2nd ed. Springer.

Haver S, Winterstein S. 2009. Environmental contour lines: A method for estimating long term extremes by a short term analysis. Transactions of the Society of Naval Architects and Marine Engineers. 116:116–127.

Hines JW, Garvey DR. 2006. Development and application of fault detectability performance metrics for instrument calibration verification and anomaly detection. Journal of Pattern Recognition Research. 1:2–15.

Huseby AB, Vanem E, Natvig B. 2013. A new approach to environmental contours for ocean engineering applications based on direct Monte Carlo simulations. Ocean Engineering. 60:124–135.

Huseby AB, Vanem E, Natvig B. 2015. Alternative environmental contours for structural reliability analysis. Structural Safety. 54:32–45.

Kohonen T. 1982. Self-organized formation of topologically correct feature maps. Biological Cybernetics. 43:59–69.

Lamaris V, Hountalas D. 2010. A general purpose diagnostic technique for marine diesel engines - application on the main propulsion and auxiliary diesel units of a marine vessel. Energy Conversion and Management. 51:740–753.

Maftei C, Moreira L, Guedes Soares C. 2009. Simulation of the dynamics of a marine diesel engine. Journal of Marine Engineering & Technology. 8:29–43.

Martin E, Kriegel HP, Sander J, Xu X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96); August. Association for the Advancement of Artificial Intelligence (AAAI).

Raptodimos Y, Lazakis I. 2018. Using artificial neural network-self-organising map for data clustering of marine engine condition monitoring applications. Ships and Offshore Structures. 13:649–656.

Serinaldi F. 2015. Dismissing return periods! Stochastic Environmental Research and Risk

Assessment. 29:1179–1189.

Vanem E. 2018a. A simple approach to account for seasonality in the description of extreme ocean environments. Marine Systems & Ocean Technology. 13:63–73.

Vanem E. 2018b. Statistical methods for condition monitoring systems. International Journal of Condition Monitoring. 8:9–23.

Vanem E, Storvik GO. 2017. Anomaly detection using dynamical linear models and sequential testing on a marine engine system. In: Proc. Annual Conference of the Prognostics and Health Management Society 2017 (PHM 2017); October. PHM Socitey.

Vanem E, Storvik GO. 2018. Dynamical linear models for condition monitoring with multivariate sensor data. International Journal of Condition Monitoring and Diagnostic Engineering Management. 21:7–18.

Yule GU. 1926. Why do we sometimes get nonsense-correlations between time-series? - a study in sampling and the nature of time-series. Journal of the Royal Statistical Society. 89:1–63.

Zacharewicz M, Kniaziewicz T. 2017. Modelling of the operating process in a marine diesel engine. Journal of Marine Engineering & Technology. 16:193–199.

Zymaris AS, Alnes ØÅ, Knutsen KE, Kakalis NMP. 2016. Towards a model-based condition assessment of complex marine machinery systems using systems engineering. In: Proc. Third European Conference of the Prognostics and Health Management Society 2016; July. PHM Society.