UiO : University of Oslo

Andreas Brandsæter

Data-driven methods for multiple sensor streams, with applications in the maritime industry

Thesis submitted for the degree of Philosophiae Doctor

Department of Mathematics
The Faculty of Mathematics and Natural Sciences

Group Technology & Research
DNV GL

2020

Doktoravhandlinger forsvart ved
Det matematisk-naturvitenskapelige *fakultet, Universitetet i Oslo.*
Nr. 2237

ISSN 1501-7710

Omslag: Hanne Baadsgaard Utigard.
Grafisk produksjon: Reprosentralen, Universitetet i Oslo.

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of *Philosophiae Doctor* at the University of Oslo. The research is carried out between August 2015 and November 2019, under supervision by Professor Ingrid K. Glad (University of Oslo) and Associate Professor Erik Vanem (DNV GL and University of Oslo).

The doctoral project is carried out in collaboration between the University of Oslo and DNV GL, funded under the Industrial Ph.D. scheme of the Norwegian Research Council (project number 251396). Furthermore, the research is conducted in close collaboration with the research-based innovation centre Big Insight, also funded by the Norwegian Research Council (project number 237718).

The thesis is a collection of five papers. The papers are preceded by an introductory part providing background, context and motivation for the work.

Acknowledgements

I truly appreciated the support, guidance, supervision and advise from my supervisors, Ingrid K. Glad and Erik Vanem. Thank you for your dedication, support and collaboration.

I would also like to thank my colleagues at DNV GL, in particular Odin Gramstad, Knut Erik Knutsen and Gabrielle Manno for their collaboration and co-authorship, and my managers Bjørn-Johan Vartal, Hans Anton Tvete, Rune Torhaug and Pierre C. Sames for their support and sponsorship.

I also thank my colleagues at the University of Oslo and representatives from Big Insight and its partners for their support and interest in my research. In particular, I would like to thank my co-supervisors Magne Aldrin (Norwegian Computing Center), Geir O. Storvik (University of Oslo) and Arne Huseby (University of Oslo), and additionally Arnoldo Frigessi (University of Oslo), Martin Tveten (University of Oslo), Mette Langaas (Norwegian University of Science and Technology) and Martin Jullum (Norwegian Computing Center).

Finally, I would like to express gratitude to my parents, my siblings and their families, and my in-laws. To my closest ones, my wife Maria and our children Jakob and Jenny, thank you for the love you bring into my life.

Andreas Brandsæter
Oslo, November 2019

List of publications

The following papers are included in this thesis:

I Brandsæter, A. and Vanem, E. (2018). Ship speed prediction based on full scale sensor measurements of shaft thrust and environmental conditions. *Ocean Engineering*, 162:316 – 330.

II Brandsæter, A., Vanem, E., and Glad, I. K. (2019). Efficient on-line anomaly detection for ship systems in operation. *Expert Systems with Applications*, 121:418 – 437.

III Vanem, E. and Brandsæter, A. (2019). Unsupervised anomaly detection based on clustering methods and sensor data on a marine diesel engine. *Journal of Marine Engineering & Technology*, 1 – 18.

IV Brandsæter, A. and Knutsen, K. (2018). Towards a framework for assurance of autonomous navigation systems in the maritime industry. In *Safety and Reliability–Safe Societies in a Changing World: Proceedings of ESREL 2018*, (pp. 449 – 457). CRC Press.

V Brandsæter, A. and Glad, I. K. (2019). Explainable artificial intelligence: How subsets of the training data affect a prediction. *Submitted for publication.*

The following papers are also written as part of the doctoral project:

VI Brandsæter, A., Manno, G., Vanem, E., and Glad, I. K. (2016). An application of sensorbased anomaly detection in the maritime industry. In *2016 IEEE international conference on prognostics and health management (ICPHM)* (pp. 1-8). IEEE.

VII Brandsæter, A., Vanem, E., and Glad, I. K. (2017). Cluster-based anomaly detection with applications in the maritime industry. In *2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)* (pp. 328–333). IEEE.

VIII Vanem, E., Brandsæter, A., and Gramstad, O. (2016). Regression models for the effect of environmental conditions on the efficiency of ship machinery systems. In *Risk, Reliability and Safety: Innovating Theory and Practice: Proceedings of ESREL 2016* (pp. 362-371). Lesley Walls.

IX Vanem, E. and Brandsæter, A. (2018). Cluster-based anomaly detection in condition monitoring of a marine engine system. In *2018 Prognostics and System Health Management Conference* (pp. 20–31). IEEE.

Contents

Chapter 1

Introduction

This thesis consists of five papers concerning the development and assurance of data-driven methods for various applications, mainly in the maritime industry, including analysis of multiple sequential sensor streams, anomaly detection, classification and regression, and explainability and interpretation of black-box models.

Use-cases are mainly selected from the maritime industry, however the methods presented are generally applicable to many industries and domains, in particular safety critical applications involving high-consequence scenarios.

Both traditional statistical methods and modern machine learning methods are studied. We avoid the (sometimes interesting) debate on the difference between statistics and machine learning (see for example Bzdok et al. (2018)), and use terminology from statistics and machine learning interchangeably. We strive to avoid repeating information from the papers. However, to enable the discussion presented in the synopsis, brief descriptions are occasionally retrieved from the papers.

In the following, we describe the main aims of the thesis. Furthermore, we discuss the scope and limitations of the work. In Chapter 2, we introduce basic theory, providing background and context for the five papers. Summaries of the five papers are provided in Chapter 3. We discuss challenges, limitations and propose topics for future research in Chapter 4. In Chapter 5, we conclude. Finally, the five papers are included.

1.1 Aims and scope

One of the main aims of this thesis is to develop methods for data-driven prediction and anomaly detection, and several modifications and enhancements are proposed to improve existing anomaly detection techniques.

We study how the proposed methods can be implemented for various applications, also in safety critical domains. When the consequences of faulty predictions are low, the path from algorithm and model development to full scale implementation can be relatively short. For such applications, increased accuracy is often enough to justify implementation. For safety critical applications however, trust and confidence in the models are required before implementation. Hence, reliable estimation of future performance is required, and we investigate different evaluation techniques.

But trust can include more than confidence that a model will perform well. Even if a model is demonstrated to be sufficiently accurate, we might be reluctant to implement it if we do not understand how it works (Lipton 2016). Therefore a second main aim is to develop methods to explain and interpret predictions

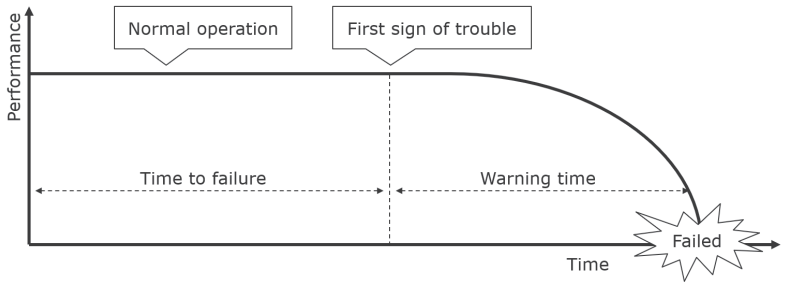


Figure 1.1: Illustrating the concept behind condition-based monitoring. (Adapted from (Knutsen et al. 2014))

and classifications of black-box models. A novel training data centric approach to explain and interpret data-driven methods is proposed.

The thesis is conducted in close collaboration with DNV GL, a global quality assurance and risk management company. DNV GL issues classification certificates and provides technical assurance, software and independent expert advisory services to different industries. A key challenge for DNV GL is to assure and verify systems which are based on data-driven methods.

1.2 Motivation

For many years, run to failure was the most common maintenance strategy. "If it ain't broke, don't fix it"[1] might be an adequate, and sometimes even preferred, maintenance strategy for many applications. For safety critical applications, however, the cost of an accident is often too high, and preventive maintenance regimes are often implemented where system components are maintained or replaced according to a time-schedule. The assumption behind such a strategy is that a component has a defined lifetime, after which its failure rate increases (Knutsen et al. 2014). However, Nowlan and Heap (1978) analysed failures on aircraft equipment and found that as much as 89 % of the failures were not age-related. Similar results are shown for the maritime industry, although slightly lower (Allen 2001). This demonstrate an important deficiency with preventive maintenance, and motivates condition-based monitoring (see Figure 1.1). In condition-based maintenance, we assume that some physical change occurs in the component or system before a failure occurs, and that this can be detected using appropriate sensors (Knutsen et al. 2014).

In the last decades, affordable sensors and data storage have enabled massive collection of sensor data in various industries, including the maritime industry. An increasing number of ships are equipped with sensor systems, offering high

[1]Widely attributed to Thomas Bertram Lance, Director of the Office of Management and Budget in Jimmy Carter's 1977 administration, who argued that the government could save billions if it adopted this simple motto. https://www.phrases.org.uk/meanings/if-it-aint-broke-dont-fix-it.html

frequency measurements which are used to monitor both the ship's performance and condition as well as the ship's operating environment. The captured sensor data can contain information which can contribute to achieve improvements both in operation and design.

Valuable information can be well hidden in the sensor data, and we need statistical methods to transform the data into insight. Increasingly complex models are used to capture the intricate relationships in large datasets. These models are often referred to as black-box models, since we do not understand their inner workings. One can however argue, that no models are intrinsically interpretable, and sufficiently high-dimensional models, for example deep decision trees, can be considered less transparent than comparatively compact neural networks (Lipton 2016).

Nevertheless, as black-box models, or machine learning models, are taking an increasingly important part in new applications, the inability of humans to understand the machine learning models seems problematic (Caruana et al. 1999; Lipton 2016). Hence, the importance of transparency, explainability and interpretability of machine learning models is growing, particularly for decision making in safety critical systems (Kim et al. 2016). If we understand the model's reasoning, it is easier to verify the model and determine when the model's reasoning is in error, and to improve the model (Caruana et al. 1999; Doshi-Velez and Kim 2017; Lundberg and Lee 2017). Doshi-Velez and Kim (2017) argue that explanations and interpretations can be important to ensure safety since we often cannot create a complete list of training scenarios in which a system can fail. Furthermore, transparency, explainability and interpretability can guard against unethical or biased predictions, such as discriminations, and we can better deal with competing objective functions of the algorithms, such as privacy and prediction quality (Doshi-Velez and Kim 2017). Interpretation also lets us learn from the model, and convert interpretations and explanations into knowledge (Shrikumar et al. 2016).

Chapter 2

Background

In this chapter, we briefly describe learning methods. We provide a general description of anomaly detection methods and frameworks, and explain how the available methods can be divided into three categories; supervised, unsupervised and semi-supervised methods. An anomaly detection method with signal reconstruction followed by residual analysis is presented in more details. We also discuss anomaly detection techniques based on clustering. In safety critical applications, a key challenge is lack of trust and confidence in the outputs of machine learning models. Therefore, an important focus throughout this thesis is reliability and robustness of data-driven methods. We discuss challenges related to explainability and interpretation and provide a brief description of the most important and popular methods. We also discuss how a method's performance should be measured, and discuss challenges related to testing and cross-validation.

2.1 Learning methods

Widespread use of artificial intelligence and machine learning is seen for a number of applications, including anomaly detection, regression and classification. In the machine learning literature, a distinction between supervised and unsupervised learning is common. In supervised learning, we denote some of the variables as inputs which affect some output variables (Hastie et al. 2009, Ch. 2). Typically, the task in supervised learning is to model the relationship between the input variables and the outputs. When the task is to determine membership of a class, and the model is trained with labelled data, we call it classification. Regression typically concerns continuous data. The models include both parametric methods such as linear models, as well as non-parametric models such as k-nearest neighbours and decision trees. Parametric models are learning models that summarize data with a set of parameters of fixed size, while models that cannot be characterized by a bounded set of parameters are called non-parametric (Russell and Norvig 2016, Ch. 18). In unsupervised learning, the properties of the data distribution are directly inferred without the use of explicitly provided labels (Hastie et al. 2009, Ch. 14). Clustering can for example be performed on unlabelled data, where the goal is to discover a natural grouping of the observed data (HajKacem et al. 2019). Semi-supervised learning usually refers to problems where only a small portion of the observed data is labelled (HajKacem et al. 2019).

2.2 Anomaly detection

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behaviour (Chandola et al. 2009). Change points are found where the distributional properties of the considered dataset change (Killick et al. 2012).

Data-driven anomaly detection techniques are alternatives to model-based approaches based on physical modelling of the system from first principles (See for example (Cipollini et al. 2018; Dimopoulos et al. 2014; Lamaris and Hountalas 2010; Zymaris et al. 2016)), which may be more difficult to use (Vanem and Brandsæter 2019). An extensive number of data-driven anomaly detection techniques are described in literature and used in a wide variety of applications in various industries. The available techniques comprise classification methods that are rule-based, or based on Neural Networks, Bayesian Networks or Support Vector Machines; nearest neighbour based methods, including k nearest neighbour and relative density; clustering based methods; statistical and fuzzy set-based techniques, including parametric and non-parametric methods (Chandola et al. 2009; Kanarachos et al. 2017; Laxhammar et al. 2009; Olson et al. 2018; Steinwart et al. 2005; Zheng et al. 2016).

In Brandsæter et al. (2019), we divide the fundamental approaches to data-driven anomaly detection into three categories (Chandola et al. 2009; Hodge and Austin 2004):

- *Supervised anomaly detection* Availability of a training dataset with labelled instances for normal and anomalous behaviour is assumed. Typically, a classifier is trained to distinguish between normal and anomalous observations, and unseen data are assigned to one of the classes.

- *Unsupervised anomaly detection* Here, the training dataset is not labelled, and an implicit assumption is that the normal instances are far more frequent than anomalies in the test data. If this assumption is not true, such techniques suffer from high false alarm rate and/or missed detection rate.

- *Semi-supervised anomaly detection* In semi-supervised anomaly detection, the training data only includes normal data. A typical anomaly detection approach is to build a model for the class corresponding to normal behaviour, and use the model to identify anomalies in the test data. Since the semi-supervised methods do not require labels for the anomaly class, they are more widely applicable than supervised techniques.

Note that the definition of semi-supervised anomaly detection differs from the definition of semi supervised learning as described in section 2.1. In this setting, semi-supervised learning refers to problems where only a small portion of the observed data are labelled, while in the anomaly detection setting, the full dataset is labelled but all samples originate from the normal class.

The choice of anomaly detection technique depends on the application. Our interest in anomaly detection is primarily motivated in condition-based maintenance and fault detection and prediction in the maritime industry. We often lack essential knowledge and data of the fault-process, and we are therefore not able to accurately and reliably predict failures. Due to this, our focus is on detecting anomalous behaviour, potentially indicating a first sign of trouble (see Figure 1.1). We strive to accurately determine when anomalous behaviour occurs. The detection delay, that is the time between the occurrence of an anomaly, and the time it is detected, should be minimized, hence on-line methods are preferred.

Loosely speaking, the fire alarm should warn you early enough before a fire, enabling you to take preventive actions. However, minimum detection delay has to be balanced with a low false alarm rate. Furthermore, transients between different operational modes should not be identified as anomalous behaviour.

2.2.1 Anomaly detection with signal reconstruction followed by residual analysis

In Brandsæter et al. (2016), Brandsæter et al. (2019), and Brandsæter et al. (2017), we use an on-line anomaly detection technique to satisfy the requirements outlined above. The technique we use consists of two steps, where first, the sensor signal is reconstructed under normal conditions, and secondly, the residuals, that is the difference between the reconstructed signal and the observed signal, are analysed to identify anomalies.

Hines and D. R. Garvey (2006) used Auto Associative Kernel Regression (AAKR) for signal reconstruction, and analysed the residuals using Sequential Probability Ratio Test (SPRT) (see Figure 2.1), for on-line monitoring of a model of a nuclear power plant steam system. Similar more recent work are performed by for example Di Maio et al. (2013) and Li et al. (2017) who also use this approach to monitor the condition of sensors on a nuclear power plant. In the latter, simulations of fault detection and identification on the sensors and components in the reactor coolant system are carried out. Boechat et al. (2012) combine AAKR and SPRT for drift correction and detection in oil well sensors monitoring, Kappaganthu et al. (2010) use the approach for model-based diagnostics of an aircraft generator, and Niu et al. (2015) integrates the on-line anomaly monitoring approach using AAKR and SPRT with a model-based strategy for system fault modelling of a multi-energy domain dynamic system. Additionally, they propose to use linear fractional transformations-based bond graph for physical parameter uncertainty modelling.

Several different methods can be used to reconstruct the signals, and to analyse the residuals. Baraldi et al. (2015a) compare the AAKR reconstruction method with two other data-driven signal reconstruction methods: fuzzy similarity (FS) (Zio and Di Maio 2010) and Elman recurrent neural networks (RNN) (Seker et al. 2003). Capabilities and drawbacks of the different methods are presented. In the evaluated cases, AAKR is reported as the fastest in triggering alarms in case of anomalous conditions. However, it is the least

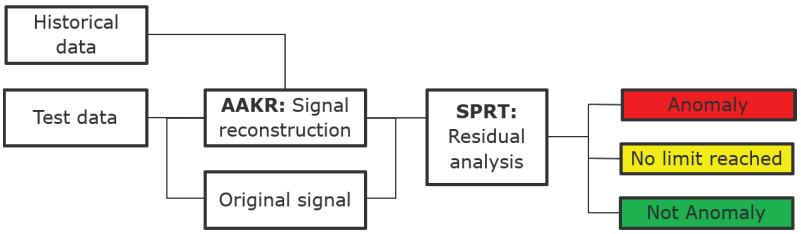


Figure 2.1: The methodology can be divided into two main steps: signal reconstruction (via AAKR) and analysis of residuals (via SPRT)

resistant to the spillover effect which occurs when anomalies are detected in signals with normal behaviour (Baraldi et al. 2015b). The recommendation of Baraldi et al. (2015a) is to use an ensemble of the three methods. Based on the examples in the study, it is reported that the ensemble method provides more satisfactory results, overcoming the limitations of each method while exploiting their strengths. However, the use of ensemble methods impose challenges related to voting strategies, and deciding which models to include, taking into consideration the individual methods accuracy and diversity (Wang 2008). Our focus is on improving the AAKR method, hopefully also leading to improved ensemble methods.

Regression models can also be used in the reconstruction step. For example, Vanem and Storvik (2017) compare the predictions produced by dynamical linear models (DLM) with the observed values, and Vanem and Brandsæter (2018) and Vanem and Brandsæter (2019) use self-organizing maps.

2.2.1.1 Signal reconstruction using AAKR

Since descriptions of Auto Associative Kernel Regression (AAKR) did not readily appear in the open literature at that time, Hines and D. R. Garvey (2006) provided a description which was derived based upon multivariate, inferential kernel regression as derived by Wand and Jones (1995). In the following, we briefly introduce the AAKR method following this description. For other excellent descriptions of the AAKR method, both comprehensive and more brief, see for example Baraldi et al. (2015a), Baraldi et al. (2011), Baraldi et al. (2012), Baraldi et al. (2015b), Brandsæter et al. (2016), Brandsæter et al. (2019), Di Maio et al. (2013), J. Garvey et al. (2007), and Hines et al. (2008).

Auto Associative Kernel Regression (AAKR) is a data-driven method where the reconstructed signal is estimated as a weighted linear combination of historical observations. The information from the current observation is used to calculate the weights. The methodology follows the following procedure: At each time t in the test data, a reconstruction of a test point $\mathbf{x}^{test}(t) = [x(t,1), ..., x(t,J)]$ is calculated as a weighted linear combination of the observations (the rows) in a training matrix $\mathbf{X}^{train}$. The weight $\mathbf{w}$ of a row k of the training data is given by

the Gaussian kernel

$$\mathbf{w}_{t,k} = \frac{1}{\sqrt{2\pi}h} e^{-\frac{\mathbf{d}_k^2}{2h^2}}, \tag{2.1}$$

where the parameter h is the bandwidth, and $\mathbf{d}_{t,k}$ is the distance between the J signal measurements in the observation $\mathbf{X}_{(t,)}^{test}$ and the k-th observation in $\mathbf{X}^{train}$, for $k = 1, \ldots, K$. Several distance functions can be used (J. Garvey et al. 2007), but the most common is the Euclidean norm

$$\mathbf{d}_k = \sqrt{\sum_{j=1}^{J} \left(\mathbf{X}_{(t,j)}^{test} - \mathbf{X}_{(k,j)}^{train} \right)^2}. \tag{2.2}$$

Finally, the reconstructed value $\hat{\mathbf{X}}_{(t,j)}^{test}$ of the j-th observation $\mathbf{X}_{(t,j)}^{test}$, is given as the weighted linear combination of the rows of the training matrix, that is

$$\hat{\mathbf{X}}_{(t,j)}^{test} = \frac{\sum_{k=1}^{K} \mathbf{w}_k \cdot \mathbf{X}_{(k,j)}^{train}}{\sum_{k=1}^{K} \mathbf{w}_k}. \tag{2.3}$$

2.2.1.2 Residuals analysis using SPRT

Once a reconstruction is produced, the residual, i.e. the difference between the observed signal and the reconstructed signal, is analysed using Sequential Probability Ratio Test (SPRT). SPRT is a statistical technique developed by Wald (1947) which we use to determine whether the residual from a prediction is caused by a faulted system or if it is due to normal process and instrumentation variations (Hines and D. R. Garvey 2006). We briefly describe the methodology in the following. For a more thorough description we suggest Brandsæter et al. (2016), Brandsæter et al. (2019), Cheng and Pecht (2012), Gross and Lu (2004, May 11), and Saxena et al. (2008).

The residuals, $\mathbf{R} = \hat{\mathbf{X}}^{test} - \mathbf{X}^{test}$, are analysed sequentially by the standard Sequential Probability Ratio Test (SPRT) to determine if the system is in normal or abnormal state. The normal state is described by a null hypothesis H_0, where each component of the residuals, $\mathbf{R}_{(t,j)}$, are assumed to be normally distributed with mean 0 and standard deviation σ. The anomalous state is described by an alternative hypothesis H_a, which assumes that the residuals are normally distributed with specified mean and/or standard deviation different from the null hypothesis.

Based on the residuals $\mathbf{R}_{(t,j)}$, an index is calculated and updated sequentially for each new observation. In order to determine the condition of the system, two threshold values are specified and at each observation the index is compared to these lower and upper decision boundaries. There are three possible outcomes at each time step:

1. the lower limit is reached, in which the null hypothesis is accepted (normal state), and the test statistic is reset.

2. the upper limit is reached, in which the null hypothesis is rejected (anomalous state), and the test statistic is reset.

3. no limit is reached, in which case the amount of information is not sufficient to make a conclusion.

For each sensor signal j, the analysis is performed independently on the sequence of residuals $\mathbf{R}_{(t_1,j)}, \ldots, \mathbf{R}_{(t_n,j)}$, where t_n denotes the current time point, and t_1 denotes the time point when the test statistic was last reset. When either of the limits are reached (outcome 1 and 2), the sequence is reset to zero. If no limits are reached (outcome 3), the sequence is extended with the new residual.

The SPRT index is given as the natural logarithm of the likelihood ratio L_a, given by

$$L_a = \frac{\text{prob of } \mathbf{R}_{(t_1,j)}, \ldots, \mathbf{R}_{(t_n,j)} \text{ given } H_a}{\text{prob of } \mathbf{R}_{(t_1,j)}, \ldots, \mathbf{R}_{(t_n,j)} \text{ given } H_0} = \prod_{t=t_1}^{t_n} \frac{f_a(\mathbf{R}_{(t,j)})}{f_0(\mathbf{R}_{(t,j)})},$$

where $f(\cdot)$ is the corresponding normal density. Note that this construction is based on an assumption of normally distributed residuals, and independence among the residuals.

Alternative hypotheses can be evaluated to detect changes in the mean, variance and/or covariance (Tveten 2017). For example, to detect positive and negative changes in the mean for each sensor j, the following indices are used:

$$SPRT_1 = \frac{m}{\sigma^2} \sum_{t=t_1}^{t_n} \left(\mathbf{R}_{(t,j)} - \frac{m}{2} \right) \tag{2.4}$$

$$SPRT_2 = \frac{m}{\sigma^2} \sum_{t=t_1}^{t_n} \left(-\mathbf{R}_{(t,j)} - \frac{m}{2} \right) \tag{2.5}$$

The standard deviation, σ, is computed from the training data. m is the mean value of the alternative hypothesis, which is decided by the user. m is usually chosen to be several times larger than σ (Cheng and Pecht 2012).

2.2.2 Anomaly detection based on clustering methods

Alternatives to the two-step process with signal reconstruction and residual analysis as described above, include methods based on clustering. Clustering refers to the division of data into groups of similar objects (Berkhin 2006), and instances in the different clusters should be as different as possible (D. Xu and Tian 2015). Numerous clustering methods exist including hierarchical methods, partitioning relocation methods, density-based methods and grid-based methods. We refer to D. Xu and Tian (2015) and Berkhin (2006) for two comprehensive surveys.

A common approach to cluster-based anomaly detection, is to first cluster the data, and then classify the data according to one of the following assumptions:

1. Normal data instances belong to a cluster in the data, while anomalies do not belong to any cluster (clusters with only one member).

2. Normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid.

3. Normal data instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters.

Various clustering techniques can be applied. For example, mixtures of Gaussian models (Hastie et al. 2009, Ch. 14) can be used to produce ellipsoid-shaped clusters of varying shapes and orientations. With this approach, a parametric model is fitted to the data, and anomalies can be identified where the sensor signals in the test data are extreme according to the established parametric model. Density-based clustering such as DBscan (Ester et al. 1996) and Hierarchical DBscan(Campello et al. 2013), can also be used. These clustering techniques group observations with many neighbours into clusters. In DBscan both the neighbourhood distance and the minimum number of core points per cluster is specified by the user. In the hierarchical extension, the neighbourhood distance can stay unspecified. Instead, a hierarchy of clusters for any neighbourhood distance are provided in a three-like structure. In both these frameworks, anomalies are identified for observations that do not belong to any of the clusters. Moreover, we can use support vector machines (SVM) (Hastie et al. 2009, Ch. 14) to formulate a classification problem with only one class representing normal data. Observations deviating from this one class are identified as anomalous. The above-mentioned methods are used and described in Vanem and Brandsæter (2019), and we refer to this paper for details and examples.

2.3 Explaining the output of any predictor

Many agree on the importance of interpretability, and explanations are sometimes required. For example, the EU General Data Protection Act (GDPR) provides individuals the right to receive an explanation for algorithmic decisions which significantly affect that individual (Goodman and Flaxman 2017). But it is not articulated precisely what interpretability means or why it is important. Lipton (2016) discusses the interpretability of human decision-makers, and what notion of interpretability these explanations satisfy, and argues that human explanations seem unlikely to clarify the mechanisms or the precise algorithms by which brains work. Nevertheless, the information conferred by human interpretation may be useful. Doshi-Velez and Kim (2017) propose to define interpretability as "the ability to explain or to present in understandable terms to a human".

Several methods are proposed and developed to interpret the black-box models and explain their predictions. Some of these methods are model-specific, that is, they can only be used on a subset of machine learning models, while other methods are model-agnostic, and these are the focus of this thesis. If a

task should be solved with machine learning methods, typically, several types of machine learning models are evaluated, and when comparing models in terms of interpretability, it is easier to work with model-agnostic explanations (Molnar 2019).

A popular and frequently used model-agnostic approach to interpret and explain the decisions and predictions is feature importance. For a linear regression model, the importance of different features is readily available, and various methods aim to provide a similar interpretation of more complex models. The available methods include perturbation methods (Breiman 2001; Fisher et al. 2018), local surrogate models such as LIME (Ribeiro et al. 2016), and Shapley values (Štrumbelj and Kononenko 2010; Štrumbelj and Kononenko 2011; Štrumbelj and Kononenko 2014). Since the predictions made by the data-driven methods rely heavily on the training data used, we also advocate explanations which convey how the training data affects the predictions. This includes case-based explanation methods which select particular points of the dataset to explain the behaviour of machine learning models (Caruana et al. 1999), and influence functions which tell us how the model parameters change when a point in the training dataset is up-weighted by an infinitesimal amount (Koh and Liang 2017) . We refer to Brandsæter and Glad (2019) for a brief description of selected popular methods.

2.4 Performance measures

When deciding if we should trust a model, we might care not only about how often a model's prediction and/or classification is right but also for which examples it is right (Lipton 2016). If the model tends to make mistakes in regions of input space where humans also make mistakes, and is typically accurate when humans are accurate, then the model may be considered trustworthy in the sense that there is no expected cost of relinquishing control. The severity of a missclassification should also be taken into consideration. For example, if a kayak is classified as a pleasure boat by one classifier and as an oil tanker by a second classifier, the performance of the first classifier can be regarded as better than the second classifier even though both classifications were wrong (Brandsæter and Knutsen 2018).

When evaluating performance in regression problems, metrics such as mean square error (MSE), mean absolute error (MAE) and R-squared are commonly used. In classification problems with few classes, error matrices (also called confusion matrices) are often used to communicate a model's performance (Stehman 1997). However, error matrices are impractical in cases with a high number of classes. When we evaluate anomaly detection methods, we are interested in the number of true and false positives (TP and FP) as well as the number of true and false negatives (TN and FN), where for example a true positive is an instance where an anomaly occurred, and the anomaly detection method successfully detected it. Anomaly detection methods should preferably achieve a high number of true positives and negatives and at the same time keep

the number of false positives and negatives at a minimum.

Two commonly used measures are sensitivity and specificity. Sensitivity is the true positive rate which has the following expression

$$TPR = \frac{TP}{TP + FN}. \tag{2.6}$$

Specificity is the probability of predicting that an instance is normal (non-anomalous) given that the true state is normal (non-anomalous). This information can also be presented as the False Positive Rate, which is given as 1 minus the specificity, that is:

$$FPR = \frac{FP}{FP + TN} = 1 - \text{specificity}. \tag{2.7}$$

The TPR and FPR are often presented in a receiver operating characteristics (ROC) graph, which is a scatterplot with the TPR on the vertical axis and the FPR on the horizontal axis. According to Fawcett (2006), the ROC graphs have properties that make them especially useful for domains with skewed class distribution and unequal classification error costs.

2.5 Performance measure estimation

Reliable estimates of the accuracy of a model on future unseen data is essential when deciding how the model should be used, especially for safety critical operations (Wolpert 1992). If we assess the accuracy of a model on the data which is used to train the model, our accuracy estimates tend to be overoptimistic (Arlot and Celisse 2010). Such practice represents an extreme dependency between the training and test datasets (they are identical) which favour over-fitted models (Hawkins 2004). Various techniques, including hold-out, bootstrap and cross validation, are proposed in the literature to tackle this problem (see for example Arlot and Celisse (2010) and James et al. (2013, Ch. 5)).

In hold-out methods, the available data $\mathcal{D}$ is divided into two mutually exclusive subsets; a training set $\mathcal{D}_{train}$ and a test set $\mathcal{D}_{test}$. The training data is used to train or fit the model. Once the training is performed, the accuracy is measured on the unseen test dataset.

Cross-validation methods are proposed to better utilize the limited amount of available data. The dataset is repeatedly divided into a training and test dataset, and the model is trained and tested repeatedly. In k-fold cross-validation, the dataset $\mathcal{D}$ is split into k mutually exclusive sets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ of approximately equal size. The training process is repeated k times, and for each iteration $t \in 1, \ldots, k$, the training is performed using a dataset $\mathcal{D} \setminus \mathcal{D}_t$, and testing is performed on $\mathcal{D}_t$. Different strategies can be applied when splitting the dataset into folds. The most common strategy is, perhaps, to randomly assign each point to a fold. Another common approach is called stratification, where points are assigned such that each fold is a good representative of the whole. For example in classification problems, each fold contains approximately the same proportion of labels as the original dataset.

The cross-validation accuracy estimate is based on the overall prediction error. Brandsæter and Vanem (2018) suggest to analyse the distribution of the fold specific accuracy estimates, for example using box plots, illustrating how the accuracy estimates are vulnerable for changes in the distribution of the test dataset. Such analyses are of particular interest when the observations are dependent, such as for example for sequential sensor data.

Dependency between the training and test dataset can result in overly optimistic estimates of model performance (Arlot and Celisse 2010). Roberts et al. (2017) argue that a similar situation can occur when there are dependence structure in the data. If the test data are drawn nearby in the dependency structure, the independence between the training and test data can be compromised. This for example applies to datasets containing sensor measurements collected in sequential time. Examples of this effect is presented by Vanem et al. (2017) and Vanem and Brandsæter (2019), where two different splitting techniques are applied. First, the data are split into two parts randomly without accounting for the autocorrelation. With this approach, the clustering on the training and test data yields very similar distribution of observations across the clusters. Secondly, parts of the analysis is repeated using a different splitting approach, where the first 75% of the data are used for training and the remaining 25% of the data are reserved for testing. This approach gives completely different results, with an anomaly rate close to 75%. This exercise both illustrates the importance of accounting for the temporal dependency, as well as the importance of representative training data.

We refer to the latter splitting approach as blocking. In k-fold blocked cross-validation, the dataset is sliced into k folds at some central points of the dependency structure, for example in time or space (Bergmeir and Benitez 2012). A year of time series data can for example be split into 12 folds such that each fold contains data from a specific month. Roberts et al. (2017) claim that block cross-validation provides accuracy estimates that are closer to the true value. Through a series of simulations and case studies, they show that block cross-validation is nearly universally more appropriate than random cross-validation if the goal is prediction to new data or predictor space, or for selecting causal predictors.

Furthermore, k-fold block cross-validation can be modified to reduce the dependency between the folds by excluding from the training data the data in the folds which are adjacent to the validation set. That is for each $k \in 1, \ldots, K$ the models that are tested on $\mathcal{D}_k$ are trained on $\mathcal{D} \setminus \{\mathcal{D}_{k-1} \cup \mathcal{D}_k \cup \mathcal{D}_{k+1}\}$.

By repeating the cross-validation multiple times using different splits into folds, a better Monte-Carlo estimate to the complete prediction accuracy can be achieved. It is assumed that repeated k-fold cross-validation stabilizes the error estimation, and therefore reduces the variance of the cross-validation estimate (Kohavi 1995). However, similar to Rodriguez et al. (2009), we have not seen proof of this.

2.6 Enhanced testing

The knowledge of a data-driven method, for example a deep neural network, is limited to the examples it has seen during training (Wood et al. 2019) and the implicit assumptions of the model. Thorough investigation and analysis of the dataset can therefore contribute to increased trust in the model.

To ensure that a model's performance is thoroughly tested, an extensive test dataset is often used. For example in the automotive industry, large amounts of real world data from ordinary operations is gathered to test autonomous navigation systems (Fei-Fei 2010; Pei et al. 2017a; Zhao and Peng 2017). Additionally, simulated real-world data is also sometimes used to massively increase the amount of data (Madrigal 2017; Zhao and Peng 2017). Pei et al. (2017a) claim that for applications involving autonomous navigation in the automotive industry, this is usually completely unguided. Hence, due to the large input space of real-world scenarios, none of these approaches can hope to cover more than a tiny fraction (if any at all) of all possible corner cases. Here, a corner case is defined as an unusual, but far from impossible, scenario. As an example, again from the automotive industry, a Tesla in autopilot mode recently crashed into a trailer because the autopilot system failed to recognize the trailer as an obstacle due to its "white color against a brightly lit sky" and the "high ride height" (Lambert 2016).

Unfortunately, deep learning methods and other data-driven methods, despite impressive capabilities, often demonstrate unexpected or incorrect behaviours in corner cases for several reasons such as biased training data, overfitting, and underfitting of the models (Pei et al. 2017a). Various methods are proposed to optimize testing and to identify erroneous behaviours of the different data driven models. In Brandsæter and Knutsen (2018) we survey different methods to increase the coverage of a test which is performed on a limited dataset by slightly perturbing the original test data. In image classification problems, the test image can for example be slightly rotated, and the brightness and contrast can also be slightly changed, see for example Pei et al. (2017a), Pei et al. (2017b), and Tian et al. (2017)). Similarly, Liu et al. (2017) propose an unsupervised image-to-image translation framework based on Coupled Generative Adverserial Networks (CoGANs), demonstrating how a scene can be transformed to another one, including transformations of images from sunny to rainy, day to night, summery to snowy, and vice versa.

Chapter 3

Summaries of the papers and main contributions

3.1 Paper I

Brandsæter, A. and Vanem, E. (2018). Ship speed prediction based on full scale sensor measurements of shaft thrust and environmental conditions. *Ocean Engineering*, 162:316 – 330.

A ship's speed through water is estimated using different statistical and machine learning models. The covariates used in the regression include shaft thrust, the ship's motions and wind measurements. Accurate estimates of ship speed are important to be able to optimize ship design and operation, and to quantify the effect of modifications. In this example, the ship's speed through water is measured with an additional sensor, and hence, the response is known. The labelled dataset allows us to train a model which captures patterns and inherent dependencies between the thrust and the environmental forces. This can also allow us to detect anomalies by analysing the difference between the measured speed and the model output.

Our main contribution in this paper is to demonstrate how regression models such as linear regression, projection pursuit and generalized linear models can be implemented for this application. We also discuss different evaluation and cross-validation techniques, and demonstrate the importance of taking time-dependency into account. Furthermore, we advocate presenting the predictor's performance on the different test sets of the cross validation, to communicate robustness and credibility in the estimates.

3.2 Paper II

Brandsæter, A., Vanem, E., and Glad, I. K. (2019). Efficient on-line anomaly detection for ship systems in operation. *Expert Systems with Applications*, 121:418 – 437.

An anomaly detection technique combining signal reconstruction and residual analysis is presented. The reconstruction is performed using Auto Associative Kernel Regression (AAKR), and Sequential Probability Ratio Test (SPRT) is used for residual analysis. The dataset used to train the model is assumed to comprise data from normal operation exclusively, and anomalies are only present in the test data. Our main contributions are the following three novel comprehensive modifications:

1. We propose a novel cluster-based method to select memory vectors to be considered by the AAKR. The advantage of the cluster based method is the increased speed. The computation time of the AAKR grows rapidly when the size of the training data increases, and we demonstrate how the presented cluster based memory vector selection technique can be used to dramatically decrease the computation time, at the same time as the performance is kept at an acceptable level. The methodology is applied to multiple imbalanced benchmarking datasets, in addition to a dataset with sensor signals from a marine diesel engine in operation. Most of the anomalies are quite subtle, restrained enough not to easily be revealed by for example analysing scatter plots of the data. Results of the cluster based methods are presented and compared to the traditional set-up, and the analyses show that comparable results are achieved, even when very few clusters are used.

2. We also propose a generalization of the distance measure used in the signal reconstruction, which enables the users to impose system-knowledge on the anomaly detection framework making it possible to distinguish response and explanatory variables, and to optimize the weighting of the different features. This generalization of the AAKR method can be particularly useful when we have reason to assume that the sensor signals correctly return the actual value (no faults in the sensors), and when we are not interested in finding anomalies in all the sensor signals. For example, if we are interested in detecting engine problems, we do not want an alarm whenever we encounter abnormal combinations of environmental conditions.

3. Finally, we introduce a credibility estimate which enables the SPRT method to reach a conclusion faster when it operates in regions close to instances which are well represented in the training dataset, and allows it to use more time to reach a conclusion when it operates in less explored regions.

3.3 Paper III

Vanem, E. and Brandsæter, A. (2019). Unsupervised anomaly detection based on clustering methods and sensor data on a marine diesel engine. *Journal of Marine Engineering & Technology*, pages 1–18.

A selection of cluster-based methods for anomaly detection are explored, including mixtures of Gaussian models, density based clustering, self-organizing maps and support vector machines.

Our main contribution in this paper is to demonstrate benefits and deficiencies with the different cluster-based anomaly detection methods. In general, the performance of the methods is found to be good. However, changing our evaluation technique from cross-validation with random splitting to blocked

cross-validation, dramatically changes the results, demonstrating the importance of having representative training data when performing data-driven anomaly detection based on sensor data.

3.4 Paper IV

Brandsæter, A. and Knusten, K. E. (2018) Towards a framework for assurance of autonomous navigation systems in the maritime industry. In *Safety and Reliability–Safe Societies in a Changing World : Proceedings of ESREL 2018*, (pp. 449–457). CRC Press.

In this paper, we discuss potential assurance frameworks for autonomous navigation of maritime surface ships, with emphasis on testing and verification of the ship's perception performance and capacities. We propose and describe a range of recommended practices and tools that can be applied to test and validate the ability, performance and robustness of safety critical systems whose decisions are based on data-driven methods. These practices and tools originate partly from traditional statistical analysis and partly from testing and assurance of autonomy in the automotive industry. Challenges related to machine perception that are unique or particularly pronounced in the maritime domain are discussed, and we suggest how the recommended practices and tools should be used and possibly adapted to suit the maritime domain.

3.5 Paper V

Brandsæter, A. and Glad, I. K. (2019). Explainable Artificial Intelligence: How Subsets of the Training Data Affect a Prediction. *Submitted for publication*

We propose a novel approach which allows us to explore and investigate how the training data affects the predictions made by any black-box method. We call the explanations Shapley values for training data subset importance. The Shapley value concept originates from coalitional game theory, developed to fairly distribute the payout among a set of cooperating players. We extend this to training data subset importance, where a prediction is explained by treating the subsets of the training data as players in a game where the predictions are the payouts.

Since a prediction made by data-driven methods relies heavily on the data used to train the model, we believe explanations should convey information about how the training data affects that prediction. Koh and Liang (2017) suggest that we can better understand a model's behaviour by studying how the model is derived from its training data, and propose to identify training points most responsible for a given prediction. Similarly, our proposed Shapley values quantify the importance of different subsets of the training data, allowing new aspects of the reasoning and inner workings of a prediction model and learning method to be conveyed. The presented methodology is suggested as a supplement

to established explanations and interpretations methods such as methods based on feature importance, influential functions and case-based explanations.

Chapter 4

Discussion

4.1 Dependent signals

Often, data from a large number of sensors are captured, and the different sensors are generally not independent of each other. For example, temperature sensors placed on different parts of an engine will often suffer from spatial dependencies. Sensor data which are captured and stored in sequential time are also prone to temporal dependencies. Preferably, the complexity of the full sensor system should be taken into consideration when designing methods and models to analyse. In many cases, standard methods can still be used, but caution should be taken when evaluating the results, and cross-validation techniques which take the dependency structure into consideration should be applied.

If the sensor signals are highly correlated, the AAKR method is not satisfactorily robust (Baraldi et al. 2012), and the reconstructed signals are less accurately estimating the values of the signals in normal conditions (Baraldi et al. 2015b). The low robustness gives rise to two problems: (1) increased detection delay, that is the time before an anomaly is detected, and (2) spillover, which occurs when anomalies are detected in signals with normal behaviour. To overcome this problem, Baraldi et al. (2015b) propose a modification of the AAKR method. Their idea is to modify the weights with a penalty vector to give less importance to the signals with large normalized residuals. One main assumption is fundamental for the proposed modification: the probability of occurrence of anomalies in a small number of signals is higher than the probability of anomalies occurring in a large number of signals.

Dependency between covariates can also make it challenging to understand and interpret even simple linear models. In observational studies and machine learning problems, it is very rare that the features are statistically independent (Aas et al. 2019). Nevertheless, several existing explanation methods assume independent features, which may give wrong explanations.

4.2 Fault free data

A key assumption in the anomaly detection methods presented in Section 2.2.1 is that all training data is fault free. It is therefore recommended that different data quality checks and outlier detection are performed before deployment. In the cluster-based methods presented in Section 2.2.2, we do not assume that the training data is fault free, but we assume that the number of faults in the training data is small. There are, however, many ways new data can fall outside the clusters without being a fault. Hence, the cluster-based techniques presented

here are recommended for initial screening, and should be used in combination with other models (Vanem and Brandsæter 2019).

4.3 High dimensions

The methods and applications demonstrated in this thesis are performed on datasets with multiple sensors or features (typically between 5 and 10 different features). However, we have not investigated datasets in really high dimensions, where measuring similarity and dissimilarity can be difficult and lack practical meaning. This effect is referred to as the curse of dimensionality, meaning the distance measures become unstable. As the dimensionality increases, the distance to the nearest data point approaches the distance to the farthest data point (Beyer et al. 1999). Furthermore, the presence of irrelevant features can eliminate the potential clustering tendency, and the number of irrelevant features grows with dimension (Berkhin 2006).

4.4 Importance of representative data

Vanem and Brandsæter (2018) demonstrate the importance of representative training data for cluster-based anomaly detection, and discuss challenges related to ensuring and testing that the data are representative. Brandsæter et al. (2016) demonstrate and discuss challenges with representative training data related to signal reconstruction using AAKR. AAKR cannot be effectively applied to reconstruct data outside the training region; no non-linear models have the ability to correctly extrapolate beyond their training region (Hines et al. 2008). Hence, whenever the system which is monitored faces operating conditions which are unseen in the training data, we cannot expect the accuracy of the reconstruction to be satisfactory. If we use parametric models, retraining with a dataset which is representative of the new operating condition must be performed, or new first-principles models must be derived. For most non-parametric models, including AAKR, the training data memory matrix may be either replaced or supplemented with new data that are characteristic of the systems current operating state (Hines et al. 2008).

4.5 Transients

Anomaly and fault detection for condition monitoring of components which are operated in different operational modes and during transients can be particularly challenging. It is for example common that several alarms are triggered during start-up of a ship engine. When reconstruction methods such as the AAKR are used, it might be advantageous to develop models dedicated to the different operational modes. This could also allow the alarm limits to vary in the different modes, depending on the operation's criticality. To achieve this, the training data should be divided and used to fit different models. This will result in reduced

computational efforts and increased model reconstruction accuracy (Baraldi et al. 2012; Al-Dahidi et al. 2014).

Baraldi et al. (2012) discuss and propose new approaches to face these challenges related to transients. The results of their proposed approaches are presented on a case study concerning condition monitoring of a gas turbine during start-up transients. The first proposed approach is to develop operational mode specific reconstruction models. This leads to remarkably reduced computational effort, however it is reported that the robustness at the borders between two operational modes is not always satisfactory. The use of a signal processing tool based on the Haar wavelet transform, which takes into account not only the present value but also the past evolution of the signal, has also been proposed. It is reported that the approach leads to more robust reconstructions in the case of abrupt changes. However, for smooth transients the reconstructions are reported to be slightly less accurate.

4.6 Lack of specification

The lack of specification is an important challenge when testing and verifying a model, especially for use in safety critical domains. A training set is necessarily incomplete, and it is not possible to guarantee that it is even representative of the space of possible inputs (Salay et al. 2017). For example, machine perception, as discussed in Brandsæter and Knutsen (2018), is a functionality which is not completely specified. What is for example the specification for recognizing a sailing boat? Problems which involve advanced functionality that are not completely specifiable has motivated the implementation of machine learning based software which learns from examples rather than being programmed from a specification (Salay et al. 2017; Spanfelner et al. 2012). Based on experimental data reviewed, Rouder and Ratcliff (2006) argue that human categorization is also dependent on stored exemplars, in addition to abstracted rules.

4.7 What is a good explanation?

Lipton (2016) claims that although interpretability is often suggested as a remedy, few articulate precisely what interpretability means or why it is important. The paper discusses the interpretability of human decision-makers, and what notion of interpretability these explanations satisfy, and argues that human explanations seem unlikely to clarify the mechanisms or the precise algorithms by which brains work. Nevertheless, the information conferred by an interpretation may be useful.

Due to their subjective nature, it is challenging to quantify and rate the quality of different interpretations and explanations (Hall and Gill 2018). An expert and a lay user might for example prefer different explanations. Miller (2018) claims that most of the research and practice in the area of explainable AI seems to use the researchers' intuitions of what constitutes a 'good' explanation. Miller et al. (2017) argue that this could lead to failure, and that the model

experts are not in the right position to judge the usefulness of explanations to lay users.

A possible approach to test the quality of an explanation, is to use human subject evaluation, assuming that good model explanations are consistent with explanations from humans who understand the model (Lundberg and Lee 2017). One can sometimes also test if explanations can guide users to select the best predictior or classifier, or to improve it (Ribeiro et al. 2016).

Chapter 5

Conclusion

In this thesis we have presented five papers concerning the development and assurance of data-driven methods, mainly with applications from the maritime industry, including analysis of sequential sensor data, anomaly detection, classification and regression, and explainability of black-box models.

An important topic throughout the work has been the importance of thorough assurance processes and appropriate cross-validation techniques for performance evaluation. In particular, we have discussed challenges and possibilities for assurance of autonomous navigation of surface ships. Because the machine perception and situational awareness algorithms are expected to be partly or fully based on machine learning algorithms, including deep learning, whose functional reasoning is challenging or even impossible to fully understand and predict, the assurance and verification of such systems are fundamentally different from a traditional assurance and verification process based on physical understanding. We have reviewed several methods for testing autonomous navigation systems, proposed and used mainly in the automotive industry, and have discussed how these methods can be adapted, combined and applied to form a framework for assurance of autonomy in the maritime industry.

We have also presented a novel data-centric method to explain individual predictions based on Shapley values for training data subset importance. The proposed method allows the user to explore and investigate how different parts of the training data affect a prediction. The use of our proposed method, in combination with other well-established methods for explainability and interpretation, can provide better understanding of a prediction made by an opaque machine learning and statistical model.

Furthermore, we have demonstrated the usefulness of data-driven methods for anomaly detection in maritime applications. Three comprehensive modifications are proposed for the anomaly detection framework based on reconstruction with auto associative kernel regression (AAKR) and residuals analysis using sequential probability ratio test (SPRT). The first modification includes clustering of the training data (memory vectors) considered by the AAKR. The training data is replaced by clusters which represent the normal operating regions. The use of this method drastically reduces the computation time. The second modification is a generalization of the distance measure. We demonstrate how this enables the possibility to distinguish between explanatory and response variables. Finally, a regional credibility estimation used in the residuals analysis is proposed. This lets the time used to identify if a sequence of query vectors represents an anomalous state or not depend on the amount of data situated close to or surrounding the query vector.

Bibliography

Aas, K., Jullum, M., & Løland, A. (2019). Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *arXiv preprint arXiv:1903.10464*.

Allen, T. M. (2001). Us navy analysis of submarine maintenance data and the development of age and reliability profiles. *Department of the Navy SUBMEPP*.

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statist. Surv.*, *4*, 40–79.

Baraldi, P., Di Maio, F., Genini, D., & Zio, E. (2015a). Comparison of data-driven reconstruction methods for fault detection. *IEEE Transactions on Reliability*, *64*(3), 852–860.

Baraldi, P., Canesi, R., Zio, E., Seraoui, R., & Chevalier, R. (2011). Genetic algorithm-based wrapper approach for grouping condition monitoring signals of nuclear power plant components. *Integr. Comput.-Aided Eng.*, *18*(3), 221–234.

Baraldi, P., Di Maio, F., Pappaglione, L., Zio, E., & Seraoui, R. (2012). Condition monitoring of electrical power plant components during operational transients. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, SAGE*, *226*, 568–583.

Baraldi, P., Di Maio, F., Turati, P., & Zio, E. (2015b). Robust signal reconstruction for condition monitoring of industrial components via a modified auto associative kernel regression method. *Mechanical Systems and Signal Processing*, *60-61*, 29–44.

Bergmeir, C., & Benitez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192–213.

Berkhin, P. (2006). A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, & M. Teboulle (Eds.), *Grouping multidimensional data: Recent advances in clustering* (pp. 25–71).

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? In *International conference on database theory* (pp. 217–235). Springer.

Boechat, A. A., Moreno, U. F., & Haramura, D. (2012). On-line calibration monitoring system based on data-driven model for oil well sensors. *IFAC Proceedings Volumes*, *45*(8), 269–274.

Brandsæter, A., & Knutsen, K. E. (2018). Towards a framework for assurance of autonomous navigation systems in the maritime industry. In *Safety and reliability–safe societies in a changing world : Proceedings of esrel 2018* (pp. 449–457). CRC Press.

Brandsæter, A., & Glad, I. K. (2019). Explainable artificial intelligence: How subsets of the training data affect a prediction. *Submitted for publication*.

Brandsæter, A., Manno, G., Vanem, E., & Glad, I. K. (2016). An application of sensor-based anomaly detection in the maritime industry. In *2016 IEEE international conference on prognostics and health management (ICPHM)* (pp. 1–8). IEEE.

Brandsæter, A., & Vanem, E. (2018). Ship speed prediction based on full scale sensor measurements of shaft thrust and environmental conditions. *Ocean Engineering*, *162*, 316–330.

Brandsæter, A., Vanem, E., & Glad, I. K. (2019). Efficient on-line anomaly detection for ship systems in operation. *Expert Systems with Applications*, *121*, 418–437.

Brandsæter, A., Vanem, E., & Glad, I. K. (2017). Cluster based anomaly detection with applications in the maritime industry. In *2017 international conference on sensing, diagnostics, prognostics, and control (SDPC)* (pp. 328–333). IEEE.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Bzdok, D., Altman, N., & Krzywinski, M. (2018). Statistics versus machine learning. *Nature Methods*, *15*.

Campello, R. J., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 160–172). Springer.

Caruana, R., Kangarloo, H., Dionisio, J., Sinha, U., & Johnson, D. (1999). Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA symposium* (p. 212). American Medical Informatics Association.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 15.

Cheng, S., & Pecht, M. (2012). Using cross-validation for model parameter selection of sequential probability ratio test. *Expert Syst. Appl.*, *39*(9), 8467–8473.

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

Al-Dahidi, S., Baraldi, P., Di Maio, F., & Zio, E. (2014). Quantification of signal reconstruction uncertainty in fault detection systems, Second European Conference of the Prognostics and Health Management Society.

Di Maio, F., Baraldi, P., Zio, E., & Seraoui, R. (2013). Fault detection in nuclear power plants components by a combination of statistical methods. *IEEE Transactions on Reliability*, *62*(4), 833–845.

Dimopoulos, G. G., Georgopoulou, C. A., Stefanatos, I. C., Zymaris, A. S., & Kakalis, N. M. (2014). A general-purpose process modelling framework for marine energy systems. *Energy conversion and management*, *86*, 325–339.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226–231).

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, *27*(8), 861–874.

Fei-Fei, L. (2010). Imagenet: Crowdsourcing, benchmarking & other cool things. In *CMU VASC Seminar* (Vol. 16, pp. 18–25).

Fisher, A., Rudin, C., & Dominici, F. (2018). All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance. *arXiv preprint arXiv:1801.01489*.

Garvey, J., Garvey, D., Seibert, R., & Hines, J. W. (2007). Validation of on-line monitoring techniques to nuclear plant data. *Nuclear Engineering and Technology*, *39*, 133–142.

Goodman, B., & Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, *38*(3), 50–57.

Gross, K. C., & Lu, W. (2004, May 11). Early detection of signal and process anomalies in enterprise computing systems. In M. A. Wani, H. R. Arabnia, K. J. Cios, K. Hafeez, & G. Kendall (Eds.), *ICMLA* (pp. 204–210). CSREA Press.

HajKacem, M. A. B., N'Cir, C.-E. B., & Essoussi, N. (2019). Overview of scalable partitional methods for big data clustering. In O. Nasraoui & C.-E. Ben N'Cir (Eds.), *Clustering methods for big data analytics: Techniques, toolboxes and applications* (pp. 1–23).

Hall, P., & Gill, N. (2018). *Introduction to machine learning interpretability*. O'Reilly Media, Incorporated.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd ed.). Springer.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, *44*(1), 1–12.

Hines, J. W., & Garvey, D. R. (2006). Development and application of fault detectability performance metrics for instrument calibration verification and anomaly detection. *Journal of Pattern Recognition Research*, *1*(1), 2–15.

Hines, J. W., Garvey, J., Garvey, D. R., & Seibert, R. (2008). *Technical review of on-line monitoring techniques for performance assessment (nureg/cr-6895). volume 3: Limiting case studies*. United States Nuclear Regulatory Commission, Office of Nuclear regulatory Research.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, *22*(2), 85–126.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.

Kanarachos, S., Christopoulos, S.-R. G., Chroneos, A., & Fitzpatrick, M. E. (2017). Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and hilbert transform. *Expert Systems with Applications*, *85*(Supplement C), 292–304.

Kappaganthu, K., Li, Y., & Hernandez, L. (2010). Model based diagnostics of an aircraft generator using aakr and sprt. *SAE International Journal of Aerospace*, *3*(2010-01-1761), 137–143.

Killick, R., Fearnhead, P., & Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, *107*(500), 1590–1598.

Kim, B., Khanna, R., & Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in neural information processing systems* (pp. 2280–2288).

Knutsen, K. E., Manno, G., & Vartdal, B. J. (2014). Beyond condition monitoring in the maritime industry. *DNV GL Strategic Research & Innovation*.

Koh, P. W., & Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1885–1894). JMLR.

Kohavi, R. (1995). *Wrappers for performance enhancement and oblivious decision graphs*. Carnegie-Mellon University Pittsburch PA dept. of Computer Science.

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*(4), 740–753.

Lambert, F. (2016). Understanding the fatal tesla accident on autopilot and the nhtsa probe. *Electrek, July*.

Laxhammar, R., Falkman, G., & Sviestins, E. (2009). Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. In *2009 12th international conference on information fusion* (pp. 756–763). IEEE.

Li, W., Peng, M.-j., Yang, M., Xia, G.-l., Wang, H., Jiang, N., & Ma, Z.-g. (2017). Design of comprehensive diagnosis system in nuclear power plant. *Annals of Nuclear Energy*, *109*, 92–102.

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Liu, M.-Y., Breuel, T., & Kautz, J. (2017). Unsupervised image-to-image translation networks. In *Advances in neural information processing systems* (pp. 700–708).

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Madrigal, A. C. (2017). Inside waymo's secret world for training self-driving cars. Retrieved August 23, 2017, from https://www.theatlantic.com/technology/archive/2017/08/inside-waymos-secret-testing-and-simulation-facilities/537648/

Miller, T. (2018). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*.

Miller, T., Howe, P., & Sonenberg, L. (2017). Explainable AI: Beware of inmates running the asylum. In *IJCAI-17 Workshop on explainable AI (XAI)* (Vol. 36).

Molnar, C. (2019). *Interpretable machine learning: A guide for making black box models explainable*.

Niu, G., Zhao, Y., Defoort, M., & Pecht, M. (2015). Fault diagnosis of locomotive electro-pneumatic brake through uncertain bond graph modeling and robust online monitoring. *Mechanical Systems and Signal Processing*, *50-51*, 676–691.

Nowlan, F. S., & Heap, H. F. (1978). *Reliability-centered maintenance*. United Air Lines Inc San Francisco Ca.

Olson, C., Judd, K., & Nichols, J. (2018). Manifold learning techniques for unsupervised anomaly detection. *Expert Systems with Applications*, *91*(Supplement C), 374–385.

Pei, K., Cao, Y., Yang, J., & Jana, S. (2017a). Deepxplore: Automated whitebox testing of deep learning systems. *arXiv preprint arXiv:1705.06640*.

Pei, K., Cao, Y., Yang, J., & Jana, S. (2017b). Towards practical verification of machine learning: The case of computer vision systems. *arXiv preprint arXiv:1712.01785*.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144). ACM.

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., ... Thuiller, W. et al. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, *40*(8), 913–929.

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2009). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Transactions on pattern analysis and machine intelligence*, *32*(3), 569–575.

Rouder, J. N., & Ratcliff, R. (2006). Comparing exemplar-and rule-based theories of categorization. *Current Directions in Psychological Science*, *15*(1), 9–13.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: A modern approach*. Malaysia; Pearson Education Limited,

Salay, R., Queiroz, R., & Czarnecki, K. (2017). An analysis of iso 26262: Using machine learning safely in automotive software. *arXiv preprint arXiv:1709.02435*.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). *Metrics for evaluating performance of prognostic techniques*.

Seker, S., Ayaz, E., & Türkcan, E. (2003). Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery. *Engineering Applications of Artificial Intelligence*, *16*(7-8), 647–656.

Shrikumar, A., Greenside, P., Shcherbina, A., & Kundaje, A. (2016). Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*.

Spanfelner, B., Richter, D., Ebel, S., Wilhelm, U., Branz, W., & Patz, C. (2012). Challenges in applying the iso 26262 for driver assistance systems. *Tagung Fahrerassistenz, München*, *15*(16), 2012.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, *62*(1), 77–89.

Steinwart, I., Hush, D., & Scovel, C. (2005). A classification framework for anomaly detection. *Journal of Machine Learning Research*, *6*(Feb), 211–232.

Štrumbelj, E., & Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, *11*(1).

Štrumbelj, E., & Kononenko, I. (2011). A general method for visualizing and explaining black-box regression models. In *International conference on adaptive and natural computing algorithms* (pp. 21–30). Springer.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, *41*(3), 647–665.

Tian, Y., Pei, K., Jana, S., & Ray, B. (2017). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. *arXiv preprint arXiv:1708.08559*.

Tveten, M. (2017). *Multi-stream sequential change detection–using sparsity and dimension reduction* (Master's thesis).

Vanem, E., & Storvik, G. (2017). Anomaly detection using dynamical linear models and sequential testing on a marine engine system. In *Proc. annual conference of the prognostocs and health management society 2017 (phm 2017)*.

Vanem, E., & Brandsæter, A. (2018). Cluster-based anomaly detection in condition monitoring of a marine engine system. In *2018 prognostics and system health management conference (phm-chongqing)* (pp. 20–31). IEEE.

Vanem, E., & Brandsæter, A. (2019). Unsupervised anomaly detection based on clustering methods and sensor data on a marine diesel engine. *Journal of Marine Engineering & Technology*, 1–18.

Vanem, E., Brandsæter, A., & Gramstad, O. (2017). Regression models for the effect of environmental conditions on the efficiency of ship machinery systems. In M. R. T̃. Bedford (Ed.), *Risk, reliability and safety : Innovating theory and practice : Proceedings of ESREL 2016* (pp. 362–371). Glasgow, Scotland: Lesley Walls.

Wald, A. (1947). *Sequential analysis*. New York: Wiley.

Wand, M. P., & Jones, M. C. (1995). *Kernel smoothing*. Chapman and Hall/CRC.

Wang, W. (2008). Some fundamental issues in ensemble methods. In *2008 IEEE International joint conference on neural networks (IEEE World congress on computational intelligence)* (pp. 2243–2250). IEEE.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*(2), 241–259.

Wood, M., Robbel, P., Maass, M., Tebbens, R. D., Meijs, M., Harb, M., … Schlicht, P. (2019). *Safety first for automated driving*. Aptiv Services US, LLC; AUDI AG; Bayrische Motoren Werke AG; Beijing Baidu Netcom Science Technology Co., Ltd; Continental Teves AG & Co oHG; Daimler AG; FCA US LLC; HERE Global B.V.; Infineon Technologies AG; Intel; Volkswagen AG.

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, *2*(2), 165–193.

Zhao, D., & Peng, H. (2017). From the lab to the street: Solving the challenge of accelerating automated vehicle testing. *arXiv preprint arXiv:1707.04792*.

Zheng, D., Li, F., & Zhao, T. (2016). Self-adaptive statistical process control for anomaly detection in time series. *Expert Systems with Applications*, *57*(Supplement C), 324–336.

Zio, E., & Di Maio, F. (2010). A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering & System Safety*, *95*(1), 49–57.

Zymaris, A. S., Alnes, Ø. Å., Knutsen, K. E., & Kakalis, N. M. (2016). Towards a model-based condition assessment of complex marine machinery systems using systems engineering. In *Proc. 3rd eur. conf. prognostics health manage. soc.* (pp. 1–15).

Papers

I

Paper I

Ship speed prediction based on full scale sensor measurements of shaft thrust and environmental conditions

Brandsæter, A., Vanem, E.

Ocean Engineering (2018), 162:316 – 330.

Ocean Engineering 162 (2018) 316–330

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Ship speed prediction based on full scale sensor measurements of shaft thrust and environmental conditions

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

[a] DNV GL, Veritasveien 1, N-1363, Høvik, Norway
[b] Department of Mathematics, University of Oslo, P.B.1053, Blindern, N-0316, Oslo, Norway

ARTICLE INFO

Keywords:
Ship speed estimation
Computational methods/numerical analysis
Sensor data analytics
Ship resistance and propulsion
Performance measure quantification
Statistical modelling
Energy efficiency

ABSTRACT

The primary goal of this study is to adapt and validate various regression models to predict a ship's speed through water based on relevant and available full scale sensor measurements from a ship, including measurements of external environmental forces. The wind is measured by on-board wind sensors, and the effect of the waves is measured by motion reference units (MRUs) installed on the ship, measuring motions in six degrees of freedom; three translational motions and rotations about these. Accurate speed estimates, which relate directly to the estimates of the propulsion efficiency, fuel efficiency and pollution, are vital to be able to optimize ship design and operation. We demonstrate how regression models such as linear regression, projection pursuit (PPT) and generalized additive models (GAM) can be easily implemented for this application. A simple regression model based on the well-established relationship between ship speed and shaft thrust represent a benchmark model towards which the other models are compared.

1. Introduction

Accurate estimates of ship propulsion and fuel efficiency are important to be able to optimize ship design and operation. Deviations between the measured ship speed and the speed predictions based on propulsion power and other internal and external conditions can be indicative of an anomaly, such as e.g. hull, propeller or engine damage. Furthermore, the effect of modifications can be quantified. This can include modifications of the ship hull, such as for example hull cleaning or bow optimization, installation of new equipment such as kites, fixed sails or batteries for machinery optimization, propeller optimization such as contra rotating propeller and various efficiency improvement devices, and operational optimization measures such as weather routing and trim and draft optimization. The logistics planning can also be optimized with accurate of time of arrival estimation.

Due to the complexity of a modern ship and its exposure to external factors such as wind, waves and currents, estimating the ship efficiency accurately is not an easy task. Various methods are described in literature and used by the industry. The methods can be divided into four main groups as suggested by (Petersen et al., 2012):

1. Traditional and standard series methods which typically rely on a set of parameters describing the hull (Savitsky, 1964; Øyan, 2012; Holtrop and Mennen, 1982),
2. regression based methods based on a set of sensor measurements (Petersen et al., 2012; Bocchetti et al., 2015; Mao et al., 2016),
3. direct model tests in test tanks (Chuang and Steen, 2011), and
4. computational fluid dynamics (CFD) (Peri et al., 2001; Sadat-Hosseini et al., 2013; Ozdemir and Barlas, 2016).

The methods are often ordered on a scale between methods that are governed by physical laws and empirical or data driven methods (sometimes referred to as black box methods) that are based on statistical inference of historical data. Due to the fact that the data driven methods require little knowledge of the physical system (Coraddu et al., 2017) and there is no need to manually build a model of the data, these methods can be easily implemented in marine operations; making such technologies a lean alternative to complex tailor-made analytics (Brandsæter et al., 2016). At the same time, the data driven methods can be unsatisfactory in terms of physical explanation and it might require a significant amount of data to be sufficiently accurate (Vanem et al., 2017; Petersen et al., 2012).

The primary goal is to survey various regression models to estimate the ship speed through water based on relevant and available sensor measurements of the shaft thrust and external environmental forces from wind, waves and currents. The wind is measured by on-board wind sensors, and the effect of the waves is measured by motion reference units (MRUs) installed on the ship, measuring motions in six

* Corresponding author. DNV GL, Veritasveien 1, N-1363, Høvik, Norway.
E-mail address: andreas.brandsaeter@dnvgl.com (A. Brandsæter).

https://doi.org/10.1016/j.oceaneng.2018.05.029
Received 8 March 2017; Received in revised form 23 April 2018; Accepted 14 May 2018
Available online 29 May 2018

degrees of freedom; three translational motions and rotations about these, and the sea currents are incorporated in the measure of ship speed through water. The speed through water y relates directly to the propulsion efficiency which is commonly defined as $e_{prop} = \frac{y}{p}$, where p is the propulsion power. It is also linked directly to the fuel efficiency $e_{fuel} = \frac{y}{f}$, where f is the energy in the consumed fuel, as defined by (Petersen et al., 2012).

We work towards a better understanding of how the external conditions affect the ship's speed, propulsion and fuel efficiency, and aim to be able to quantify these effects. Several case studies using different methods are described in the recent literature, both with main focus on propulsion efficiency estimation (Petersen et al., 2012; Vanem et al., 2017; Chuang and Steen, 2011; Øyan, 2012; Holtrop and Mennen, 1982; Mao et al., 2016) and fuel efficiency and emission estimation (Trodden et al., 2015; Bialystocki and Konovessis, 2016; Coraddu et al., 2017; Rakke, 2016; Bocchetti et al., 2015).

2. Data description

This study is based on an extended version of the dataset used in (Vanem et al., 2017). For completeness, parts of the data description provided in (Vanem et al., 2017) is rendered in the following with minor modifications.

The dataset contains variables associated with the efficiency of the ship machinery system, such as the speed through water (knots), propulsion power [kW] and shaft thrust (N). The shaft thrust is assumed to be proportional to the propulsion power over speed over ground. Other variables included in the dataset are related to the ship's motions, wind speed relative to the ship and trim and draft. These variables represent external factors and are used to explain variation in the efficiency and ship speed.

The ship is installed with two motion reference units (MRUs) measuring the ship's motion in all six degrees of freedom (heave, surge, sway, roll, yaw and pitch). From the raw motion data recorded at 5 Hz various integrated parameters are stored every 30 s, calculated from the preceding 15 min time record of the motions. The integrated parameters reported by the system include the first five spectral moments of each motion signal (m_0, m_1, , m_3 and m_4), the mean, standard deviation, skewness and kurtosis of the signal as well as the maximum and minimum values during the time window. Also the spectral peak period T_p and zero crossing period T_z were recorded.

Out of these parameters many are not relevant for the present analysis and are not considered here. Besides, some of the parameters carry redundant information and can be derived from other parameters, such as the standard deviation of the signal $\sigma = \sqrt{m_0}$ and $T_z = T_{02} = \sqrt{m_0/m_2}$. We therefore limit ourselves to consider m_0 and T_z for each of the degrees of freedom. The zeroth spectral moment $m_0 = \sigma^2$ is the total energy of the motion spectrum and indicates the magnitude of the ship motion. Note that for a wave record the significant wave height is usually defined as $H_s = 4\sqrt{m_0}$. Likewise, T_z indicates the typical period of the different motions. Since the periodic ship motions are primarily an effect of the waves, both m_0 and T_z can be considered as proxies for the real wave conditions in the sense that m_0 will be proportional to the real significant wave height and T_z will be similar to the typical period of the wave field. Moreover, the ship response in the different degrees of freedom will to a certain extent reflect the wave direction relative to the ship.

In addition to the ship motions, representing the effect of the waves, the wind speed relative to the ship is recorded; the wind component perpendicular to the ship (Wind-y) and the wind parallel to the ship (Wind-x). Wind-x is defined so that a positive value means wind blowing in the same direction as the ship speed. Two other parameters that are important for the hydrodynamic resistance of a ship is the draft and trim, which have also been recorded and included in the present analyses. The draft is defined as the vertical distance between the waterline and the bottom of the hull and is naturally related to the cargo level of the ship, while the trim is the difference between the forward and aft drafts.

For the analysis presented in this paper the original 30 s data were down-sampled to 5 min resolution by calculating the average values within each 5 min window. This makes the dataset smaller and easier to handle, and reduces the time dependency.

The data have been collected from an ocean-going ship over approximately 10 months in normal operation. Due to data quality issues, a large fraction of the data were removed in initial cleaning and outlier removal. For example when the difference between the measured speed through water and speed over ground is significantly larger than reasonable current, at least one of the measured speed sensors must report wrong values. Although the measurement of the speed through water is known not to be most reliable (van den Boom and Hasselaar, 2014), it is difficult to know which reading is wrong, hence we remove the data point. After the initial filtering and outlier removal, the dataset used in the analysis contains about 33000 data points, which correspond to about 115 days of data.

Initially, 18 selected variables are included in the analysis. Trace plots of the data are shown in Fig. 1. The upper plot shows the speed and thrust data series, the next shows all the ship motion data and the two lower plots show the wind and the trim and draft data, respectively. Each point on the horizontal axis represents the average sensor value in the previous 5 min.

A correlation plot showing the linear correlation between the various variables in the data set is shown in Fig. 2. It is seen that the a highly correlated structure is present in the dataset.

3. Methodology

3.1. Regression models

We employ various regression models to describe and predict the data, i.e., linear regression models, generalized additive models (GAM) and projection pursuit regression (PPR) models. Other models including various regression trees and kernel density estimation were also explored to some extent. We were not able to tune these models to provide accurate predictions, hence they are omitted. In the following, a brief introduction to each of the models will be provided. Reference is made to textbooks such as (Hastie et al., 2009) for a more thorough introduction.

The response variable will be denoted Y and the explanatory variables will be denoted $\mathbf{X} = (X_1, X_2, ..., X_P)$. The basic problem is to construct a prediction rule for predicting Y conditioned on the explanatory variables based on a stochastic model on the form

$$Y = f(\mathbf{X}) + \varepsilon, \tag{1}$$

where ε represents stochastic white noise and is often modelled as a zero-mean Gaussian variable. Different models for the $f(\cdot)$ function give rise to different regression models. Assuming N observations, the observed values are y_j and $\mathbf{x}_j$ for $j = 1, ..., N$.

3.1.1. Linear regression models

In linear regression one assumes a linear model on the following form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_P X_P + \varepsilon. \tag{2}$$

The error made in such predictions are referred to as the residuals, and the residual sum of squares (RSS) will be a function of the model parameters. It is defined as

$$RSS(\hat{\beta}) = \sum_{j=1}^{N} (y_j - \hat{y}_j)^2 \tag{3}$$

and the fitted model parameters $\hat{\beta}_p$, $p = 0, ..., P$ are estimated from the

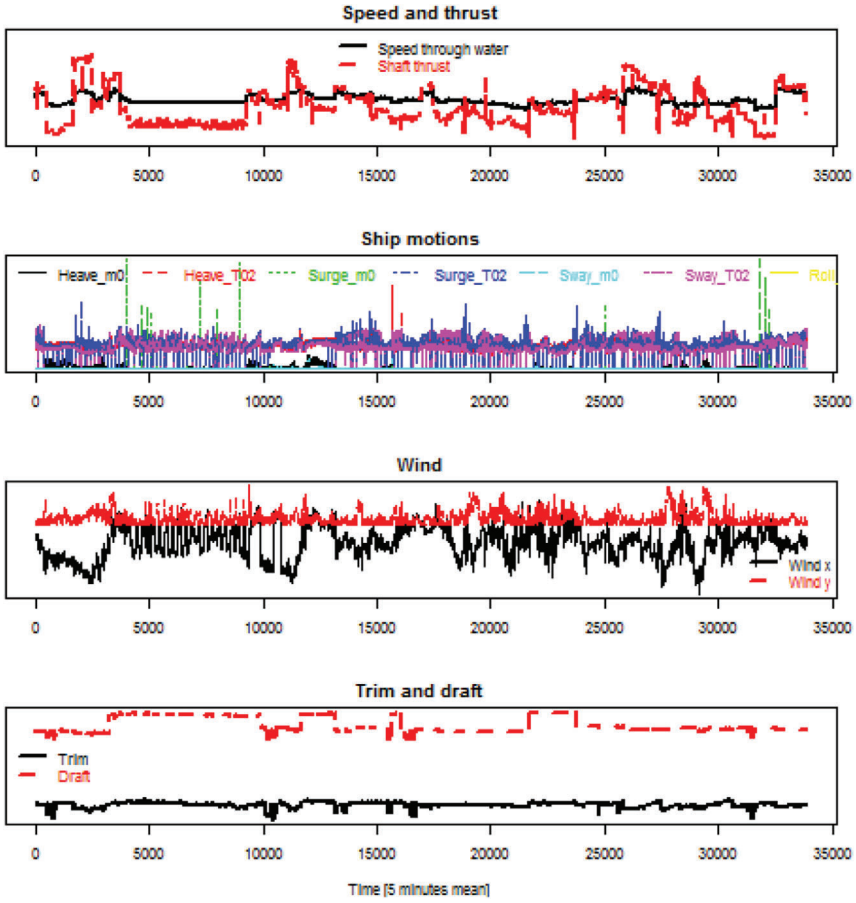


Fig. 1. Trace plot of the data. Each point on the horizontal axis represents the average sensor value in the previous 5 min.

data. In ordinary least squares, these estimators are found as

$$\hat{\beta}^{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{4}$$

In some cases one may obtain better predictions by setting some of the coefficients to zero. This will introduce bias, but may reduce mean square error by reducing variance. Deleting some coefficients gives a simpler model that may also be easier to interpret if P, the number of explanatory variables, is large, but interpretation could be affected by highly correlated covariates. Variable subset selection keeps the intercept in the model and selects a subset of the explanatory variables and ignores the rest. Compared to the standard linear regression, the accuracy when applying best subset linear regression was found to be similar, but somewhat poorer, and we omit the results here. Another possibility is to shrink some of the coefficients by for example using ridge regression, Lasso, least angle regression, principle component regression and partial least squares (Vanem et al., 2017).

3.1.2. Generalized additive models (GAM)

The generalized additive model has the form

$$Y = \alpha + \sum_{p=1}^{P} f_p(X_p) + \varepsilon \tag{5}$$

where $f_p(\cdot)$ are smooth functions of one covariate (Wood, 2006). Estimation of the smooth functions to fit the data as well as possible and to be as smooth as possible can be formulated as minimization of a penalized sum of squares, where a tuning parameter λ is introduced to control the degree of smoothing as follows:

$$PRSS(\alpha, f_p) = \sum_{j=1}^{N}\left(y_j - \alpha - \sum_{p=1}^{P} f_p(x_{jp})\right) + \lambda \sum_{p=1}^{P} \int_t f''_p(t_p)^2 dt_p. \tag{6}$$

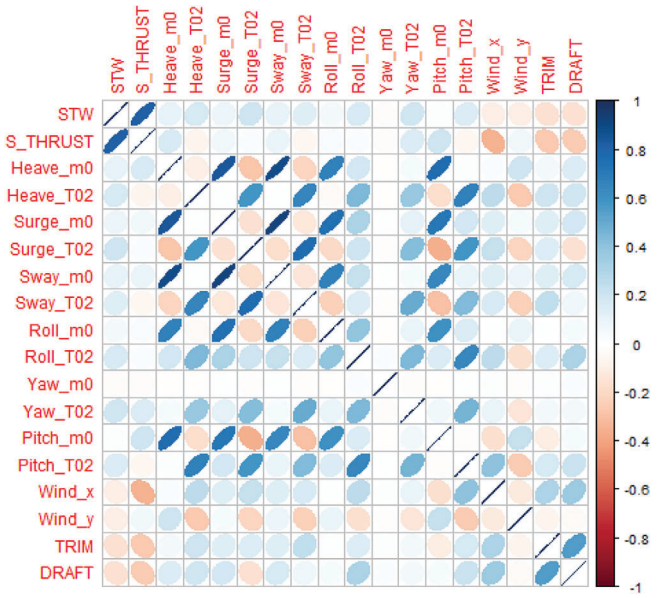


Fig. 2. Correlation plot of the linear correlations in the data from the sensors signals used in the models. Dark blue color indicates strong correlations, white indicates no correlations, and dark red indicates strong negative correlation. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

If the tuning parameter $\lambda \to \infty$, the estimate for f will approach a straight line, while choosing $\lambda = 0$ results in an un-penalized regression spline estimate. We choose the tuning parameter λ using generalized

cross validation (GCV), applying the mgcv package in R (Wood, 2017).

To impose additional smoothing, we inflate the model degrees of freedom in the GCV by choosing a constant multiplier γ (Wood, 2017). We report results using

$$\gamma = 0.5 * log(P) \tag{7}$$

where P is the number of explanatory variables. We refer to this model as GAM smooth.

A GAM model based on a subset of the variables is also fitted and the results are reported. The subset contains the variables believed to be most important, but it is not necessarily the optimal subset. The following variables are selected: Shaft thurst, Heave (M0), Heave (T02), Surge (T02), Sway (M0), Roll (T02), Yaw (T02), Wind-x, Wind-y and trim. This model is referred to as GAM selected.

3.1.3. Projection pursuit regression (PPR) models

A further generalization of the generalized additive model can be obtained where the smooth functions are allowed to be non-linear smooth functions of some linear combinations of the covariates $\mathbf{X}$. Projection pursuit regression models are such models on the form

$$f(\mathbf{x}) = \beta_0 + \sum_{m=1}^{M} g_m(\omega_m^T \mathbf{X}) = \beta_0 + \sum_{m=1}^{M} g_m\left(\sum_{j=1}^{p} \omega_{mj} X_j\right). \tag{8}$$

The g_m's are smooth functions (called ridge functions, varying only in the direction defined by the vector ω_m) estimated from the data together with the direction vectors ω_m. The direction vectors are normalized, $\|\omega_m\| = 1$ and one must choose parameter M and smoothness of each of the g_m's (tuning parameters). Note that the scalar variable $V_m = \omega_m^T \mathbf{X}$ is the projection of $\boldsymbol{X}$ onto the unit vector ω_m. Projection pursuit regression models are able to handle interaction effects between variables and are in fact universal approximators in that they can approximate almost any function for large M. We report results from projection pursuit models using 3 and 10 terms.

3.2. Baseline regression method

The above mentioned methods use wind and waves measurements as well as shaft thrust as explanatory variables. We compare the accuracy of these methods with predictions produced using a baseline method which does not take weather effects into account. The baseline model is based on the following well known relationship between the ship's speed and power demand in calm sea conditions (Tupper, 2004)

$$A_c \approx \frac{\Delta^{2/3} Y^3}{P} \propto \sim \frac{Y^2}{S}. \tag{9}$$

Here, A_c denotes the admiralty coefficient which is assumed to be constant for the ship, and Δ denotes the ships weight displacement which we also assume to be constant. The speed through water [knots] is denoted by Y and the installed power [kW] is denoted by P. The shaft thrust [kN], denoted by S, is assumed to be approximately proportional to power over speed.

Based on the this relationship we construct the following simple baseline method for the ship's speed through water:

$$Y_j = \beta \sqrt{S_j} + \varepsilon_j \tag{10}$$

where S_j is the measured shaft thrust, ε_j is the prediction error, and β is a constant calculated from the training data

$$\beta = \frac{\sum_{i \in \mathscr{D}_t} Y_j^2}{\sum_{i \in \mathscr{D}_t} S_i} = \frac{\overline{Y^2}}{\overline{S}} \tag{11}$$

where $\mathscr{D}_t$ denotes the training dataset. This β minimizes the sum of squares:

$$\sum_{i \in \mathscr{D}_t} (Y_i - \beta \sqrt{(S_i)})^2. \tag{12}$$

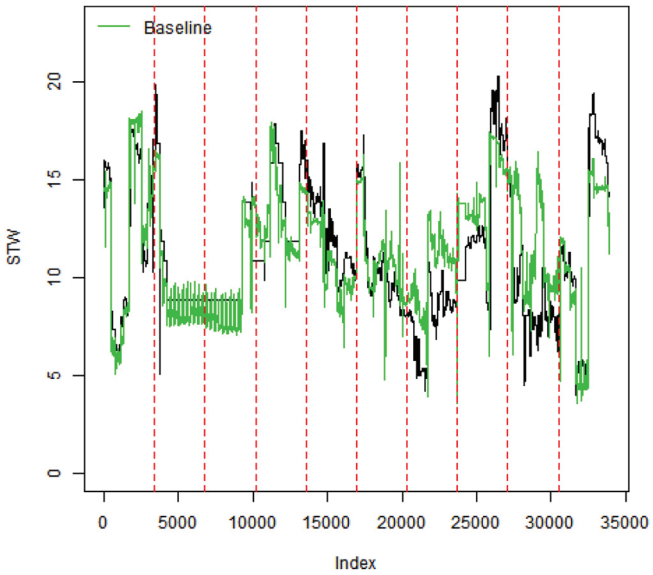


Fig. 3. Prediction using the baseline method. Each point on the horizontal axis represents the average sensor value in the previous 5 min. The black line shows the observed speed through water, and the green line shows the speed through water predicted by the baseline method. The red dotted vertical lines indicates the boundry between the different folds. Here we use 10-fold cross validation. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

We refer to this method as the baseline method. Due to the difficulties comparing our results with previous work (Petersen et al., 2012), we use this method to benchmark the different models explored in this paper.

The speed through water (STW) predictions produced by the baseline method are displayed in Fig. 3. Here, we use 10-fold cross validation, and the 10 folds are distinguished in the figure by the vertical red dotted lines, see section 3.3 for further explanation.

3.3. Cross validation

It is well known that when we evaluate predications from a statistical model on the dataset used to train the model, our accuracy estimates tend to be overoptimistic (Arlot and Celisse, 2010). To address this issue, a basic approach is to divide the dataset $\mathscr{D}$ into two exclusive parts $\mathscr{D}_t$ and $\mathscr{D}_k$; where one part $\mathscr{D}_t$ is used to train the model, and the other $\mathscr{D}_k$ is reserved for validation. To build robust and accurate models we ideally want to use all data available. The same applies to testing; we want to test our models in all situations, not only on a subset. Cross validation introduces various methods of repetitively splitting the data into training and validation datasets. Each of the splits provides a validation estimate, and by averaging over all the estimates we get a cross validation estimate. A range of different splitting techniques can be applied, providing different cross validation estimates. See for example (Arlot and Celisse, 2010; Kohavi, 1995) for a brief overview of the most common splitting techniques.

3.3.1. Standard K-fold cross validation

In this study we restrain to K-fold cross validation, which in its standard form splits the original dataset $\mathscr{D}$ into K subsets (folds) $\mathscr{D}_1, ..., \mathscr{D}_K$, as described in (Arlot and Celisse, 2010). Here, we choose the sets to be mutually exclusive with equal size. For each $k \in 1,2, ..., K$ the models are trained on $\mathscr{D}_t = \mathscr{D} / \mathscr{D}_k$, and tested on $\mathscr{D}_k$. Furthermore, we experiment with several K-s and report results with $K = 10$, 20, 50 and 3385. When $K = 3385$, each fold contains 10 points.

3.3.2. Modified K-fold cross validation

Cross validation is applicable to almost any algorithms and frameworks, involving regression, classification and many others (Arlot and Celisse, 2010). The only assumptions needed is that the data is identically distributed and that the training and validation sets are independent. The data itself do not need to be independent (Arlot and Celisse, 2010). In the case where the data is not independent, as is often the case with date retrieved from continuous sensor measurements, the independence between the training and validation datasets can be controlled by choosing $\mathscr{D}_t$ and $\mathscr{D}_k$ such that

$$min_{i\in\mathscr{D}_t, j\in\mathscr{D}_k}|i-j| > h$$

where h is a parameter $h > 0$.

Modified cross validation excludes from the training data the data in the folds which are adjacent to the validation set, that is for each $k \in 1, ..., K$ the models that are tested on $\mathscr{D}_k$ are trained on $/\{\mathscr{D}_{k-1} \cup \mathscr{D}_k \cup \mathscr{D}_{k+1}\}$.

3.3.3. Repeated K-fold cross validation

To make sure that the results are not strongly dependent on how the folds are selected, we repeatedly run the K-fold cross validation with new selections. When we run K-fold cross validation with L repetitions, we divided the original dataset $\mathscr{D}$ into K subsets (folds) in L different ways, such that no folds selected are equal.

That is, we choose $\mathscr{D}_k^l$, the dataset of the k-th in the l-th repetition, such that $\mathscr{D}_k^l \neq \mathscr{D}_{\dot{k}}^{\dot{l}}$ for all $\dot{l} \neq l$ and $\dot{k} \neq l$.

3.4. Model comparison and evaluation

3.4.1. Root mean squared error (RMSE)

After fitting the various models on the training data $\mathscr{D}_t$, the performance of the predictions are evaluated based on the root mean squared error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\hat{Y}_j - Y_j)^2} \tag{13}$$

where $\hat{Y}$ denotes the predicted speed through water based on shaft thrust and external forces, while Y is the direct speed measurement. N denotes the number of data points in the validation set. The model with the lowest RMSE will be preferred.

3.4.2. R-squared

For initial model checking we use the coefficient of determination, R squared

$$R^2 = 1 - \frac{\sum_{i\in\mathscr{D}_t}(y_i - \hat{y}_i)^2}{\sum_{i\in\mathscr{D}_t}(y_i - \bar{y}_i)^2}. \tag{14}$$

This is a measure of how well the regression model explains the total variation of the response variable. The coefficient varies from 0 to 1, where 1 indicates a perfect fit of the model to the data. The adjusted R^2 introduces a penalty for increasing (effective) number of parameters in order to avoid overly complex models.

4. Analysis and results

If we fit the regression models introduced above using only shaft thrust as explanatory variable, the prediction accuracy, in terms of RMSE based on predictions using 10-fold cross validation, are similar but still worse than the accuracy achieved with the baseline method. Out of the analysed methods, the linear model achieves the highest average RMSE of 2.055, some 0.8 % higher than the baseline method. When we instead use the square root of the shaft thrust as explanatory variable when fitting a linear model, the obtained RMSE is even slightly higher. This suggests that we need more elaborate methods to be able to increase our accuracy.

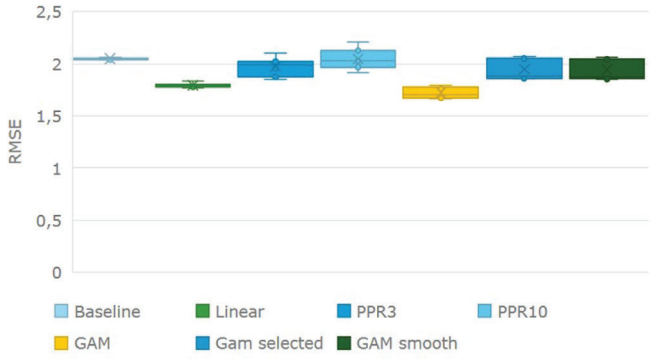


Fig. 4. Boxplot of the RMSE per repetition $l \in 1, ..., 7$ for the different regression models, using 10-fold cross validation. For each regression method the box displays the inner quartile range, and the median is marked with a horizontal line. The mean value is indicated with a cross, while the actual values are marked with circles. The whiskers in this plot displays the maximal and minimal values.

The total RMSE of the different regression models where all the explanatory variables are utilized, also variables which describe external conditions, are shown in the boxplot in Fig. 4. The RMSE values per repetition $l \in 1, ..., 7$ are shown, using 10-fold cross validation. For each regression method the box displays the inner quartile range (the range between the 25th percentile and the 75th percentile), and the median is marked with a horizontal line. The mean value is indicated with a cross, while the actual values are marked with circles. The whiskers in this plot displays the maximal and minimal values.

We observe that the spread in the results are not very large, which indicate that the results are not heavily dependent on how we chose to select the folds. The RMSE of both the linear model and the full GAM model achieves RMSE values significantly below the baseline model, with mean values 12 and 16% below the baseline model respectively. The smoothed GAM and the GAM on selected explanatory variables achieves average values about 5% below the baseline method. The same applies to the PPR with 3 terms, while the average RMSE of the PPR with 10 terms are about the same as the baseline method.

Fig. 5 shows the density of the residuals of the predictions by the baseline, linear, PPR with 3 terms and GAM models. Here we use 10-fold cross validation without repetition. The figure indicate that the predictions are not biased, except for the baseline method which tend to predict too high values.

Fig. 6 shows a scatter plot with shaft thrust on the horizontal axis and speed through water on the vertical axis. The observed values are marked in black, while the predicted values based on the baseline, linear, GAM and PPR with 3 terms models are marked in green, red, blue and grey respectively. We note that the predictions based on the linear, PPR and GAM models have a good spread, while the baseline model does not take notice of the variations which we believe are induced by the external conditions.

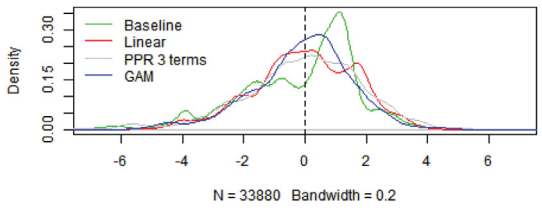


Fig. 5. Densityplot of the residuals for the different models, using 10-fold cross validation without repetition. The bandwidth is set to 0.2.

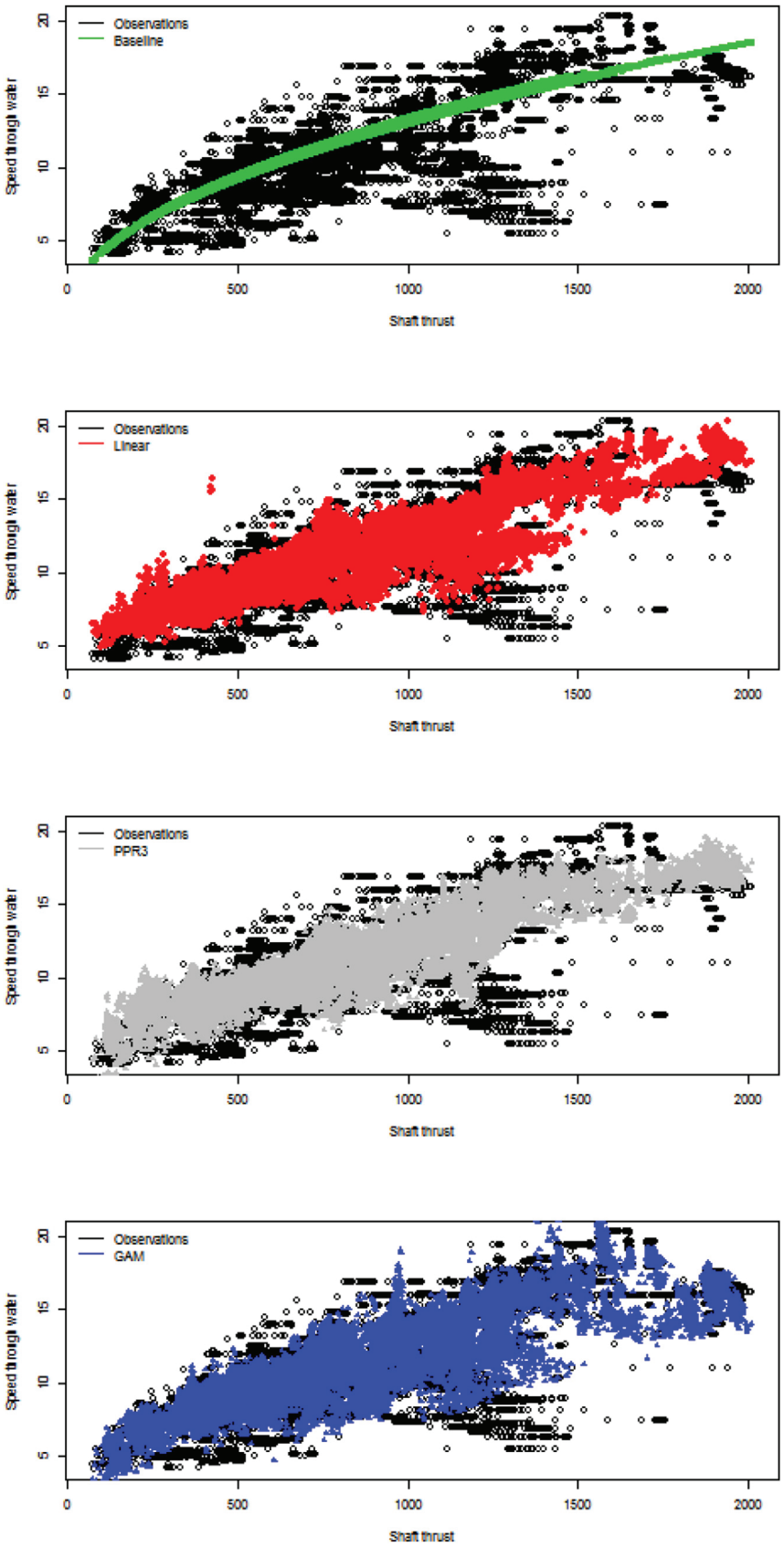


Fig. 6. Scatter plot of the shaft thrust vs the speed through water and the values predicted using the different models, using 10-fold cross validation.

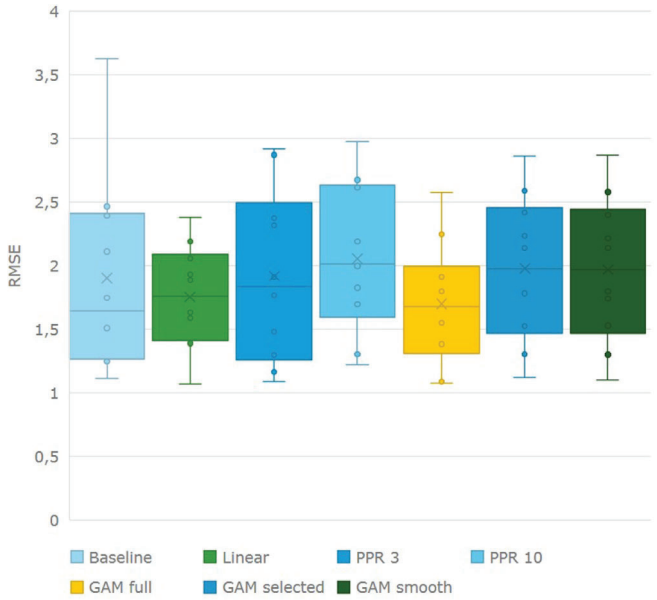


Fig. 7. Boxplot of the RMSE per fold $k \in 1, ...,10$, without repetition, for each regression model.

4.1. Fold specific results

A boxplot of the fold specific RMSE values of the different regression models is displayed in Fig. 7. Here we have used 10-fold cross validation without repetition. The GAM model achieves the lowest 75th percentile, at 1.9, while the baseline method has a slightly lower median and 25th percentile compared to the GAM model. The linear model achieves the lowest maximal RMSE value at 2.4, which indicates robust performance. The results indicate that the baseline method performs well in many cases, but as we observe from the high maximal value, which is some 50 % higher than the maximum value reported for the linear model, it is not robust in all situations.

The R^2 values, which are calculated for each fold, are displayed in Fig. 8. The coefficient varies from 0 to 1, where 1 indicates a perfect fit of the model to the data. The PPR10 achieves the highest value, followed by the full GAM model, and the other PPR and GAM's. The lowest R^2 calculated for PPR10 and the GAM model is 0.89, which indicate that the total variation of the response variable is quite well explained by the regression model.

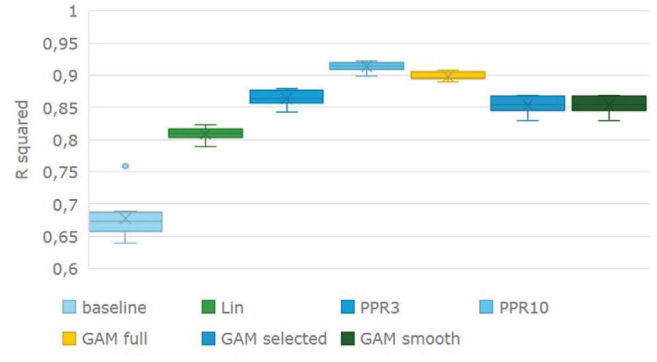


Fig. 8. Boxplot of the adjusted R^2 per fold $k \in 1, ...,10$, without repetition, for each regression model. The whiskers extend upwards to the largest element that is less than 1.5 times the inner quartile range higher than the upper quantile, and downwards to the lowest element. Elements outside this range are considered outliers, and are marked with a dot, as seen for one of the values using the baseline model.

4.2. Comparing folds with calm and harsh weather

To demonstrate the importance of representative training data, and the different regression models' strengths and weaknesses in this respect, we select two periods which represent calm and harsh weather and investigates these in somewhat greater depth.

Data from three selected weather sensors; heave, pitch and wind perpendicular to the ship are displayed in Fig. 9. Fig. 10 shows a scatter plot of the wind component and the pitch motion. The figures indicate that calm weather are well represented by fold 2, while the weather conditions seam to be more severe in fold 9. Fold 2 and 9 are marked in magenta and cyan respectively.

Fig. 11 shows a scatter plot with shaft thrust on the horizontal axis and speed through water on the vertical axis. Again, fold 2 and 9 are highlighted. The observed values from the other folds are marked in black, while the predicted values based on the baseline models are marked in green. We observe that the observed values of fold 2 lie close to the baseline predictions. This indicates that the baseline method performs well in many cases, including calm weahter (fold 2), but we also observe that the baseline method is unable to achieve accurate predictions in harsh weather (fold 9). This is supported by the fold specific RMSE calculation for the baseline method, as reported in Fig. 12. It shows that the RMSE of fold 2 is low, while the reported RMSE of fold 9, where the weather conditions are more severe, is high.

The observed speed through water and corresponding predictions produced by the baseline, linear and GAM models on fold 2 and 9 are shown in Fig. 13.

From Table 1, we see that in terms of RMSE, the baseline method outperforms both the GAM and the linear model in calm weather (fold 2), while in harsh weather (fold 9), the RMSE of both the linear and the GAM method is more than 40% lower than the baseline method.

4.3. Modified K-fold cross validation

In the modified version of K-fold cross validation the data in the folds which are adjacent to the validation set are excluded from the training set. That is for each $k \in 1,2, ...,K$ the models that are tested on $\mathscr{D}_k$ are trained on $/\{\mathscr{D}_{k-1} \cup \mathscr{D}_k \cup \mathscr{D}_{k+1}\}$. For example, the training set used for predictions of the data points in fold 11 in modified 30-fold cross validation consists of data from fold 1–9 and 13–30.

We observe that the predictions for the corresponding folds in the standard 10-fold and the modified 30-fold cross validation are similar. This is supported by the fold specific RMSE calculations, displayed in Table 2. We note, however, that the RMSE for the predictions made by the GAM model on fold 4–6 in the 30-fold cross validation differ significantly from the RMSE calculation using standard 10-fold cross validation for the corresponding fold.

5. Discussion

5.1. Changing the numbers of folds in the cross validation

In Fig. 14 the RMSE of the different regression models are displayed with standard $K-$fold cross validation for K=10, 20, 50 and 3385. In addition results using 30-fold modified cross validation is presented.

Clearly, the calculated RMSE values decrease with increasing number of folds in the cross validations. This is not surprising, due to the implicit increase in available training data. But it illustrates, however, that insufficiency in training data can lead to inaccurate predictions (Vanem et al., 2017; Petersen et al., 2012).

5.2. Ensemble methods

By combining the before-mentioned methods we are able to achieve slightly increased accuracy. For example, on average the GAM model achieves the best predictions, but we have observed that in cases where

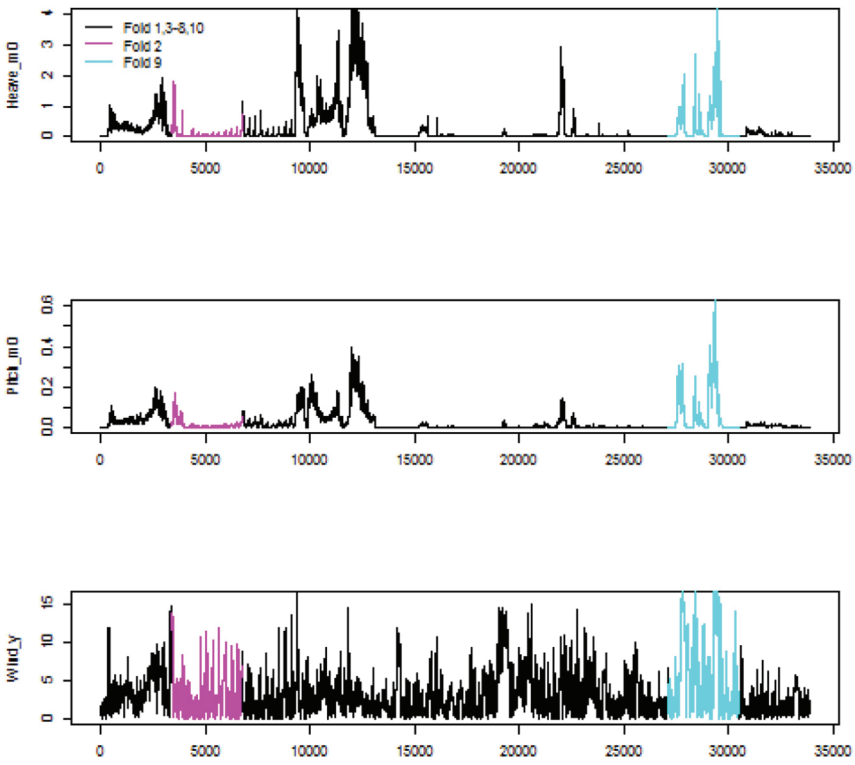


Fig. 9. Traceplot of data from selected weather sensors; heave, pitch and wind perpendicular to the ship. The dataset is divided into 10 mutually exclusive folds with equal size. Fold 2 and 9 of are higlighted in magenta and cyan respectively. Each point on the horizontal axis represents the average sensor value in the previous 5 min. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

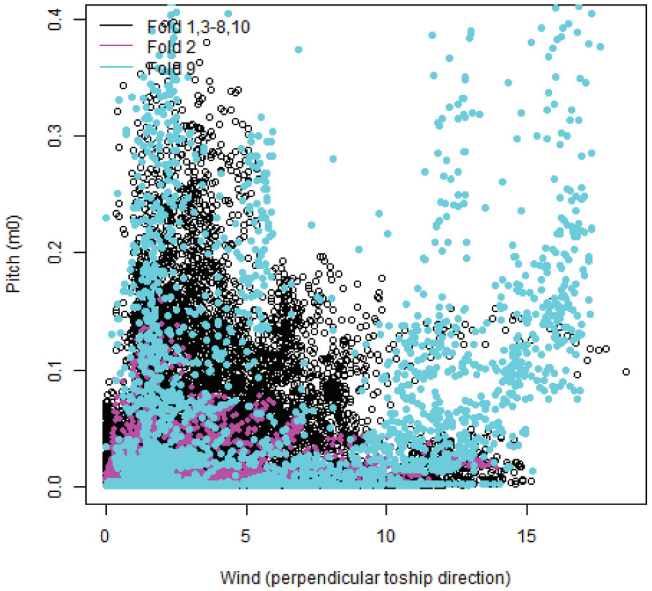


Fig. 10. Scatter plot of the wind component perpendicular to the ship and the pitch motion. The dataset is divided into 10 mutually exclusive folds with equal size. Fold 2 and 9 of are higlighted in magenta and cyan respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

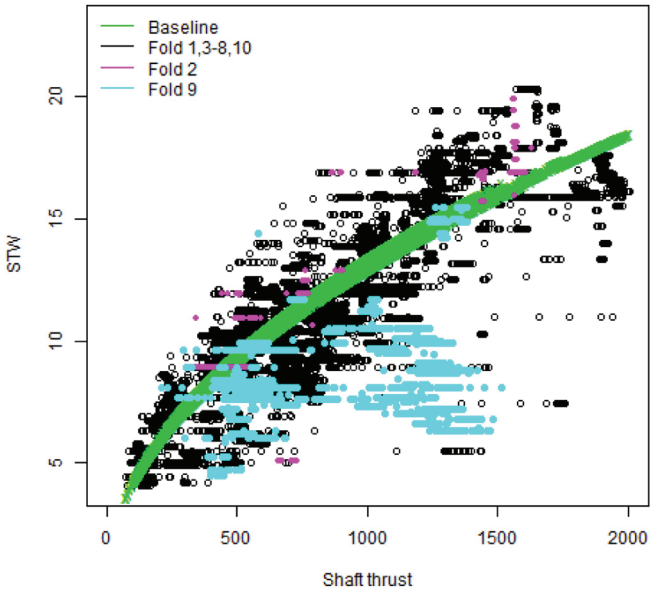


Fig. 11. Scatter plot of the shaft thrust and speed through water. The green makers are the predictions based on the baseline method. The magenta and cyan markers show the observed data in fold 2 and 9 respectively. The data from the remaining folds are marked in black. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

the pitch motion is higher than usual, as is the case in fold 9, the linear model performs better. The ensemble model selects the prediction based on the GAM model when the pitch motion is below the mean value, and reversely, when the pitch motion is above the mean value, the prediction based on the linear model is selected.

Furthermore, if we are in an anomaly detection setting, we might want to make use of the information we have on the speed through water. We might then create an ensemble method which uses one model for speed through water above a given threshold, and another model for speeds below. As an example, we observe that the baseline model achieves better prediction accuracy at high ship speeds compared to the results of the GAM model. Hence we can use the baseline model at high speeds (for example above 11 knots) and use the GAM for lower speeds (below 11 knots).

Table 3 reports the RMSE values for the two ensemble methods described above. As reference, the results of the baseline model, the full GAM model and the linear model are also reported. The percentage

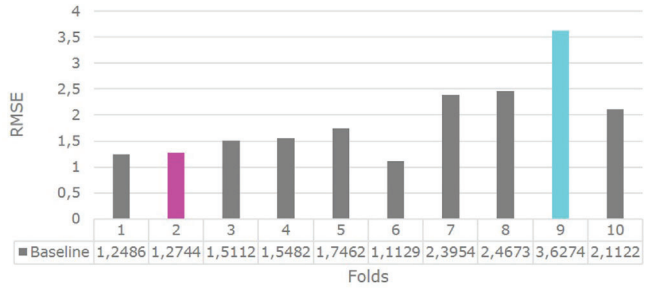


Fig. 12. RMSE of the baseline prediction, for 10 different folds. The RMSE value of fold 2, representing calm weahter conditions, and fold 9, representing harsh weather conditions, are highlighted in magenta and cyan respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

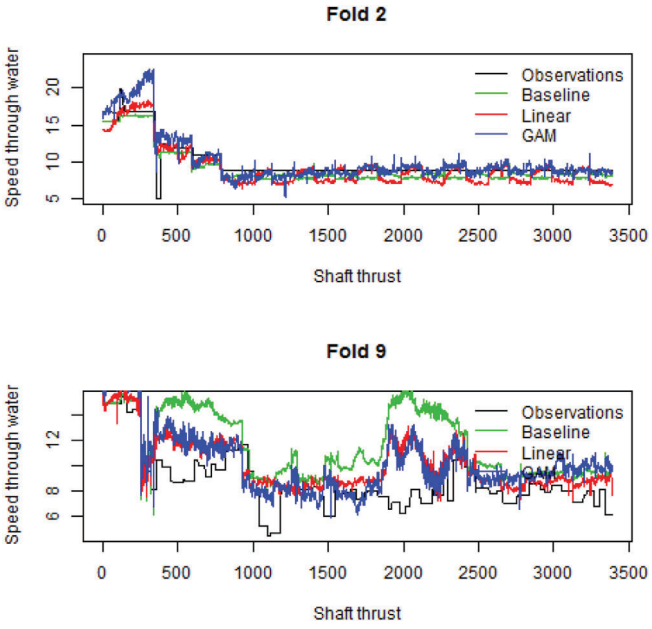


Fig. 13. Prediction with different regression models on fold 2 (calm weather) and 9 (harsh weather), using 10-fold cross validation.

Table 1
RMSE for fold 2 and 9, using 10-fold cross validation, without repetition.

Regression method	Fold 2		Fold 9		Total	
	RMSE	%	RMSE	%	RMSE	%
Baseline	1.27	0	3.80	0	2.06	0
Linear	1.39	9	2.06	−43	1.80	−12
GAM full	1.39	9	2.25	−38	1.76	−14

Table 2
RMSE for fold 4–6 and 25–27, in modified 30-fold cross validation.

Regression method	Fold 4-6		Fold 25-27		Total	
	RMSE	%	RMSE	%	RMSE	%
Baseline	1.26	0	3.57	0	2.04	0
Linear	1.40	11	2.00	−44	1.81	−11
GAM full	1.58	25	2.16	−40	1.82	−11

relative difference between the results based on the baseline method and the other methods are reported in right hand column.

This method seems a bit ad-hoc and might be prone for over fitting.

Fig. 14. RMSE for the different regression mehods, with $K = 5$, 10, 20, 50 and 3385 folds. The green bars, marked with 30 m, is the results of the modified 30 fold corss validation. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Table 3
RMSE values, including ensemble method, using standard 10-fold cross validation.

Method	RMSE	%
Baseline	2.04	0
Linear	1.80	11.9
GAM	1.76	13.7
Ensemble: GAM-Linear (based on pitch)	1.74	−14.7
Ensemble: GAM-Baseline (based on STW)	1.70	−16.7

Hence, we merely mention it here, and have not investigated it to great depth. When prediction is not the aim, and we do have information about the speed, a simple ensemble method could prove to be useful. Also, comparing the predictions produced by the different methods could be used as an indicator of the prediction accuracy.

5.3. Insufficient training data

As illustrated above, the amount and quality of the training data available are critical for the methods explored here. For example, when a ship is entering a type of operation that is not well represented in the training data this will often cause inaccurate predictions. Training data can possibly be ”borrowed” from sister ships or other ships with similar design. When the ships are not identical by design, the training data can possibly be reused after some modifications detailed by for example simulation software such as (Dimopoulos et al., 2014; Tillig et al., 2016). Notwithstanding, it should be noted that this of course can both be work intensive and might introduce biases and inaccuracies.

5.4. Operational mode selection

Typical operational modes of a ship include transit (in different speeds and loading conditions), harbour, stand by (with or without anchor) and dynamic positioning. During the different modes, the behaviour of the ship changes substantially, and it might therefore be advantageous to select among different methods based on the current operational mode. The training data should be divided and used to fit

different models. This might result in reduced computational efforts and increased accuracy (Al-Dahidi et al., 2014; Baraldi et al., 2012; Brandsæter et al., 2016).

6. Conclusions

Accurate ship speed estimates, and accompanied propulsion and fuel efficiency estimates, are vital to be able to optimize ship design and operation. This paper points at some of the strengths and weaknesses associated with the use of standard statistical methods for speed predictions. The results are compared and benchmarked with respect to a simple model based on the well established relationship between ship speed and shaft thrust.

In many cases, especially in calm weather conditions, the baseline method performs well in terms of prediction accuracy. Furthermore, the various regression models explored were not able to outperform the simple baseline method when shaft thrust was the only explanatory variable, but when environmental conditions were included, the accuracy of the predictions were significantly increased.

By the models investigated in this work both the generalized additive model (GAM) and the linear models prove most useful, with increased accuracy of 16 and 12% compared to the baseline model respectively, using 10-fold cross validation.

When the cross validation was performed on higher number of folds, the relative difference increased significantly, essentially for the GAM model, which when validated on the 50-fold and the 3385-fold cross validation, achieved an accuracy increase of 26 and 41% respectively, compared to the baseline method.

The lower RMSE achieved by the GAM model in the case where the $K-$fold cross validation were performed on higher K-s might indicate that the GAM would perform even better with a more extensive and more representative dataset available. Also the projection pursuit models were satisfactory with respect to accuracy when evaluated on a high number of folds, but they were not able to produce accurate predictions when the number of folds in the K-fold cross validations were in the lower range.

It is not surprising that the accuracy increases when the number of folds increase, since this makes more data available when training the model. Nevertheless, the large accuracy increase points at the importance of large, relevant, high quality datasets, which is difficult to obtain.

Acknowledgements

The research is partly funded by the Research Council of Norway, project number 237718 and 251396. We would like to thank Ingrid Kristine Glad (University of Oslo) and Magne Aldrin (Norwegian Computing Center) for good discussions.

Appendix A. Additional figures

Appendix A.1. Increasing the number of folds

Fig. A.15 shows a boxplot of the fold specific RMSE using 50-fold cross validation without repetition

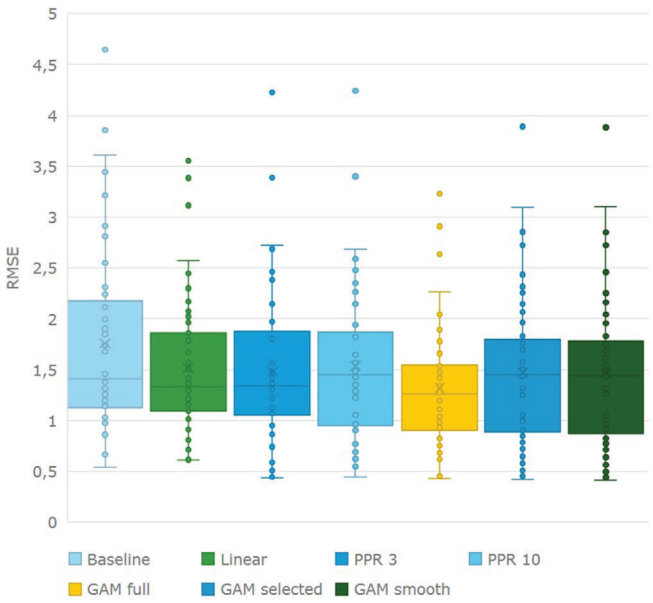


Fig. A.15. Boxplot of the RMSE per fold $k \in 1, ..., 50$, without repetition, for each regression model.

Appendix A.2. Other performance measures

The mean error $\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})$ is reported in the boxplot in Fig. A.16. Again, we use 10-fold cross validation with 7 repetitions. The figure indicate that the predictions are not biased, except for the baseline method which tend to predict too high values.

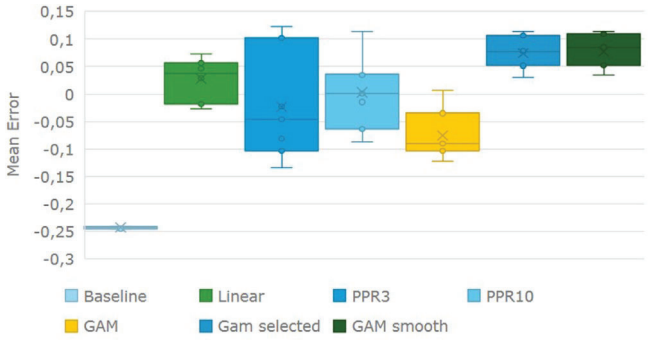


Fig. A.16. Boxplot of the total mean error per repetition $l \in 1, \ldots, 7$ for the different regression models, using 10-fold cross validation.

The relative mean absolute error (MAE) is reported in the boxplot in Fig. A.17. Here we report the results per $l \in 1, \ldots, 7$ for the different regression models, using 10-fold cross validation. The GAM model achieves the lowest value, for each of the different cross validations.

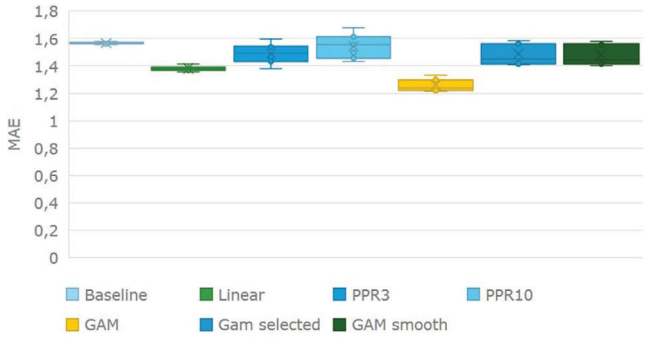


Fig. A.17. Boxplot of the mean absolute error (MAE) per repetition $l \in 1, \ldots, 7$ for the different regression models, using 10-fold cross validation.

The relative mean absolute percentage error (MAPE) is reported in the boxplot in Fig. A.18. Again, we report the results per repetition $l \in 1, \ldots, 7$ for the different regression models, using 10-fold cross validation.

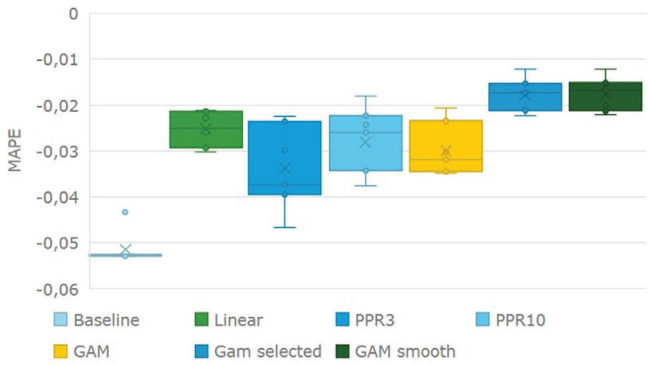


Fig. A.18. Boxplot of the mean absolute percentage error (MAPE) per repetition $l \in 1, \ldots, 7$ for the different regression models, using 10-fold cross validation.

Appendix A.3. Intermediate results

In Fig. A.19 a diagnostics plot of the linear model is shown. Here, all data are used for training, leaving no data for model validation. In this situation the obtained adjusted R squared value is 0.807.

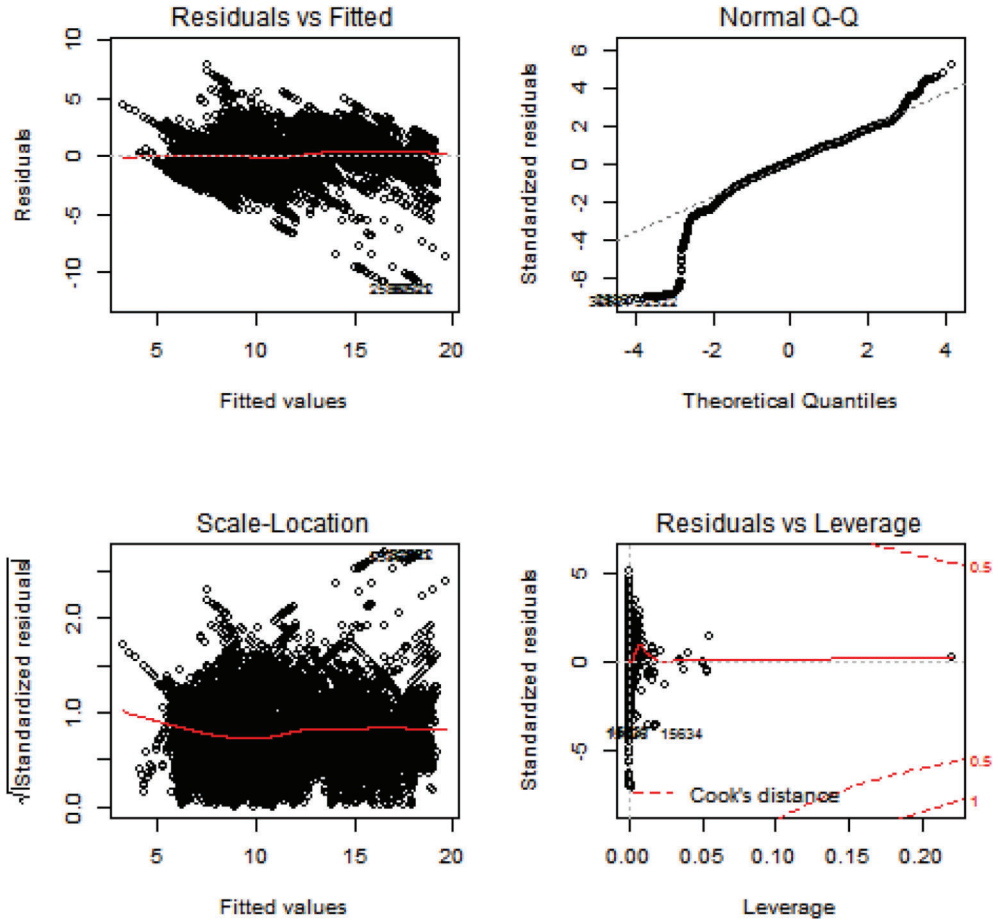


Fig. A.19. Diagnostics plot of the linear model.

A diagnostics plot for the GAM model is shown in Fig. A.20. Again, all data are used for model fitting. When this is done for the GAM model, the adjusted R squared value is 0.894. The estimated functions for the GAM model using a selection of the available explanatory variables are displayed in Fig. A.21.

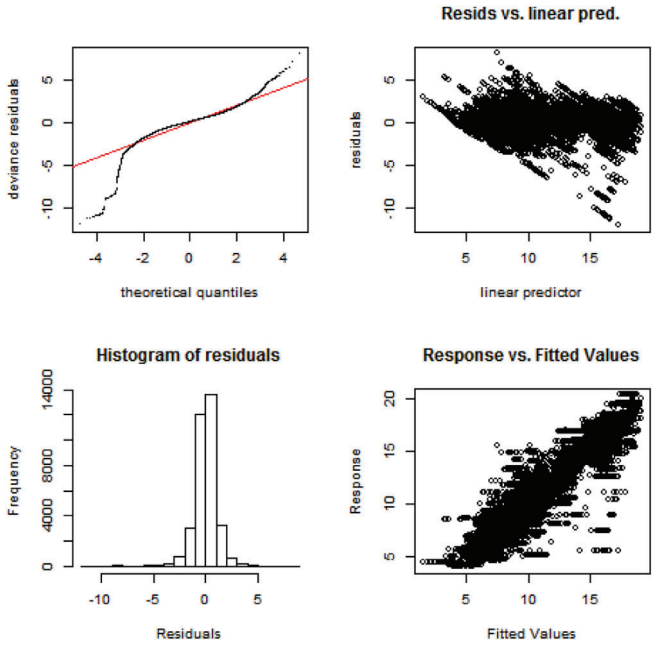


Fig. A.20. Diagnostics plot of the GAM model.

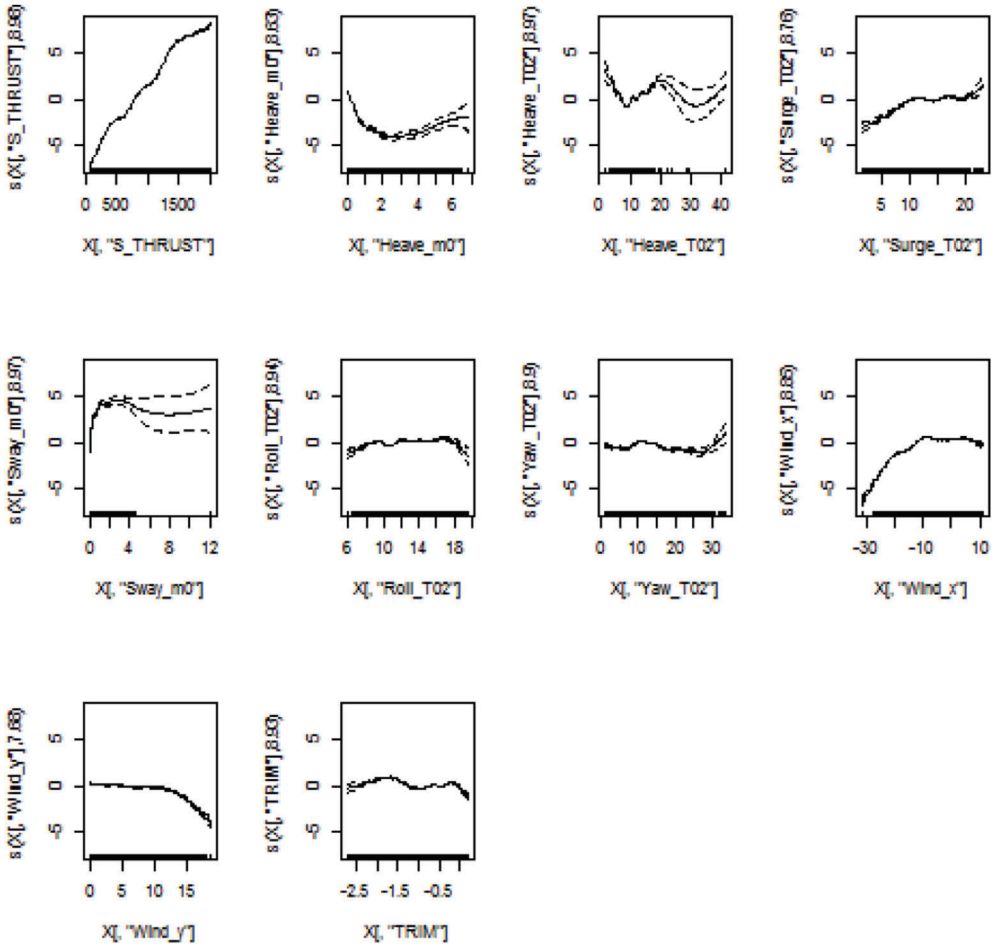


Fig. A.21. Estimated functions for the GAM model using a selection of the available explanatory variables.

Fig. A.22 displays the estimated functions for the PPR model with 10 terms. The adjusted R squared value is 0.891.

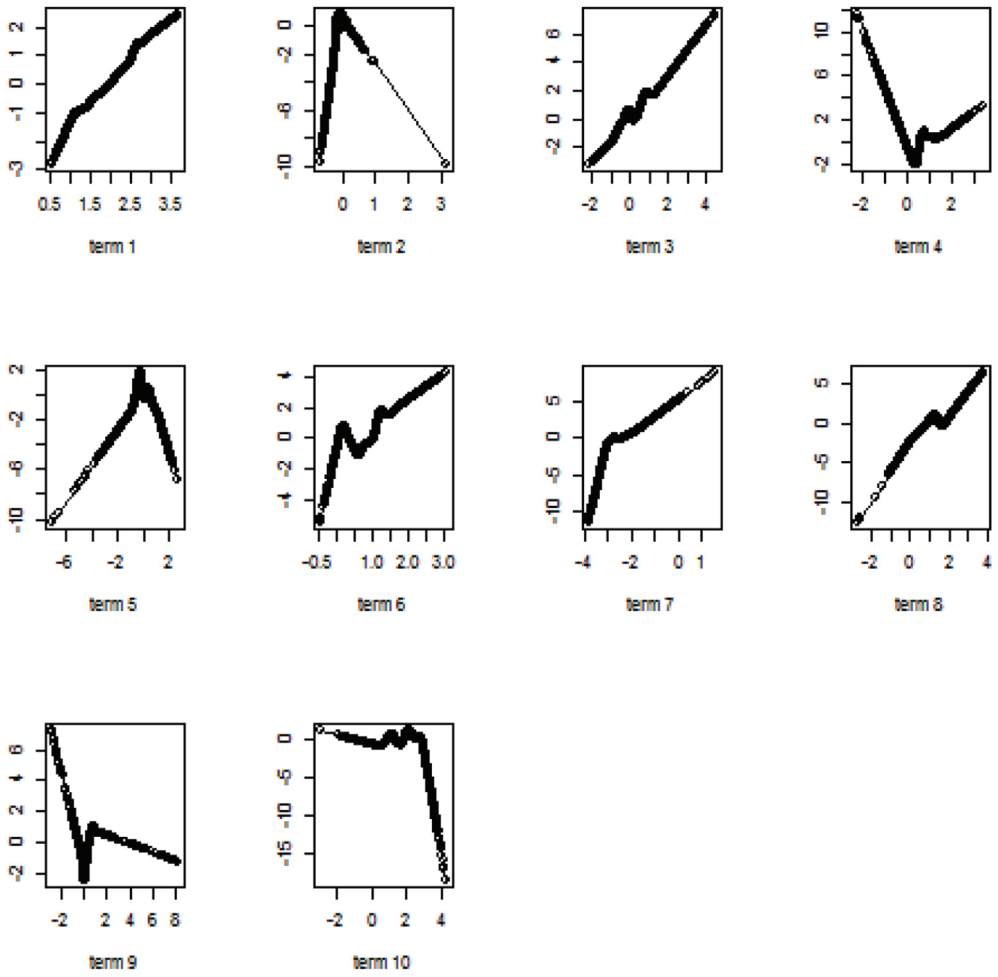


Fig. A.22. Estimated functions for the PPR model with 10 terms.

References

Al-Dahidi, S., Baraldi, P., Di Maio, F., Zio, E., 2014. Quantification of signal reconstruction uncertainty in fault detection systems. In: Second European Conference of the Prognostics and Health Management Society.

Arlot, S., Celisse, A., 2010. A survey of cross-validation procedures for model selection. Stat. Surv. 4, 40–79. https://doi.org/10.1214/09-SS054.

Baraldi, P., Di Maio, F., Pappaglione, L., Zio, E., Seraoui, R., 2012. Condition monitoring of electrical power plant components during operational transients. In: Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 226. SAGE, pp. 568–583.

Bialystocki, N., Konovessis, D., 2016. On the estimation of ship's fuel consumption and speed curve: a statistical approach. J. Ocean Eng. Sci. 1 (2), 157–166. www.sciencedirect.com/science/article/pii/S2468013315300127.

Bocchetti, D., Lepore, A., Palumbo, B., Vitiello, L., 2015. A statistical approach to ship fuel consumption monitoring. J. Ship Res. 59, 162–171.

Brandsæter, A., Manno, G., Vanem, E., Glad, I.K., June 2016. An application of sensor-based anomaly detection in the maritime industry. In: 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 1–8.

Chuang, Z., Steen, S., 2011. Prediction of speed loss of a ship in waves. In: Proceedings of Second International Symposium on Marine Propulsors Smp'11. Institute for Fluid Dynamics and Ship Theory (FDS) - Hamburg University of Technology (TUHH), German Society for Maritime Technology (STG).

Coraddu, A., Oneto, L., Baldi, F., Anguita, D., 2017. Vessels fuel consumption forecast and trim optimisation: a data analytics perspective. Ocean Eng. 130, 351–370. www.sciencedirect.com/science/article/pii/S0029801816305571.

Dimopoulos, G.G., Georgopoulou, C.A., Stefanatos, I.C., Zymaris, A.S., Kakalis, N.M., 2014. A general-purpose process modelling framework for marine energy systems. Energy Convers. Manag. 86, 325–339.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference and Prediction, second ed. Springer. http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

Holtrop, J., Mennen, G., 1982. An approximate power prediction method. Int. Shipbuild. Prog. 29.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume. 2. IJCAI'95. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1137–1143. http://dl.acm.org/citation.cfm?id=1643031.1643047.

Mao, W., Rychlik, I., Wallin, J., Storhaug, G., 2016. Statistical models for the speed prediction of a container ship. Ocean Eng. 126, 152–162. http://www.sciencedirect.com/science/article/pii/S0029801816303699.

Øyan, E., 2012. Speed and Powering Prediction for Ships Based on Model Testing.

Ozdemir, Y.H., Barlas, B., March 2017. Numerical study of ship motions and added resistance in regular incident waves of {KVLCC2} model. International Journal of Naval Architecture and Ocean Engineering 9 (2), 149–159. http://www.sciencedirect.com/science/article/pii/S2092678216305076.

Peri, D., Rossetti, M., Campana, E.F., 2001. Design optimization of ship hulls via cfd techniques. J. Ship Res. 45 (2), 140–149. http://www.ingentaconnect.com/content/sname/jsr/2001/00000045/00000002/art00006.

Petersen, J.P., Jacobsen, D.J., Winther, O., 2012. Statistical modelling for ship propulsion efficiency. J. Mar. Sci. Technol. 17 (1), 30–39. https://doi.org/10.1007/s00773-011-0151-0.

Rakke, S.G., 2016. Ship Emissions Calculation from Ais.

Sadat-Hosseini, H., Wu, P.-C., Carrica, P.M., Kim, H., Toda, Y., Stern, F., 2013. {CFD} verification and validation of added resistance and motions of {KVLCC2} with fixed and free surge in short and long head waves. Ocean Eng. 59, 240–273. http://www.sciencedirect.com/science/article/pii/S0029801812004258.

Savitsky, D., 1964. Hydrodynamic design og planning hulls. Mar. Technol. 1.

Tillig, F., Ringsberg, J., Mao, W., Ramne, B., 2016. A generic energy systems model for efficient ship design and operation. In: Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment, . https://doi.org/10.1177/1475090216680672.

Trodden, D., Murphy, A., Pazouki, K., Sargeant, J., 2015. Fuel usage data analysis for efficient shipping operations. Ocean Eng. 110 (Part B), 75–84. energy Efficient Ship Design and Operations. www.sciencedirect.com/science/article/pii/S0029801815005004.

Tupper, E. (Ed.), 2004. Introduction to Naval Architecture, fourth ed. Butterworth-Heinemann, Oxford. www.sciencedirect.com/science/article/pii/B9780750665544500008.

van den Boom, H., Hasselaar, T., 2014. Ship Speed-power Performance Assessment. No. T04. SNAME Annual Meeting, Houston, TX.

Vanem, E., Brandsæter, A., Gramstad, O., 2017. Regression models for the effect of environmental conditions on the efficiency of ship machinery systems. In: Bedford, M.R.T. (Ed.), Risk, Reliability and Safety : Innovating Theory and Practice : Proceedings of ESREL 2016 (Glasgow, Scotland, 25–29 September 2016. Lesley Walls.

Wood, S., 2017. Mixed gam computation vehicle with gcv/aic/reml smoothness estimation. february 8, 2017. https://cran.r-project.org/web/packages/mgcv/mgcv.pdf accessed: 2017-02-22.

Wood, S.N., 2006. Generalized additive models: an introduction with R. Texts in Statistical Science. Chapman Hall/CRC. http://opus.bath.ac.uk/7011/.

Paper II

Efficient on-line anomaly detection for ship systems in operation

Brandsæter, A., Vanem, E., Glad, I. K.

Expert Systems with Applications (2019), 121:418 – 437

II

Expert Systems With Applications 121 (2019) 418–437

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Efficient on-line anomaly detection for ship systems in operation

Andreas Brandsæter[a,b,*], Erik Vanem[a,b], Ingrid K. Glad[b]

[a] DNV GL, Veritasveien 1, Høvik N-1363, Norway
[b] Department of Mathematics, University of Oslo, P.B.1053, Blindern, Oslo N-0316, Norway

ARTICLE INFO

Article history:
Received 6 December 2017
Revised 20 December 2018
Accepted 21 December 2018
Available online 22 December 2018

Keywords:
Anomaly detection
Condition monitoring
Maritime industry
Auto Associative Kernel Regression (AAKR)
Cluster based AAKR
Sequential Probability Ratio Test (SPRT)

ABSTRACT

We propose novel modifications to an anomaly detection methodology based on multivariate signal reconstruction followed by residuals analysis. The reconstructions are made using Auto Associative Kernel Regression (AAKR), where the query observations are compared to historical observations called memory vectors, representing normal operation. When the data set with historical observations grows large, the naive approach where all observations are used as memory vectors will lead to unacceptable large computational loads, hence a reduced set of memory vectors should be intelligently selected. The residuals between the observed and the reconstructed signals are analysed using standard Sequential Probability Ratio Tests (SPRT), where appropriate alarms are raised based on the sequential behaviour of the residuals.

The modifications we introduce include: a novel cluster based method to select memory vectors to be considered by the AAKR, which gives an extensive reduction in computation time; a generalization of the distance measure, which makes it possible to distinguish between explanatory and response variables; and a regional credibility estimation used in the residuals analysis, to let the time used to identify if a sequence of query vectors represents an anomalous state or not, depend on the amount of data situated close to or surrounding the query vector.

We demonstrate how the anomaly detection method and the proposed modifications can be successfully applied for anomaly detection on a set of imbalanced benchmark data sets, as well as on recent data from a marine diesel engine in operation.

1. Introduction

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behaviour (Chandola, Banerjee, & Kumar, 2009). In other words, anomalies can be defined as observations, or subsets of observations, which are inconsistent with the remainder of the data set (Hodge & Austin, 2004). Depending on the field of research and application, anomalies are also often referred to as outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants (Chandola et al., 2009; Hodge & Austin, 2004). Anomaly detection is related to, but distinct from noise removal (Chandola et al., 2009).

Traditionally, sensor based component control is typically rule-based. A temperature threshold might for example be predefined, forcing the system to automatically shut-down if the temperature surpasses a predefined threshold. The problem with the rule-based approach emerges when we want to analyse multiple signals, and base our decisions on the combined behaviour. To illustrate this, we can consider two signals, x_1 and x_2, where normal behaviour is located on a circle, with an anomaly in the centre of the circle (see Fig. 1). While the anomalous point can be easily identified when we analyse both signals together, it will not be detected as anomalous if we analyse the signals separately. When we want to monitor and analyse a system with many signals, the problem space grows rapidly, making it almost impossible to describe rules that cover every permutation (Flaherty, 2017). Hence, more sophisticated anomaly detection methods are needed.

An extensive number of anomaly detection methods are described in the literature and used extensively in a wide variety of applications in various industries. The available techniques comprise (Chandola et al., 2009; Kanarachos, Christopoulos, Chro-

* Corresponding author at: Strategic Research and Innovation, Veritasveien 1, Høvik, Norway.
E-mail addresses: andreas.brandsaeter@dnvgl.com (A. Brandsæter), erik.vanem@dnvgl.com (E. Vanem), glad@math.uio.no (I.K. Glad).

https://doi.org/10.1016/j.eswa.2018.12.040
0957-4174/

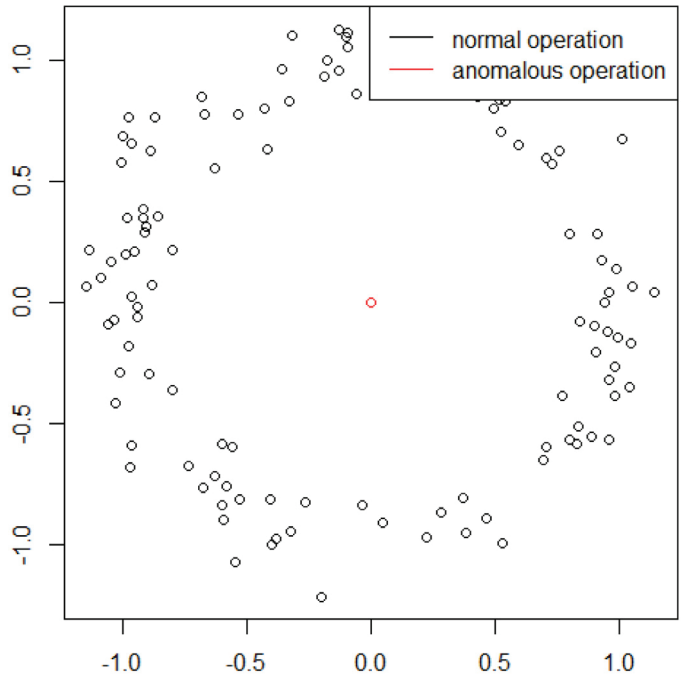


Fig. 1. Points representing normal behaviour is located on a circle. An anomaly is located in the middle.

neos, & Fitzpatrick, 2017; Olson, Judd, & Nichols, 2018; Zheng, Li, & Zhao, 2016): classification methods that are rule-based, or based on Neural Networks, Bayesian Networks or Support Vector Machines; nearest neighbour based methods, including k nearest neighbour and relative density; clustering based methods; and statistical and fuzzy set-based techniques, including parametric and non-parametric methods based on histograms or kernel functions.

The fundamental approaches to the problem of anomaly detection can be divided into three categories (Chandola et al., 2009; Hodge & Austin, 2004):

- *Supervised anomaly detection:* Availability of a training data set with labelled instances for normal and anomalous behaviour is assumed. Typically, predictive models are built for normal and anomalous behaviour, and unseen data are assigned to one of the classes.
- *Unsupervised anomaly detection:* Here, the training data set is not labelled, and an implicit assumption is that the normal instances are far more frequent than anomalies in the test data. If this assumption is not true then such techniques suffer from high false alarm rate.
- *Semi-supervised anomaly detection:* In semi-supervised anomaly detection, the training data only includes normal data. A typical anomaly detection approach is to build a model for the class corresponding to normal behaviour, and use the model to identify anomalies in the test data. Since the semi-supervised methods do not require labels for the anomaly class, they are more widely applicable than supervised techniques.

Our main motivation in this study is related to anomaly detection in the maritime industry. Modern ships are a highly complex systems, often equipped with thousands of sensors to monitor various features of the system. Our aim is eventually to identify anomalies and unexpected system behaviour that can represent faults in the system, but in principle, any behaviour that deviates from the behaviour represented in the training data can be discovered, not only faults.

We repeatedly refer to the maritime case study in many of the examples and demonstrations. However, the methods we envisage and the modifications we propose are widely applicable to anomaly detection problems concerning time series data.

In most industries, including the maritime industry, data from normal operating conditions are continuously collected on a large and increasing number of assets. However, comprehensive fault data are more rare, hence we pursue a semi-supervised approach, and present a kernel function based non-parametric statistical anomaly detection technique.

We use an on-line anomaly detection technique, consisting of two steps. In the first step, the observed signal is reconstructed under normal conditions. Secondly, the residuals, i.e. the difference between the observed signal and the reconstructed signal, are analysed. In this study, the signal reconstruction is performed using Auto Associative Kernel Regression (AAKR), (see Section 2.1), and the residual analysis is performed sequentially, with Sequential Probability Ratio Test (SPRT), (see Section 2.2).

One of the main drawbacks with the AAKR signal reconstruction method becomes evident when the set of historical observations grows large. Then the crude approach where all observations are used as memory vectors will lead to unacceptable large computational loads. Therefore, a reduced set of memory vectors should be intelligently selected (Hines, Garvey, & Seibert, 2008; Hines, Garvey, Seibert, & Usynin, 2008), and in this paper we suggest a novel approach to memory vector selection, where the original dataset is represented by sets surrounding a selection of clusters.

In Baraldi, Di Maio, Genini, and Zio (2015), the AAKR signal reconstruction method is compared with other popular signal reconstruction techniques, including Fuzzy Similarity (FS), and Elman Recurrent Neural Network (RNN), and capabilities and drawbacks are discussed. Hence, in this paper we will restrain to comparing the results of the modifications we propose to the crude AAKR method.

The remaining of the paper is structured as follows: The anomaly detection framework mentioned above will be briefly presented in Section 2. In Section 3, we propose three modifications of the standard framework:

A. *Cluster based memory vector selection method:* Perform a cluster analysis on the training data set, which represent normal conditions. Replace the original training data set with rectangular boxes - one for each cluster, centred at the cluster means - and define everything inside the boxes as normal condition.
B. *Modified distance measure between the query vector and the memory vectors:* Modifying the distance measure to enable the possibility of treating the variables differently based on the credibility of the signals, and distinguish between explanatory and response signals.
C. *Credibility estimation:* Regard some regions in the sample space more credible or trustworthy than others. Assume that the reconstruction of a response signal is more credible if the corresponding explanatory signals are similar to previously observed signals.

In Section 4, the performance of the proposed cluster based method is demonstrated on 14 different data sets - 13 benchmark data sets from the KEEL database (Alcalá-Fdez et al., 2011), and one data set from a marine engine in operation, and the results of the proposed cluster based method are compared to the results of the original (crude) method without memory vector selection. To further demonstrate the methodology and the proposed modifications, a more comprehensive study of the data set with the marine engine is presented in Section 5. A short discussion of the assumptions and results is presented in Section 5.8. Finally, in Section 6 some concluding remarks are offered, together with a discussion on further work.

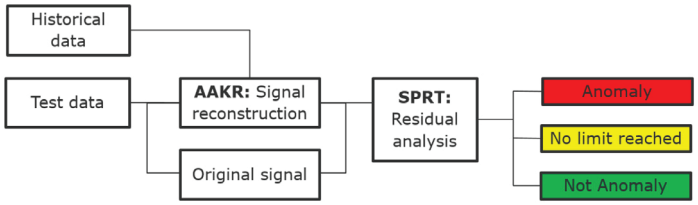


Fig. 2. The methodology can be divided into two main steps: signal reconstruction (via AAKR) and analysis of residuals (via SPRT).

The analysis is conducted in R version 3.3.3 (2017-03-06), using RStudio Version 1.0.136, on a single computer running Windows 10 Enterprise, version 1607, with Intel Core i5-6600 CPU @ 3.30 GHz processor, and 3.02GB installed RAM.

2. Standard framework for anomaly detection with AAKR and SPRT

The classical framework can be divided into two main steps: signal reconstruction and residual analysis (see Fig. 2). In particular, Auto Associative Kernel Regression (AAKR) is used for the reconstruction, and Sequential Probability Ratio Test (SPRT) is used to analyse the residuals between the reconstructed and the observed signal.

At each new time t of the on-line anomaly detection monitoring, both the reconstruction and the residuals analysis are performed in a sequential manner. In the signal reconstruction step, the values of the monitored signals are reconstructed as an estimate of the signals under normal conditions. AAKR is a data driven method where the reconstructed signal is estimated as a weighted linear combination of historical observations. The information from the current observation is used to calculate the weights. In the second step, the residuals, i.e. the difference between the observed test points (queries) and the reconstructed signals, are analysed sequentially, building evidence that the sensors report possibly anomalous behaviour.

2.1. Signal reconstruction using Auto Associative Kernel Regression (AAKR)

Many excellent descriptions of the AAKR method, both comprehensive and more brief, are given in the literature (Baraldi, Canesi, Zio, Seraoui, & Chevalier, 2011; Baraldi, Di Maio, Genini et al., 2015; Baraldi, Di Maio, Pappaglione, Zio, & Seraoui, 2012; Baraldi, Di Maio, Turati, & Zio, 2015; Di Maio, Baraldi, Zio, & Seraoui, 2013; Garvey, Garvey, Seibert, & Hines, 2007; Hines, Garvey, & Seibert, 2008; Hines, Garvey, Seibert, & Usynin, 2008). In the following we will render a basic description, following Brandsæter, Manno, Vanem, and Glad (2016).

The historical observations are collected in an $L \times J$ matrix, where L is the total number of time points of historical observations, and J is the number of sensors. If all historical observations should be taken into account by the AAKR, the reconstruction process will be very computationally expensive when the data set of historical observations grows large. Therefore, more or less intelligent selection methods (Hines, Garvey, & Seibert, 2008; Hines, Garvey, Seibert, & Usynin, 2008) are used to select some $K < L$ historical observations, or memory vectors, and collect them in a new $K \times J$ matrix $\mathbf{X}^{train}$, to be used in the reconstruction procedure.

Note that the reconstruction method does not consider time ordering, not even the sequentiality, of the observations in the training data.

At each test point t, a reconstruction of the test point $\mathbf{x}^{test}(t) = [x(t,1), \ldots, x(t,J)]$ is calculated as a weighted linear combination of the observations (the rows) in the training matrix $\mathbf{X}^{train}$. The weight $\mathbf{w}$ of a row k is given by the Gaussian kernel

$$\mathbf{w}_k = \frac{1}{\sqrt{2\pi}h} e^{-\frac{\mathbf{d}_k^2}{2h^2}}, \tag{1}$$

where the parameter h is the bandwidth, and $\mathbf{d}_k$ is the distance between the J signal measurements in the observation $\mathbf{X}^{test}_{(t,)}$ and the kth observation in $\mathbf{X}^{train}$, for $k = 1, \ldots, K$. Several distance functions can be used (Garvey et al., 2007), but the most common is the Euclidean norm

$$\mathbf{d}_k = \sqrt{\sum_{j=1}^{J} \left(\mathbf{X}^{test}_{(t,j)} - \mathbf{X}^{train}_{(k,j)}\right)^2}. \tag{2}$$

Finally, the reconstructed value $\hat{\mathbf{X}}^{test}_{(t,j)}$ of the jth observation $\mathbf{X}^{test}_{(t,j)}$, is given as the weighted linear combination of the rows of the training matrix, that is

$$\hat{\mathbf{X}}^{test}_{(t,j)} = \frac{\sum_{k=1}^{K} \mathbf{w}_k \cdot \mathbf{X}^{train}_{(k,j)}}{\sum_{k=1}^{K} \mathbf{w}_k}. \tag{3}$$

The methodology processes the various signals together. To avoid numerical instabilities due to possibly very different range of magnitudes in the different signals, the signal values need to be normalized. Without normalization, the effect of a deviation in one signal cannot be directly compared to the other signals. In the present work we have used the following normalization procedure, sometimes referred to as the z score normalization, encouraged by Di Maio et al. (2013). Having measured a signal $\mathbf{X}_{(t,j)}$, the normalized signal, $\tilde{\mathbf{X}}_{(t,j)}$ is given by

$$\tilde{\mathbf{X}}_{(t,j)} = \frac{\mathbf{X}_{(t,j)} - \hat{\mu}_j}{\hat{\sigma}_j}, \tag{4}$$

where

$$\hat{\mu}_j = \frac{\sum_{k=1}^{K} (\mathbf{X}^{train}_{(k,j)})}{K}, \tag{5}$$

$$\hat{\sigma}_j = \sqrt{\frac{\sum_{k=1}^{K} \left(\mathbf{X}^{train}_{(k,j)} - \hat{\mu}_j\right)^2}{K}}. \tag{6}$$

Alternative normalization procedures should also be investigated, such as the min max-normalization or the decimal scaling, see e.g. Saranya and Manikandan (2013). It is noted that in some situations the choice of normalization technique can influence the results significantly.

2.2. Residuals analysis using Sequential Probability Ratio Test (SPRT)

The residuals, i.e. the differences between the reconstructed value under normal conditions, and the observed test value, $\mathbf{R}_{(t,)} =$

$\hat{\mathbf{X}}^{test}_{(t,)} - \mathbf{X}^{test}_{(t,)}$, are analysed sequentially by the standard SPRT to determine if the system is in normal or abnormal state. The methodology will be briefly described in the following. For a more thorough description we suggest (Brandsæter et al., 2016; Cheng & Pecht, 2012; Gross & Lu, 2002; Saxena et al., 2008).

The normal state is described by a null hypothesis H_0, where each component of the residuals, $\mathbf{R}_{(t,j)}$, are assumed to be normally distributed with mean 0 and standard deviation σ. The anomalous state is described by an alternative hypothesis H_a, which assumes that the residuals are normally distributed with specified mean and/or standard deviation different from the null hypothesis. The SPRT is performed for each signal $j = 1, \ldots, J$ independently.

Based on the residuals $\mathbf{R}_{(t,j)}$, an index is calculated and updated sequentially for each new observation. In order to determine the condition of the system, two threshold values, A and B, are specified and at each observation the index is compared to these lower and upper decision boundaries. There are three possible outcomes at each time step:

1. The lower limit is reached, in which the null hypothesis is accepted (normal state), and the test statistic is reset.
2. The upper limit is reached, in which the null hypothesis is rejected (anomalous state), and the test statistic is reset.
3. No limit is reached, in which case the amount of information is not sufficient to make a conclusion.

For each sensor signal j, the analysis is performed on the sequence of residuals $\mathbf{r}_{(i_1,j)}, \ldots, \mathbf{r}_{(i_n,j)}$. When either of the limits are reached (outcome 1 and 2), the sequence is reset to zero. If no limits are reached (outcome 3), the sequence is extended with the new residual.

The SPRT index is given as the natural logarithm of the likelihood ratio L_a, given by

$$L_a = \frac{\text{prob of } \mathbf{r}_{(i_1,j)}, \ldots, \mathbf{r}_{(i_n,j)} \text{ given } H_a}{\text{prob of } \mathbf{r}_{(i_1,j)}, \ldots, \mathbf{r}_{(i_n,j)} \text{ given } H_0} = \prod_{i=i_1}^{i_n} \frac{f_a(\mathbf{r}_{(i,j)})}{f_0(\mathbf{r}_{(i,j)})},$$

where $f(\cdot)$ is the corresponding normal density. Note that this construction is based on an assumption of independence among the residuals.

We consider two alternative hypotheses, i.e. deviations in either direction of the mean, leading to the following indices, for each sensor j

$$SPRT_1 = \frac{M}{\sigma^2} \sum_{i=1}^{n} \left(\mathbf{r}_i - \frac{M}{2} \right) \tag{7}$$

$$SPRT_2 = \frac{M}{\sigma^2} \sum_{i=1}^{n} \left(-\mathbf{r}_i - \frac{M}{2} \right) \tag{8}$$

The standard deviation, σ, is computed from the training data. M is the mean value of the alternative hypothesis, which is decided by the user. M is usually chosen to be several times larger than σ (Cheng & Pecht, 2012).

2.3. Limitations associated with the standard framework

There are some well-known challenges and limitations related to the anomaly detection framework presented above.

An important challenge relates to the efficiency of the AAKR method. When the data set of historical observations grows large, the signal reconstruction procedure becomes very computationally costly (Michau, Palme, & Fink, 2017). To encounter this, various memory vector selection techniques are used (Hines, Garvey, & Seibert, 2008; Hines, Garvey, Seibert, & Usynin, 2008). In this paper, we present a novel cluster based memory vector selection technique, see Section 3.1.

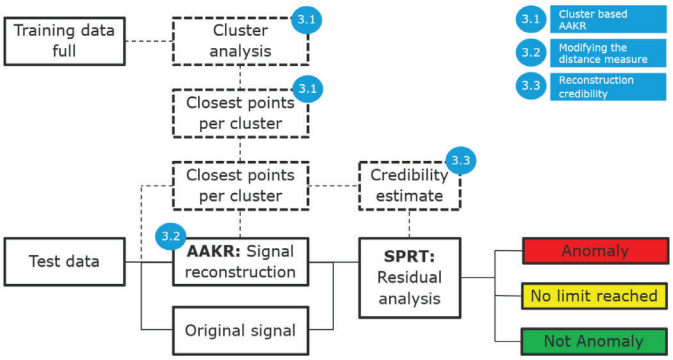


Fig. 3. The modified anomaly detection framework.

When the relative importance of the various signals is known and understood, for example based on physical meaning or by subject matter expert's experience, it should be possible to incorporate this information in the model. We propose to impose the relative importance on the AAKR model by changing the distance measure, see Section 3.2. The proposed generalization of distance measure provides the possibility to distinguish between explanatory and response signals. This also makes it more natural to compare the reconstructions produced with AAKR, with reconstructions based on other regression methods.

With the standard framework, all regions in the sample space are considered equally credible. We suggest to assume that the reconstruction of a response signal is more credible if the corresponding explanatory signals are similar to previously observed signals. In Section 3.3, we describe one possible approach to encounter this.

Other challenges associated with the anomaly detection framework, such as challenges related to time dependency and the need for representative training data, as well as problems associated with evaluating the accuracy when labelled data is lacking, are of general nature and is not addressed here.

3. Proposed modifications

In the following, we propose three novel modifications aiming to improve the anomaly detection framework as presented above, and to address associated challenges. A sketch of the suggested modified anomaly detection framework is shown in Fig. 3, with the new boxes marked with dashed borders.

3.1. Cluster based memory vector selection for AAKR

In the maritime industry, as in many other industries, the amount of available and potentially interesting data is large and growing. In the AAKR method, the distance between the observed query vector and each of the memory vectors have to be calculated, as well as the weights associated with each memory vector and eventually the weighted linear combination of all the memory vectors. Consequently, if we use a naive approach, and let all training data points be represented in the set of memory vectors, the algorithm will be very computationally costly for large training data sets. Hence, intelligent memory vector selection methods are needed.

Several memory vector selection methods exist, including vector ordering, min-max selection, combination of vector ordering and min-max selection (Boechat, Moreno, & Haramura, 2012; Coble, Humberstone, & Hines, 2010; Hines, Garvey, & Seibert, 2008). The methods all strive to adequately represent the operating conditions expected in future fault free operations. If variants of

normal operating conditions, such as changes in weather, seasonal variations, are not included in the memory vectors, no confidence can be given to predictions of the model and the memory matrix must either be appended or replaced with new data (Boechat et al., 2012; Hines & Garvey, 2006).

In our experience, a ship's operation pattern can be divided into relatively few sub-operations, such as for example harbour, transit (in a few different speeds) and manoeuvring. This relatively simple operation pattern is typically also reflected in related systems such as the machinery. Hence, we propose to use a memory selection method based on clustering, which exploits this property of the data. Our first experiences with this method was presented in Brandsæter, Vanem, and Glad (2017). Here we elaborate and systematically investigate the methodology.

3.1.1. Clustering for anomaly detection

Several clustering based anomaly detection techniques have been developed (see e.g. Chandola et al., 2009), and various categories of clustering methods for anomaly detection are suggested in the literature. One common approach is to cluster the data first, and then classify the data according to one of the following assumptions:

1. Normal data instances belong to a cluster in the data, while anomalies do not belong to any cluster.
2. Normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid.
3. Normal data instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters.

The approach we propose in this paper, is somewhat inspired by both 1 and 2 above. In brief, we suggest to first cluster all historical observations. Secondly, the regions surrounding the cluster centroids are identified. The clustering and identification of surrounding sets are performed off-line, prior to operation. Then, during operation, for each new query point, one memory vector from each of the surrounding sets are selected such that the distance between the query point and the representative of the surrounding set is minimized. Finally, the selected memory vectors are used in the AAKR reconstruction procedure. In this way, a new set of memory vectors is selected for each query vector.

3.1.2. Prediction based on representatives from the surrounding sets

After the clustering process is executed on the training data, and the surrounding sets are identified, the reconstruction of the test data can take place. The reconstruction of the query vector, $\hat{\mathbf{X}}^{test}_{(t,)}$, is produced using AAKR as described in Section 2.1, but now the training data $\mathbf{X}^{train}$ which contains selected or all historical observations, is replaced by a matrix $\mathbf{X}^{closest}$ containing the unique closest point per cluster, i.e. the ith row of $\mathbf{X}^{closest}$ is given by

$$\mathbf{p}^* = \underset{\mathbf{p}\in O_i}{\arg\min} \sum_{j=1}^{J} \left(\mathbf{p}_j - \mathbf{X}^{test}_{(t,j)}\right)^2, \tag{9}$$

where O_i is the surrounding set of cluster i. Uniqueness follows in the Euclidean space for surrounding sets that are closed and convex (Dattorro, 2010).

Hence, if a test point $\mathbf{X}^{test}$ lies inside a surrounding set O_i, the distance between the test point and the closest point in that surrounding set is 0. If on the other hand, the test point lies outside the surrounding set, the distance between the test point and the closest point in that surrounding set is strictly greater than 0, and the closest point will be on the surrounding set's border. This is illustrated in Fig. 4 a simplistic example in 2 dimensions.

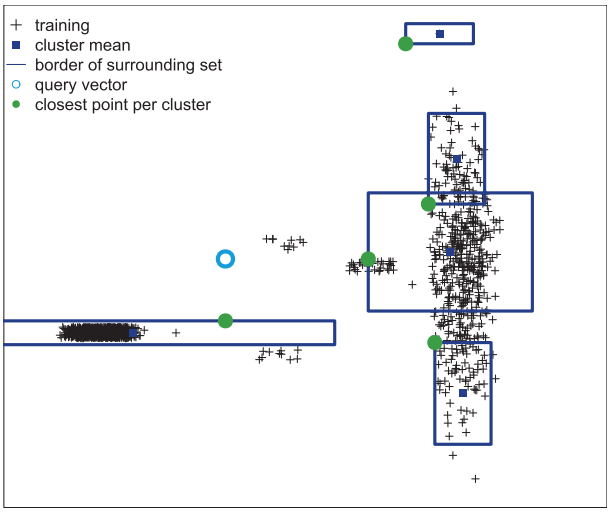


Fig. 4. Illustration of the surrounding hyperrectangles, and their unique closest points to a query vector.

3.1.3. Surrounding sets

One candidate for the surrounding set of a cluster is the convex hull of its members (see left hand plots of Fig. 5). Another suggestion is to use an ellipsoid, centred at the cluster mean with shape parameters based on the standard deviation of the cluster members, for each sensor signal (see the centre plots of Fig. 5). Furthermore, the clustering can be performed using clustering techniques such as Density-based spatial clustering of applications with noise (DBSCAN) (Ester, Kriegel, Sander, Xu et al., 1996), CLARA (Ng & Han, 1994) and CLARANS (Ng & Han, 2002). Such techniques enable identification of clusters with arbitrary shape, that are non-linearly separable, which cannot be adequately clustered with k-means or Gaussian Mixture EM clustering (Ester et al., 1996).

However, for simplicity, and due to the computational cost of calculating the distance between a query vector and the boundary of more complex shapes (Cameron, 1997; Jarvis, 1973), we chose to use axis-aligned hyperrectangles/boxes.

If the data set is in $\mathbb{R}^2$, it is possible to find the set of k axis-aligned rectangles of minimum area that covers the points in the data set using optimization techniques such as for example mixed integer and linear programming (see Ahn et al.; Park & Kim). But to our knowledge, no efficient method exists that applies to large data sets in high dimensions.

Fortunately, we do not need to determine the optimal set of hyperrectangles/boxes and can be satisfied with a good selection. Hence, we will explore the use of well-known clustering techniques to cluster the data. When the data set is divided into clusters the size and position of the hyperrectangles are determined in one of the following ways:

1. *Centred:* The boxes are centred at the mean value of the members of the cluster (in each dimension), where the distance between cluster centroids and boundary are given by the standard deviation.
2. *Enclosed:* The boxes are placed such that they cover all points assigned to each specific cluster.

In addition, a rectangle scaling factor γ is used to increase or decrease the size of the surrounding set.

Four different surrounding sets for a simplistic two dimensional example are illustrated in Fig. 5: convex hulls, ellipses, rectangles centred at the cluster mean and rectangles placed such that they cover all points assigned to each specific cluster. In the upper and lower plots, the number of clusters is set to 7 and 15 respectively.

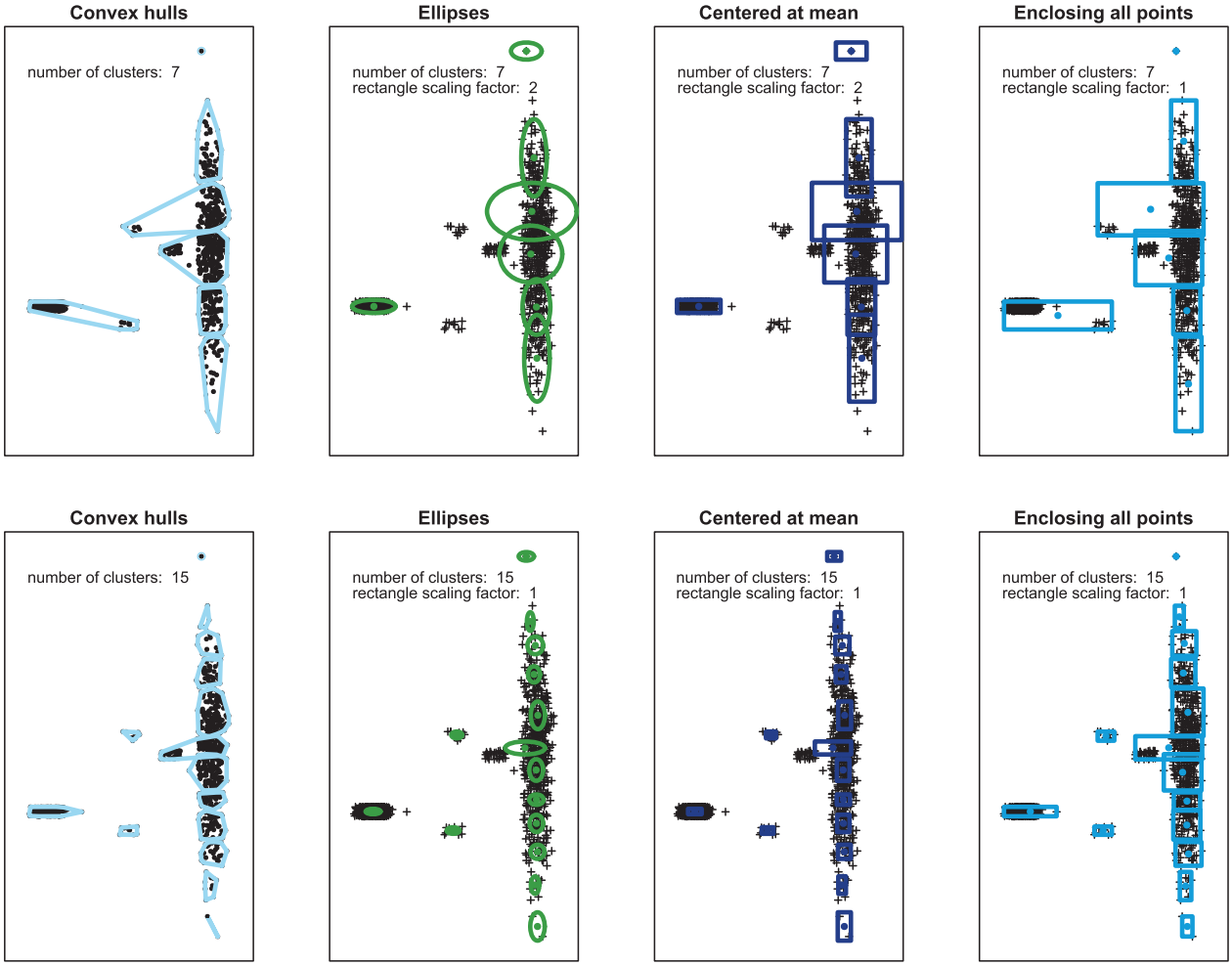


Fig. 5. Illustration of different surrounding sets, with 7 and 15 clusters (upper and lower).

Hierarchical clustering has been used to find the cluster centroids, with the complete linkage criterion, see (Section 3.1.4). The rectangle scaling factor, γ, which adjusts the shape and size of the ellipses and the rectangles is set to 2. In the lower plot, the rectangle scaling factor is set to 1, and the number of clusters is increased to 15.

3.1.4. Clustering techniques

The following clustering techniques are explored (See e.g. Cord & Cunningham, 2008; Friedman, Hastie, & Tibshirani, 2009):

1. *Standard k-means clustering:* In the initialization, k cluster centroids are chosen randomly. Then for each iteration, the observations are reassigned to the closest cluster centroid, before the cluster centroids are updated to reflect the new cluster mean. The iterations continue until the cluster centroids no longer change from one iteration to another.
2. *Agglomerative hierarchical clustering:* Each observation starts in its own cluster, and the pair of clusters with minimum distance, according to a linkage criterion, are merged. To calculate the distance between two points, we use Euclidean distance. We explore two different linkage criteria:
   - *Single:* Where the distance between two clusters A and B, is given as $\mathbf{min}\{d(a, b): a \in A, b \in B\}$, where a and b are observations assigned to cluster A and B respectively.
   - *Complete:* Where the distance between two clusters A and B, is given as $\mathbf{max}\{d(a, b): a \in A, b \in B\}$, where a and b are observations assigned to cluster A and B respectively.

3.1.5. Choosing the number of clusters

Unlike in classification tasks, cluster analysis procedures will generally be unable to refer to predefined class labels when employed in real-world applications. Consequently, there is usually no clear definition of what constitutes a correct clustering for a given data set (Cord & Cunningham, 2008). However, since the final goal of our analysis in this study is anomaly detection, which is a classification task, we can claim that the best number of clusters is the one which provides the most accurate anomaly detection. However in practice, this approach can only be utilized through cross validation, on a training set with labelled anomalies.

For standard clustering analysis, not involving classification, a wide variety of validation methods have been proposed (For an overview, see for example Cord & Cunningham, 2008; Friedman et al., 2009; Guha & Mishra, 2016; Wilks, 2011). Cord and Cunningham (2008) organize them into three distinct categories:

1. *Internal validation:* Compare clustering solutions based on the goodness-of-fit between each clustering and the raw data on which the solutions were generated.

2. *External validation:* Assess the agreement between the output of a clustering algorithm and a predefined reference partition that is unavailable during the clustering process.
3. *Stability-based validation:* Evaluate the suitability of a given clustering model by examining the consistency of solutions generated by the model over multiple trials.

In this study, we concentrate on internal validation, which means that we compare the various combinations of clustering methods and number of clusters, based on the goodness-of-fit according to some evaluation function. In addition to well-known methods such as the elbow, silhouette and gap statistic methods, there are more than thirty other indices and methods that have been published for identifying the optimal number of clusters (Charrad, Ghazzali, Boiteau, Niknafs, & Charrad, 2014). We can for example use the NbClust package (Charrad et al., 2014) in R, which provides 30 of the most popular indices for determining the number of clusters for a given data set. The number of clusters is chosen according to the majority rule. However, to allow easy comparison between the various clustering methods, and to illustrate the effect of using different number of clusters, we use a fixed array of number of clusters in the demonstration in Section 4.

As described above, choosing the optimal number of clusters is often ambiguous. Fortunately however, the cluster based AAKR method proposed in this paper, does not require that the optimal number of clusters is found. The motivation behind the clustering is to increase the computational speed. If we increase the number of clusters, we know that we should retain more of the information in the original data. But the number of clusters to use is a trade-off between computational speed and accuracy. With too few clusters, a lot of the information in the data is lost, but with sufficiently many clusters, the assumption is that we can approximate the information in the full training data with sets surrounding the clusters. The aim is to find the right balance between model performance and model run time (Hines, Garvey, & Seibert, 2008). If the model performance turns out to be poor, more clusters should be included to expand the memory matrix coverage of the operational region (Coble et al., 2010).

That being said, we see that in some of the cases presented in Section 4, the results show that the cluster based AAKR outperforms the crude method, where no clustering has been performed. We believe this is due to insufficient training data, and do not regard this performance improvement significant.

3.2. Modified distance measure to distinguish explanatory and response signals

When reconstructions are produced using AAKR, usually all signals are weighted equally when the distance between the query vector and the memory vectors is calculated. In Baraldi, Di Maio, Turati et al. (2015), a new procedure for determining the distance is proposed, where the data are projected into a new signal space, by defining a penalty vector which reduces the contribution of signals affected by malfunctioning. The procedure is motivated by the conjecture that faults or malfunctions causing variations of a small number of signals are more frequent than those causing variations of a large number of signals.

In this paper, we propose to modify the distance calculation, in a fashion inspired by Baraldi, Di Maio, Turati et al. (2015), such that the contribution of the various signals can be weighted differently. Instead of the standard Euclidean norm (see Eq. (2)), we propose to use a weighted version by multiplying the difference in each direction with a penalty vector which we refer to as the distance scaling

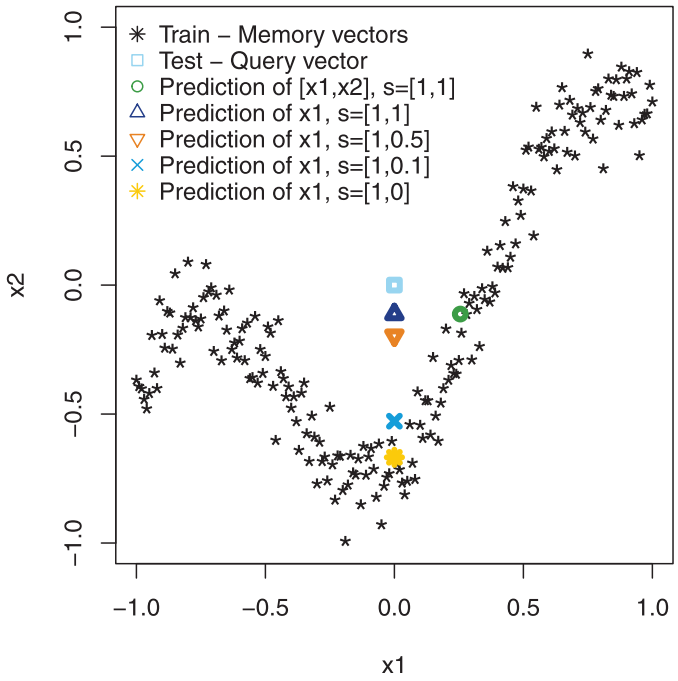


Fig. 6. Illustrating the usage of the modified distance measure, with different distance scaling vectors **s**. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

vector $\mathbf{s} = [s_1, \ldots, s_J]$. This gives the following distance measure

$$\mathbf{d}_k^{mod} = \sqrt{\sum_{j=1}^{J} \{\left[(\mathbf{X}_{(t,j)}^{test} - \mathbf{X}_{(k,j)}^{train} \right] \cdot \mathbf{s}_j\}^2}. \tag{10}$$

If all elements of **s** are equal to 1, the classical distance measure is used. Note that if one of the signals is completely disregarded, i.e. the weight is set to 0, and the weights of the other signals are not changed, then the AAKR reconstruction resembles the traditional Nadaraya–Watson estimator, where the signal with 0 weight is the response variable, and the remaining signals are the explanatory variables. This choice of **s**, also makes comparisons to other regression methods more natural.

This generalization of the AAKR method can be particularly useful when we are not interested in finding anomalies in all the sensor signals, such as sensors measuring environmental conditions. For example, if our aim is to detect anomalies that could be caused by or lead to engine failure, we might find it uninteresting to search for anomalies in the outside air temperature sensor. As long as there is nothing wrong with the sensor, there is obviously nothing wrong with the air temperature, and we are not interested in alarms regarding this. At the same time, this sensor signal could be important in explaining the behaviour in other signals, such as engine temperature or bearing temperature. Hence, we do want to be able to include it in the analysis as an explanatory variable.

In Fig. 6 the usage of the modified distance measure is illustrated with a simplistic example in two dimensions. The black coloured stars are the training data (also referred to as memory vectors), and the light blue coloured square is a query vector (also referred to as test data), located at $[x_1, x_2] = [0, 0]$. The AAKR method with the standard Euclidean distance measure would reconstruct the signal at [0.43,−0.24], as shown by the green circle. If signal x_1 measures an environmental parameter, such as for example outside temperature or wind speed, and we assume that the sensor recordings are without faults, we are not interested in

residuals in this dimension. Hence, we would regard signal x_1 as an explanatory variable, and place the reconstruction at the query vector, in this dimension. This is represented by the dark blue triangle. If we reduce the second entry of the distance scaling vector **s**, we reduce the contribution of observations that are near to the query point in the x_2 direction, and far away in the x_1 direction. The orange triangle shows the reconstructions produced with distance scaling vector **s** equal to [1,0.5], while the blue cross, and the yellow star shows the reconstructions produced using distance scaling vector [1,0.1] and [1,0] respectively.

In many real-life applications, the choice of explanatory and response variables is determined by the subject matter experts. Often, it is natural to let **s** take values 0 or 1, but other values are also acceptable. The distance scaling vector can be chosen to achieve acceptable levels of expected detection delay (EDD) and average run length (ARL), as described and demonstrated in Section 5.

3.3. Reconstruction credibility

As the training data is not evenly distributed in the data space, we propose to regard reconstructions from some regions of the sample space more credible or trustworthy than others. The idea is that we should have more confidence in our reconstructions when the query vector is close to, or at least not too far away from, the historical observations for the subset of the signals which we can treat as explanatory variables, such as environmental conditions or similar.

If reconstructions are made using AAKR with the cluster based memory vector selection method presented in Section 3.1, the number of members of a nearby cluster can also be taken into consideration when assessing the credibility of a reconstruction. One can argue that a high number should lead to higher confidence.

To illustrate the idea, we look at the simplistic example in 2 dimensions, shown in the upper plot of Fig. 7. The signal on the horizontal axis, x_1, can for example represent an environmental variable such as wind speed and we decide to treat this as an explanatory variable. Furthermore, the vertical axis, x_2, can for example represent the bearing temperature, and we decide to treat this as a response variable. Now, if we observe a value $[x_1, x_2] = [-0.75, 1.00]$ (see the leftmost red point in Fig. 7), we will be confident that this is an anomaly, since we have many historical observations of x_1 in the area around -0.75, and no corresponding values of x_2 near 1.00. However, for $[x_1, x_2] = [-0.25, 1.00]$ (rightmost red point) we have very few historical observations, hence our confidence in the reconstructions in this area is decreased.

A credibility estimate can be taken into account when the residuals are analysed in the Sequential Probability Ratio Test (SPRT). We suggest to multiply the credibility estimate with the SPRT index (see Eqs. (7) and (8)). This enables the anomaly detection framework to reach a conclusion faster when our confidence in the reconstruction is high, and use more time when our confidence is low. It should be noted, however, that the statistical properties of the SPRT will change.

3.3.1. Suggested formula for credibility estimate calculation

Different estimates can be used to calculate the credibility estimates, and we believe that different estimates should be used in different applications and cases. In the case presented here, we have used the following credibility estimate, ψ, of a query vector $\mathbf{X}^{test}_{(t,)}$,

$$\psi = 1 - \frac{1}{1 + \log(\eta^{\kappa} + 1)} \tag{11}$$

where η denotes the sum of the number of points in the surrounding sets which are close to $\mathbf{X}^{test}_{(t,)}$. A surrounding set is regarded as close if the distance between the point and the cluster centre is less than a predefined parameter λ. We experiment with different values for λ, and in the following section we show results using the following values: inf, 0.5, 0.25, 0.1, 0.05 and 0.01. When λ is infinite, all data points are regarded as close, and the credibility estimate will be constant throughout the data set. A parameter, κ, is set to control the importance of the number of points. Here, for simplicity, we fix κ to 0.1.

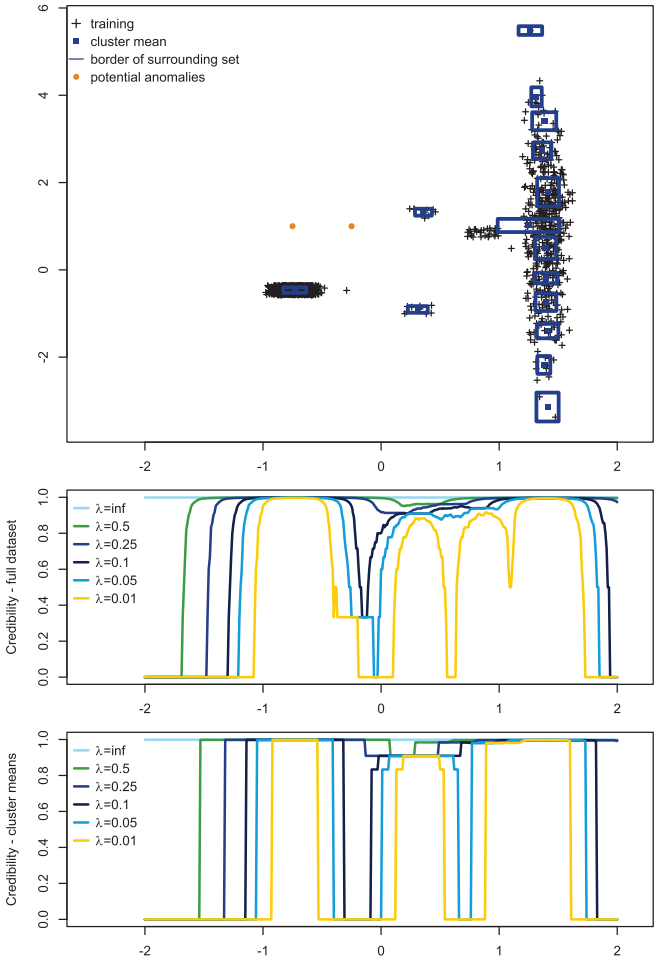


Fig. 7. The upper plot shows a simplistic data set, in two dimensions. In the two lower plots the credibility estimate is calculated for points along the horizontal axis, with different bandwidths. In the middle plot, the distances to all historical observations has been calculated, while the estimates in the lower plot are based on the distance to the unique closest point per cluster and the number of cluster members in that cluster. The number of clusters used in this figure is 15. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We see that the credibility estimate in Eq. (11) requires that the distances between $\mathbf{X}^{test}_{(t,)}$ and all the historical observations are calculated. To avoid this, we replace the full training data set with the clusters as explained in the earlier section. Also the number of points in each cluster is taken into consideration. Hence, the credibility is given by Eq. (11) where η is substituted by $\tilde{\eta}$, the sum of cluster members in clusters with nearby centres, i.e. the distance is less than a specified bandwidth.

The lines in the middle and lower plot of Fig. 7 show the proposed credibility estimates, obtained with different values of λ. The estimates in the middle plot are based on the full training data set, and the estimates in the lower plot are based on the 15 clusters and their surrounding data sets.

Table 1
Data sets used in the analysis.

Data set no.	Data set name	Imbalance ratio	No. of features	No. of training samples
1	vehicle0	3.23	18	428
2	yeast6	53.89	8	963
3	ecoli-0-1-3-7_vs_2-6	14.50	6	186
4	glass5	6.89	9	142
5	shuttle-c0-vs-c4	3.99	9	1218
6	dermatology-6	13.88	32	226
7	shuttle-6_vs_2-3	18.00	9	147
8	winequality-red-4	24.33	11	1034
9	poker-9_vs_7	12.50	10	160
10	yeast1	2.89	8	687
11	segment0	5.99	18	1319
12	vehicle2	3.23	18	409
13	vehicle3	3.04	18	415
14	engine1	1.50	5	10,000[a]

[a] Data set 14 originally includes 175,558 training samples. Due to this high number, computing the results of the crude methods is impractical. Hence, we sample 10,000 training samples without replacement, and use the result of this as an approximation of the crude method.

4. Demonstration on benchmark data sets

In this section we demonstrate the cluster based AAKR method on multiple imbalanced data sets. We present results using different clustering techniques and surrounding sets (see Section 3.1.3), and compare them to the results obtained with the crude AAKR method.

4.1. Data sets

We use 13 imbalanced data sets from the KEEL database (Alcalá-Fdez et al., 2011) (See Table 1). The rows in the data sets are pre-labelled, such that all anomalies are known, and we assume that all datapoints that are not marked as anomalies, represent normal behaviour.

The imbalanced data sets we envisage here, are data sets originated from data sets of multiple classes, where one (or more) of the classes are labelled as anomalous. For example, the imbalanced data set *yeast6* is based on the classification data set *yeast*, which contains information about a set of yeast cells, for predicting the cellular localization sites of proteins. In the classification data set, each instance is classified in 10 different localizations. In the imbalanced version, *yeast6*, the positive examples consist of class EXC and the negative examples consist of the other 9 classes. See Appendix A for a description of the other data sets.

We train on 2/3 of the data, and test on the remaining 1/3. Rows with anomalies occurring in the fraction of the data set used for training are removed.

In addition to the benchmark data sets from the KEEL database, we include another imbalanced data set from a marine engine in operation. The data set originally includes 175,558 rows. Due to the high number of rows, computing the results of the crude methods is impractical. Hence, we sample 10,000 rows without replacement, and use the result of this as an approximation of the crude method. A thorough description of this data set, together with a comprehensive analysis, is provided in Section 5.

The data sets represent various real world applications. In this section, we do not take into account any possible knowledge of the real application, and all columns of the data set are treated as equally important for detecting anomalous behaviour.

4.2. Algorithms

We present results based on the combinations of clustering algorithms and surrounding sets as presented in Table 2. The *k*-means clustering is performed with the *kmeans* implementation in the *stats* package in R (R Core Team, 2017), with the Lloyd algorithm (Lloyd, 1982). For hierarchical clustering we use the *hclust* implementation, also from the *stats* package, with the following two linkage criteria: single and complete.

Even for the largest data set, the clustering with *k*-means is performed in less than a second, hence we will not report the time to perform the clustering. For the *engine*1 data set, with 175,558 rows, the hierarchical clustering method cannot be performed due to memory restrictions. It requires that the dissimilarity structure (as produced by the *dist* function in R) is provided, which needs allocation of more than 100GBs memory.

4.3. Simple threshold based residual analysis

Many of the data sets considered in this section are not time dependent, and many of the anomalies occur alone, i.e. the observation imminently before and after are not anomalous. Due to this, we will not use the Sequential Probability Ratio Test (SPRT) when comparing the methods here. A comprehensive demonstration of SPRT will be provided in the maritime case study in Section 5. Here, we will restrain to a simple threshold method when we analyse the residuals. Again to ease the comparison between the methods, we adjust the threshold limit for each feature with a parameter τ, which controls the false alarm rate.

Furthermore, for the data sets we investigate, we have no knowledge about which signals are causing the anomaly, hence we do not distinguish this here. If an alarm is triggered in one of the signals, we consider all signals anomalous at this row/time instance.

The threshold limits obtained using this procedure should be similar to the limits we can obtain with cross validation on a training set, assuming we have known anomalies present in the training set.

4.4. Results

We present results using a range of different number of clusters, k, and a range of 50 different threshold values, τ, between 0.7 and 1. In the following, we highlight a selection of the results. The full table of results can be found in the supplementary material.

4.4.1. Decreased computation time for the cluster based methods

The main goal of the proposed cluster based method is to decrease the computation time of the different methods, and at the same time keeping the performance at an acceptable level. Fig. 8 shows savings in prediction time relative to the crude method

Table 2
Combinations of clustering algorithms and surrounding sets in the presented results.

	Clustering algorithm	Surrounding set
1	**crude**, no clustering	**points**, every point is represented
2	**k-means**, Lloyd's algorithm	**points**, centred at mean with $\gamma = 0$, i.e. every cluster is represented with a single point
3	**k-means**, Lloyd's algorithm	**centred**, centred at mean with $\gamma = 1$, i.e. every cluster is represented with a box centred at the mean of the cluster members, with size based on the standard deviation
4	**k-means**, Lloyd's algorithm	**enclosed**, every cluster is represented with a box which encloses the cluster members
5	**hierarchical**, **complete** linkage criteria	**enclosed**, every cluster is represented with a box which encloses the cluster members
6	**hierarchical**, **single** linkage criteria	**enclosed**, every cluster is represented with a box which encloses the cluster members

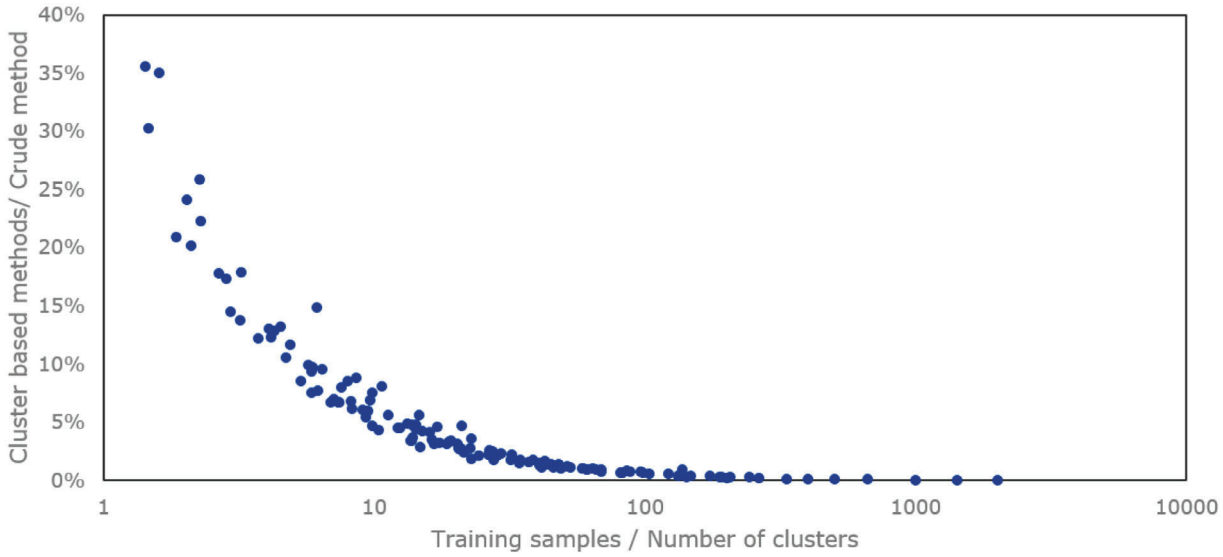


Fig. 8. Decreased computation time per prediction: The vertical axis of the figure shows the maximum computation time, when using the cluster based methods, relative to the computation time when the crude method is used. The horizontal axis represents the number of samples in the training divided by the number of clusters.

achieved with the proposed methods. The horizontal axis in the figure shows the number samples in the original training set divided by the number of clusters. As expected, as this ratio increases, i.e. when we have fewer clusters than training samples, we achieve greater time savings.

4.4.2. Comparing performance

When comparing the different methods ability to classify the anomalies, we have to balance the number of:

- True Positives (TP) - anomalous instance which is correctly identified as anomalous,
- False Positives (FP) - normal instances which are incorrectly identified as anomalous,
- False Negatives (FN) - anomalous instance which is incorrectly identified as normal
- True Negatives (TN) - normal instances which are correctly identified as normal

In this analysis, it is often useful to examine the sensitivity, which is also called the True Positive Rate. It is a measure of the probability of predicting that an instance is anomalous given that the true state is anomalous (Friedman et al., 2009). The True Positive Rate has the following expression

$$TPR = \frac{TP}{TP + FN}. \tag{12}$$

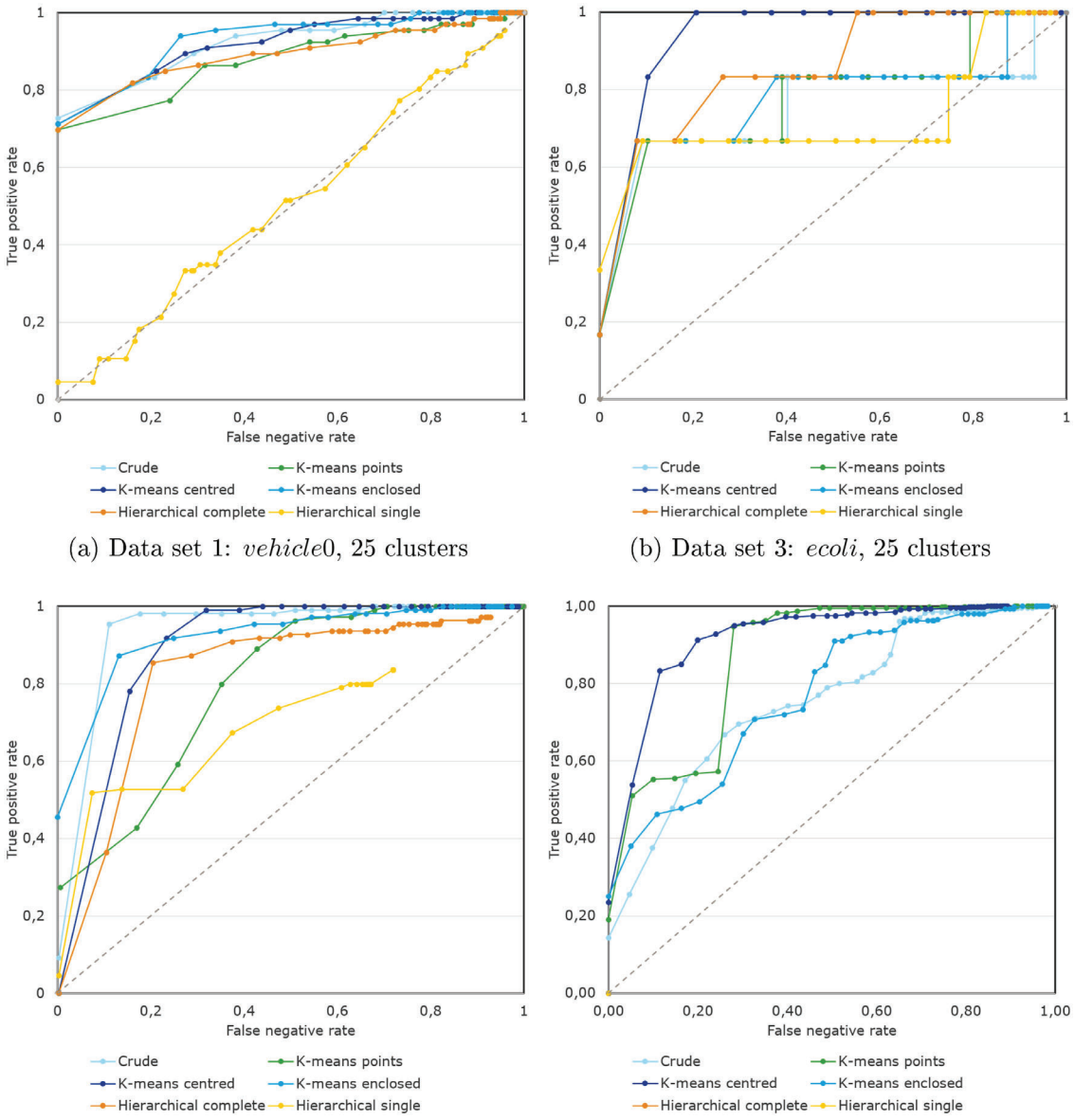
Another useful measure is the specificity, which is the probability of predicting that an instance is normal (non-anomalous) given that the true state is normal (non-anomalous). This information can also be presented as the False Positive Rate, which is given as 1 minus the specificity, that is:

$$FPR = \frac{FP}{FP + TN} = 1 - \text{specificity} \tag{13}$$

The TPR and FPR are often presented in a receiver operating characteristics (ROC) graph, which is a scatterplot with the TPR on the vertical axis, and the FPR on the vertical axis. The ROC graphs have properties that make them especially useful for domains with skewed class distribution and unequal classification error costs, which is important for cost-sensitive learning and learning in the presence of imbalanced classes (Fawcett, 2006).

The ROC graphs of four selected data sets are shown in Fig. 9. We find the most favourable results, of a ROC graph, in the upper left corner, where the FNR is low at the same time as the TPR is high. Similarly, the least favourable results are found in the lower right corner.

From Fig. 9, we observe that the different methods' performance is quite similar, except for the hierarchical clustering method with the single linkage criterion, which is clearly outperformed by the other methods especially on the *vehicle0* and *semgnet0* data sets. On the *engine1* data set, the hierarchical methods are not used due to the computational burden of performing the clustering.



(a) Data set 1: *vehicle*0, 25 clusters

(b) Data set 3: *ecoli*, 25 clusters

(c) Data set 11: *segment*0, 25 clusters

(d) Data set 14: *engine*1, 100 clusters

Fig. 9. The ROC graph for four selected data sets. Results are shown for 50 threshold values τ between 0.7 and 1. Straight lines are drawn between the points for increased readability.

In model selection, the area under the ROC curve is a popular measure, where the model with the highest area under the ROC curve will be selected. The area under the ROC-curve for the 14 data sets is provided in Table 3.

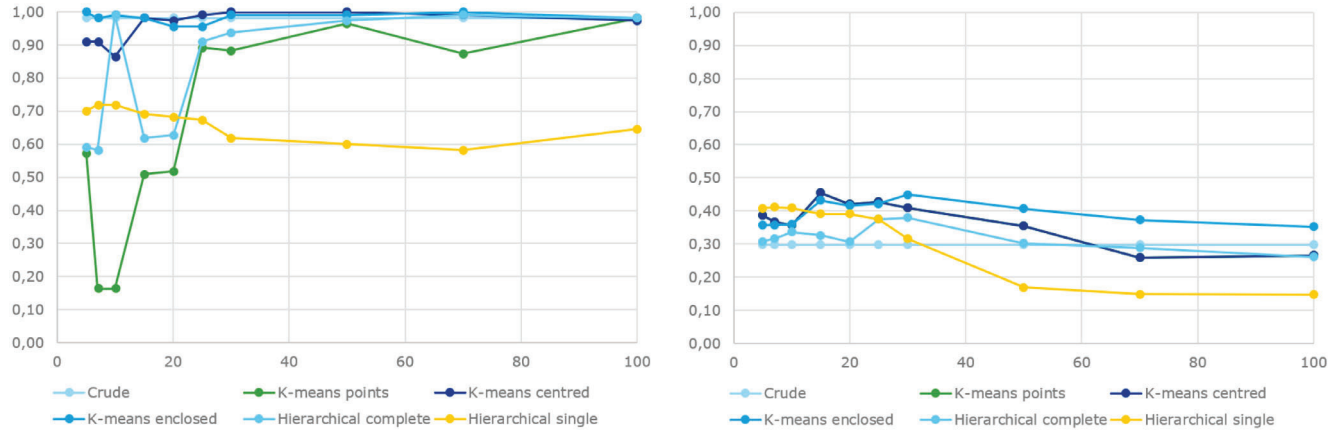
We observe that the area under the curve for the different methods are quite similar, again with a somewhat decreased performance for the hierarchical clustering with the single linkage criterion. The performance differs extensively on the different data sets, with area under the curve as high as 1.00 on some data sets, meaning that all instances are correctly labelled, both the true normal and the true anomalous. On other data sets, however, the performance is quite low, and for some data sets even close to 0.5. That being said, we have not investigated how subtle the anomalies are in the different data sets. In some of the data sets, the anomalies can be very obvious, and in others they can be well-hidden. Hence, the numbers presented here are intended for comparison of performance of the proposed methods with each other, and with the crude method. Our claim is not that the proposed cluster based methods are specifically suitable to solve the particular problems of the specific data sets, but we aim to demonstrate that the best proposed cluster based methods efficiently can achieve performance results comparable to the crude method, while inducing considerable reduction in computation time.

4.4.3. Number of clusters

Fig 10 illustrates how changes in the number of clusters used affects the performance. In figure (a) and (b) respectively, the True Positive Rate and True Negative Rate for the *segment0* data set are shown for various number of clusters. The threshold value τ is kept constant at 0.97. We observe, as expected, that the results converge towards the result of the crude method, as the number of clusters increases. However, we also observe surprisingly good results with very few clusters for all methods, except the hierarchical clustering method which uses the single linkage criterion.

Table 3
Area under the ROC curve. (Hierarchical clustering is not performed for data set 14 due to the large size of the training set). The number of clusters is 25 for data set 1–13. For data set 14, 100 clusters are used.

Dataset	Crude	K-means points	K-means centred	K-means enclosed	Hier. complete	Hier. single	Time crude	Time cluster	Relative time
1	0.92	0.88	0.92	0.91	0.89	0.47	179	5.2	2.9%
2	0.59	0.71	0.52	0.52	0.35	0.16	371	4.2	1.1%
3	0.75	0.77	0.93	0.76	0.83	0.69	11	0.6	5.4%
4	0.41	0.62	0.57	0.55	0.60	0.51	9	0.7	8.3%
5	1.00	1.00	0.98	0.97	1.00	0.71	629	5.8	0.9%
6	1.00	1.00	1.00	1.00	1.00	0.86	64	3.9	6.2%
7	1.00	0.99	0.99	0.97	1.00	1.00	10	0.7	7.5%
8	0.59	0.70	0.59	0.60	0.59	0.33	538	6.0	1.1%
9	0.96	0.86	0.95	0.97	0.93	0.14	12	0.9	7.5%
10	0.61	0.51	0.54	0.48	0.53	0.24	265	4.3	1.6%
11	0.91	0.79	0.88	0.90	0.75	0.45	1511	15.3	1.0%
12	0.94	0.81	0.88	0.88	0.79	0.46	159	5.5	3.5%
13	0.73	0.74	0.77	0.74	0.68	0.55	158	5.0	3.2%
14	0.73	0.82	0.81	0.75			5834	19.6	0.3%



(a) True Positive Rate: TP/(TP+FN) (b) False Positive Rate: FP/(FP+TN)

Fig. 10. The True Positive Rate and False Positive Rate of data set *segment0* is shown, where the number of clusters used by the cluster based methods vary on the horizontal axis. The threshold value τ is kept constant at 0.97.

5. Marine engine case study with comparisons

In this section, the anomaly detection framework using AAKR in combination with SPRT, both with and without the modifications proposed in Section 3, are applied on the data set consisting of sensor measurements from a large marine diesel engine. The data is collected from a large ocean going ship in operation.

We limit the further analysis to only consider the surrounding sets that are centred at the cluster mean. The size of the surrounding sets are determined by the standard deviation of the cluster members, multiplied with the rectangle scaling factor γ. We present results using three different sizes of γ, and refer to them as points ($\gamma = 0$), small rectangles ($\gamma = 0.5$) and large rectangles ($\gamma = 1$).

5.1. Data description

The data is collected over a period of 10 months, starting in December 2014. A total of 333,144 observations are recorded, which includes idling. In this study, we concentrate on normal operation and use a simple filter based on engine speed [rpm] to remove the idling states, leaving us with a data set consisting of 175,558 rows.

We consider the following sensors:

- engine speed [rpm],
- lubricant oil inlet pressure [bar],
- lubricant oil inlet temperature [C],
- engine power [kW]
- engine bearing temperature [C]

The bearing temperature is considered the response signal, and the others are used as explanatory variables, when this is distinguished. The time series are shown in Fig. 11.

5.2. Operational mode

The ship investigated in this study, is operated in different operational modes, such as transit (in different speeds), port and stand by (with or without anchor), in addition to transient modes. A ship is in a transient mode when its operation changes from one defined mode to another. According to our experience, these modes are the most challenging ones, in respect to anomaly detection.

5.3. Cross validation

When predictions from a statistical model is evaluated on the data set used to train the model, the accuracy estimates tend to be overoptimistic (Arlot & Celisse, 2010). Hence, the data set $\mathcal{D}$ should be divided into exclusive parts where one part, $\mathcal{D}_{train}$, is used to train the model, and the other, $\mathcal{D}_{test}$, is reserved for testing. To build robust and accurate models we ideally want to include all data available in the training data set. The same applies to testing; we want to test our models in many situations. Cross validation introduces various methods of repetitively splitting the

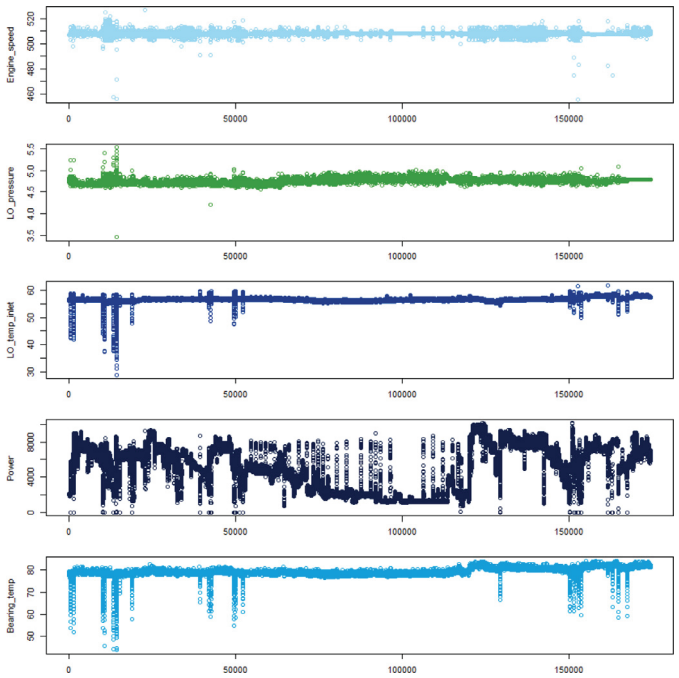


Fig. 11. Time series with training data for the evaluated signals.

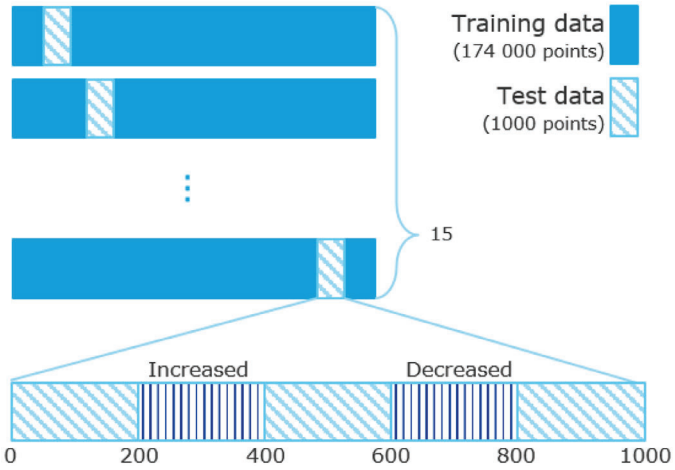


Fig. 12. Illustration of the test set up.

data into training and test data sets. A range of different splitting techniques can be applied. See for example (Arlot & Celisse, 2010; Kohavi, 1995) for a brief overview of the most common splitting techniques. We also note that repeated k-fold cross validation can be used to stabilize the error estimation and reduce the variance (Jiang & Wang, 2017; Kohavi, 1995; Rodriguez, Perez, & Lozano, 2010).

In this study, we repeatedly select folds or time intervals containing 1000 query vectors, which constitute the test data set, $\mathcal{D}_{test}$. The remaining 174,000 points constitute the training data set $\mathcal{D}_{train}$. We repeat this procedure 15 times, leaving us with a total of 15,000 tested points.

5.4. Fault simulation

To our knowledge, no faults or anomalies are registered and reported by the crew, shipowner, etc. for the data set we envisage. Hence, we assume that the data set represent normal behaviour and we define normal states based on this data.

To be able to test the anomaly detection framework, we alter some of the signals to simulate faulty states. The anomaly we induce in the test data, is a temperature change in one of the main bearings of the engine. The other signals remain unchanged. For each test set $\mathcal{D}_{test}$, we increase the temperature with A^+ degrees Celsius in the area 200:400, and decrease the temperature with A^- degrees Celsius in the area 600:800. The set up is illustrated in Fig. 12.

The signals are only altered slightly. Fig. 13 shows a scatter plot comparing the training and the test data set, with both A^+ and A^- set to 1.0. The training data are shown in purple, and the test data are shown in blue, green and red, to mark the normal state and the two states with increased and decreased temperatures respectively. On the diagonal, a density plot of each individual signal are shown. The correlations are shown in the upper triangle. We observe that the test values, both in the regions with normal condition, and in the regions were we have altered the signals, lie within the normal operating mode of that specific signal. Hence, a rule based anomaly detection method based on a single threshold would not be able to detect the anomaly.

5.5. Evaluating the signal reconstruction

First, we evaluate the signal reconstructions, by comparing the root mean squared error (RMSE) under various conditions. When no anomalies or faults are present in the data, we want the difference between the observed signals and their reconstructions to be as small as possible. The RMSE of the reconstructed temperature signal using the proposed cluster based AAKR is shown in Fig. 14. Due to high computational cost, for very large number of clusters, we select a subset of the available data consisting of 20,000 points, and produce predictions combining different number of clusters and rectangle scaling factors. Here, no anomalies are simulated (A^+ and A^- are set to 0), and the data are assumed to be collected from normal operation.

Note that a rectangle scaling factor of 0 corresponds infinitely small rectangles, i.e. points. Hence, if the rectangle scaling factor is 0, and the number of clusters is equal to the number of historical observations, the reconstruction method resembles the standard AAKR method with the crude memory vector selection where all historical observations are included. The RMSE, using this method, is shown in the lower right hand corner in Fig. 14.

The choice of number of clusters depends on the requirements in calculation time. More clusters will increase accuracy, but computation time will also increase. In this study, we chose to use 100 clusters, and experiment with three rectangle scaling factors 0, 0.5, and 1. We refer to these three options as points, rectangles and large rectangles respectively.

5.5.1. Difference in RMSE with and without anomalies

For the Sequential Probability Ratio Test (SPRT) to be able to successfully detect anomalies, the residuals, i.e. the difference between the observed and the reconstruction signals, should be more pronounced for observations from the anomalous states, compared to observations from normal state. To indicate how the residuals change when we induce anomalies, we reconstruct the signals on the 15 different folds, and calculate the RMSE before and after the anomalies are induced.

The results are shown in the box plots in Fig. 15, for the 15 different folds. Results based on the crude AAKR, where all historical observations are included as memory vectors, and the cluster based version with points (infinitely small rectangles), rectangles and large rectangles are shown. We observe that the calcu-

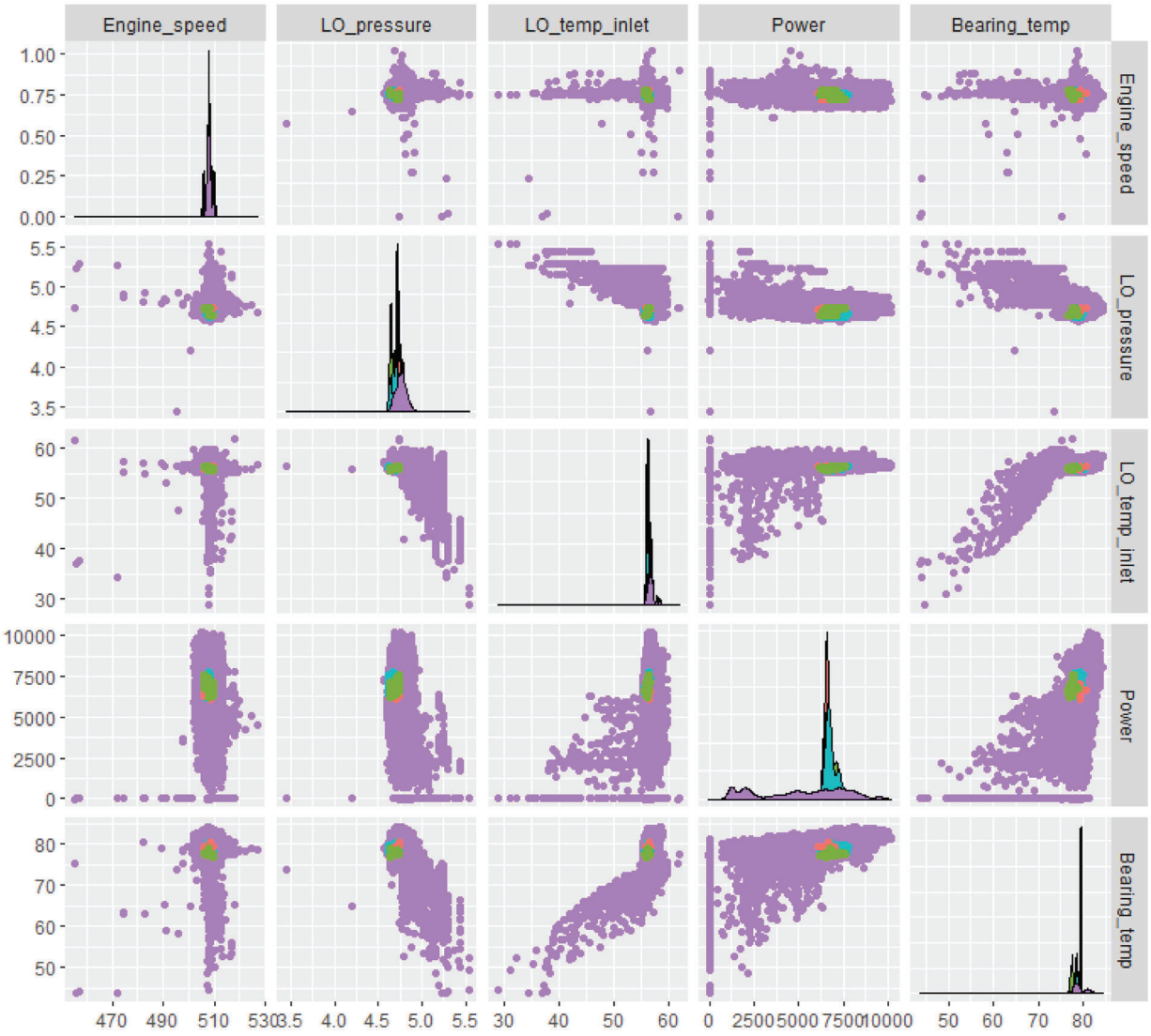


Fig. 13. A scatter plot comparing the training (purple) and the test data set from one of the tested folds, which contains two regions with anomalies (red and green), and the remaining points are considered normal (blue). In this illustration, the training and test data consists of 174,000 and 1000 points respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

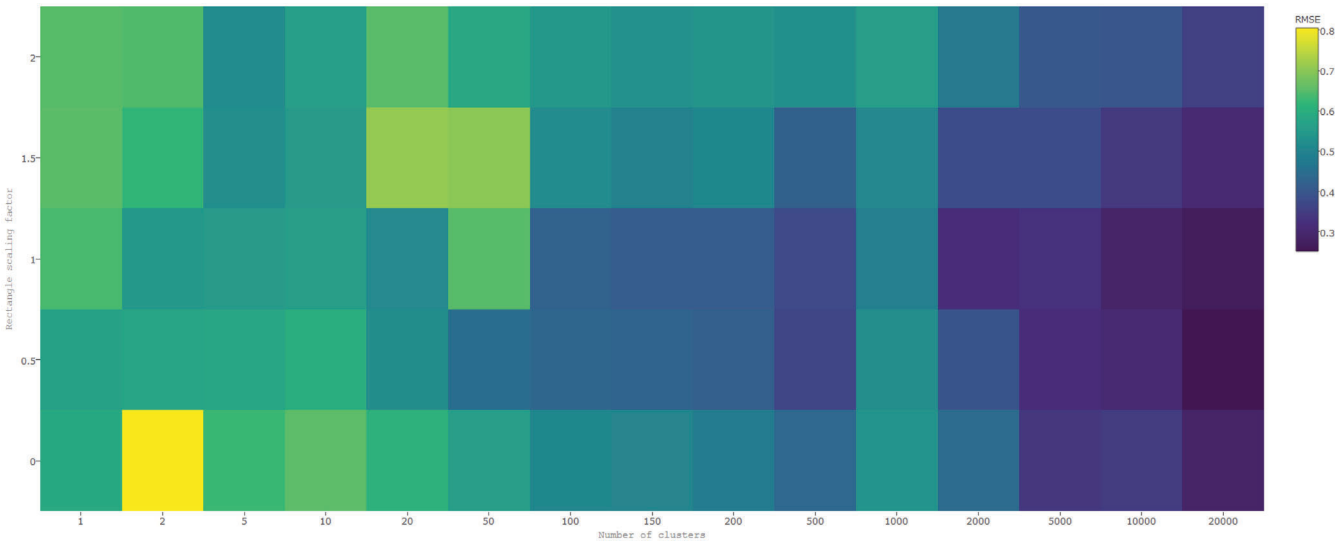


Fig. 14. The root mean squared error (RMSE) of the cluster based AAKR, with different number of clusters and different rectangle scaling factors. Note that when the number of clusters is equal to the number of points, in this example 20,000, and the rectangle scaling factor is set to 0, it resembles the crude AAKR. The kernel bandwidth h is set to 0.2.

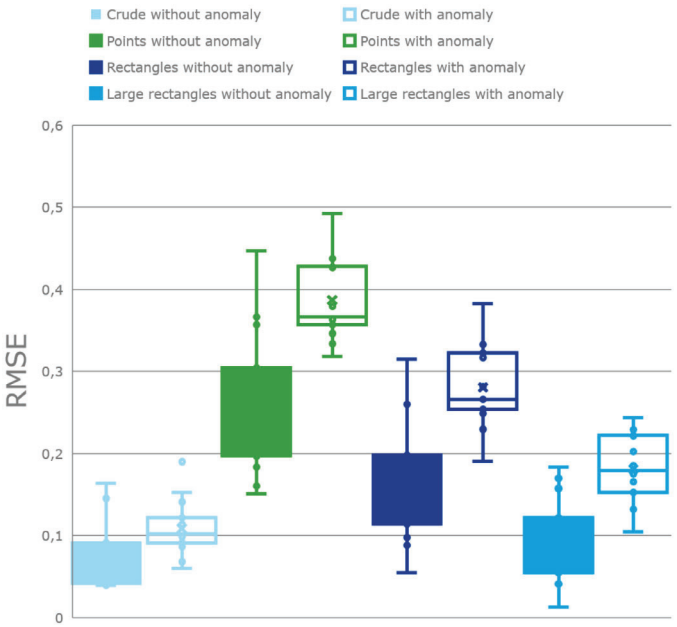


Fig. 15. Box plot of RMSE values calculated with the different memory vector selection methods with and without induced anomalies, on 15 different folds or folds.

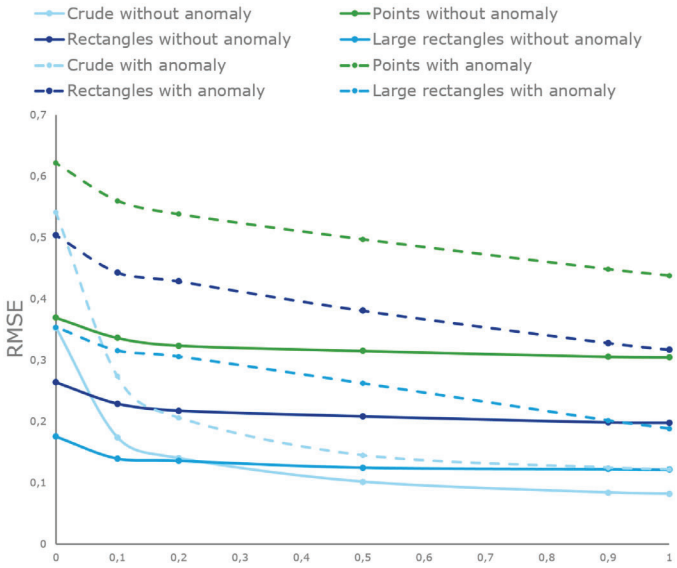


Fig. 16. RMSE values calculated based on reconstructions using the different memory vector selection methods. Values based on calculations with and without anomalies induced are showed with in filled and dotted lines respectively. Here, we vary the Jth component of the distance scaling vector **s**, and keep the other distance scalings factors constant at 1. The Jth signal is the bearing temperature.

lated RMSE is greater after anomalies are introduced, which indicates that it should be possible to detect the anomalies. The lowest RMSE is achieved with the crude method, closely followed by the method which use large rectangles. We observe that the differences between RMSE before and after anomalies are induced are more pronounced for reconstructions based on the cluster based methods.

5.5.2. Distance scaling vector

Now we analyse how the distance scaling vector **s**, as introduced in Section 3.2, effects the RMSE before and after anomalies are induced. Fig. 16 shows the average of the RMSE calculated from the different 15 folds. The filled and dotted lines are based on calculations before and after anomalies are induced respectively. Here, we only vary the Jth component of the distance scaling vector **s**, and keep the other distance scaling vectors constant at 1. The Jth signal is the bearing temperature.

When the Jth component of the distance scaling vector is 0, the results of both the crude method and the cluster based methods are small and similar, with values in the range [0.12,0.15]. For larger values of the Jth component of the distance scaling vector, we observe a significant difference in favour of the cluster based version. Remember, when anomalies are induced we want the AAKR method to produce reconstructions resulting in large residuals, and large RMSE values, while for fault-free signals, without anomalies, we want the RMSE values to be as low as possible.

5.5.3. Analysing the empirical distributions of the residuals

The empirical distribution of the residuals based on reconstructions made with the crude AAKR and the cluster based AAKR, with large rectangles, rectangles and points as surrounding sets, are shown in Fig. 17. As described in Section 5.4, a positive and negative change in mean has been induced in the time intervals 200:400 and 600:800 respectively. Outside of these two time intervals, no anomalies are induced.

The vertical dotted lines in the figure show the means of the three hypotheses; H_0 in the middle, where no anomalies are induced, and the two chosen alternative hypotheses, H_1 on the right hand side and H_2 on the left hand side, for respectively positive and negative changes in mean.

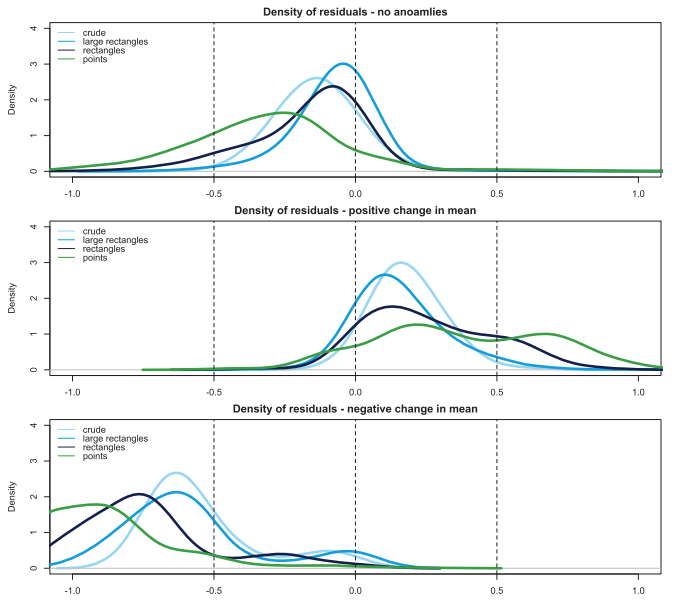


Fig. 17. Estimated densities of the residuals based on the reconstructions from the crude AAKR and the cluster based AAKR, with large rectangles, rectangles and points as surrounding sets. In the upper plot, the densities are based on signals that are not changed. In the middle and lower plot, the densities are based on values from signals that are altered in the positive and negative direction respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

When no anomalies are introduced, we expect the residuals to be small, and centred around zero. The estimated densities of the residuals, when no anomalies are induced, are shown in the upper plot of Fig. 17. We observe that the residuals are mainly situated around zero, but especially the density of the residuals based on

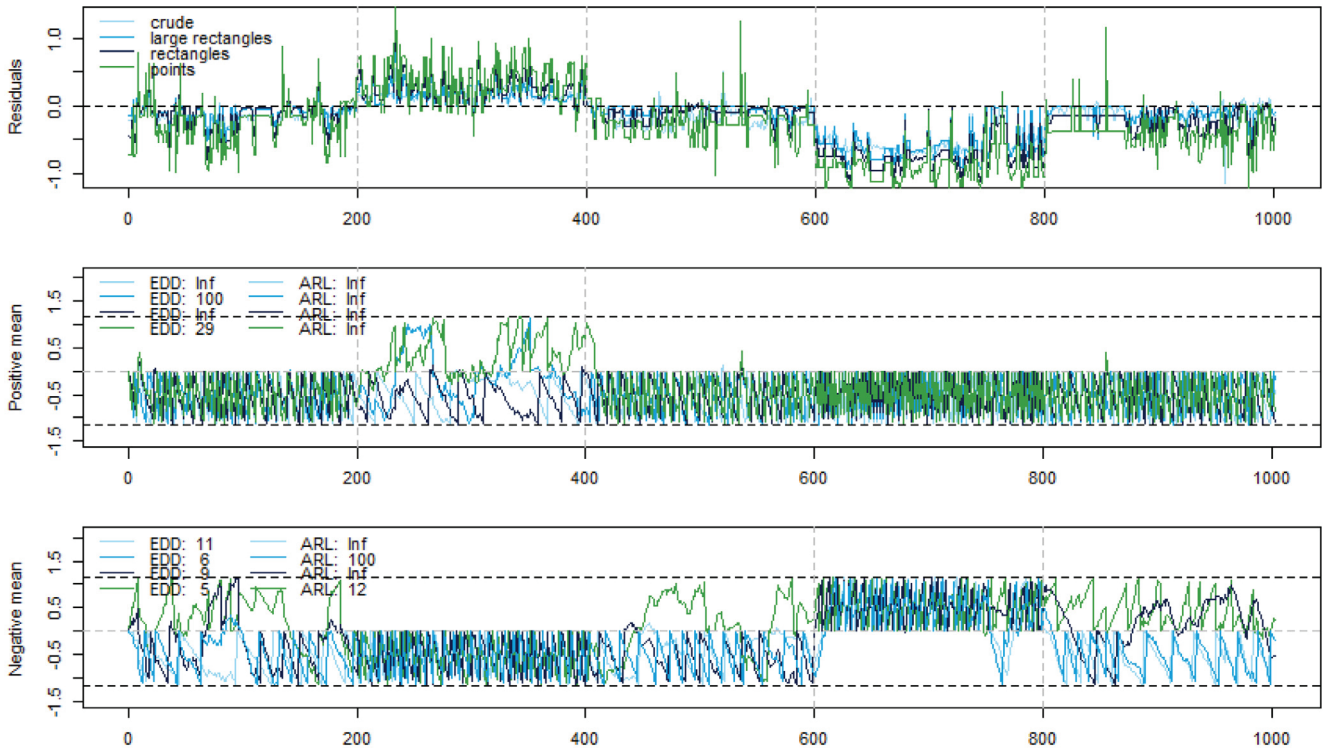


Fig. 18. The residuals are shown in the upper plot. The middle and lower plot show the SPRT indices for positive and negative changes in the mean.

reconstructions using points as surrounding sets (green line) seems to be shifted in the negative direction.

The middle and lower plots show estimated densities from signals which are altered to mimic anomalies. Residuals based on a positive and a negative change in mean are shown in the middle and lower plots respectively. The middle plot shows a slight shift in the positive direction. The shift is most evident in the residuals from reconstructions using the cluster based AAKR with points as surrounding sets. Also the residuals based on reconstructions using rectangles as surrounding sets are quite noticeable. In the lower plot, a shift in negative direction is indisputable, for all reconstructions.

5.5.4. Computation time

The computation time of producing 1000 reconstructions with 175,000 historical observations is about 22 minutes using the crude memory vector selection method. In comparison, the cluster based version, with 100 clusters, produces the 1000 reconstructions in less than 5 s. The time to perform the clustering, using *K*-means clustering, with the Lloyd algorithm, is about 95 s. However, the clustering only needs to be performed once, and does not need to be performed on-line, hence we believe the time to perform clustering should not be an issue.

5.6. Illustration of the sequence of residuals and the SPRT indices

An example of the residuals analysis using SPRT is displayed in Fig. 18. The residuals are displayed in the upper plot, while the middle and lower plots show the SPRT indices of the positive and negative change in mean respectively. If a value exceeds the upper horizontal dotted line, an alarm is raised, either for positive or negative change in mean, and the sequential test is reset. Similarly, if the value is below the lower horizontal line, the sequential test is reset. But now, confidence of normal state is reached, and no alarm is raised.

The approximated expected detection delay (EDD) and average run length (ARL) of the various reconstruction methods are reported in the figure. The EDD is the expected number of time points from an anomaly is introduced until it is detected, and ARL is the expected number of time points between false alarms.

The induced fault in the example presented in Fig. 18 is a temperature change of +1 °C in the first anomalous time interval and −1 °C in second anomalous time interval. Furthermore, the kernel bandwidth, h, is 0.1, the mean value of the two alternative hypothesis, for positive and negative change in mean, M, is set to 1, and the standard deviation, σ, is extracted from the training data. The distance scaling factor $\mathbf{s}$ is fixed at [1,1,1,1,0.1]. Note that if the last entry is 1, the original AAKR reconstruction will be performed, while if the last entry is 0, a standard Nadaraya–Watson regression will be used. See Figs. 19 and 21 for results with other choices of $\mathbf{s}$.

For positive change in mean, an EDD of 29 is returned when points are used as surrounding sets, while it is 100 when large rectangles are used. Otherwise no alarms for positive change in mean are raised in this example. Neither, no false alarms are raised. For negative change in mean, more alarms are raised. We observer that the lowest EDD is achieved by the use of points as surrounding sets, but this also provides a low ARL of 12. We note that the results are well aligned with Fig. 17.

5.7. Results using multiple surrounding sets, distance scaling vectors and credibility factors

Results of the proposed anomaly detection framework are presented in Figs. 19–21. Multiple surrounding sets are used for the cluster based AAKR reconstruction, and this is combined with multiple distance scaling vectors and credibility factors. All entries in

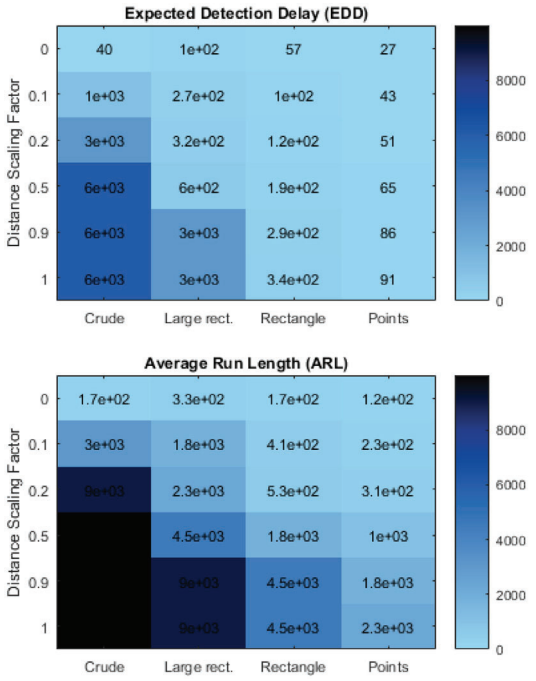


Fig. 19. Surrounding set and distance scaling vector: EDD and ARL at various surrounding sets and distance scaling vectors. If no alarms are raised, the EDD and ARL cannot be calculated. These are represented with black colour. The bandwidth of the credibility estimation is set to infinity, which means that all areas are considered equally credible.

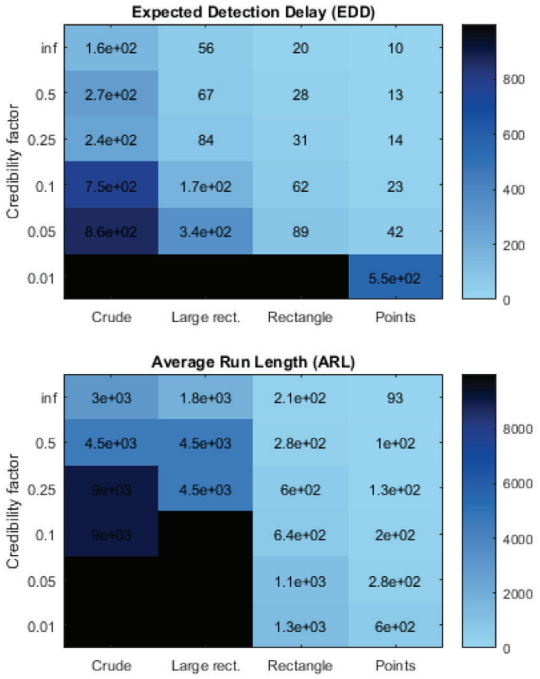


Fig. 20. Surrounding set and credibility factors: EDD and ARL at various surrounding sets and various credibility estimate factors. If no alarms are raised, the EDD and ARL cannot be calculated. These are represented with black colour. The distance scaling vector, $\mathbf{s}_J$, is 0.1.

the distance scaling vector can be adjusted, but here we concentrate on the Jth component. The values in the tables represent approximations of the mean EDD and mean ARL, taken over the whole test period of 15 folds, with 1000 points in each. The presented results are well aligned with our expectations, and show consistent behaviour.

In Fig. 19, the anomaly detection capability of the methodology using the crude and the cluster based AAKR with different surrounding sets for reconstruction, combined with residuals analysis using a range of different distance scaling factors, are presented. We observe that the lowest EDD is achieved by combining points (infinitely small rectangles) as surrounding sets with distance scaling vector 0. Furthermore, the EDD increases when the distance scaling vector is increased. Also, the EDD seems to increase when the size of the surrounding sets is increased. As expected, the ARL follows the same pattern. This illustrate the usual trade-off between EDD and ARL; we want low EDD, but this will of course cause a decrease in the ARL.

Fig. 20 illustrates how changes in credibility factor effects the EDD and ARL. Again, we apply reconstructions produced both with the crude and cluster based AAKR. Here, we fix the distance scaling vector $\mathbf{s}$ at $[1, 1, \ldots, 1, 0.1]$, and concentrate on the change in credibility factor. We observe, as expected, that both the EDD and the ARL decreases with when the credibility factor increases.

In Fig. 21, EDD and ARL based on various combinations of distance scaling vectors and credibility factor are presented. We chose to use the reconstruction version with large rectangles as surrounding set.

5.8. Discussion and suggestions for further research

In the following, we discuss some key challenges and suggestions for anomaly detection, with emphasis on the maritime industry.

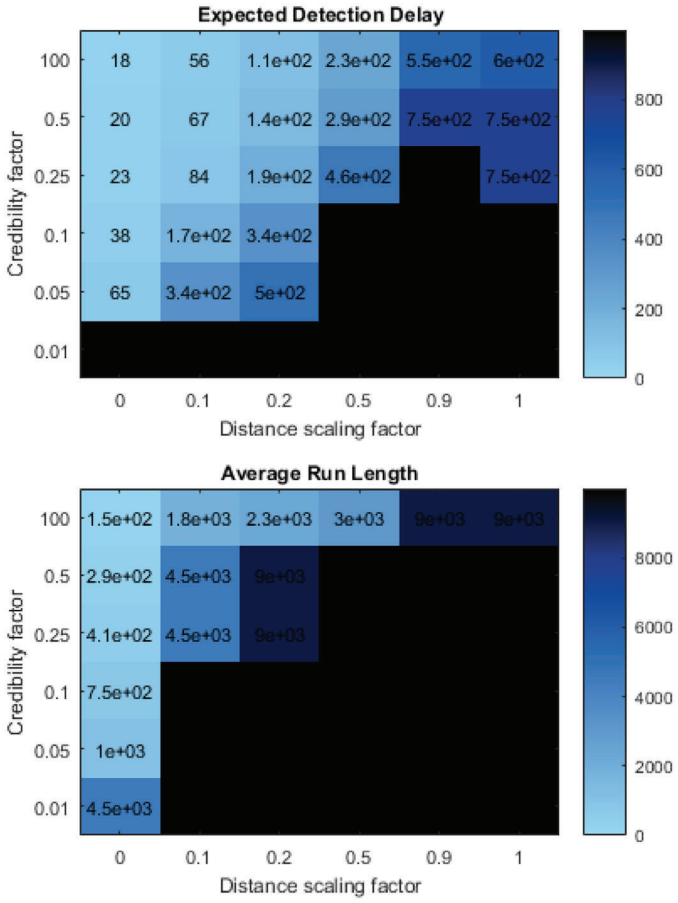


Fig. 21. Distance scaling vectors and credibility factors: EDD and ARL at various distance scaling vectors, and credibility factors. The figure is based on reconstructions produced using cluster based AAKR, with large rectangles.

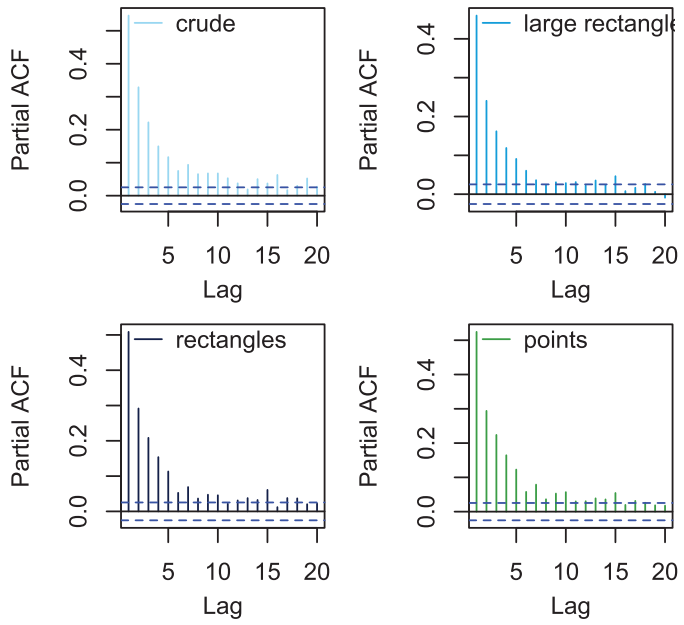


Fig. 22. Partial auto correlation function of the residuals in the bearing temperature sensor.

5.8.1. Extensions to high-dimensional sensor data

In this paper we apply the anomaly detection framework on data sets containing a very limited amount of sensor signals and performed the reconstruction of the measured signals based on distances from the training data in low-dimensional space. However, sensor monitoring of typical ship systems will often consist of hundreds of sensors and it remains to be seen how well the proposed approach scales in higher dimensions. The method will suffer from the curse of dimensionality (Keogh & Mueen, 2011), which will make it more challenging to establish similar models for high-dimensional data. Sensible techniques for dimension reduction will have to be carried out before the signals are analysed with AAKR. Additionally, feature extraction should be investigated further. We believe this is an interesting and important topic for further research.

5.8.2. Operational mode selection

During the different operating modes the behaviour of a ship changes substantially, and it might therefore be advantageous to develop reconstruction models dedicated to the different operational modes. This could also allow the alarm limits to vary in the different modes, depending on the operations criticality. To achieve this, the training data should be divided and used to fit different models. This will result in reduced computational efforts and increased model reconstruction accuracy (Al-Dahidi, Baraldi, Di Maio, & Zio, 2014; Baraldi et al., 2012).

5.8.3. Partial auto correlation in the residuals

The partial auto correlation function of the residuals, made with crude AAKR and cluster based AAKR, with large rectangles, rectangles and points as surrounding sets are shown in Fig. 22. The figure reveals that some time dependence is present in the residuals, for time lags below 5–10 s. We also observe that the dependency structure is similar in the four cases.

5.8.4. Training data extension

Sometimes training data are not available. For instance when a ship is entering a type of operation that has not been tested before, or if a ship is moved to a new geographical area, where it has never operated before, the training data might need to be modified to represent the "new" normal conditions. If the sensors are affected in a deterministic way, new training data can be simulated, based on the other training data. Ships are usually built in sister series. The sensor data collected by the first ship in a series, can possibly be reused by a later ship in the series. Also when the ships are not identical, it is possible that the training data from the first ship can be used on the later one, after necessary calibrations and modifications detailed by simulation software such as for example Dimopoulos, Georgopoulou, Stefanatos, Zymaris, and Kakalis (2014).

6. Conclusion

The paper introduces three generalizations and modifications of an on-line anomaly detection framework consisting of signal reconstruction with Auto Associative Kernel Regression (AAKR) and residuals analysis using Sequential Probability Ratio Test (SPRT).

We demonstrate the ability of the cluster based memory vector selection method for AAKR, which is successfully used for faster signal reconstruction. The methodology is applied to multiple imbalanced benchmarking data sets, in addition to the data set with sensor signals from a marine diesel engine in operation. Many of the anomalies are quite subtle, restrained enough not to easily be revealed by for example analysing scatter plots of the data. Results of the crude and the cluster based methods are presented and compared, and the analysis show that comparable results are achieved, even when very few (<25) clusters are used. The advantage of the cluster based methods is the increased speed. The computation time of the AAKR grows rapidly when the size of the training data increases, and we demonstrate how the presented cluster based memory vector selection technique can be used to dramatically decrease the computation time, at the same time as the performance is kept at an acceptable level.

We also show how the cluster based AAKR can be used in combination with the SPRT, which is used for residuals analysis, to construct a robust and fast anomaly detection framework. The results are well aligned with our expectations, and show consistent behaviour. A generalization of the distance measure used in the signal reconstruction process is proposed, which enables the users system-knowledge to be imposed on the anomaly detection framework to distinguish response and explanatory variables and optimize the weighting of the different features. The distance scaling vector can be chosen to achieve acceptable levels of expected detection delay (EDD) and average run length (ARL).

We also introduce a credibility estimate which enables the SPRT method to reach a conclusion faster when it operates in regions close to instances which are well represented in the training data set, and allows it to use more time to reach a conclusion when it operates in less explored regions.

CRediT authorship contribution statement

Andreas Brandsæter: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Project Administration. **Erik Vanem:** Validation, Writing - Review & Editing, Supervision, Project Administration, Funding Acquisition. **Ingrid K. Glad:** Validation, Writing - Review & Editing, Supervision, Project Administration, Funding Acquisition.

Acknowledgements

The work is carried out in collaboration with the Big Insight project, and is partly funded by the Research Council of Norway, project number 237718 and 251396.

Appendix A

Abstracts of the original classification data sets is provided below, together with a description of how anomalies are defined for each of the data sets. The descriptions are collected here: Alcalá-Fdez et al., 2011 and Dua and Efi (2017).

Data set	Abstract	Description of anomaly
vehicle0	3D objects within a 2D image by application of an ensemble of shape feature extractors to the 2D silhouettes of the objects	Positive examples belong to class 0 (Van) and the negative examples belong to the rest.
yeast6	Predicting the Cellular Localization Sites of Proteins	Positive examples belong to class EXC and the negative examples belong to the rest.
ecoli-0-1-3-7_vs_2-6	This data contains protein localization sites	Positive examples belong to classes pp and imL and the negative examples belong to classes cp, im, imU and imS.
glass5	From USA Forensic Science Service; 6 types of glass; defined in terms of their oxide content (i.e. Na, Fe, K, etc.)	Positive examples belong to class 5 and the negative examples belong to the rest.
shuttle-c0-vs-c4	The shuttle data set contains 9 attributes all of which are numerical. Approximately 80% of the data belongs to class 1	Positive examples belong to class 0 and the negative examples belong to class 4.
dermatology-6	Aim for this data set is to determine the type of Eryhemato-Squamous Disease.	Positive examples belong to the class 6 and the negative examples to the rest of the classes.
shuttle-6_vs_2-3	The shuttle data set contains 9 attributes all of which are numerical. Approximately 80% of the data belongs to class 1. The task is to decide what type of control of the vessel should be employed.	Positive examples belong to the class 6 and the negative examples belong to the classes 2–3.
winequality-red-4	The data set is related to red variant of the Portuguese Vinho Verde wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).	Positive examples belong to the class 4 and the negative examples belong to the rest of classes.
poker-9_vs_7	Each record of this data set is an example of a hand consisting of five playing cards drawn from a standard deck of 52. Each card is described using two attributes (suit and rank), for a total of 10 nominal attributes. The class attribute describes the Poker Hand obtained	Positive examples belong to the class 9 and the negative examples belong to the class 7.
yeast1	Predicting the Cellular Localization Sites of Proteins	Positive examples belong to class NUC and the negative examples belong to the rest.
segment0	This data set is an image segmentation database similar to a database already present in the repository (Image segmentation database) but in a slightly different form.	Positive examples belong to class 1 and the negative examples belong to the rest.
vehicle2	3D objects within a 2D image by application of an ensemble of shape feature extractors to the 2D silhouettes of the objects	Positive examples belong to class 2 (Bus) and the negative examples belong to the rest.
vehicle3	3D objects within a 2D image by application of an ensemble of shape feature extractors to the 2D silhouettes of the objects	Positive examples belong to class 3 (Opel) and the negative examples belong to the rest.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.eswa.2018.12.040.

References

Ahn, H.-K., Bae, S. W., Demaine, E. D., Demaine, M. L., Kim, S.-S., Korman, M., et al. (2011). Covering points by disjoint boxes with outliers. *Computational Geometry, 44*(3), 178–190.

Al-Dahidi, S., Baraldi, P., Di Maio, F., & Zio, E. (2014). Quantification of signal reconstruction uncertainty in fault detection systems. *The second European conference of the prognostics and health management society*.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., et al. (2011). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing, 17*, 255–287.

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys, 4*, 40–79. doi:10.1214/09-SS054.

Baraldi, P., Canesi, R., Zio, E., Seraoui, R., & Chevalier, R. (2011). Genetic algorithm-based wrapper approach for grouping condition monitoring signals of nuclear power plant components. *Integrated Computer-Aided Engineering, 18*(3), 221–234.

Baraldi, P., Di Maio, F., Genini, D., & Zio, E. (2015). Comparison of data-driven reconstruction methods for fault detection. *IEEE Transactions on Reliability, 64*(3), 852–860. doi:10.1109/TR.2015.2436384.

Baraldi, P., Di Maio, F., Pappaglione, L., Zio, E., & Seraoui, R. (2012). Condition monitoring of electrical power plant components during operational transients.. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, SAGE, 226*, 568–583.

Baraldi, P., Di Maio, F., Turati, P., & Zio, E. (2015). Robust signal reconstruction for condition monitoring of industrial components via a modified auto associative kernel regression method. *Mechanical Systems and Signal Processing, 60–61*, 29–44. doi:10.1016/j.ymssp.2014.09.013.

Boechat, A. A., Moreno, U. F., & Haramura, D. (2012). On-line calibration monitoring system based on data-driven model for oil well sensors. *IFAC Proceedings Volumes, 45*(8), 269–274.

Brandsæter, A., Manno, G., Vanem, E., & Glad, I. K. (2016). An application of sensor-based anomaly detection in the maritime industry. In *2016 IEEE international conference on prognostics and health management (ICPHM)* (pp. 1–8). doi:10.1109/ICPHM.2016.7811910.

Brandsæter, A., Vanem, E., & Glad, I. K. (2017). Cluster based anomaly detection with applications in the maritime industry. *2017 international conference on sensing, diagnostics, prognostics, and control, Shanghai, China*.

Cameron, S. (1997). EnhancinG GJK: Computing minimum and penetration distances between convex polyhedra. In *Robotics and automation, 1997. proceedings., 1997 IEEE international conference on: 4* (pp. 3112–3117). IEEE.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR), 41*(3), 15.

Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A., & Charrad, M. M. (2014). Package 'nbclust'. *Journal of Statistical Software, 61*, 1–36.

Cheng, S., & Pecht, M. (2012). Using cross-validation for model parameter selection of sequential probability ratio test. *Expert Systems with Applications, 39*(9), 8467–8473. doi:10.1016/j.eswa.2012.01.172.

Coble, J., Humberstone, M., & Hines, J. W. (2010). Adaptive monitoring, fault detection and diagnostics, and prognostics system for the iris nuclear plant. In *Annual Conference of the Prognostics and Health Management Society*.

Cord, M., & Cunningham, P. (2008). *Machine learning techniques for multimedia: Case studies on organization and retrieval*. Springer Science & Business Media.

Dattorro, J. (2010). *Convex optimization & Euclidean distance geometry*. USA: Meboo Publishing.

Di Maio, F., Baraldi, P., Zio, E., & Seraoui, R. (2013). Fault detection in nuclear power plants components by a combination of statistical methods. *IEEE Transactions on Reliability, 62*(4), 833–845. doi:10.1109/TR.2013.2285033.

Dimopoulos, G. G., Georgopoulou, C. A., Stefanatos, I. C., Zymaris, A. S., & Kakalis, N. M. (2014). A general-purpose process modelling framework for marine energy systems. *Energy Conversion and Management, 86*, 325–339.

Dua, D., & Efi, K.T., (2017). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise.. In *KDD: 96* (pp. 226–231).

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874.

Flaherty, N. (2017). Frames of mind. *Unmanned Systems Technology, 3*(3).

Friedman, J., Hastie, T., & Tibshirani, R. (2009). The elements of statistical learning. *Springer series in statistics*: 1 (2). New York, NY, USA: Springer-Verlag.

Garvey, J., Garvey, D., Seibert, R., & Hines, J. W. (2007). Validation of on-line monitoring techniques to nuclear plant data. *Nuclear Engineering and Technology, 39*, 133–142.

Gross, K. C., & Lu, W. (2002). Early detection of signal and process anomalies in enterprise computing systems. . In M. A. Wani, H. R. Arabnia, K. J. Cios, K. Hafeez, & G. Kendall (Eds.), *ICMLA* (pp. 204–210). CSREA Press.

Guha, S., & Mishra, N. (2016). Clustering data streams. In M. Garofalakis, J. Gehrke, & R. Rastogi (Eds.), *Data stream management: Processing high-speed data streams* (pp. 169–187)). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-28608-0_8.

Hines, J. W., & Garvey, D. R. (2006). Development and application of fault detectability performance metrics for instrument calibration verification and anomaly detection. *Journal of Pattern Recognition Research*.

Hines, J. W., Garvey, D. R., & Seibert, R. (2008). Technical review of on-line monitoring techniques for performance assessment (NUREG/CR-6895). Volume 3: Limiting case studies. *Technical Report*. United States Nuclear Regulatory Commission, Office of Nuclear regulatory Research.

Hines, J. W., Garvey, D. R., Seibert, R., & Usynin, A. (2008). Technical review of on-line monitoring techniques for performance assessment (NUREG/CR-6895). Volume 2: Theoretical issues. *Technical Report*. United States Nuclear Regulatory Commission, Office of Nuclear regulatory Research.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review, 22*(2), 85–126.

Jarvis, R. A. (1973). On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters, 2*(1), 18–21.

Jiang, G., & Wang, W. (2017). Error estimation based on variance analysis of k-fold cross-validation. *Pattern Recognition, 69*, 94–106.

Kanarachos, S., Christopoulos, S.-R. G., Chroneos, A., & Fitzpatrick, M. E. (2017). Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and hilbert transform. *Expert Systems with Applications, 85*(Supplement C), 292–304. doi:10.1016/j.eswa.2017.04.028.

Keogh, E., & Mueen, A. (2011). Curse of dimensionality. In *Encyclopedia of machine learning* (pp. 257–258). Springer.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on artificial intelligence*. In *IJCAI'95: 2* (pp. 1137–1143). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory, 28*(2), 129–137.

Michau, G., Palme, T., & Fink, O. (2017). Deep feature learning network for fault detection and isolation. In *Proceedings of the annual conference of the prognostics and health management society* (pp. 108–118). Citeseer.

Ng, R. T., & Han, J. (1994). Efficient and effective clustering methods for spatial data mining. In *Proceedings of VLDB* (pp. 144–155). Citeseer.

Ng, R. T., & Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering, 14*(5), 1003–1016.

Olson, C., Judd, K., & Nichols, J. (2018). Manifold learning techniques for unsupervised anomaly detection. *Expert Systems with Applications, 91*(Supplement C), 374–385. doi:10.1016/j.eswa.2017.08.005.

Park, S. H., & Kim, J.-Y.. Unsupervised clustering with axis-aligned rectangular regions. http://cs229.stanford.edu/proj2009/ParkKim.pdf.

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical ComputingVienna, Austria.

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2010). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(3), 569–575.

Saranya, C., & Manikandan, G. (2013). A study on normalization techniques for privacy preserving data mining. *International Journal of Engineering and Technology, 5*, 2701–2704.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., et al. (2008). Metrics for evaluating performance of prognostic techniques. In *2008 international conference on prognostics and health management* (pp. 1–17). doi:10.1109/PHM.2008.4711436.

Wilks, D. (2011). Chapter 15 – Cluster analysis. In D. S. Wilks (Ed.), *Statistical methods in the atmospheric sciences*. In *International Geophysics: 100* (pp. 603–616). Academic Press. http://www.sciencedirect.com/science/article/pii/B9780123850225000154. doi:10.1016/B978-0-12-385022-5.00015-4.

Zheng, D., Li, F., & Zhao, T. (2016). Self-adaptive statistical process control for anomaly detection in time series. *Expert Systems with Applications, 57*(Supplement C), 324–336. doi:10.1016/j.eswa.2016.03.029.

Paper III

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

Vanem, E., Brandsæter, A.

Journal of Marine Engineering & Technology (2019)

III

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 13, 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.

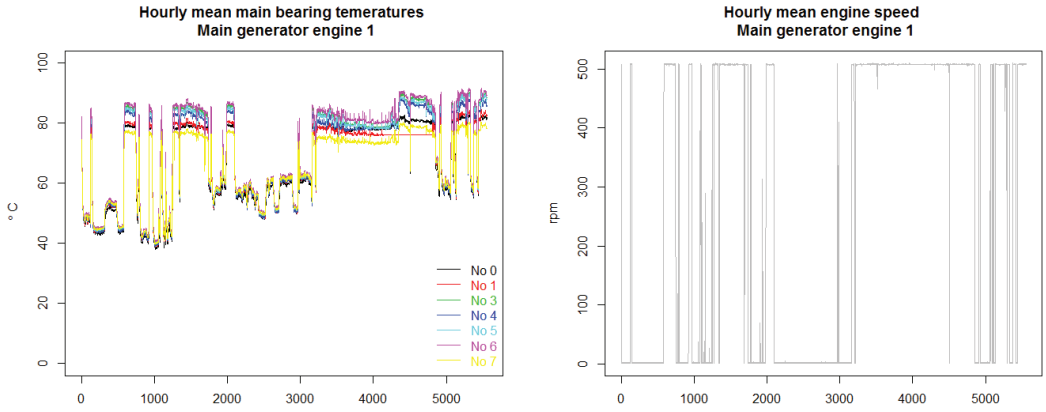


Figure 1. Traceplots illustrating strong correlation between some of the signals (engine 1); the various temperature readings for main bearing temperatures (left) and the engine speed (right); hourly averaged data

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.

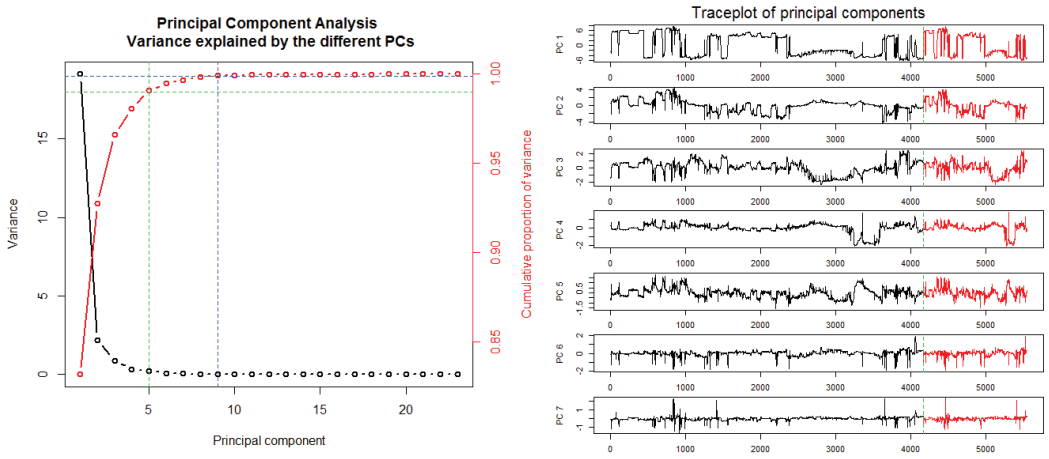


Figure 2. Variance explained by the principal components in the training data (left) and 7 first principal components (right)

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.

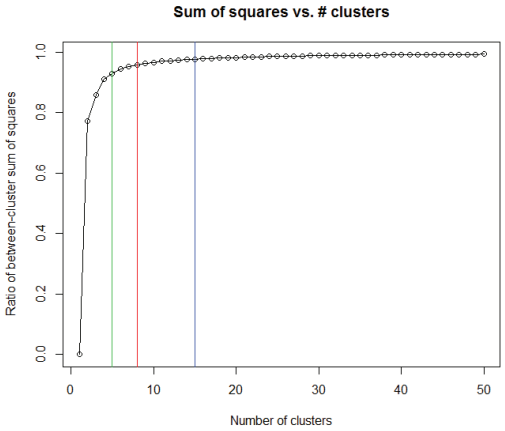


Figure 3. Estimating the number of clusters in the data, K

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

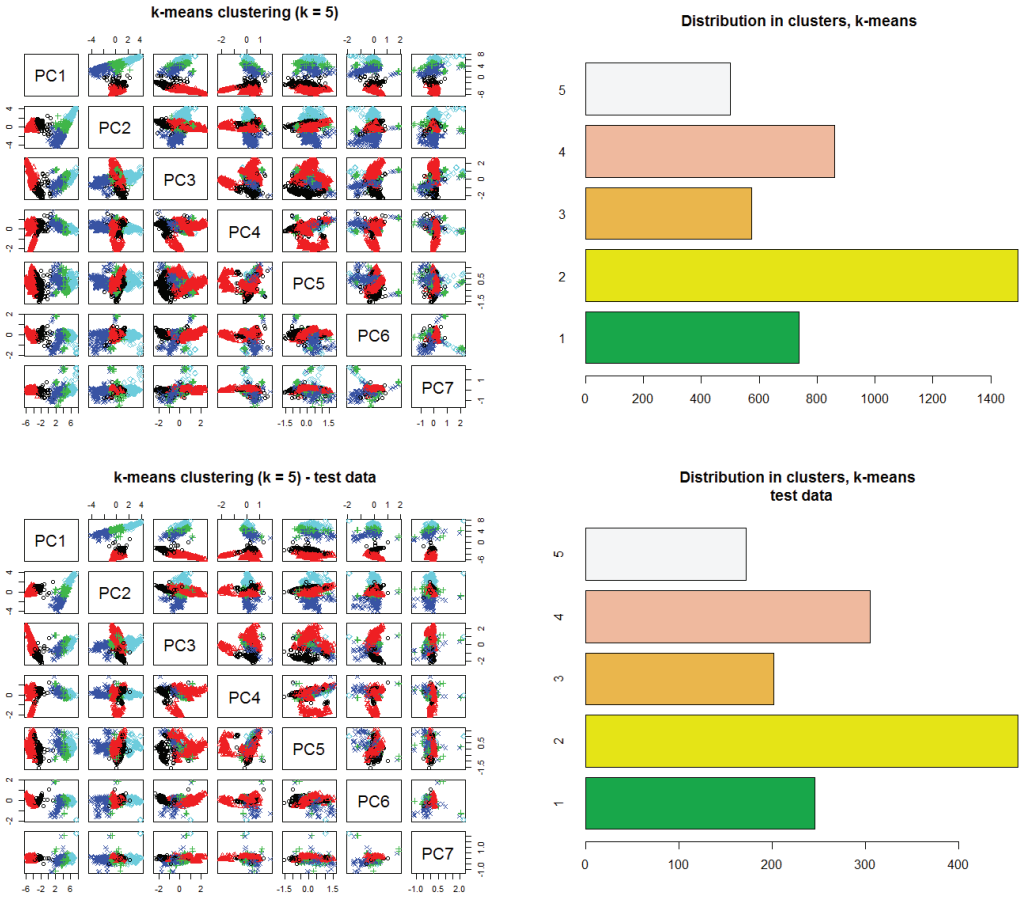


Figure 4. Data clustered in $K = 5$ clusters with K-means (left) and distribution of observations within the clusters (right); Clustering on training data (top) and applied to the test data (bottom)

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

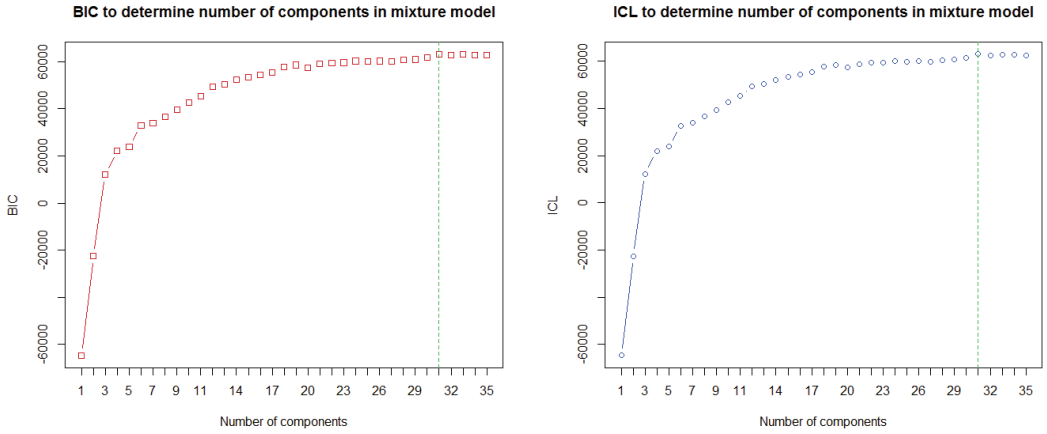


Figure 5. Estimating the number of components of the mixture model by BIC (left) and ICL (right)

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.

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

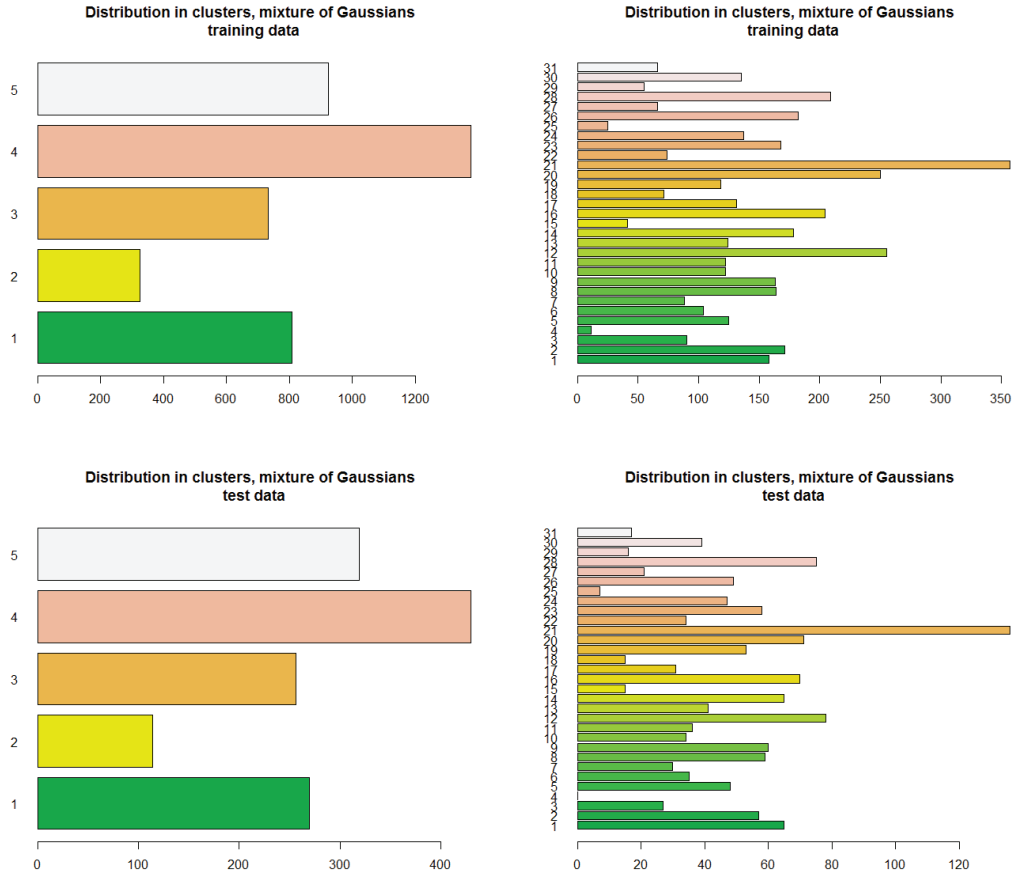


Figure 6. Distribution of observations assigned to each cluster by the mixture of Gaussian models with $K = 5$ (left) and $K = 31$ (right); training data (top) and test data (bottom)

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

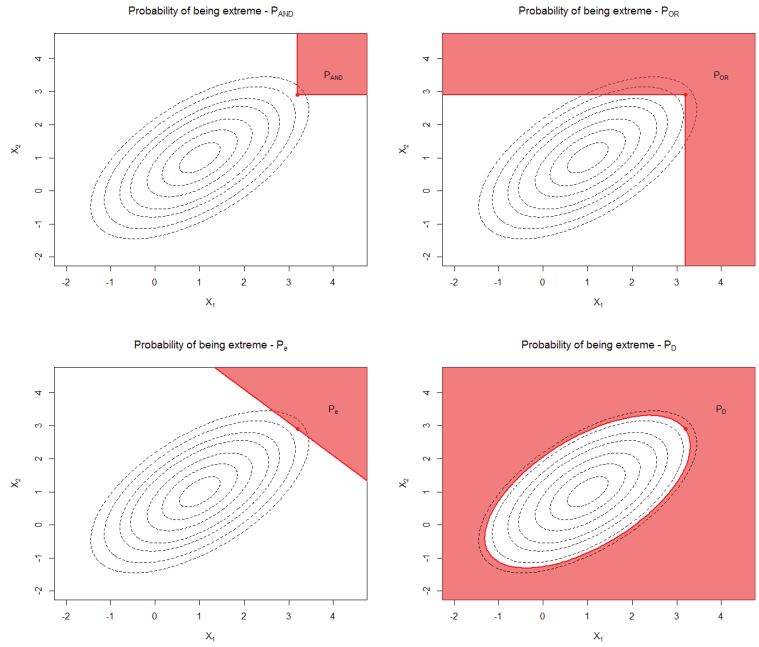


Figure 7. Different definitions of characterizing extremeness in the bivariate case

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.

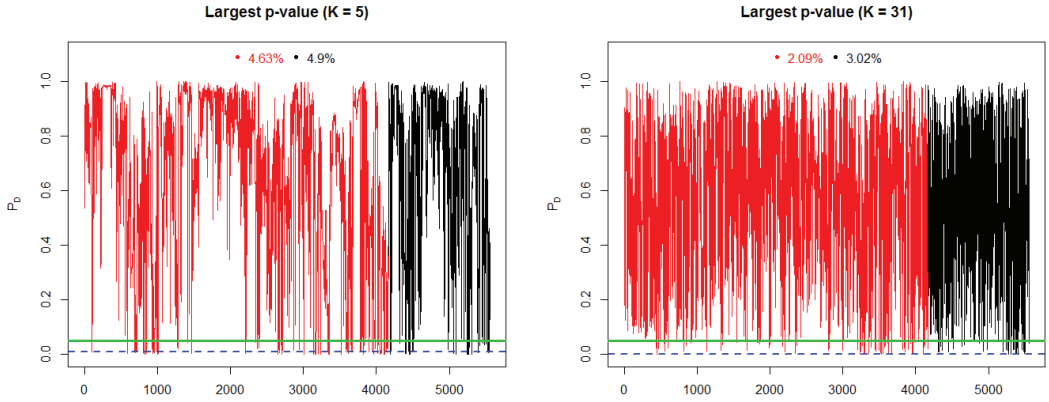


Figure 8. Maximal p-values for anomaly detection; mixture model with $K = 5$ (left) and $K = 31$ (right)

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.

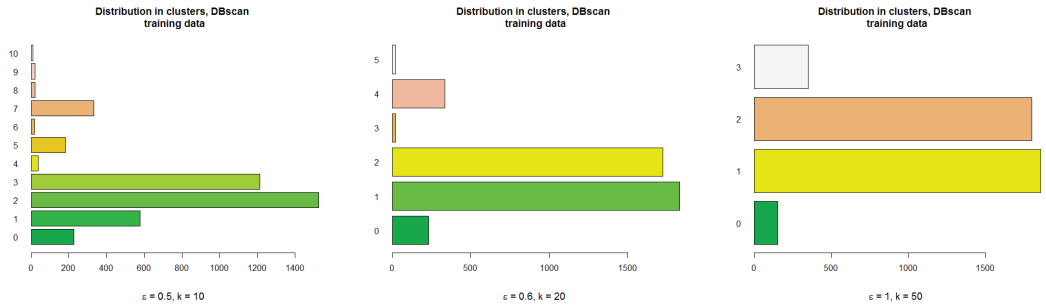


Figure 9. Distribution of data points in each clusters for the training data for different values of ε and k

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.

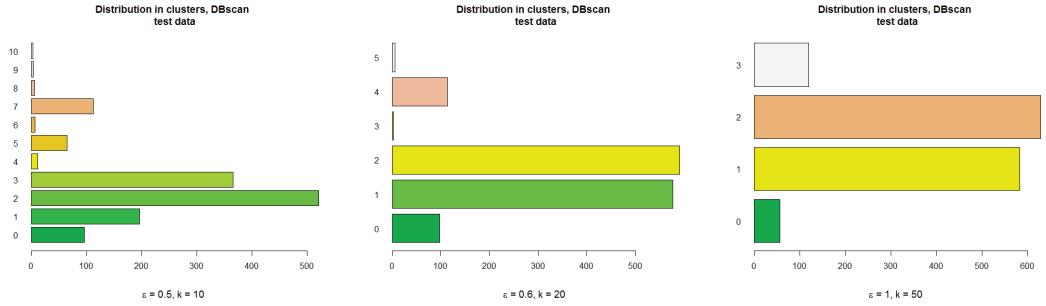


Figure 10. Distribution of number of points in each clusters for the test data for different values of ε and k

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

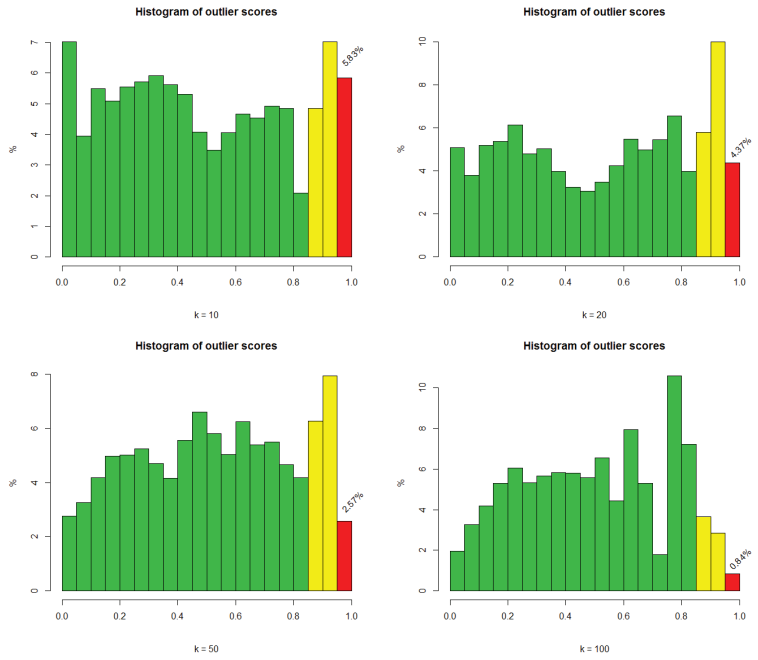


Figure 11. Histograms of outlier scores as calculated from HDBscan clustering for different values of k

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.

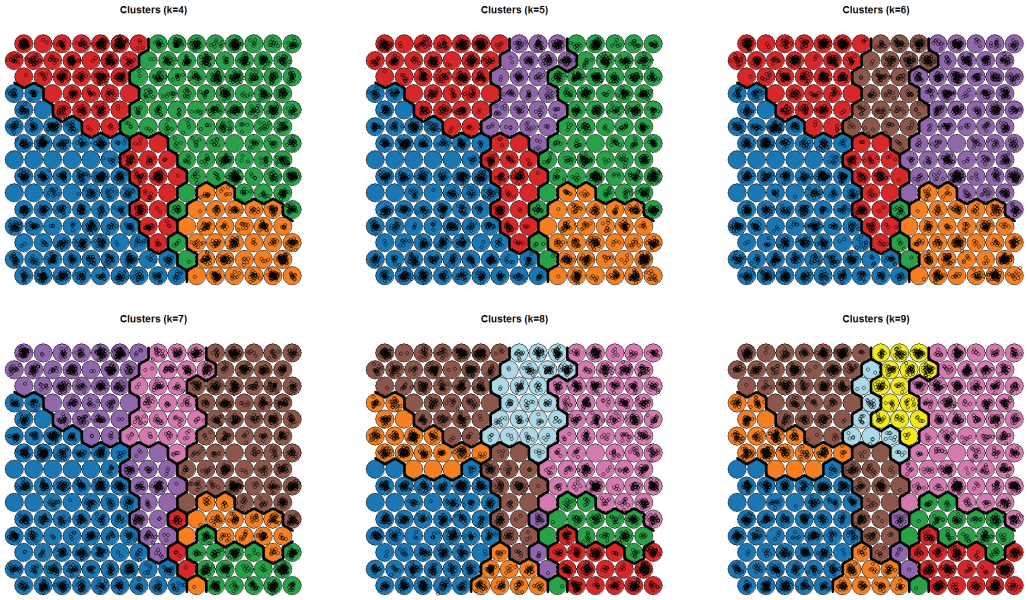


Figure 12. Clustering of the self organizing maps for different number of clusters; hierarchical clustering

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.

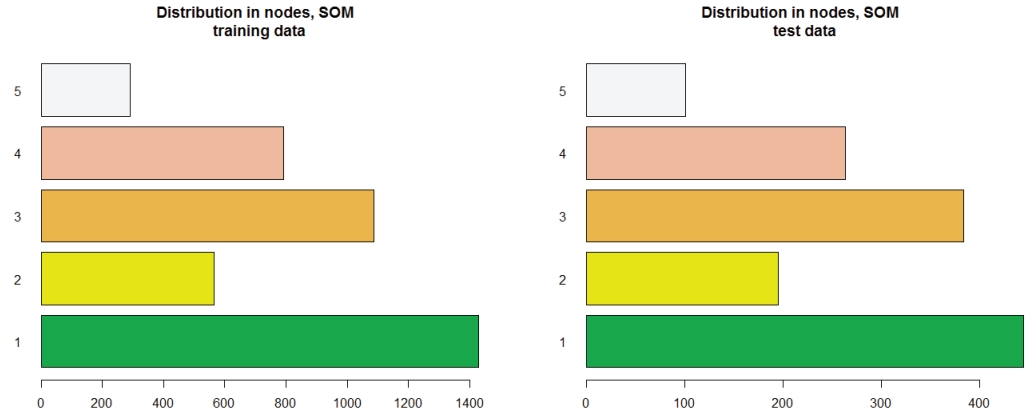


Figure 13. Distribution of observations within each cluster for the training (left) and test (right) data with clustering performed by self organizing maps; 5 clusters based on map with 15×15 nodes

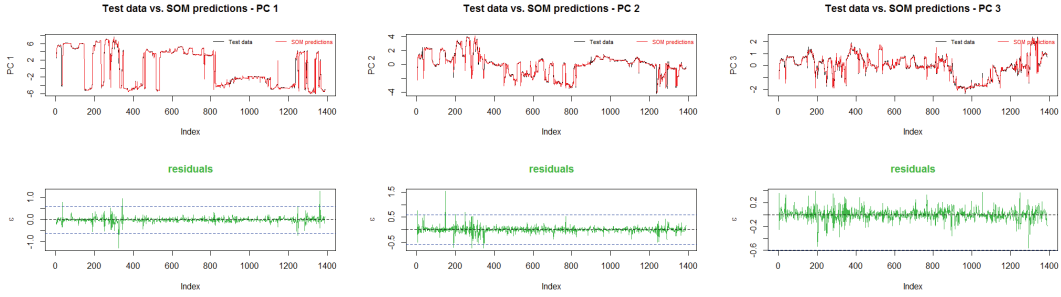


Figure 14. Anomaly detection can be performed by studying the residuals between the observed signals and the ones predicted by the trained map; Traceplots of test data and SOM prdictions (top row) and traceplots of residuals (bottom row)

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	
Training data	190	123	95	48	9	6	1	0	472
Test data	80	46	35	23	6	1	1	0	192

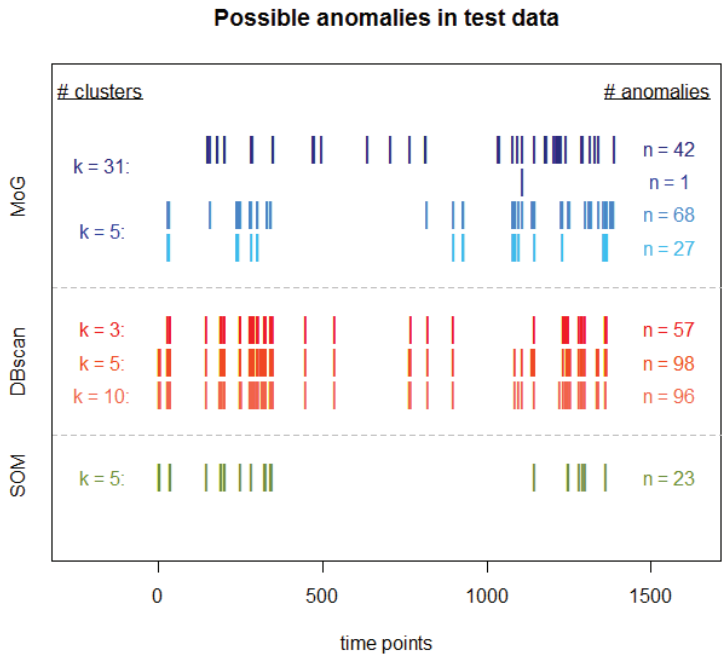


Figure 15. Comparing the possible anomalies detected by the different clustering methods. Vertical lines corresponds to time points where the different methods would flag an alarm.

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) or DLM (Vanem and Storvik 2017) 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.

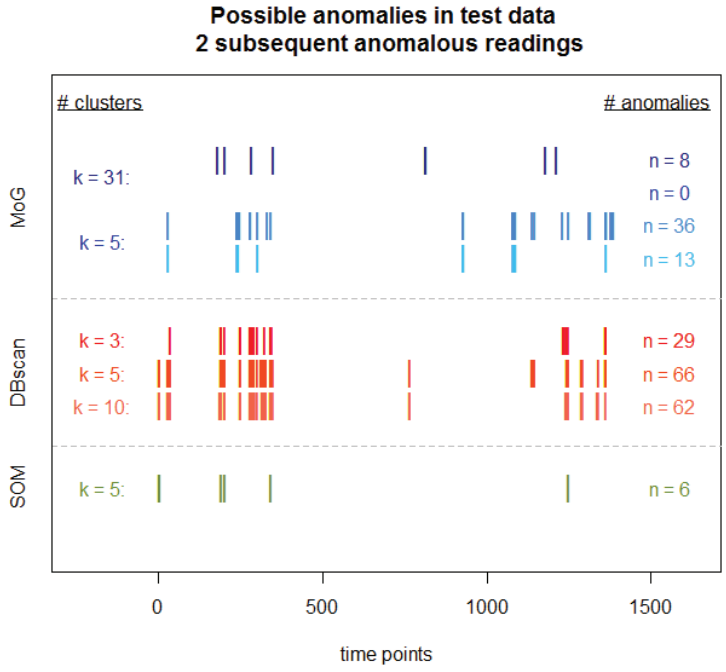


Figure 16. Comparing the possible anomalies detected by the different clustering methods when requiring 2 subsequent anomalous readings to flag an anomaly. Vertical lines corresponds to time points where the different methods would flag an alarm.

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.
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.
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.

Paper IV

Towards a framework for assurance of autonomous navigation systems in the maritime industry

Brandsæter, A., Knutsen, K. E.

In *Safety and Reliability–Safe Societies in a Changing World : Proceedings of ESREL 2018*, (pp. 449–457). CRC Press.

IV

Towards a framework for assurance of autonomous navigation systems in the maritime industry

A. Brandsæter
DNV GL, Høvik, Norway
University of Oslo, Oslo, Norway

K. E. Knutsen
DNV GL, Høvik, Norway

ABSTRACT: We discuss potential assurance frameworks for autonomous navigation systems in the maritime industry, with emphasis on testing and verification of the system's perception performance and capacities. Ongoing research in this field has revealed profound challenges related to artificial situation awareness and machine perception specific to the marine environment. The lack of a clear and transparent framework and methodologies to assure the safety associated with the usage of such solutions, have been identified as key barriers for the implementation of autonomous navigation solutions at scale. Because the machine perception and situational awareness algorithms are expected to be partly or fully based on machine learning algorithms, including deep learning, whose functional reasoning is challenging or even impossible to understand and predict, the verification of such systems is fundamentally different from a traditional verification process based on physical understanding and theory. We review several methods for testing autonomous navigation systems, proposed and used mainly in the automotive industry, and discuss how these methods can be adapted, combined and applied to form a framework for assurance of autonomy in the maritime industry.

1 INTRODUCTION

Autonomous transport on land, in the air and at sea has been coined the technology trend with the highest potential to disrupt the transport sector in the future. It has the potential for making transport solutions more cost effective, safe and environmentally friendly, but also to disrupt entire business models and value chains associated with the mode of transport. Given the disruptive potential of this technology trend, increasing research efforts are being invested to realize the technological solutions.

1.1 *The autonomy revolution*

Technologies and methods for autonomous systems is a very active area of research both in the industry and in academia. However, the majority of the research being done for autonomous vehicle navigation is focused around the automotive industry. The amount of test data for such vehicles is becoming abundant and is considered an important contributor to the current state of the art in the research field. Major advances in object detection, classification and image analysis have been made in recent years, with extensive use of artificial intelligence related technologies such as feature extraction, artificial neural networks, deep learning models such as convolutional neural networks (CNNs), gradient-based and derivative-based matching approaches (see for example (Hofmann 2013, Rout 2013, MathWorks 2017c)). Research is needed to identify if and how the algorithms, methods and sensors, developed for the automotive industry, can be utilized in the maritime domain.

1.2 *Opportunities in the maritime industry*

Several studies have shown that human error contributes to a majority of marine casualties (Rothblum 2000, Harrald et al. 1998). However, automated systems and autonomy can also introduce new challenges, and existing challenges might be amplified (Lützhöft & Dekker 2002). Nevertheless, we expect that if the interaction between the humans and machines are treated carefully, with thorough testing and verification, autonomy can contribute significantly to increase safety in many maritime operations.

Unmanned ships will enable optimization of energy efficiency due to changes in design constraints and freeing of space, previously used to accommo-

date crew. In addition, more hydrodynamic and aerodynamic designs may in turn lead to less fuel consumption and reduced emissions. Furthermore, autonomous ships might be able to compete with road transportation and contribute to reduced emission from road transportation as well as reduced road wear and tear.

If autonomous ships are successfully implemented, it will most probably enable fundamentally new types of ship transportation operations, such as for example single container shipment (Woodgate 2017); extremely slow speed transportation with very low emissions (Tvete 2017); container feeder to replace road transport (Kongsberg 2017); and unmanned patrol ships (Fingas 2017).

Several demonstrators have already proven that it is feasible for a transport solution to be operated by sensors and software either partially or fully based on deep learning algorithms (Huval et al. 2015, Ackerman 2017). Among others, a company Drive.ai, has an ambition to use deep learning fully from sensory input to decision making, while others usually use deep learning in parts of the system, e.g. situational awareness, while relying on traditional control system logic in other parts of the system (Huval et al. 2015, Ackerman 2017, Muoio 2016). Nevertheless, the solutions are yet to be deployed at scale. One of the reasons for the lack of deployments is that the solutions are still not proven to be sufficiently safe.

1.3 *Early rule development as an enabler for innovation*

A key element required to keep the autonomous system safe, is the ability of the system to achieve situational awareness. Situational awareness algorithms are usually partly or fully based on machine learning algorithms whose functional reasoning are challenging or even impossible to understand and predict. Hence, the verification of such a system is fundamentally different from a traditional verification process based on physical understanding and theory. The machine learning algorithms are data driven, and completely dependent on the quality of the training data. Therefore, verification will likely be carried out by a combination of testing, simulations and benchmarking against real and synthetic data sets. Furthermore, adaptive methods, where data are automatically collected and used to retrain the system, will also be considered.

For a manned system, awareness is achieved by the human operator by using his or her senses and perceptive abilities to interpret instrument signals and input from surroundings. An unmanned ship should use a priori information, such as maps, combined with sensor readings to make observations relative to the environment, and use software to perceive the situation based on this input. This digital perception will be used as input to a decision-making algorithm. In turn, this controls the actuators of the vessel which are effectuating the decision made. For the autonomous system to make safe decisions, the situational awareness must be sufficiently accurate for all feasible situations and conditions which the vessel may encounter.

System functional and performance requirements necessary to obtain a required safety level of an automated situational awareness system should be established as early as possible, as this will offer the technology providers a standard to be met by their solutions. If requirements are not set before or early in the technology development phase, developers risk spending significant efforts and money on developing solutions which in the end may not meet the safety standard. However, establishing such a standard is difficult when no solutions exist to evaluate the standard against.

In addition to a standard for required system and component performance, tools are needed for verifying that the technology meets the requirements set in the standard. For a situational awareness system, this will include verifying that the sensors adequately detect objects affecting the safety of the vessel and its surroundings under various conditions, and that the perception algorithm can use this information together with other a priori information to adequately understand the situation.

1.4 *Focus of this study*

In this paper, we discuss rules and regulations related to autonomous navigation systems in a maritime context, with focus on autonomous perception and situational awareness. However, we believe that a framework for approval developed for autonomous applications will also be applicable to other systems that are based on machine learning algorithms and artificial intelligence.

The remainder of the paper is structured as follows. In section 2, we propose and describe a range of recommended practices and tools that can be applied to test and validate the ability, performance and robustness of safety critical systems which decisions are based on data-driven methods. These practices and tools originate partly from traditional statistical analysis and are suggested and applied for testing and assurance of autonomy in the automotive industry. In section 3, we discuss challenges related to machine perception that are unique or particularly pronounced in the maritime domain, and suggest how the recommended practices and tools should be used and possibly adapted to suit the maritime domain. Furthermore, we present a possible scope for assurance framework, and discuss potential implications of autonomy such as for example operational dependent requirements. We also describe the IMO guidelines for approval of alternatives and equivalents. We conclude in section 4.

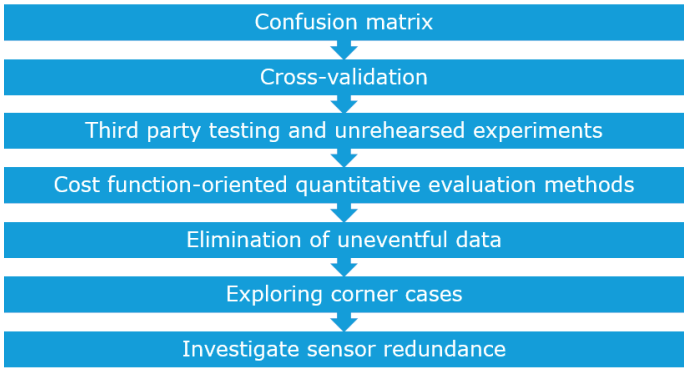


Figure 1: Proposed components in an assurance framework for safety critical systems.

2 TOOLS AND RECOMMENDED PRACTICES FOR ASSURANCE

In the following, we propose and describe a different recommended practices and tools that can all be applied to test and validate the ability, performance and robustness of safety critical systems which decisions are based on data-driven methods. See Figure 1 for an overview of the methods.

2.1 *Confusion matrix*

It is not obvious how to measure and evaluate the performance of autonomous navigation systems. If two systems provides divergent predictions or decision, it is difficult to define and quantify which reaction was most correct. In classification problems, the results are often presented in a confusion matrix, where the predicted class is compared with the actual or the true class. With two classes, for example object detection with two objects, it is straight forward to define the confusion matrix, which inhere the number of

- true positives (TP), hits
- true negatives (TN), correct rejections
- false positives (FP), false alarms, Type I errors
- false negatives (FN), misses, Type II errors

When more classes are needed, defining the criteria for performance evaluation becomes more challenging. For example, how should we quantify the performance of a perception system which correctly detects a vessel, but misclassifies it as a ferry? And how should this be compared to misclassifying it as a kayak? Or what if the system is not even able to recognize it as an object?

To be able to make the above-mentioned comparison, we are fully dependent on a correctly labelled dataset of the ground truth. An autonomous ship will likely be equipped with multiple sensors, including multiple daylight and IR cameras in various directions, radars with different settings, in addition to automatic identification system (AIS) and other satellite data, etc. The labelling process should take all these sources into account when labelling the data. For example, if an object is not visible in the video stream due to thick fog or other difficult weather conditions, but we know the objects position from AIS data or other sources, the object will be labelled in the ground truth data set. All relevant information should be correctly labelled, in all datasets.

Data collection, and especially data annotation or labelling, are surprisingly time consuming and costly tasks for vehicle classification (Schöning et al. 2015, Chen and Ellis 2014). However, various tools and methods for semi-automatic ground truth labelling on video streams designed for the automotive industry are available, such as for example (MathWorks 2017b, MathWorks 2017a, Cuevas et al. 2015, Lopez-Villa et al. 2015, Schöning et al. 2015). Another approach is to crowd source the data annotation like Mighty AI has done in automotive, where they have developed a mobile app in which users may annotate images manually and get paid for it, whereupon Mighty AI makes a business out of selling annotated datasets [https://mty.ai/]. The available solutions should be explored, and if necessary adapted for our use in a maritime context.

2.2 *Cross-validation*

It is well known that when we evaluate predictions from a statistical model on the dataset used to train the model, our accuracy estimates tend to be overoptimistic (Arlot & Celisse 2010). To build robust and accurate models we ideally want to use all data available. The same applies to testing; we want to test our models in all situations, not only on a subset. Cross-validation introduces various methods of repetitively splitting the data $\mathcal{D}$ into two exclusive parts $\mathcal{D}_t$ and $\mathcal{D}_v$; where one part $\mathcal{D}_t$ is used to train the model, and the other $\mathcal{D}_v$ is reserved for validation.

A range of different splitting techniques can be applied, providing different cross-validation estimates. See for example Arlot & Celisse 2010, Kohavi 1995 for a brief overview of the most common splitting techniques.

One of the most widely used splitting technique is called K-fold cross-validation, which in its standard form splits the original dataset $\mathcal{D}$ into K subsets (folds) $\mathcal{D}_1, \ldots, \mathcal{D}_K$, as described in (Arlot and Celisse 2010, Brandsæter and Vanem 2016). For each $k \in 1, 2, \ldots, K$ the models are trained on $\mathcal{D}_t = \mathcal{D} \setminus \mathcal{D}_k$, and tested on $\mathcal{D}_k$. To make sure that the results are not strongly dependent on how the folds are selected, we repeatedly run the K-fold cross-validation with new selections. The sets are often chosen to be mutually exclusive with equal size.

2.3 *Extensive testing*

The standard approach for assurance of autonomous navigation in the automotive industry is extensive testing (Pei et al. 2017a, Zhao and Peng 2017, Waymo 2016, Fei-Fei 2010), where large amounts of real world data from ordinary operation is gathered and

manually labelled, and data on driver performance, behaviour, environment, driving context and other factors that were associated with critical incidents, near misses and crashes are analysed and used in evaluating the system performance (Zhao and Peng 2017).

Simulated real-world data is also sometimes used to massively increase the amount of data (Madrigal 2017, Zhao and Peng 2017), but usually this is completely unguided, and due to the large input space of real-world scenarios, none of these approaches can hope to cover more than a tiny fraction (if any at all) of all possible corner cases (Pei et al. 2017a). Here, a corner case is defined as an unusual, but far from impossible, scenario. In particular, if each individual parameter, such as temperature, fog, daylight, driving speed, number of other vehicles involved, etc. are well within the normal range for that parameter, but still the combined scenario is unusual. As an example, again from the automotive industry, a Tesla in autopilot recently crashed into a trailer because the autopilot system failed to recognize the trailer as an obstacle due to its white color against a brightly lit sky and the high ride height (Lambert 2016). Such corner cases were not part of Waymos (Googles) or Teslas test set (Pei et al. 2017a) and thus never showed up during testing.

2.4 *Third party testing and unrehearsed experiments*

In 2003 the Defense Advanced Research Projects Agency (DARPA) announced the first Grand Challenge with the goal of developing vehicles capable of autonomously navigating desert trails and roads at high speeds. In Krotkov et al. 2007, the conduct of six evaluation experiments for the DARPA PerceptOR program is described. Key distinctions of the testing methodology include conduct of the experiments by an independent third party, and the use of unrehearsed experiments that provide little advance knowledge of and access to the test courses. The article also presents quantified, objective performance metrics for the systems evaluated. Furthermore, it includes blind experiments that do not allow the system operators to see the test courses until all tests are completed.

The test environment and the test content are described in detail; however, the evaluation approach are not thoroughly discussed (Sun et al. 2011).

2.5 *Cost function-oriented quantitative evaluation methods*

Wei and Dolan 2009 claims that most teams in the 2007 DARPA Urban Challenge preferred to avoid difficult manoeuvres in high-density traffic by stopping and waiting for a clear opening instead of interacting with it and operating the vehicle and human drivers. To encounter this, researchers at Beijing Institute of Technology, propose a design method for a scientific and comprehensive test and evaluation system for autonomous ground vehicles competitions, to better guide and regulate the development of autonomous ground vehicles. The evaluation method proposed by Sun et al. 2014, Sun et al. 2011 aims to evaluate the quality of completion with a cost function-oriented quantitative evaluation method. This evaluation method can presumably evaluate the overall technical performance and individual technical performance of autonomous ground vehicles. A complete test system that includes the test contents, the test environment, and the test methods to meet the demands of testing for autonomous ground vehicles is developed, and a fuzzy evaluation method is combined with an analytic hierarchy process to solve fuzzy and hard-to-quantify problems (Sun et al. 2014).

2.6 *Elimination of uneventful data*

Recently, a new approach to testing autonomous cars was proposed by researchers affiliated with the University of Michigans Mcity connected and automated vehicle center. Zhao and Peng 2017 presents an accelerated evaluation process which aims to eliminate the uneventful driving activity, and filter out only the potentially dangerous driving situations where an automated vehicle needs to respond, creating a faster, less expensive testing program. It is claimed that this approach can reduce the amount of testing needed by a factor of 300 to 100,000.

Four methodologies that form the basis of the accelerated evaluation process are listed (Zhao and Peng 2017):

1. Evaluate how frequently a significant driving event happens on the road, and stripe out the more common, uneventful safe driving situations.
2. Use importance sampling to statistically increase the number of critical driving events in a way that still accurately reflects real-world driving situations.
3. Construct a formula that accurately distils those critical events, tests the formula, and apply it to further reduce the amount of testing required.
4. Analyse interactions between human-driven vehicles and robotic vehicles and optimize the random occurrences of significant driving events in the most complex scenarios.

2.7 *Exploring corner cases in deep learning systems*

In Pei et al. 2017a, Tian et al. 2017, Pei et al. 2017b prepared by researchers at Colombia University, Lehigh University and University of Virginia, a method for automated whitebox testing of deep learning systems is proposed. Deep Learning (DL)

has made tremendous progress, achieving or surpassing human-level performance for a diverse set of tasks including image classification (He et al. 2016, Simonyan and Zisserman 2014), which has led to widespread adoption and deployment of DL in security- and safety-critical systems such as self-driving cars (Bojarski et al. 2016). Unfortunately, DL systems, despite their impressive capabilities, often demonstrate unexpected or incorrect behaviours in corner cases for several reasons such as biased training data, overfitting, and underfitting of the models (Pei et al. 2017a).

The proposed method aims to identify erroneous behaviours of a DL system without manual labelling/checking, by jointly maximizing a joint objective function combining a metric called neural coverage, and differential behaviour between multiple tested methods. The objective function is maximized by changing the input variable x, under some physical constraints. For example, an input image can be rotated or scaled differently, brightness and contrast can be changed, and rain and fog can be added to the input image.

With *differential behaviour* we mean that when different deep neural networks (DNNs) are tested, the same input will be classified into different classes by the different DNNs. The aim is to maximize the probability that a randomly selected DNN provides an output that differs from the output of the other DNNs. Suppose we have N different DNNs, then each DNN has its own function model $F_k : x \rightarrow y$ for $k \in 1 \ldots N$, where x and y are the input and output values respectively. If $F_k[c]$ is the class probability that the output of the k-th neural network is c, and the j-th neural network is randomly chosen, the objective function (which will be maximized) is formulated as

$$obj_1(x) = \sum_{k \neq j} F_k(x)[c] - \lambda_1 \cdot F_j(x)[c] \quad (1)$$

where λ_1 is a parameter to balance the objective terms between the DNNs that maintain the same class outputs as before ($F_{k \neq j}$) and the DNN that produce different class outputs (F_j).

Neural coverage is a measure of how many rules in a DNN are exercised by a set of inputs. The neuron coverage of a set of test inputs is defined as the ratio of the number of unique activated neurons for all test inputs and the total number of neurons in the DNN.

To maximize the neural coverage, Pei et al. 2017a and Tian et al. 2017 propose to iteratively pick inactivated neurons and modify the input such that output of that neuron goes above a predefined threshold t. Hence, for a given neuron n, we maximize the following function

$$obj_2(x) = G_n(x) \text{ such that } G_n(x) > t \quad (2)$$

where G_n is the output value of neural n.

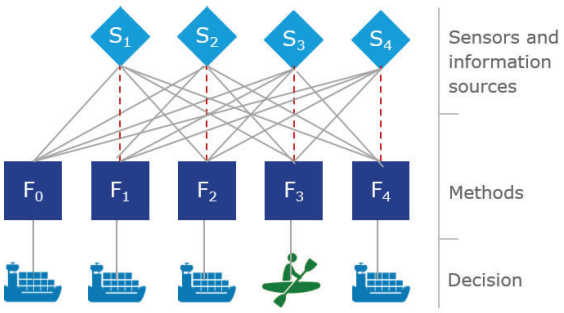


Figure 2: Illustrating how we invoke differential behaviour by repeatedly excluding one information source from the sensor fusion machinery.

The neural coverage and the differential behaviour is jointly maximized, by slightly changing the input values using gradient ascent. The joint objective function is

$$obj_{joint}(x) = \sum_{k \neq j} F_k(x)[c] - \lambda_1 \cdot F_j(x)[c] + G_n(x) \quad (3)$$

By changing the input variables x to maximize this joint objective function, the paper claims that the method finds thousands of erroneous behaviours in fifteen state-of-the-art DNNs trained on five real-world datasets. Hence, new corner cases are explored and different types of erroneous behaviours are uncovered. In addition, test inputs generated by the proposed method can be used to retrain the corresponding deep learning model to improve classification accuracy, and also identify potentially polluted training data (Pei et al. 2017a).

2.8 *Demonstrate need for sensor redundancy*

Inpired by Pei et al. 2017a, as introduced above, we propose to invoke differential behaviour by repeatedly exclude one information source from the sensor fusion machinery. In addition to revealing differential behaviour, we believe this method will be useful to demonstrate the importance of sensor redundancy. If differential behaviour often occurs when a specific information source is removed from the set of explanatory variables, it can indicate that redundancy of this information is needed to achieve adequate robustness.

To illustrate the idea, we consider a method which fuses four information sources: S_1 a day-light camera; S_2 an IR camera; S_3 a radar; and S_4 AIS (Automatic Identification System). F_0 is the standard method which uses all information sources, while the methods F_k for $k > 0$ cannot use the information from information source S_k. The goal is to change the input variable x in way to invoke differential behaviour as illustrated in Figure 2, where the output of method F_3, which does not take information source S_k into account, diverges from the other methods.

In the same way as in section 2.7, we let $F_k[c]$ be the class probability that the output of the k-th method is c. Now the k-th method is the method where information source k is excluded as an explanatory vari-

able. In addition, we propose to include method F_0 which includes all variables. Now the objective function (which will be maximized) is formulated as

$$obj_3(x) = \sum_{k \neq j} F_k(x)[c] - \lambda_3 \cdot F_j(x)[c] \quad (4)$$

where j is randomly chosen, and λ_3 is a parameter to balance the objective terms between the method that maintain the same class outputs as before ($F_{k \neq j}$) and the method that produce different class outputs (F_j).

3 ASSURANCE IN THE MARITIME DOMAIN

The assurance of systems which safety is dependent on the accuracy and reliability of data driven models needs to be thoroughly tested. In this chapter, we present challenges related to machine perception that are unique or particularly pronounced in the maritime domain. We describe potential requirements, and discuss how the recommended practices and tools should be used and possibly adapted to form a framework for assurance in the maritime domain.

3.1 *Important technical challenges in the maritime domain*

One of the major differences, relevant for autonomous navigation, between the automotive and the maritime industry is machine perception. Machine perception, also referred to as artificial or digital perception, is the process where information from sensing, maps, satellite data and the vessel condition, are transformed into situation awareness (see Fig. 3).

The requirements of a machine perception system in the maritime industry will most likely concern both what should be *detected*, such as object types, sizes, distances, reflexibilities, etc.; and what should be *classified*, such as ship types, number of ships, seamarks, complexity, etc. The requirements should be evaluated under various external conditions such as weather and daylight.

Several technical challenges, particularly prominent in the maritime domain, remain open as described by for example (Prasad et al. 2016, Prasad et al. 2017):

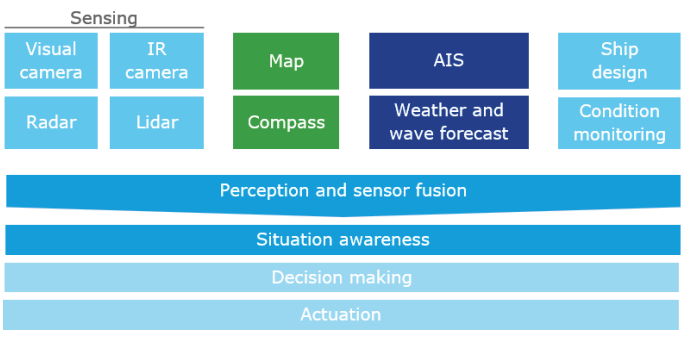


Figure 3: Key components in autonomous navigation in the maritime industry.

– *Vessel movements effect on sensors:* For sensors like video cameras which are mounted on-board ships, the unpredictable motion of the ship complicates the object detection.
– *Background subtraction:* The water background is dynamic due to waves. Hence, background learning methods which recognizes background when a pixel stays constant for at least some time, fails. Also, waves and foam are often misinterpreted as foreground objects when using standard methods.
– *Weather and illumination conditions:* The maritime environment is exposed to a variety of different weather and illumination conditions such as fog, rainfall, clouds, bright sunlight, twilight and night. The different solar angles pose significant challenges with speckle and glint which makes it difficult to distinguish background and foreground.
– *Insufficient training data from the maritime domain:* Very limited work has been carried out to develop object classification algorithms for objects relevant to the maritime environment. The objects of interest include other ships, leisure boats, kayaks, land marks, buoys, ice bergs, etc.
– *Uneventful sailing:* On ocean going ships especially, a very large fraction of the collected data from a voyage are uneventful, hence a very large portion of the corner cases are left unproven.

3.2 *Operational specific requirements*

Operational specific requirements are not considered in current class rules. We foresee that this might change in the future, especially for ships with autonomous navigation systems, as the operation will be embedded into the technology rather than being the responsibility of the human operator. For example, if the ships perception is limited due to fog, the permitted speed might be lowered, or the ship might be denied access to specific geographical areas, until the weather conditions improve. This decision might also be based on ship type, cargo, manoeuvring capabilities, etc.

3.3 *Triple modular redundancy*

The tools presented in 2.7 and 2.8 above, both search for differential behaviour from multiple algorithms or sensor selections, using a majority organ (or voting circuit). This concept was first described by Von Neumann 1956. Today, the concept is often referred to as *triple modular redundancy* and is perhaps most widely used in space and aeronautics applications (Wu et al. 2017, Yeh 1996), where reliability requirements sometimes are very high. Using the majority vote out of three (or more) methods ensures that a single failure will not cause a system failure. We believe this concept is highly relevant for autonomous navigation, as well as other black box AI algorithms, and believe the use of this should be required, in some form,

to ensure sufficient system reliability and robustness.

3.4 *Approval of alternatives and equivalents*

According to the International Maritime Organizations guidelines for the approval of alternatives and equivalents (IMO Maritime Safety Committee 2013), the approval of an alternative and/or equivalent design can be performed by comparing the alternative design to existing designs to demonstrate that the design has an equivalent level of safety. Hence, the approval of autonomous systems used in shipping, including everything from smaller automated tasks to fully autonomously navigated ships, will be based on the equivalence principle: The autonomous functionality must make the operation safer or at least as safe as the conventional operation.

3.5 *Automatic assessment of human perception ability*

To enable the comparison of human and autonomous perception, evaluation metrics and measures, and performance thresholds should be identified. To achieve this, the human perception in representative real ship operations has to be studied. Research in the field of human errors have shown that a large number of investigated maritime accidents are related to loss of situation awareness (Grech et al. 2002).

It should be noted that the perception ability is not necessarily the ambition. We know that the perception performance can be influenced by many factors such as for example stress, distractions, monotony, boredom, etc. (Horrey et al. 2017, Brodsky and Slor 2013, Schwebel et al. 2012), but our aim is to measure the perception performance in practice.

Simulation tools might be applied to provide more extensive data sets, to complement the data collected from real operation. In addition to increasing the data set, the simulation tool offers the possibilities to create controlled situations, and explore changes in weather, rotated objects, etc. as well as the possibility to explore potentially dangerous situations. Another advantage with the simulated data is that it is prelabelled, and one will therefore avoid spending time and effort to establish the ground truth on the simulated datasets.

4 CONCLUSIONS

A framework and tentative guidelines for assurance of autonomous systems in the maritime industry are proposed and discussed, with additional focus on the perception and situation awareness functionality. Because vital parts of the autonomous systems, such as the machine perception and situational awareness algorithms, are expected to be partly or fully based on machine learning algorithms, including deep learning, whose functional reasoning is challenging or even impossible to understand and predict, we believe the assurance of such systems are fundamentally different from a traditional assurance process based on physical understanding and theory. Hence, we believe new guidelines, framework and methodologies are needed.

We propose and describe a range of recommended practices and tools that can be applied to test and validate the ability, performance and robustness of safety critical systems which decisions are based on data-driven methods. We discuss challenges related to machine perception that are unique or particularly pronounced in the maritime domain, and suggest how the recommended practices and tools should be used and possibly adapted to constitute an assurance framework for autonomous navigation in the maritime domain. Furthermore, we discuss potential implications of autonomy such as for example operational dependent requirements. We also discuss the assurance framework for autonomous systems relative to the IMO guidelines for approval of alternatives and equivalents.

ACKNOWLEDGEMENT

We thank Per Ove Husøy (Kongsberg Digital), Michael Link (Kongsberg Digital), Erlend Vågsholm (Kongsberg Maritime) and Jørgen Ernstsen (University College of Southeast Norway) for interesting discussions related to the topics of this paper.

REFERENCES

Ackerman, E. (2017). How drive.ai is mastering autonomous driving with deep learning. `https://spectrum.ieee.org`. Date: 10.5.2017.

Arlot, S. & A. Celisse (2010). A survey of cross-validation procedures for model selection. *Statist. Surv. 4*, 40–79.

Bojarski, M., D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Brandsæter, A. & E. Vanem (2016). Ship speed prediction based on full scale sensor measurements of shaft thrust and environmental conditions. *Submitted*.

Brodsky, W. & Z. Slor (2013). Background music as a risk factor for distraction among young-novice drivers. *Accident Analysis & Prevention 59*, 382–393.

Chen, Z. & T. Ellis (2014). Semi-automatic annotation samples for vehicle type classification in urban environments. *IET Intelligent Transport Systems 9*(3), 240–249.

Cuevas, C., E. M. Yáñez, & N. García (2015). Tool for semiautomatic labeling of moving objects in video sequences: Tslab. *Sensors 15*(7), 15159–15178.

Fei-Fei, L. (2010). Imagenet: crowdsourcing, benchmarking & other cool things. In *CMU VASC Seminar*, Volume 16, pp. 18–25.

Fingas, J. (2017). Rolls-royce unveils plans for an autonomous patrol ship. `https://www.engadget.com/2017/09/12/rolls-royce-autonomous-patrol-ship/`. Retrieved 10.10.2017.

Grech, M. R., T. Horberry, & A. Smith (2002). Human error in maritime operations: Analyses of accident reports using the leximancer tool. In *Proceedings of the human factors and*

ergonomics society annual meeting, Volume 46, pp. 1718–1721. Sage Publications Sage CA: Los Angeles, CA.

Harrald, J. R., T. Mazzuchi, J. Spahn, R. V. Dorp, J. Merrick, S. Shrestha, & M. Grabowski (1998). Using system simulation to model the impact of human error in a maritime system. *Safety Science 30*(1), 235–247.

He, K., X. Zhang, S. Ren, & J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hofmann, P. (2013). Object detection and tracking with side cameras and radar in an automotive context. Master's thesis, Institute of Computer Science of Freie Universitt Berlin.

Horrey, W. J., M. F. Lesch, A. Garabet, L. Simmons, & R. Maikala (2017). Distraction and task engagement: how interesting and boring information impact driving performance and subjective and physiological responses. *Applied ergonomics 58*, 342–348.

Huval, B., T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, et al. (2015). An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*.

IMO Maritime Safety Committee (2013). Guidelines for the approval of alternatives and equivalents as provided for in various imo instruments (24 june 2013 ed.). *IM Organization, Ed. London: International Maritime Organization*.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, San Francisco, CA, USA, pp. 1137–1143. Morgan Kaufmann Publishers Inc.

Kongsberg (2017). Autonomous ship project, key facts about yara birkeland. `https://www.km.kongsberg.com/`. Retrieved 8.10.2017.

Krotkov, E., S. Fish, L. Jackel, B. McBride, M. Perschbacher, & J. Pippine (2007). The darpa perceptor evaluation experiments. *Autonomous Robots 22*(1), 19–35.

Lambert, F. (2016). Understanding the fatal tesla accident on autopilot and the nhtsa probe. *Electrek, July*.

Lopez-Villa, J., H. Insuasti-Ceballos, S. Molina-Giraldo, A. Alvarez-Meza, & G. Castellanos-Dominguez (2015). A novel tool for ground truth data generation for video-based object classification. In *Signal Processing, Images and Computer Vision (STSIVA), 2015 20th Symposium on*, pp. 1–6. IEEE.

Lützhöft, M. & S. W. Dekker (2002). On your watch: automation on the bridge. *The Journal of Navigation 55*(1), 83–96.

Madrigal, A. C. (2017). Inside waymos secret world for training self-driving cars. `https://www.theatlantic.com/technology/archive/2017/08/inside-waymos-secret-testing-and-simulation-facilities/537648/`.

MathWorks (2017a). Automated driving system toolbox for matlab. `https://se.mathworks.com/products/automated-driving.html`. Retrieved: 29.9.2017.

MathWorks (2017b). Computer vision system toolbox for matlab. `https://se.mathworks.com/products/computer-vision.html`. Retrieved: 29.9.2017.

MathWorks (2017c). Object recognition methods in computer vision. `https://in.mathworks.com/discovery/object-recognition.html`. Retrieved: 25.9.2017.

Muoio, D. (2016). A start-up born out of stanford just entered the driverless car race with a radical approach. `http://www.businessinsider.com/driveai-using-deep-learning-for-its-autonomous-cars-2016-8?r=US&IR=T&IR=T`. Retreived: 30.8.2016.

Pei, K., Y. Cao, J. Yang, & S. Jana (2017a). Deepxplore: Automated whitebox testing of deep learning systems. *arXiv preprint arXiv:1705.06640*.

Pei, K., Y. Cao, J. Yang, & S. Jana (2017b). Towards practical verification of machine learning: The case of computer vision systems. *arXiv preprint arXiv:1712.01785*.

Prasad, D. K., C. K. Prasath, D. Rajan, L. Rachmawati, E. Rajabaly, & C. Quek (2016). Challenges in video based object detection in maritime scenario using computer vision. *arXiv preprint arXiv:1608.01079*.

Prasad, D. K., D. Rajan, L. Rachmawati, E. Rajabally, & C. Quek (2017). Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

Rothblum, A. M. (2000). Human error and marine safety. In *National Safety Council Congress and Expo, Orlando, FL*.

Rout, R. K. (2013). *A survey on object detection and tracking algorithms*. Ph. D. thesis, The department of Computer Science and Engineering of National Institute of Technology Rourkela.

Schöning, J., P. Faion, & G. Heidemann (2015). Semi-automatic ground truth annotation in videos: An interactive tool for polygon-based object annotation and segmentation. In *Proceedings of the 8th International Conference on Knowledge Capture*, pp. 17. ACM.

Schwebel, D. C., D. Stavrinos, K. W. Byington, T. Davis, E. E. ONeal, & D. De Jong (2012). Distraction and pedestrian safety: how talking on the phone, texting, and listening to music impact crossing the street. *Accident Analysis & Prevention 45*, 266–271.

Simonyan, K. & A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sun, Y., G. Xiong, W. Song, J. Gong, & H. Chen (2014). Test and evaluation of autonomous ground vehicles. *Advances in Mechanical Engineering 6*, 681326.

Sun, Y., G. M. Xiong, H. Y. Chen, S. B. Wu, J. W. Gong, & Y. Jiang (2011). A cost function-oriented quantitative evaluation method for unmanned ground vehicles. In *Advanced Materials Research*, Volume 301, pp. 701–706. Trans Tech Publ.

Tian, Y., K. Pei, S. Jana, & B. Ray (2017). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. *arXiv preprint arXiv:1708.08559*.

Tvete, H. A. (2017). The revolt, a new inspirational ship concept. `https://www.dnvgl.com/technology-innovation/revolt/index.html`. DNV GL, Retreived: 8.10.2017.

Von Neumann, J. (1956). Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata studies 34*, 43–98.

Waymo (2016). Report on autonomous mode disengagements for waymo self-driving vehicles in california. Technical report, Waymo.

Wei, J. & J. M. Dolan (2009). A robust autonomous freeway driving algorithm. In *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 1015–1020. IEEE.

Woodgate, E. (2017). Students design autonomous containers to disrupt sea freight of aquaculture products. `https://www.dnvgl.com/`. DNV GL Press release, 15.8.2017.

Wu, C.-H., T.-J. Chen, T.-Y. Hsu, S.-H. Tsai, & H.-P. Chang (2017). Design of applying flexray-bus to federated archiectecture for triple redundant reliable uav flight control system. In *Dependable and Secure Computing, 2017 IEEE Conference on*, pp. 73–78. IEEE.

Yeh, Y. C. (1996). Triple-triple redundant 777 primary flight computer. In *Aerospace Applications Conference, 1996. Proceedings., 1996 IEEE*, Volume 1, pp. 293–307. IEEE.

Zhao, D. & H. Peng (2017). From the lab to the street: Solving the challenge of accelerating automated vehicle testing. *arXiv preprint arXiv:1707.04792*.