National Technical University of Athens
School of Naval Architecture and Marine Engineering

Diploma Thesis

Operational Anomaly Detection Using Clustering Methods and Machine Learning Models

Georgios Apostolakos

Supervisor: Assistant Professor Nikos Themelis

September 2023

Acknowledgments

I want to express my gratitude to the people who provided unwavering support and encouragement during my journey to complete my studies and thesis at the School of Naval Architecture and Marine Engineering of the National Technical University of Athens.

I express my thanks to my thesis advisor, assistant professor Nikos Themelis, for his guidance and help throughout this process. His insights and feedback significantly shaped the direction of this work.

Additionally, I would like to express my gratitude to Laskaridis Shipping Co. Ltd for their support in providing the ship's operational data which made this study possible.

I am also grateful to my family. Their belief in me has been my constant motivation during my studies. Their support and encouragement provided a solid foundation to complete my academic journey.

I would like to thank my friends with whom, through mutual support, managed to overcome obstacles, solve problems and share experiences that seemed insightful during the course of my studies as well as this thesis.

Abstract

This thesis addresses the critical issue of anomaly detection in marine operational systems using clustering methods and machine learning models. The study aims to develop an effective methodology for identifying anomalies within complex cargo vessel main engine data. The research investigates the application of various unsupervised machine learning techniques to detect anomalies, with a particular emphasis on the practical implementation of K-Means Clustering, Gaussian Mixture Models, Density-Based Spatial Clustering of Applications with Noise, and Self Organising Maps. Following the introductory section, where the purpose of this study is stated, an extensive literature review of the marine engine anomaly detection topic is presented. All the fundamental theories and insights concerning the algorithms employed are found in the theoretical background. In this section are also presented the methodologies behind the machine learning algorithms which are utilized for anomaly detection. The methodology section delves into data preparation, encompassing data cleaning, de-noising, steady state identification, normalization, and dimensionality reduction. The anomaly detection framework for each model is then presented followed by a presentation of the simulated anomalies which are utilized for model testing purposes. In the case study and results are presented detailed descriptions of the data used, the data preparation procedure, and the implementation of anomaly detection algorithms. The results showcase the effectiveness of each algorithm in identifying anomalies within operational data. All models delivered good results. These results are critically analyzed in the discussion section, along with potential improvements. Additionally addressed are few data-related challenges such as the importance of maintaining the time sequence in time-series data and overall data quality as a results affecting parameter. Also examined, are the impact of dimensionality reduction on the accuracy of anomaly detection and the detection of simulated anomalies. The key findings along with future research proposals in this field are summarized in the conclusion section.

Περίληψη

Η παρούσα εργασία εξετάζει το ζήτημα της ανίχνευσης λειτουργικών ανωμαλιών βασισμένες σε μεθόδους συσταδοποίησης (clustering) και σε μοντέλα μηχανικής μάθησης (machine learning). Η μελέτη στοχεύει στην ανάπτυξη μιας αποτελεσματικής μεθοδολογίας για τον εντοπισμό ανωμαλιών σε περίπλοκα δεδομένα που προέρχονται από την κύρια μηχανή εμπορικών πλοίων. Κατά τη διάρκεια της εργασίας διερευνάται η εφαρμογή πολλών μη επιβλεπόμενων (unsupervised) τεχνικών μηχανικής μάθησης για την ανίχνευση ανωμαλιών, με ιδιαίτερη έμφαση στην πρακτική εφαρμογή της συσταδοποίησης K-Means, των Gaussian Mixture Models, Density-Based Spatial Clustering of Applications with Noise και Self Organising Maps. Μετά την εισαγωγή, που αναφέρεται ο σκοπός της εργασίας, ακολουθεί μια εκτενής βιβλιογραφική ανασκόπηση πάνω στο θέμα της ανίχνευσης ανωμαλιών σε ναυτικούς κινητήρες. Στη συνέχεια, παρουσιάζεται το θεωρητικό υπόβαθρο που αφορά τις έννοιες που χρησιμοποιούνται στην πορεία της εργασίας. Παράλληλα αναπτύσσονται οι θεωρίες και οι μεθοδολογίες των αλγορίθμων μηχανικής μάθησης που χρησιμοποιούνται. Η ενότητα που αφορά τη μεθοδολογία εμβαθύνει στην προετοιμασία των δεδομένων που περιλαμβάνει τα εξής: λεπτομερείς περιγραφές των δεδομένων που χρησιμοποιήθηκαν, τη διαδικασία προετοιμασίας δεδομένων και την εφαρμογή αλγορίθμων ανίχνευσης ανωμαλιών. Στα αποτελέσματα που ακολουθούν, φαίνεται η αποτελεσματικότητα κάθε αλγορίθμου στον εντοπισμό ανωμαλιών με βάση τα επιχειρησιακά δεδομένα. Όλα τα μοντέλα παρουσίασαν αρκετά καλά αποτελέσματα. Αυτά αναλύονται στην ενότητα επεξεργασίας και κριτικής των αποτελεσμάτων, μαζί με πιθανές βελτιώσεις για αυτά. Επιπροσθέτως, αντιμετωπίζονται κάποιες προκλήσεις που σχετίζονται με τα δεδομένα, όπως η σημασία της διατήρησης της χρονικής σειράς σε τέτοιου τύπου δεδομένα αλλά και η συνολική ποιότητα των δεδομένων σαν παράμετρος που επηρεάζει τα αποτελέσματα. Εξετάζονται επίσης ο αντίκτυπος των μεθόδων μείωσης διαστάσεων (dimensionality reduction) των δεδομένων στην ακρίβεια της ανίχνευσης ανωμαλιών αλλά και η ανίχνευση προσομοιωμένων ανωμαλιών. Τα βασικά ευρήματα μαζί με μελλοντικές προτάσεις σε αυτό το περιβάλλον συνοψίζονται στην ενότητα των συμπερασμάτων.

Contents

List of Figures

List of Tables

Acronyms

AAKR Auto Associative Kernel Regression

AI Artificial Intelligence

AIC Akaike Information Criterion

AMS Alarm Monitoring System

ANN Artificial Neural Networks

BIC Bayesian Information Criterion

BMU Best Matching Unit

CBM Condition Based Maintenance

CH Calinski-Harabasz

CNN Convolutional Neural Networks

DAQ Data Acquisition

DBSCAN Density-Based Spatial Clustering of Applications with Noise

EAI Explainable Artificial Intelligence

EM Expectation Maximization

ESN Echo State Network

EWMA Exponentially Weighted Moving Average

FN False Negative

FP False Positive

GMM Gaussian Mixture Models

k-NN k Nearest Neighbors

LL Last Leap

LML Last Major Leap

LO Lubricant Oil

ME Main Engine

ML Machine Learning

MSE Mean Square Error

NRMSE Nomalised Root Mean Square Error

PCA Principle Component Analysis

PHM Prognostics & Health Management

RMSE Root Mean Square Error

SG Savitzky-Golay

SHAP Shapley Additive Explanation

SOM Self Organising Map

SPRT Sequential Probability Ratio Test

SSE Sum of Squared Error

SVM Support Vector Machine

TN True Negative

TP True Positive

VAE Variational Autoencoder

1 Introduction

The development of new applications which collect and analyze data to reach meaningful conclusions is continuously increasing in this rapidly evolving technological landscape. The integration of Machine Learning (ML) and Artificial Intelligence (AI) has emerged as a fundamental catalyst, reshaping industries and redefining problem-solving paradigms. One industry that has been notably affected is shipping. These technologies have assumed a pivotal role in revolutionizing the maritime cluster, catalyzing advancements in operational efficiency, safety protocols, and sustainability measures. Indicative applications utilizing these technologies within the industry include prediction of fuel consumption (Gkerekos et al., 2019), speed loss (Karagiannidis & Themelis, 2021), propulsion power (Kim et al., 2020a), weather routing (Gkerekos & Lazakis, 2020), development of monitoring systems (Guanglei et al., 2019), energy efficiency analysis (Beşikçi et al., 2016), and predictive maintenance algorithms (Jimenez et al., 2020).

Furthermore, a rise in investment focused on the four primary categories delineating the smart shipping sector is anticipated (Economics & International, 2021). The categories are namely:

1. smart port
2. autonomous vessels
3. on-board technologies
4. professional services technologies

Collectively, it is widely acknowledged within the industry that, achieving seamless integration of smart applications across these domains, holds paramount significance for stakeholders and industry players alike.

Among the various smart technologies, this study places particular emphasis on smart maintenance, recognizing the need for further research in ship operations. While the potential benefits of AI applications in the shipping sector, such as achieving optimal and cost-effective maintenance practices, have been acknowledged, as well as its potential positive impacts on personnel safety and security, a definitive technological solution for these challenges remains elusive. Nonetheless, some endeavors have been observed concerning the advancement of Prognostics & Health Management (PHM) for shipping systems (Velasco-Gallego & Lazakis, 2022a).

When it comes to Condition Based Maintenance (CBM) applications within the sector, it is estimated that only around 2% of vessels operate under that scheme (Jimenez et al., 2020), indicating limited integration of this aspect of ML and AI based technologies (Velasco-Gallego & Lazakis, 2022c). A CBM program may be data-driven or physics based, such as the models developed by Lamaris & Hountalas (2010) and Dimopoulos et al. (2014). The main disadvantage of the latter lies in their high modeling complexity and difficulty of use (Vanem & Brandsæter, 2021).

In the case of a data-driven model, it is comprised of three main algorithms: anomaly detection, fault identification, and prognostics. Anomaly detection involves monitoring data streams to identify deviations from the typical system behavior, which serve as indicators of system changes (Chandola et al., 2009). In fault identification, a diagnostic tool is applied to discern the nature of the detected anomaly—distinguishing between actual faults and unexpected yet normal system behavior and precisely identifying the type of fault. The prognostics task strives to predict the future behavior of the system based on its current state and estimate its remaining useful life Vanem & Brandsæter (2021).

1.1 Objective of the Study

The focus of this study centers on the comprehensive analysis of multidimensional time series data originating from the main engine of a merchant vessel. The primary objective of this study

is the development and application of anomaly detection techniques to the available dataset. One significant challenge emerges: the boundaries between abnormal and normal operating states cannot be set, or such task proves to be very difficult. This challenge is primarily attributed to the intricate complexity of the data and the correlation between various variables. As a result, the establishment of a rule-based system relying on extreme values for anomaly detection appears to be an unfeasible alternative.

Hence, this study adopts a data-driven perspective, deviating from the physics-based approaches. The proposed methodology revolves around the development of algorithms which utilize clustering models and machine learning techniques to address the anomaly detection challenge. Specifically, the examined methods include K-Means clustering, clustering with GMM, DBSCAN, and Self Organising Maps (SOM) forming a comprehensive framework for tackling this intricate problem.

Furthermore, special attention is paid to the data preparation phase, aiming to maximize the utility of the provided dataset. This phase involves tasks as data cleaning and refinement to remove noise, steady state identification to ensure proper training of the algorithms, data normalization, and dimensionality reduction to reduce complexity of the datasets.

1.2 Purpose of the Study

Four fundamental pillars form the purpose of the study. First, it aims in a presentation of practical applications associated with anomaly detection. Second, various methods are developed in the research, thereby providing a structured approach to addressing the anomaly detection problem. Third, multiple anomaly detection models are developed, thus reflecting a proactive approach to confronting this challenge. Fourth, the study is involved in the comparison of those anomaly detection models, enabling a critical assessment of their practical utility and effectiveness.

1.3 Structure of Thesis

This paragraph showcases how the rest of the study is structured. The second chapter involves a literature review about the topic of anomaly detection in marine machinery and its applications. The next chapter contains the theoretical background required for the comprehension of the methodologies which are employed in the thesis. This also includes the theory behind the employed clustering methods. Chapter four is named "Methodology". It comprises of three parts: methodology of data preparation, of anomaly detection, and of how anomalies are simulated. The case study and the results are shown in the fifth chapter, followed by a discussion in the sixth. The final chapter, "Future Work", consists of potential concepts and ideas for enhancement of the methodologies presented in the previous chapters.

2 Literature Review

The concept of anomaly detection based on data driven methods has been already researched in the maritime sector. Multiple studies have been identified implementing several ML methodologies.

Brandsæter et al. (2016) developed a framework based on Auto Associative Kernel Regression (AAKR) and Sequential Probability Ratio Test (SPRT). AAKR was responsible for signal reconstruction of each data point and SPRT was implemented for the anomaly detection task by performing residuals analysis. All historical observations were collected in a matrix from which the reconstructed signals were created. Initially the signals were calculated as a linear combination of each training point. Gaussian kernels were used as weights. To modify them, an alternative distance measure was implemented which gave less importance to instances with significant difference between the observation and training point. An adequate reconstruction of the signal was expected to have a small Mean Square Error (MSE). The anomaly detection by SPRT was performed sequentially for each point. The assumption was that normal-state residuals must be normally distributed with mean 0 and standard deviation σ whereas the anomalous are to have non-zero mean and/or different standard deviation from normal-state. The system's condition was determined by two decision variables which act as lower and upper boundaries for an index which was calculated directly from a sequence of residuals. In case the lower boundary was reached, the normal-state hypothesis was accepted. For instances between the two boundaries, no conclusion can be reached, and, if the upper limit was exceeded, anomalous state was accepted. As highlighted by the authors, for each point's reconstruction with AAKR all training points must be used and, as a result, this method is not suitable for large-dimensional datasets.

In their next study (Brandsæter et al., 2017), the authors presented a modified version of AAKR. The modifications applied to the method include a modified distance measure and a k-means clustering based approach to enhance the algorithm's performance when reconstructing the signal. The first modification was the addition of a distance scaling factor (vector) to the distance measure. Authors treated all signals except one as explanatory variables and the one remaining as a response variable. This dictated the values inserted in the vector. As for the clustering, it was selected to replace the training points in the reconstruction phase with a predetermined number of clusters resulting in an increased computational speed while maintaining similar results.

Brandsæter et al. (2019) published another updated version of their research. A significant modification between Brandsæter et al. (2019) and Brandsæter et al. (2017) was the use of clustering based techniques for the detection of anomalies rather than SPRT. Data were characterized as anomalous if:

- they did not belong to a cluster,
- there was a great distance between the cluster centroid and the data point,
- they belonged to clusters with small population and density.

Centered and enclosed clusters surrounding sets were also explored. Centered sets are those where the boundary is defined by the distance between the centroid and standard deviation, adjusted by a scaling factor. Enclosed sets, surround all points within the cluster. Agglomerative hierarchical clustering has been examined in parallel to k-means. The optimal number of clusters was defined after an internal validation of the obtained results. This means that the goodness of fit between each cluster and the raw data from which it is generated is examined. The last addition was a reconstruction credibility estimator which was based on the principle that reconstructions from regions of the space, where the query vector is closer to dense historical observations of explanatory variables, were considered more credible. The dataset consisted

of five parameters. To perform this methodology in large datasets, dimensionality reduction techniques need to be implemented. To simulate anomalies, a change in a parameter was forced. Results show increased Root Mean Square Error (RMSE) which indicates that SPRT would have been able to capture the anomalies. Other performance metrics introduced include Expected Detection Delay (EDD) and Average Run Length (ARL) that express the expected time from when an anomaly is introduced until detected and the expected time points between false alarms respectively. The presented methodology showed similar results to Brandsæter et al. (2016) even with relatively small number of clusters.

Vanem & Brandsæter (2021) explored anomaly detection exclusively through unsupervised learning models. A comparative analysis between five methods is presented. These included k-means, mixture of Gaussian models, density based clustering, self organising maps, and support vector machines. Dimension reduction was applied to the data through Principle Component Analysis (PCA) resulting in a seven dimensional dataset from twenty five initially. Superior results may be achieved through a combination of different methods in one algorithm.

In all of the previously presented studies, Brandsæter et al. (2016), Brandsæter et al. (2017), Brandsæter et al. (2019), and Vanem & Brandsæter (2021), time stamp, sequence, and dependencies within the data have been neglected. This operation is not recommended (Bergmeir et al., 2018), eventhough it delivered adequate results in the examined cases.

Guanglei et al. (2019) developed an algorithm for engine monitoring purposes based on Gaussian Mixture Models and Principle Component Analysis. The aim is that this system can detect machinery faults in their early stages, while providing an easy to implement and robust methodology. This study also took into consideration ship navigational data in order to connect ship machinery performance to weather or sea conditions.

A SOM based approach was introduced by Raptodimos & Lazakis (2018). Data were clustered by implementing a two level methodology. In first level, the SOM algorithm ran until minimal changes were noticed in the resulting map. The objective of the second level was to group neighboring clusters by calculating the distance between cluster centres and comparing it with four predetermined values. If no neighbor was found in the first loop, then it would search for neighbors in a larger area, defined by the second value etc. Low frequency data from a single ME cylinder were used.

Cheliotis et al. (2022) developed a fault detection and diagnostic model. The methodology was divided in data collection, data preparation, fault detection, and the diagnostic module. Data preparation was handled by DBSCAN. Also, corrections to ISO and other standard conditions were performed. An Expected Behaviour approach was used with multiple Polynomial Ridge Regression models. The residuals were analysed by Exponentially Weighted Moving Average (EWMA) and were characterized as normal if they lied between control levels. The diagnostic module presented was a Bayesian Network. Conditional probability distribution of occurrence of a fault given the clustering results are calculated based on the assumption of independence. The model may be used to monitor system degradation and trends while identifying the root cause of faults.

Velasco-Gallego & Lazakis (2022c) addressed the anomaly detection and diagnosis issue with a Long Short-Term Memory-Based Variational Autoencoder Neural Network in parallel with multi-level Otsu's thresholding. Pre-processing mainly constituted of steady state identification via transforming the input time series into an image using first-order Markov chain (Velasco-Gallego & Lazakis, 2022d). De-noising was combined with dimension reduction using VAE, an algorithm which learns the parameters' probability distribution and the time dependencies in the dataset. For anomaly detection, Nomalised Root Mean Square Error (NRMSE) matrix is calculated and converted into an image with distinct regions. The thresholds are defined by Otsu's method resulting in classes within the image whose number is defined from Gaussian mixture models. Since Otsu's method could not handle multiple classes, it was decided to keep one for normal and one for anomalous behaviour.

In their next study (Velasco-Gallego & Lazakis, 2022a), the authors explored fault classification through time series imaging by first-order Markov chain and image analysis with ResNetSOV2 (a type of deep residual network) and Convolutional Neural Networks (CNN). Pre-processing was handled as in Velasco-Gallego & Lazakis (2022c) with the addition of data imputation step as described in Velasco-Gallego & Lazakis (2022b) and data normalisation with sliding window algorithm. Time series images are then generated with first-order Markov chain by estimating the transition matrix through a stochastic process. To validate the results, also, Gramian Angular Field algorithm was used to transform time series data into an image. The size was set to 50x50 pixels to avoid the risk of over-fitting. Best results found when using the Markov-CNN method. Authors suggest that ResNetSOV2 may not be suitable with this study.

Cai et al. (2017) developed a fault diagnosis model for a marine ME. The engine was divided into four subsystems (fuel, lubrication, intake and exhaust, cooling) and diagnosis was performed in each one separately. If all variables from each subsystem were used, their existing high correlation would have lead to complexity and redundancy. Thus, a selection of variables has been used in the algorithm. A Support Vector Machine (SVM) model handled the classification part and the association between fault features was analysed by the association rule mining algorithm.

A predictive anomaly detection tool was proposed by Qu et al. (2022). The method is based on an Echo State Network (ESN) for prediction of future time series and a deep auto-encoder for the anomaly detection of the predicted data. Prediction of one minute's data is based on the previous three minutes. This minute is used as input to the anomaly detection algorithm. Promising results were found from the study when compared to other methodologies.

Another predictive anomaly detection tool has been developed by Makridis et al. (2020). Time-series forecasting methods have been created in this study by utilizing LSTM neural networks, one class SVMs, Gradient Boosting Classification, and Weighted Permutation Entropy. The anomaly detection part of the algorithms was handled by rules and threshold values on the residuals (actual - predicted). The study aimed to detect faults in the crosshead bearings of main engines.

Kim et al. (2020b) came up with an ensemble approach to the problem. After data pre-processing, which included removal of out-of-range values, idle periods, 10-minute averaging and dimension reduction due to high correlation between certain parameters, multiple anomaly detectors were trained. The detectors were based on a modified k-nearest neighborhood method (Local Outlier Factor). The ensemble process identified local regions derived from LOF, through a method called Locally Selective Combination in Parallel Outlier Ensembles (LSCP). It constructs competitive ensemble anomaly detectors for each of these local regions within the dataset, thus ensuring the provision of resilient predictions. Then the anomalous regions were clustered with k-means resulting in four clusters. Authors suggest that an Explainable Artificial Intelligence (EAI) framework should be included in future work to improve analysis outcomes.

The effect of using an EAI technique such as Shapley Additive Explanation (SHAP) is examined in Kim et al. (2021). Pre-processing remained the same as in previous study (Kim et al., 2020b). Then, anomalies were detected with the use of Isolation Forest algorithm. Similar to Random Forest, this method consists of multiple binary decision trees. SHAP was applied to the anomalous instances with a goal of measuring the contribution of each sensor to those. Also a SHAP value was calculated as a metric of anomalousness. Hierarchical clustering was applied on the SHAP values to capture similar anomalous behaviour.

3 Theoretical Background

This section involves the exploration of fundamental theories and insights which are required for the comprehension of methodologies employed in this study. The significance of engine monitoring is initially discussed with focus paid on the the sub-systems that constitute the marine engine. Then, necessary information regarding the machine learning models is presented. The text is structured as follows: A thorough dissection of supervised, semi-supervised, and unsupervised machine learning models is presented accompanied by an explanatory analysis of how an anomaly detection model based on each one would operate. A survey of clustering algorithms, encompassing their diverse typologies follows. Finally, the methodology of models utilized in the study is demonstrated.

3.1 Importance of Engine Modeling

The two purposes of engine modeling are performance evaluation or prediction and interpretation of experimental results and phenomena occurring in an engine (Kyrtatos, 1993). Anomaly detection falls within the first category since the objective of such methodologies is to enhance engine monitoring by understanding previous faults and taking necessary actions to ensure operational reliability and good performance.

3.1.1 Sub-systems of Marine Engine

In that context, Kyrtatos (1993) presented the systems of a marine engine that may be modeled for either of the two purposes.

- Combustion chamber
- Piston rings assembly
- Piston-Connecting rod assembly
- Intake & Exhaust system
- Turbocharger
- Fuel system
- Cooling system
- Lubrication system
- Exhaust valves control system
- Bearings
- Coupled auxiliary machinery

In the case of condition monitoring, the sub-systems are utilized in order to better monitor the overall operation of complex systems, such as those found in marine engines. By dividing the engine in systems, one may focus in the within-system phenomena and also monitor the interactions between systems that influence the overall performance (NASA, 2007; Dimopoulos et al., 2014).

3.1.2 Categories of Maintenance

When it comes to maintenance types, there two main categories: Corrective and Preventive. The last is further divided in Condition Based Maintenance and Predetermined Maintenance (PM).

When corrective maintenance is employed, the philosophy is to replace parts only when failure occurs. This type of maintenance is used in applications where sudden fails in equipment do not affect the overall performance or risk the life of employees.

On the other hand, the philosophy behind preventive maintenance is to replace parts before their failure points. In the case of predetermined maintenance, the replacement intervals are standard and have been determined based on historical observations or by information gathered through similar machinery. This maintenance strategy often leads to waste of resources, since the parts may have remaining useful life by he time they are replaced. Predetermined maintenance is associated with increased reliability. CBM comes to optimize the maintenance field by continuously monitoring the equipment's condition. By doing so, it can be decided what part replacement actions need to by taken at every time instance based on current conditions. These techniques can rely on continuous monitoring and analysis and/or inspection and testing. The first option usually requires sensor data which are gathered and transmitted to databases where they are later analyzed and decisions are taken (Jimenez et al., 2020).

Anomaly detection comes to fulfill the requirement for robust condition analysis. In the particular study this is done through machine learning methods. Other solutions for the same task in marine engines include thermodynamic models, similar to what Lamaris & Hountalas (2010) and Dimopoulos et al. (2014) have presented.

3.2 Categories of Machine Learning Models

3.2.1 Supervised Learning

In the domain of supervised learning, an underlying presumption exists regarding the anticipated outcome. The input data arrives with labels, and the primary objective of the model involves either establishing a correspondence between the output and input labels, or delineating a continuous output when mapped against the input label. The former approach is termed classification, while the latter is referred to as regression. The datasets crafted for this purpose function as guides, steering the algorithms towards the completion of their designated tasks. These datasets encompass feedback mechanisms that facilitate this process, thus giving rise to the term "Supervised Learning."

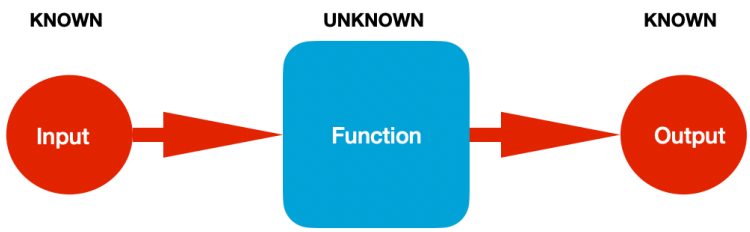

Figure 1: Schematic representation of supervised learning algorithms. Source: Raj (2023).

3.2.2 Unsupervised Learning

On the contrary, unsupervised learning operates within a context where data lacks labels. Its goal revolves around revealing inherent patterns embedded within the data points of the given dataset. In contrast to supervised learning, it lacks an integrated feedback mechanism, thereby earning the label of unsupervised learning. Common unsupervised learning techniques are clustering and dimensionality reduction.

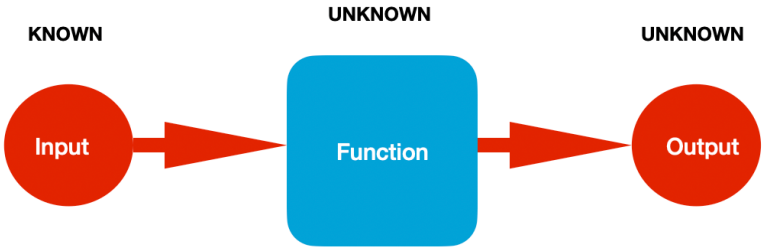

Figure 2: Schematic representation of unsupervised learning algorithms. Source: Raj (2023).

3.2.3 Semi-supervised Learning

An intermediate position is occupied by semi-supervised learning, bridging the gap between supervised and unsupervised paradigms. In this context, during the model's training phase, the training dataset comprises a limited amount of labeled data combined with an extensive collection of unlabeled data. This approach can also be characterized as an instance of weak supervision. Semi-supervised learning models may utilize supervised or unsupervised learning techniques (Raj, 2023).

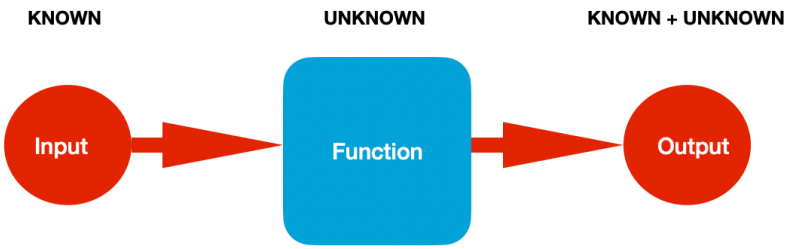

Figure 3: Schematic representation of semi-supervised learning algorithms. Source: Raj (2023).

3.2.4 Supervised, Unsupervised & Semi-supervised Anomaly Detection

Previous researchers have utilized all three learning approaches to construct anomaly detection algorithms in several application fields. The following apply when referring to the data required in each learning type:

Supervised learning applications involves possessing a training dataset with labeled instances for both normal and anomalous behaviors. Typically, prognostic models are constructed for both normal and anomalous behaviors, and unseen data are subsequently assigned into one of these categories.

In unsupervised anomaly detection, the training dataset lacks labeling, assuming implicitly that instances demonstrating normal behavior significantly outnumber anomalies in the test data. Should this assumption prove incorrect, such techniques tend to experience an elevated false alarm rate.

In semi-supervised anomaly detection, the training dataset exclusively consists of normal data instances. A common methodology for anomaly detection is to establish a model specifically for the category associated with normal behavior, subsequently employing this model to identify anomalies within the test data. Since semi-supervised techniques do not mandate anomaly class labels, they hold broader applicability compared to their supervised counterparts (Chandola et al., 2009).

3.3 Taxonomy of Clustering Algorithms

The main parameters responsible for different types of clustering algorithms are initialization conditions and measures of performance (Bindra & Mishra, 2017). An acceptable classification of clustering methods is:

- Hierarchical clustering
- Partitional clustering
- Density based clustering

This categorization stems from a multitude of factors, and certain algorithms have emerged to bridge the gap between these distinct approaches. A plethora of algorithms has been developed to address diverse challenges across various domains. Yet, despite this surge in algorithmic solutions, a universally applicable approach that comprehensively resolves all prevalent clustering problems remains elusive. Constructing an integrated framework (clustering) at an expert level has proven to be a difficult task, resulting in diverse and specialized algorithms (Bindra & Mishra, 2017; Rodriguez et al., 2019).

3.3.1 Hierarchical Clustering

These types of clustering algorithms generate a sequence of progressively nested partitions, or clusters. This sequence can be visualized as a tree, commonly referred to as a cluster dendrogram, offering a hierarchical depiction of clusters. This hierarchical structure (tree) provides a view of the data at each level of abstraction. Each data point residing within a leaf node forms its own cluster, while the root encompasses all points within a singular cluster. By segmenting the dendrogram at different levels, meaningful information can be extracted. The hierarchical approach to clustering is categorized into agglomerative, and divisive classes. The majority of hierarchical clustering algorithms have been primarily obtained using agglomerative methods.

Agglomerative clustering techniques operate with a bottom-up approach, where the merging of the most similar cluster pair is initiated by treating each of the K points as separate clusters. This iterative procedure continues until all data points become members of a unified cluster. Variations of agglomerative algorithms abound, differing mainly in how the resolution of similarity discrepancies between existing clusters and merged clusters is adjusted. Numerous agglomerative algorithms are available, contingent upon the distance measurement between two clusters (Bindra & Mishra, 2017).

3.3.2 Partitional Clustering

Partitional clustering diverges significantly from the hierarchical approach, which progressively generates clusters through iterative mergers or divisions. Partitional clustering, in contrast, assigns a collection of objects into k clusters without establishing a hierarchical arrangement. Partitional algorithms are the preferred choice for handling extensive datasets due to

their relatively modest computational demands. However, in terms of clustering coherence, this method proves less effective than the agglomerative approach. These algorithms deduce cluster shapes as hyper-ellipsoidal and essentially experiment with segmenting data into a predetermined number of clusters to optimize a given criterion through data partitioning.

Centroid-based techniques allocate certain points to clusters in a manner that minimizes the mean squared distance between points and the centroid of the designated cluster. The sum of squared error function reigns supreme as the predominant criterion in the partitional approach, serving as a measure of within-cluster variance. Among the most popular partitional clustering algorithms that employ the sum of squared error function is K-Means and ISODATA algorithms (Bindra & Mishra, 2017; Rodriguez et al., 2019).

3.3.3 Density Based Clustering

This category of clustering algorithms is based on the concept of the neighborhood. Clusters are identified by ensuring that each given N-neighborhood (number of points that form a neighborhood), with a specified $N > 0$, contains a minimum number of points, signifying that the density within the N-neighborhood of points must surpass an initial criterion (Ester et al., 1996). In this context, proximity between objects is not the defining factor; rather, the primary focus rests on assessing local density. A cluster is perceived as an assemblage of data points dispersed throughout the data space.

Within the domain of density-based clustering, the presence of contiguous regions featuring low object density is pivotal, and the measurement of distance between them is conducted. Objects situated within these regions of low density contribute to outliers or noise. These methods exhibit heightened resilience to noise and possess the capacity to uncover clusters characterized by non-convex shapes. Representative methods of density based clustering include DBSCAN and OPTICS (Ordering Points To Identify the Clustering Structure) (Bindra & Mishra, 2017).

3.4 Characteristics of Clustering Algorithms

Bindra & Mishra (2017) consider four characteristics that an efficient clustering algorithm should have in order to solve its assigned problem. These include scalability, the necessity for user's domain knowledge, the ability to discover arbitrary shaped clusters, and the presence of similarity and dissimilarity metrics.

When referring to scalability, the algorithm's objective is to perform with large amount of data within a respectable time.

Another issue is the requirement for prior domain knowledge. Many algorithms require a predefined number of clusters or other parameters as input. However, the user may not be in a position to estimate these parameters beforehand. This results in performance degradation of the algorithms due to the dependence on user input.

Discovering arbitrary shaped clusters presents a formidable challenge, particularly in the identification of clusters exhibiting diverse shapes and sizes. Certain algorithms, such as K-means, fail to do that. Data attributes might have different dimensions, and an effective clustering algorithm must possess the capacity to cluster this type of data. Density-based algorithms like DBSCAN, employing the concept of Minpts, adapt to this challenge. However, a multitude of algorithms based on either centroid or medoid-based methodologies struggle to meet the two clustering criteria of developing varied clusters and converging concave shaped clusters.

Similarity and dissimilarity metrics are measures of quantification the similarity between two objects. A capable algorithm should be able to merge two similar clusters and segregate objects that display dissimilarity, even if that was not expected initially.

3.5 Theoretical Part of Employed Methodologies

An exploration of the methodologies employed to create the anomaly detection models is conducted. The mathematics behind each algorithm and the way it works are examined to ensure proper understanding of how these algorithms will be transformed into anomaly detection models. Furthermore, metrics of finding the optimal parameter combination, advantages and disadvantages of each method are presented along with a simple example of clustering in a sample 2-D dataset.

3.5.1 K-Means Clustering

K-Means is considered a common method for clustering (Makwana et al., 2013). The algorithm was developed by Lloyd (1957) and the process may be summarized by the pseudocode of Figure 4. The technique aims to group similar points into distinct clusters. K-Means requires the number of clusters, K, to be specified before the algorithm's initialization. It operates by assigning data points to their nearest cluster centroid and adjusting the centroid positions based on the assigned points in each iteration. The algorithm strives to minimize the within-cluster variance and maximize the separation between clusters, thus optimizing the clustering outcome. Given the algorithm's nature, the random initialization of centroids at the beginning may lead to different clusters in every run, giving overall non consistent results (Riveiro et al., 2018; Farahnakian et al., 2023). Multiple methods and criteria have been developed in order to establish the optimal number of clusters. Other than the criteria, this operation also requires domain knowledge to evaluate the results (Makwana et al., 2013). The algorithm's objective function is formulated as:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \cdot ||x^i - \mu_k||^2 \tag{1}$$

where $w_{ik} = 1$ for data point x^i if it belongs to cluster k, otherwise $w_{ik} = 0$. μ_k is defined as the centroid of x_i's cluster.

K-Means employs an EM approach to tackle the problem which is considered to be minimization in two-parts. Initially J is minimized with respect to (w.r.t.) w_{ik} and μ_k is fixed. Technically, cluster assignments get updated (E-step) after differentiating J w.r.t. w_{ik}. In the second phase, J is minimized w.r.t. μ_k and w_{ik} remains fixed. The M-step comprises of differentiating J w.r.t. μ_k and recomputing the centroids after cluster assignment. The E-step is formulated as:

$$\begin{aligned} \frac{\partial J}{\partial w_{ik}} &= \sum_{i=1}^{m} \sum_{k=1}^{K} ||x^i - \mu_k||^2 \\ \implies w_{ik} &= \begin{cases} 1 & \text{if} k = \text{argmin}_j ||x^i - \mu_k||^2 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

The M-step may be formulated as:

$$\begin{aligned} \frac{\partial J}{\partial \mu_k} &= 2 \cdot \sum_{i=1}^{m} w_{ik} \cdot (x^i - \mu_k) = 0 \\ \implies \mu_k &= \frac{\sum_{i=1}^{m} w_{ik} \cdot x^i}{\sum_{i=1}^{m} w_{ik}} \end{aligned} \tag{3}$$

Algorithm 1 *K*-means clustering algorithm

Require: *K*, number of clusters; *D*, a data set of *N* points
Ensure: A set of *K* clusters

1. Initialization.
2. **repeat**
3. **for** each point *p* in *D* **do**
4. find the nearest center and assign *p* to the corresponding cluster.
5. **end for**
6. update clusters by calculating new centers using mean of the members.
7. **until** stop-iteration criteria satisfied
8. **return** clustering result.

Figure 4: K-Means clustering algorithm pseudocode. Source: Jin & Han (2010).

As previously mentioned, several methods have been proposed to determine the optimal number of clusters in K-Means. Gupta et al. (2018) found that more than 30 methods exist. The two principles based on which these are created may be summarized in the two following points.

- Compactness: Focus is given on dense and compact clusters
- Separation: Different clusters to be well separated in space.

The examined methods in this study are summarized below.

Elbow Method

The elbow is considered to be the oldest method of determining the number of clusters. The idea is described as repeating the K-Means algorithm with different number of clusters and then plot the Sum of Squared Error (SSE) of distances over the number of clusters. The elbow approach is based on the premise that the explained variation varies quickly for a few clusters, then it slows down, forming a visible elbow in the curve. The amount of clusters we may employ for our clustering process is represented by the elbow point (Yuan & Yang, 2019). Sometimes, it may be difficult to choose the elbow because no clear elbow or multiple elbows may exist in a particular dataset.

Silhouette Coefficient

The silhouette is mathematically expressed as follows:

$$\mathrm{S(i)} = \frac{\mathrm{b(i)} - \mathrm{a(i)}}{\max\{\mathrm{a(i)},\ \mathrm{b(i)}\}} \tag{4}$$

where b(i) is the smallest average distance of point i to all points in any other cluster and a(i) is the average distance of i from all other points in its cluster.

The silhouette was introduced in Kaufman & Rousseeuw (1990). The rationale behind this method is that if only 3 clusters A,B and C are in the dataset and i belongs to cluster C, then b(i) is calculated by measuring the average distance of i from every point in cluster A, the average distance of i from every point in cluster B and taking the smallest resulting value. The Silhouette Coefficient for the dataset is the mean of the Silhouette Coefficients of individual points. The value lies between -1 and 1. If the value is zero, it means that the point may be assigned also to another cluster. If the value is S(i) $\rightarrow$ 1, the entity is well clustered whereas if S(i) $\rightarrow$ –1 it is not assigned to the correct cluster. The average coefficient of all data points

is the Silhouette of the dataset. When comparing different number of clusters, the superior is this with the highest coefficient Makwana et al. (2013).

Calinski-Harabasz Index

The Calinski-Harabasz index was developed based on the principle that good clusters are those that are both highly compact and well-separated from one another. These two principles are expressed through the index which divides the variance of the sums of squares of the distances of individual entities to their cluster center by the sum of squares of the distance between the cluster centers. The selected number of clusters is this with the maximum index value (Calinski & Harabasz, 1974). The CH index is mathematically expressed as follows.

$$\mathrm{CH_k} = \frac{\mathrm{BCSM}}{\mathrm{k-1}} \cdot \frac{\mathrm{n-k}}{\mathrm{WCSM}} \tag{5}$$

with k being the number of clusters, n the number of points, BCSM: Between Cluster Scatter Matrix, calculates separation between clusters, WCSM: Within Cluster Scatter Matrix, calculates compactness within clusters.

Last Leap & Last Major Leap

Most of the previous examined metric required a visual inspection of the results before confirming the value of k. (Gupta et al., 2018) proposed two methods of identifying the number of "natural" clusters of a dataset: the Last Leap (LL) and the Last Major Leap (LML). The first estimates the number that corresponds to well-separated clusters whereas the second aims to determine equal clusters in terms of size. An equal number between both criteria satisfies compactness and separation of clusters.

The LL criterion is expressed by the following index:

$$\begin{aligned} \mathrm{LL(k)} &= \frac{\mathrm{d_k - d_{k-1}}}{\mathrm{d_k}} \\ \bar{\mathrm{k}}_{\mathrm{LL}} &= \mathrm{argmax\ LL(k)} \end{aligned} \tag{6}$$

where $\mathrm{d_k} = \mathrm{min_{i \neq j}} ||\mathrm{v_i - v_j}||^2$ is the minimum distance between cluster centers.

The LML method defines the optimal number of clusters $\mathrm{k_{LML}}$ through the following operation:

$$\mathrm{I_{LML}(k)} = \begin{cases} 1 & \mathrm{if} \frac{1}{2}\mathrm{d_k} > \mathrm{max_{l=k+1,\ldots,k_{max}}\ d_l} \\ 0 & \mathrm{otherwise} \end{cases} \tag{7}$$

$$\mathrm{K} = \{\mathrm{k} | \mathrm{I_{LML}} == 1\} \,\mathrm{and} \tag{8}$$

$$\bar{\mathrm{k}}_{\mathrm{LML}} = \begin{cases} \max \mathrm{K} & \mathrm{if\ K} \neq 0 \\ 1 & \mathrm{otherwise} \end{cases} \tag{9}$$

Advantages & Disadvantages of K-Means

As with every method, several advantages and disadvantages of K-Means have been found. According to Google course for developers (Google, 2022), the significant pros of K-Means are its ability to scale to large data sets due to its efficiency, the way that it adapts to new data, and the simplicity to implement. On the contrary, the disadvantages of this method are summarized in the manual selection of k, the initialization of centroids, and that clusters often may contain outliers or that they can be assigned to a different outlier-only cluster.

Simple Example of K-Means Implementation

A 2-dimensional case is considered as an example. It is know from the beginning of the analysis that belong to three clusters. First, all methods of determining the number of clusters are examined and the respective graphs are developed.

In Figure 5 the elbow is visible at k = 3, so that is the optimal number found by this method. Similarly, in Figure 6 (Silhouette Coefficient), Figure 7 (Calinski-Harabasz Index), the maximum is observed at k = 3. Finally, in Figure 8 the minimum distance between cluster centres, d_k, is plotted against the number of clusters. Since LL and LML identify leaps in d_k to determine the number of clusters and, as seen in the graph, the maximum distance is recorded at k = 3, this is the optimal number of clusters.

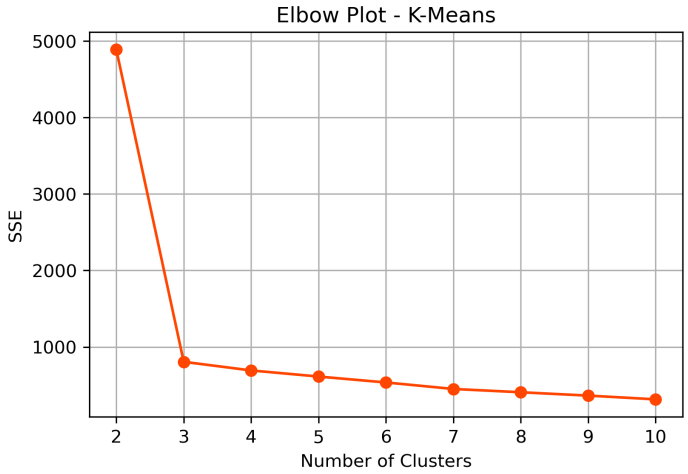

Figure 5: Determination of number of clusters through the elbow method.

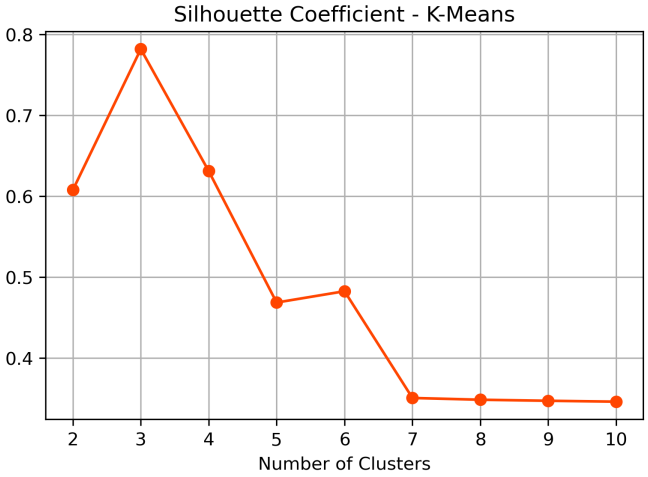

Figure 6: Visual determination of number of clusters through the silhouette coefficient with K-Means.

As for the clustering itself, the result can be observed in Figure 9. This example featured a dataset of three well separated clusters. Each cluster is plotted in a different color and the centroids are seen in black. The usual clusters produced by K-Means have circular (2-D) or hyper-spherical shape (in higher dimensional spaces). This can make K-Means unsuitable for certain types of data sets.

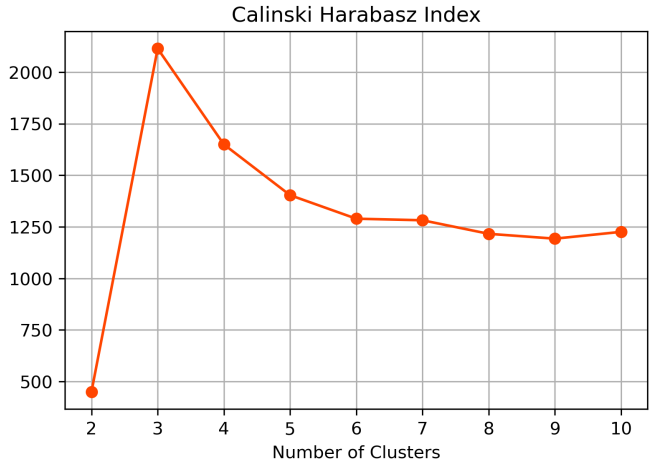

Figure 7: Visual determination of number of clusters through the Calinski-Harabasz index.

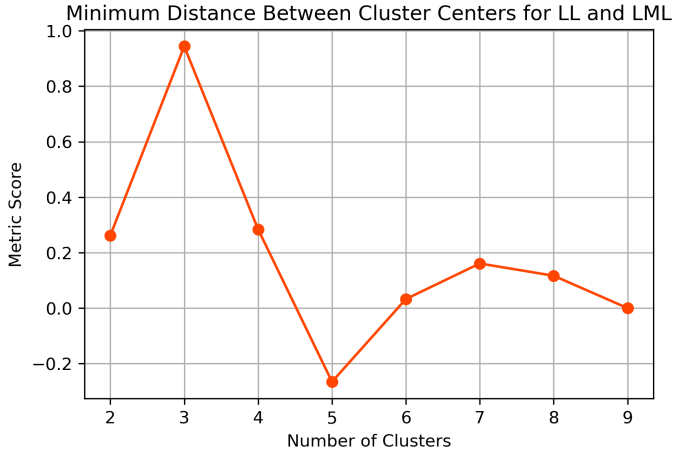

Figure 8: Visual inspection of minimum distance between cluster centroids for different numbers of k for LL and LML metrics.

Another drawback of the use of K-Means, as mentioned previously, is the handling of outliers. Since the method cannot classify points as noise or outliers, like DBSCAN, these are enclosed into clusters and are either located far away of the centroid, or in completely different cluster. In the first case, these points have the ability to move the centroid in a non representative position. Few such points can be seen in all clusters of Figure 9 although their ability to move the centroid was minimal.

3.5.2 Gaussian Mixture Models

The assumption behind GMM may be thought as that each cluster is generated from a mixture of Gaussian distributions. k is the predetermined number of desired clusters. GMM is a probabilistic model which is capable of capturing complex patterns in the data and assign them to a cluster based on a probability. The clusters produced by this method take hyper-ellipsoid shapes and have various orientations inside the hyper-space due to the Gaussian distributions (Vanem & Brandsæter, 2021). Similarly to K-Means, an expectation maximization approach is used.

The first step of the algorithm is initialization of parameters for the Gaussian components. These include mean (μ_k), covariance (Σ_k), and mixing coefficient (π_k) of each of the k com-

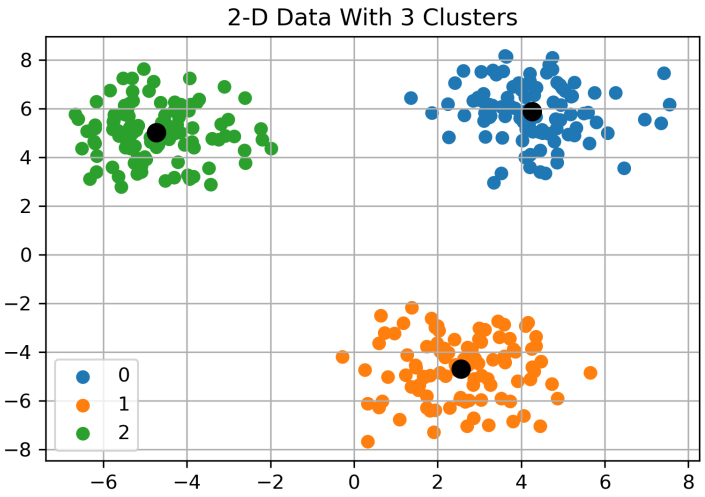

Figure 9: Representation of clustering with K-Means in sample dataset.

ponents. Additionally, $\sum_k \pi_k = 1$. A common approach is to initialize the means through K-Means. The density function is formulated as:

$$f(x) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x; \mu_k, \Sigma_k) \tag{10}$$

Following the initiation step, in the **E** (Expectation) step the probabilities of each data point belonging to a particular component are calculated. Let i be the data-point index and l the component that it belongs the probability is calculated as:

$$p_{il} = \frac{\pi_l \cdot \mathcal{N}(x_i; \mu_i, \Sigma_l)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x; \mu_k, \Sigma_k)} \tag{11}$$

In the **M**-step, or the Maximization step, the algorithm updates the parameters of each component based on the responsibilities calculated in the E-step. This iterative process refines the model parameters. Specifically, the M-step involves updating the mixing coefficients (π), means (μ), and covariances (Σ) as follows:

$$\begin{aligned} \pi_l &= \frac{1}{N} \sum_{i=1}^{N} p_{il} \\ \mu_l &= \frac{\sum_{i=1}^{N} p_{il} \cdot x_i}{\sum_{i=1}^{N} p_{il}} \\ \Sigma_l &= \frac{\sum_{i=1}^{N} p_{il} \cdot (x_i - \mu_l) \cdot (x_i - \mu_l)^T}{\sum_{i=1}^{N} p_{il}} \end{aligned} \tag{12}$$

The algorithm iterates between **E** and **M** steps until convergence is achieved. Convergence can be determined using various methods, such as monitoring changes in the log-likelihood function or assessing the stability of the model parameters across iterations. Once the model parameters stabilize or the maximum number of iterations is reached, the EM algorithm stops.

In order to find the optimal number of Gaussian components or clusters the model is ran repeatedly for several k until is found. The silhouette coefficient (has been discussed previously), the Bayesian Information Criterion (BIC), and the Akaike Information Criterion (AIC) are chosen as metrics to evaluate the different models. Another metric named "Integrated Complete

Likelihood", which is similar to BIC, has also been proposed by Vanem & Brandsæter (2021) but is not used in this study. Hence, the concepts behind BIC and AIC are analyzed in the following paragraphs.

Bayesian Information Criterion

BIC was developed as a metric for model selection. Models with lower score are preferred. This criterion works under the assumption that adding parameters usually increases maximum likelihood but may possibly result in over-fitting. Thus, a penalty term is introduced which increases together with the number of parameters in the model (Schwarz, 1978). BIC is mathematically defined as:

$$\mathrm{BIC} = \mathrm{k} \cdot \ln(\mathrm{n}) - 2 \cdot \ln(\hat{\mathrm{L}}) \tag{13}$$

where:

- $\hat{\mathrm{L}}$: the maximized value of the likelihood value of the model
- n: the number of data-points
- k: the number of parameters estimated in the model

Akaike Information Criterion

AIC is a similar metric to BIC. Presented in Akaike (1973) it estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model. The criterion rewards goodness of fit (as assessed by the likelihood function), but introduces a penalty term that is an increasing function of the number of estimated parameters. The penalty discourages over-fitting similarly to BIC. AIC is mathematically formulated as:

$$\mathrm{AIC} = 2 \cdot \mathrm{k} - 2 \cdot \ln(\hat{\mathrm{L}}) \tag{14}$$

where:

- $\hat{\mathrm{L}}$: the maximized value of the likelihood value of the model
- k: the number of parameters estimated in the model

Advantages & Disadvantages of GMM

GMMs have notable strengths in clustering. One strength is that they provide a probabilistic estimate of all data-points with each cluster. This is useful with ambiguous data points that fall at the border of two clusters. They produce non-spherical clusters with variable cluster shapes and sizes. Also, they are less sensitive to scaling and large datasets. However, GMMs struggle with non-normally distributed variables. Due to the Gaussian distribution assumption, the results are elliptical clusters. Adequate data per cluster is crucial for good clustering results. Also, the number of clusters needs to be specified in advance. GMM are sensitive to outliers and initialization conditions which results to relatively slower convergence rates than other methods (Ellis, 2023; Gao, 2012).

Simple Example of GMM Implementation

The same dataset as in K-Means is considered to showcase the abilities of GMM in this example. All methods need to identify the three clusters in the dataset. As seen in Figure 10, Figure 11, Figure 12 this is true. These figures verify the optimal number of clusters through the metrics and visually. The three clusters are clearly visible in the 2-D space. The clustering result is identical to K-Means, which is expected.

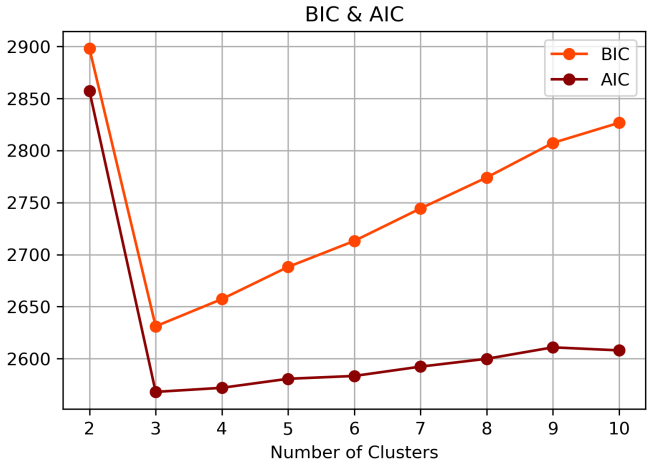

Figure 10: Visual determination of number of clusters through BIC & AIC scores with GMM.

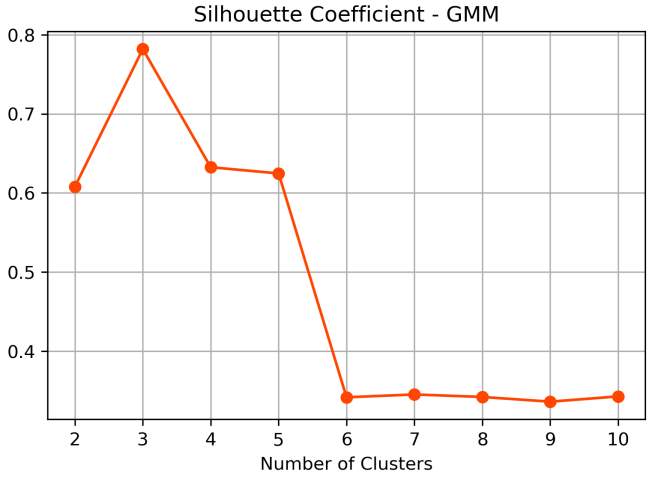

Figure 11: Visual determination of number of clusters through the silhouette coefficient with GMM.

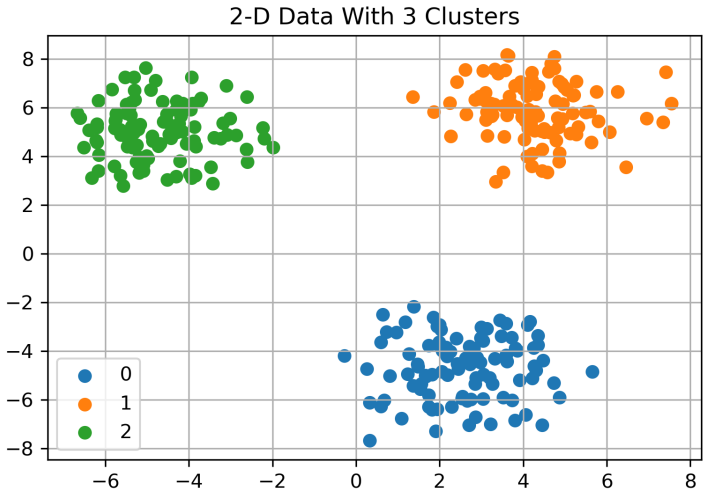

Figure 12: Representation of clustering with GMM in sample dataset.

3.5.3 Density-Based Spatial Clustering of Applications with Noise

DBSCAN, as the name suggests, is a density based clustering algorithm. Originally proposed by Ester et al. (1996), operates by considering regions of high data density as potential clusters, while identifying points in sparse areas as noise. This algorithm offers a robust solution for clustering spatial data, capable of discovering clusters of arbitrary shape and handling outliers effectively.

The algorithm of DBSCAN requires two critical components to be defined by the user: ε and MinPts. The size of neighborhood (ε) defines the maximum distance within which points are considered neighbors. The algorithm makes a distinction between core points, border points, and noise. Core points are those which have at least MinPts neighbors at distance less or equal than ε. Points categorized as bordering are within ε from a core point but have less than MinPts neighbors within the same distance. As noise are considered these points whose distance to their closest neighbor is greater than ε. Therefore, each cluster must have one or more boundary points and at least one core point. Results produced by DBSCAN have irregular hyper-shapes which differ from the more standardized hyper-spheres or ellipsoids produced by K-Means and GMM.

ALGORITHM 1: Pseudocode of Original Sequential DBSCAN Algorithm

```
Input: DB: Database
Input: ε: Radius
Input: minPts: Density threshold
Input: dist: Distance function
Data: label: Point labels, initially undefined
1  foreach point p in database DB do                          // Iterate over every point
2      if label(p) ≠ undefined then continue                  // Skip processed points
3      Neighbors N ← RangeQuery(DB, dist, p, ε)               // Find initial neighbors
4      if |N| < minPts then                                   // Non-core points are noise
5          label(p) ← Noise
6          continue
7      c ← next cluster label                                 // Start a new cluster
8      label(p) ← c
9      Seed set S ← N \ {p}                                   // Expand neighborhood
10     foreach q in S do
11         if label(q) = Noise then label(q) ← c
12         if label(q) ≠ undefined then continue
13         Neighbors N ← RangeQuery(DB, dist, q, ε)
14         label(q) ← c
15         if |N| < minPts then continue                      // Core-point check
16         S ← S ∪ N
```

Figure 13: DBSCAN algorithm pseudocode. Source: Schubert et al. (2017).

Parameter Selection & Metrics of Evaluation

Determining these two parameters of the method is not straightforward and in most scenarios domain knowledge is required (Vanem & Brandsæter, 2021; Schubert et al., 2017). The approach developed in this study predicted ε through the value of MinPts (Rahmah & Sitanggang, 2016).

The first step is to give potential values for MinPts. An initial approach of than value is given as MinPts $= 2 \cdot$ Dimensions (Ester et al., 1996; Sander & Ester, 1998). Based on that, more potential values are added, resulting in an array of MinPts. The next step is to find the optimal value of ε for each MinPts. k Nearest Neighbors (k-NN) is used for this task, an algorithm capable of identifying the k points closest to each point, where k is predefined by the user. Additionally, the distances of k closest points are also returned. Based on that,

k = MinPts is used. The mean distance of each point to its neighbors is calculated and then sorted in ascending order and plotted. In that k-distance graph, the optimal value of ε is at the point of maximum curvature Figure 14.

Subsequently, the dataset is clustered using DBSCAN with these hyper-parameter values. In each iteration, the silhouette coefficient is calculated as a metric of evaluation. The results are gathered at the end of this process. In order to find the best combination of hyper-parameters the following approach has been developed.

Each iteration of the algorithm is ranked based on the following criteria:

- Bigger Silhouette Coefficient
- Lowest rank of noise "cluster" in terms of points size
- Lowest number of noise points
- Highest number of points in clusters with more than $0.5\% \cdot$ (Total Data) points.

Following each iteration of the algorithm, the choice of hyper-parameters is ranked based on these four metrics. A final score is calculated based on the average of the positions each hyper-parameter combination got. The final parameter selection is based on the higher scoring combination.

Advantages & Disadvantages of DBSCAN

According to Gupta (2023), advantages of this method include the ability to cluster data of arbitrary shapes in comparison to K-Means which produces more spherical clusters. The robustness to noise and the ability to exclude them from clusters is also highlighted. Also, not requiring the number of clusters in advance is considered positive. Cons of DBSCAN include the algorithm's sensitivity to parameter selection, its ineffectiveness on datasets of varying density distributions and high computational cost.

Simple Example of DBSCAN Implementation

The same case as in K-Means is considered to demonstrate the ability of DBSCAN. It is noted that the method can cluster complex shaped data achieving good performance (Ester et al., 1996). After defining the optimum number of MinPts with the previously described scoring method, the point of maximum curvature of the k-distance graph is found through a computational method. The optimal value of MinPts in that example is found to be 21, which is significantly more than what Ester et al. (1996) suggested. The graph for the particular example is shown in Figure 14.

The algorithm managed to identify the three clusters. Points identified as noise are labeled as cluster -1. These points can also be treated as anomalies after further testing (Vanem & Brandsæter, 2021).

3.5.4 Self Organising Maps

Self Organising Maps are an unsupervised learning technique. They are a type of ANN which has a feed-forward structure with a single computational layer. SOMs are based on competitive learning and topological organization principles, and they are commonly used for tasks like clustering, visualization, and feature extraction (Akinduko & Mirkes, 2012). The algorithm was initially proposed in Kohonen (1982) and its methodology is explained in the text that follows.

An SOM consists of a grid of neurons, usually in 1D or 2D formats (Akinduko & Mirkes, 2012). Each neuron in the grid is associated with a weight vector of the same dimensionality as

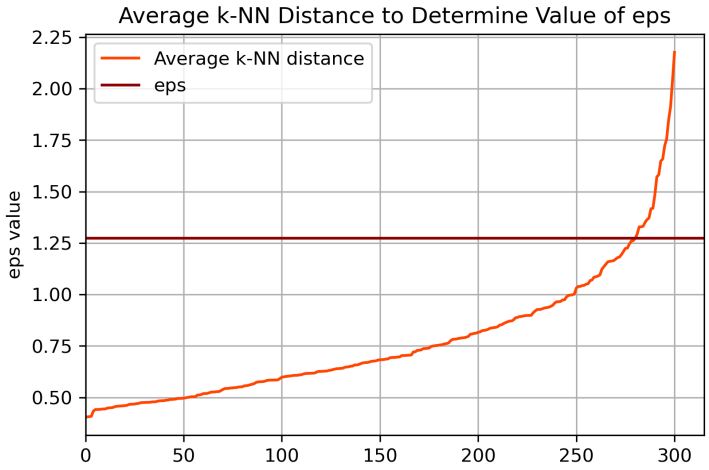

Figure 14: Average k-NN distance graph to specify optimal value of ε in DBSCAN.

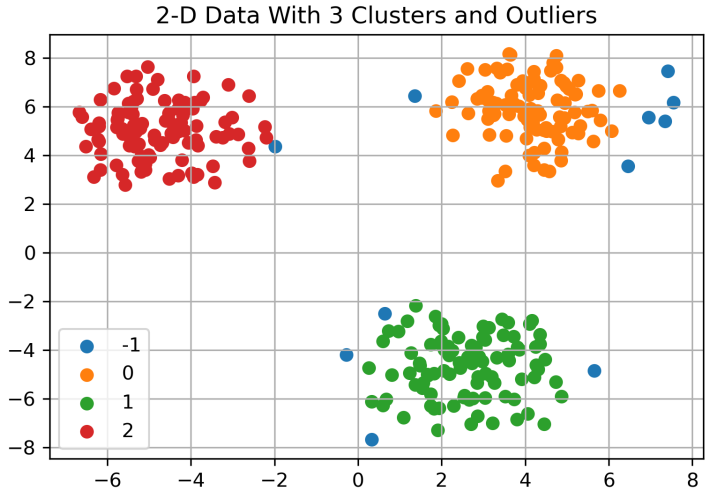

Figure 15: Representation of clustering with DBSCAN in sample dataset.

the input data. The goal of the SOM is to map the high-dimensional input data onto this grid in a meaningful way (Günter & Bunke, 2002).

The number of neurons is initially estimated as a function of the total observations (n). In a 1D map, is equal to $M = \text{round}(5 \cdot \sqrt{n})$. In a 2-dimensional example, keeping in mind that the required neuron number is still equal to M, the grid size should be $\sqrt{M} \times \sqrt{M}$ (Tian et al., 2014).

The training process involves iteratively adjusting the weights of the neurons to match the input data distribution. The key idea is that neurons that are spatially close to each other in the grid will respond similarly to similar input patterns. During that training phase, the SOM adapts its weights to the input data distribution in such a way that nearby neurons respond to similar inputs.

The training process, also shown in Figure 17, runs by iterating through a few steps. First, the map's neurons weights are initialized randomly or via a predetermined method, such as through Principal Components. By utilizing that methodology one may ensure faster results (Akinduko & Mirkes, 2012). Secondly, a data point from the training dataset is selected and the distance between it all the neurons is calculated (usually Euclidean distance is used). The neuron's weight vector which was found to be the closest to the data point is updated. This

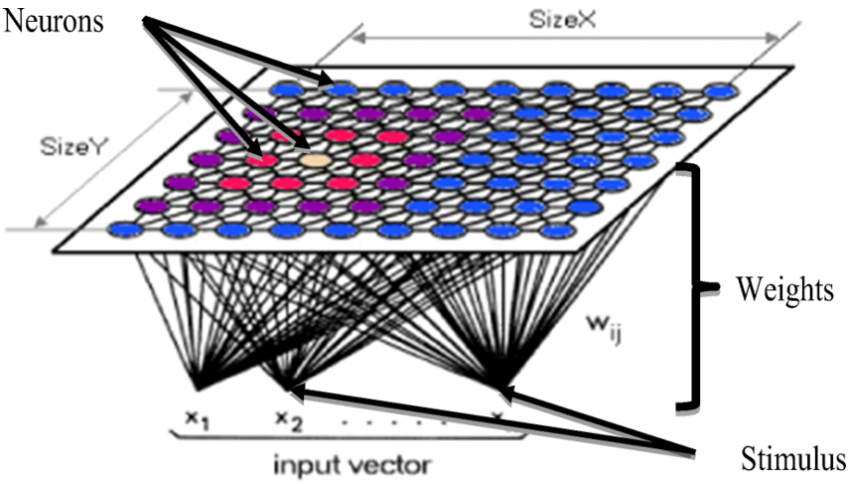

Figure 16: Schematic representation of a SOM. Source: Alia et al. (2020).

process moves the winning neuron and its closest neurons towards that data point. This winning neuron is called Best Matching Unit (BMU). The BMU's weights are updated significantly more than those of the other neurons. The learning rate and the neighborhood size are adjusted over time, by being gradually decreasing as training progresses. These steps are repeated for a predefined number of iterations or until convergence (Kohonen, 1982; Akinduko & Mirkes, 2012; Günter & Bunke, 2002).

The weight of neuron j is updated based on input x with i dimensions and learning rate η by the following formula:

$$w_{ji}^{new} = w_{ji}^{old} + \eta \cdot h_{j^*j} \cdot (x_i - w_{ji}^{old}) \tag{15}$$

with j^* being the BMU and j another neuron.
h_{j^*j} is the neighborhood function centered at the BMU and j. If it is Gaussian it is expressed as:

$$h_{j^*j} = \exp\left(\frac{d(j^*,j)^2}{2 \cdot \text{radius}^2}\right) \tag{16}$$

$d(j^*,j)$ is the distance between neurons j^* and j, and radius is the current neighborhood radius.

som-algorithm
(1) **input:** a set of patterns, $X = \{x_1, \ldots, x_N\}$
(2) **output:** a set of prototypes, $Y = \{y_1, \ldots, y_M\}$
(3) **begin**
(4) initialize $Y = \{y_1, \ldots, y_M\}$ randomly
(5) **repeat** select $x \in X$ randomly
(7) find y^* such that $d(x, y^*) = \min\{d(x, y)|y \in Y\}$
(8) **for** all $y \in N(y^*)$ **do**
(9) $y = y + \gamma(x - y)$
(10) reduce learning rate γ
(11) **until** termination condition is true
(12) **end**

Figure 17: SOM algorithm pseudocode. Source: Günter & Bunke (2002).

An additional input parameter required to train a SOM is the number of epochs or number of iterations that the algorithm will go through during training. One may monitor the improvements of the map in the training process through two metrics: Quantization Error and Topographic Error.

Quantization Error

The quantization error refers to the discrepancy between a data point and its corresponding BMU. It measures how well each data point is represented by its nearest neuron on the map. The metric is calculated by using the Euclidean norm. x is the input vector and w is the weight vector of the BMU.

$$\mathrm{Q} = ||\mathrm{x} - \mathrm{w}|| \tag{17}$$

The calculation of quantization error involves finding the mean distance between the sample vectors and the cluster centroids that represent them. In the context of the Self-Organizing Map (SOM), these cluster centroids correspond to the weight vectors. A lower quantization error indicates that the model is doing a good job of representing data points close to their respective BMUs. This may be simply achieved by increasing the number of nodes although this may lead to distortion of the map's topology (Pölzlbauer, 2004).

Topographic Error

The topographic error measures the extent to which the topology of the original data is preserved on the grid. It quantifies the cases where the BMUs of neighboring input vectors are not mapped to neighboring neurons on the grid. It reflects how well are the spatial relationships maintained between data points. This process unfolds in the following manner: Across all data samples, the best-matching unit and the second-best-matching unit are identified. When these units are not neighboring on the map lattice, it is deemed an inconsistency. The cumulative inconsistency is subsequently adjusted to fit within a scale of 0 to 1, where 0 signifies impeccable preservation of topology.

The topographic error is mathematically expressed as:

$$\mathrm{T} = \frac{\text{Number of BMUs without topographic neighbors}}{\text{Total number of data points}} \tag{18}$$

Similarly to the quantization error, with a lower topographic error is shown that the SOM is maintaining the neighbor relationships between BMUs that were close in the original high-dimensional space (Pölzlbauer, 2004).

Advantages & Disadvantages of SOM

The most advantageous aspect of SOMs lies in their simplicity and ease of comprehension. Their operational logic is straightforward: proximity and "connection" within the map indicate similarity, while gaps represent dissimilarity. Furthermore, they exhibit remarkable efficacy. Demonstrably adept at data classification, they can also be easily evaluated for their quality. This allows for the quantification of map effectiveness and the strength of object similarities.

A notable challenge associated with SOM is acquiring suitable data. Generating a map requires having values for every dimension of each sample element. Another drawback of SOM is the difficulty in joining similar groups of data together, resulting in two or more clusters for almost identical points. The last disadvantage is that SOM are considered to be computationally expensive as a methodology (Kaski, 1997).

Simple Example of SOM Implementation

Similarly to what previously presented, the same example is considered to demonstrate how SOMs work. A 2D 2×2 map is employed even though the dataset contains three distinct clusters. The selection of the particular map predicts that the algorithm might detect four clusters instead of three.

In this example it is chosen to iterate thirty times through the dataset to train the algorithm. By monitoring the plot which contains the quantization and topographic errors, it is clear that this number of iterations results in good results. As shown in Figure 18, the quantization

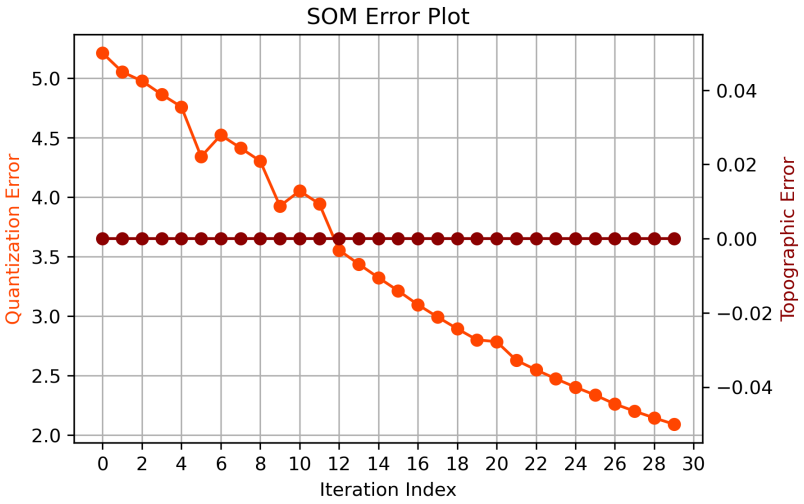

Figure 18: SOM algorithm quantization and topographic errors.

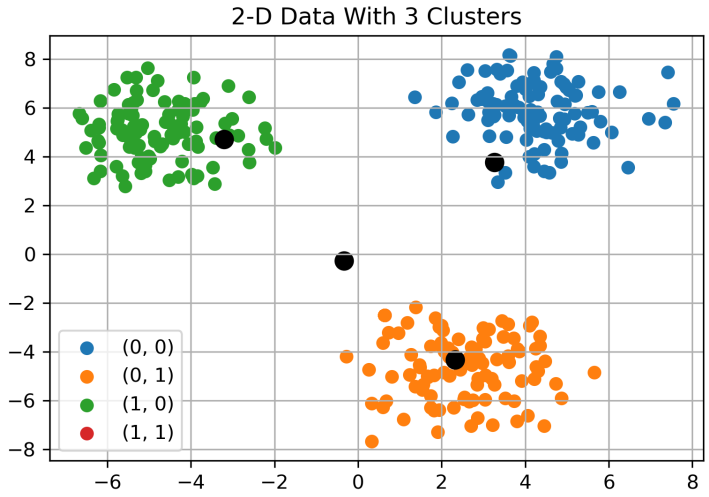

Figure 19: Representation of clustering with SOM in sample dataset. Clusters are named after their assigned neurons position in the map. Black points represent the four neurons.

error decreases and is equal to less than half of its original value at the 30th iteration. On the contrary, the topographic error is equal to zero at all iterations and does not provide information. this occurs due to the well separated clusters that, result in an almost perfect map without topographic errors (Cabanes & Bennani, 2010). Additionally the low dimensionality of the dataset might also be responsible for that.

In Figure 19 can be seen that the SOM algorithm managed to detect the three clusters without utilizing all four neurons. This is a proof of the method's capabilities. If a 3×1 map was used instead, the expected result would have been similar to what shown in Figure 19 but without the 4th neuron.

When taking into account the neuron positions, it is visible that they shape a cluster around them. In the case of (0, 0) and (1, 0) the neurons are located at the border point of these clusters. Although nodes are not equivalent to the cluster centroids of K-Means, in an ideal scenario they should be be position close to the centroid. In this particular case, their resulting positions after 30 training iterations are moderately out of place. Potential reasons include wrong estimation of learning rate or neighborhood radius, or excessive training that led to over-fitting.

4 Methodology

The proposed framework aims to detect operational anomalies from a vessel's ME in an efficient manner. The study employs several unsupervised learning techniques to conduct the anomaly detection task. This study's methodology can be summarized in two steps. Data preparation is the first major step. After that, the dataset is ready to be used as input in the anomaly detection models, which is the second step. Several models are examined utilizing methods such as K-Means clustering, DBSCAN, clustering with GMM, and SOMs. All of the implemented methods theoretical framework and background are described in the previous section.

Another step which is performed and refers mostly to the discussion section of the thesis is the simulation of anomalies. This is performed in order to verify the performance of the anomaly detection algorithms in almost real operational scenarios.

The structure of this section is as follows: presentation of all methodologies relevant to the data preparation phase followed by a description of the way ML are transformed to anomaly detection models. Then, follows a section which concerns anomalies typology and the methodology based on which anomalies will be simulated.

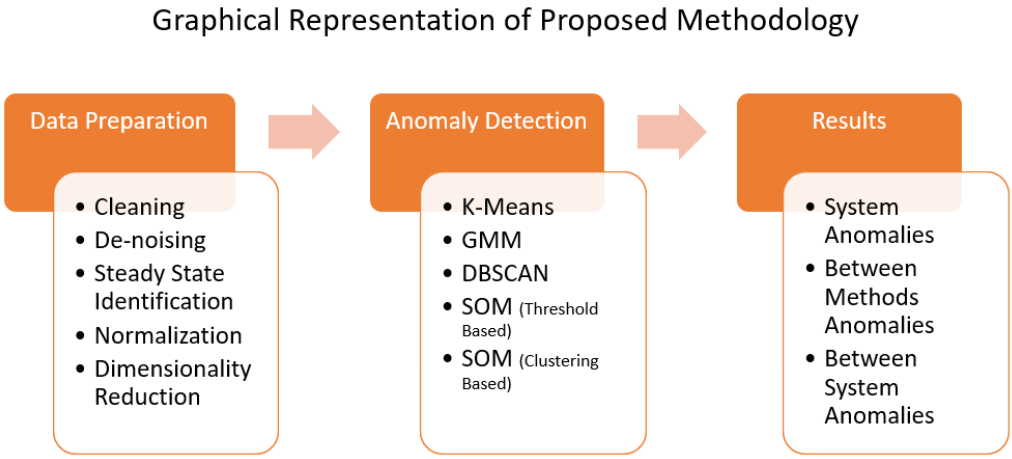

Figure 20: Graphical representation of proposed methodology.

4.1 Data Preparation

This section deals with the initial treatment of the data up until the point when they are ready to be imputed into the model. The purpose is to convert them from raw form in one which extracts the most out of them. Øyvind Øksnes Dalheim & Steen (2020b) consider the extensively used term "data pre-processing" to be highly related with "data preparation". Data preparation encompasses a broader set of activities that include data pre-processing but also extends beyond it. It involves the entire process of collecting, cleaning, transforming, and organizing the data to make it ready for analysis.

Several people have tried organize data preparation in steps. Pyle (1999, p. 112) considered eight which are listed below. Not all of them apply to the current study, since the source is written based on data preparation for a broader field of applications.

1. Accessing the data
2. Auditing the data
3. Enhancing and enriching the data

4. Looking for sampling bias
5. Determining data structure
6. Building the Prepared Information Environment
7. Surveying the data
8. Modeling the data

Masmoudi et al. (2021) summarized the process into three steps, data cleaning, reduction, and normalization.

This study follows the approach of Masmoudi et al. (2021) since both are in a similar field of application. Some additions and modifications to that framework are presented and described in the text that follows. Data preparation is structured in seven steps. The first step is identical to what Masmoudi et al. (2021) considered. The second is data de-noising and the third involves steady state identification of critical parameters. The fourth step is data normalization. At this point, the features of the dataset are categorized in subsystems (fifth step) and dimensionality reduction (sixth step) is performed within each subsystem. The seventh and final step is the train-test data split.

4.1.1 Data Cleaning

In this step the raw dataset goes through the first reduction process. Issues being addressed are the removal of unwanted, duplicate features, and abnormal values. In this context, "abnormal" refers to the existence of values in parameters that are physically impossible to occur or contradictory (such as negative values in Shaft Power).

The decision to implement an imputation algorithm may be examined by considering the quantity and frequency of missing values (NaN). The need to deploy such algorithm comes after assessing their occurrence patterns. An imputation method previously used in the maritime sector is with the use of ANN (Lazakis et al., 2018; Martinez-Luengo et al., 2019). Outside the sector, several methods have been utilized, such us MSE, Support Vector Regression, and Simple Linear Regression (Richman et al., 2009).

4.1.2 Data De-noising

This step is performed in order to eliminate noise of certain sensor signals. Several methods are examined to handle this task. Noise reduction algorithms are classified in three categories based on how they treat the data to reach the desired outcome.

- Filtering: Estimation of value at a given point, t, by utilizing the data preceding t.
- Smoothing: Estimation of value at point t by incorporating both preceding and subsequent data.
- Prediction: Signal value prediction at point t+1 or beyond by observing the data prior to t.

Smoothing may yield superior results in terms of de-noising, it necessitates access to past and future samples of the signal, which may not always be feasible. Smoothing improves data quality by replacing the noisy and irregular signal with a smoothed version that likely provides a more accurate description of the observed phenomena (Pawel & Smyk, 2018).

Three smoothing and one prediction method that have previously been examined in similar applications are explored in order to conclude which fits best to the data. These include:

1. Simple Moving Average, used by Pawel & Smyk (2018).

2. Exponentially Weighted Moving Average (prediction), used by Velasco-Gallego & Lazakis (2022d).
3. Savitzky-Golay Filter, used by Wen et al. (2023); Pawel & Smyk (2018).
4. Gaussian Filter, used by Pawel & Smyk (2018).

Simple Moving Average

Simple moving average is the simplest way to smooth data containing noise. A centered moving average is examined in this case, meaning that the average will be calculated at each window's center. Mathematically this is expressed as:

$$\mathrm{SMA(i)} = \frac{1}{\mathrm{w}} \cdot \sum_{\mathrm{j=i-\frac{w}{2}}}^{\mathrm{i+\frac{w}{2}}} \mathrm{x_j} \tag{19}$$

where,

- w: the moving window size
- $\mathrm{x_j}$: average of the j-th window

Exponentially Weighted Moving Average

The EWMA is a prediction based method used to de-noise the dataset. It gives more weight to recent observations while exponentially decreasing the weights of older observations. The EWMA at a given time point is calculated as a weighted sum of the current value and the previously calculated EWMA, with weights determined by a smoothing factor. A higher smoothing factor gives more emphasis to the current observations, making EWMA more responsive. Mathematically, it is expressed as follows:

$$\mathrm{EWMA(t)} = \alpha \cdot \mathrm{x(t)} + (1 - \alpha) \cdot \mathrm{EWMA(t-1)} \tag{20}$$

Where,

- α: smoothing factor, $0<\alpha<1$
- x(t): current value at time t

Savitzky-Golay Filter

Savitzky & Golay (1964) developed a filter which smooths data by applying a polynomial function to a window of points. The approach is based on least-squares. This aims to minimize noise and keep the important features in the smoothed representation. To apply the filter, window size and polynomial order must be given.

$$\mathrm{SAVGOL(i)} = \sum_{\mathrm{k=0}}^{\mathrm{n}} \mathrm{a_k i^k} \tag{21}$$

Where,

- n: polynomial degree
- a: vector of coefficients which is calculated by minimizing the error

$$\mathrm{E} = \sum_{\mathrm{i=-M}}^{\mathrm{M}} (\mathrm{SAVGOL(i)} - \mathrm{x_i})^2 \tag{22}$$

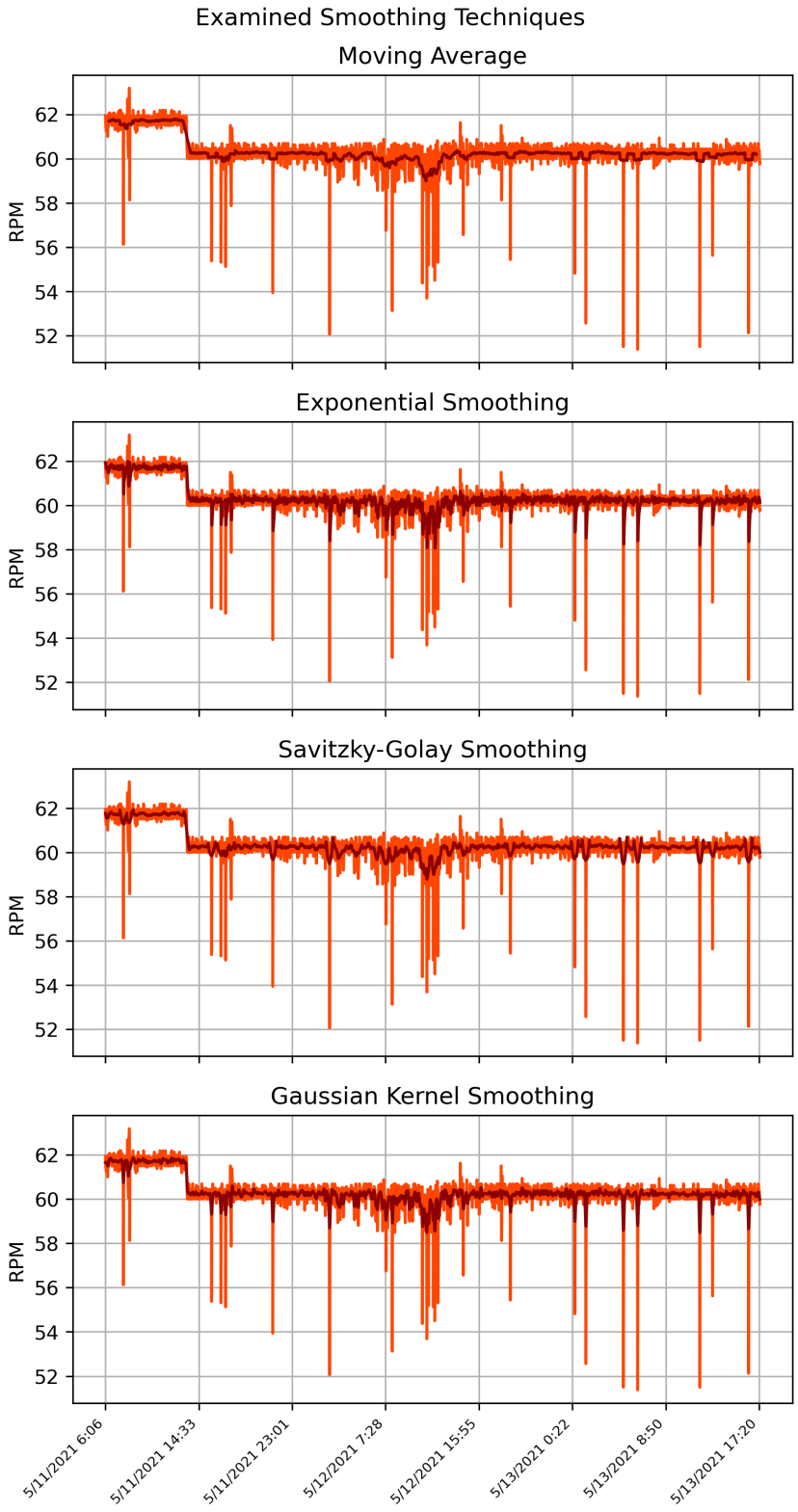

Figure 21: Examined Smoothing Techniques on Sample Dataset.

Gaussian Filtering

This smoothing algorithm is based on the normal/Gaussian distribution. The input data are convoluted with a Gaussian kernel to achieve smoothing to a level determined by the standard

deviation of the distribution.

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{23}$$

Where,

- σ: standard deviation
- μ: average

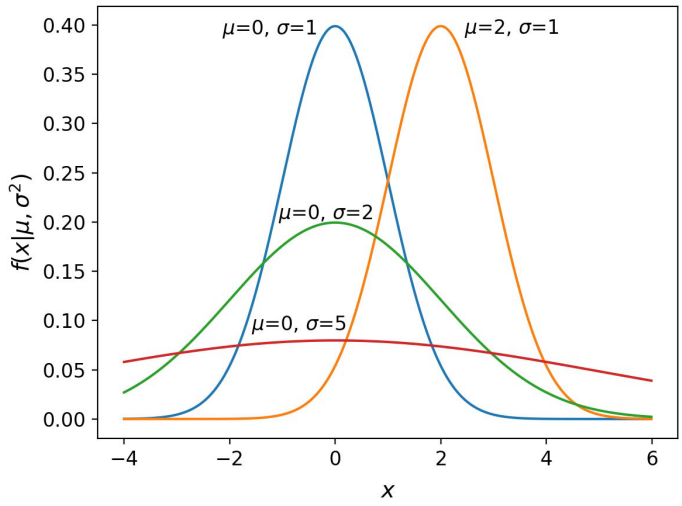

Figure 22: Impact of parameters σ and μ on shape of Normal distribution probability density function.

4.1.3 Steady State Identification

The aim of this study is to conduct an operational anomaly detection analysis in sensor signals from a ME. To do that, the engine should be running on steady operating conditions. Thus, except of the removal of not physically possible values from the data (first step), there is a need to remove idle and transient states (Velasco-Gallego & Lazakis, 2022a,c). Only two studies have been found in the maritime sector which deal with this issue. The first implemented image generation through first order Markov chains and connected component analysis to identify steady states. The method was compared to other frequently used approaches such as K-Means clustering and GMM with Expectation Maximization algorithm and delivered superior results (Velasco-Gallego & Lazakis, 2022d). The second method was developed based on a rolling window approach and the assumption that steady state can be modeled by a linear trend model (Øyvind Øksnes Dalheim & Steen, 2020a). The algorithm is to be applied individually in critical variables of the examined system. A modified version of this algorithm is used in this study. The original approach is described below.

There are four inputs needed: vector containing the data, rolling window size, significance level, and steady state probability threshold. The algorithm is designed in such way that the rolling windows have maximum overlap.

In each window, the model describing the data can be expressed as a linear function.

$$z_t = b_0 + b_1 t + a_t \tag{24}$$

With,

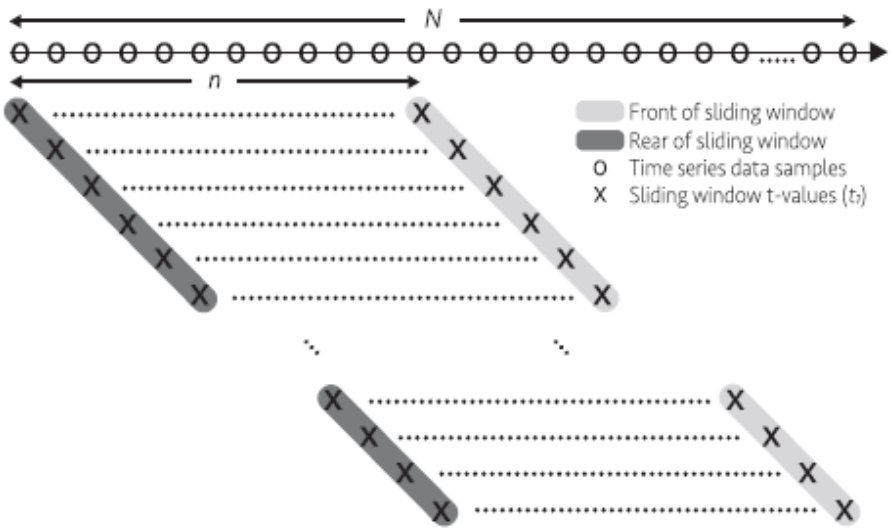

Figure 23: Illustration of the steady state identification algorithm. Each point is a member of more than one rolling windows. The probability of a point being in a steady state is calculated from the times it was participating in a steady window. Source: Øyvind Øksnes Dalheim & Steen (2020a).

- a_t: zero mean white noise with constant variance ${\sigma_a}^2$
- b_0: intercept of line
- $b_1 t$: linear drift component formed by the slope b_1 and relative time t inside the window.

b_1 and b_0 are estimated by a linear regression model within each window.

$$b_1 = \frac{\sum t \cdot z_t - \frac{1}{n}\sum t \cdot \sum z_t}{\sum t^2 - \frac{1}{n}(\sum t)^2} \tag{25}$$

$$b_0 = \frac{1}{n} \cdot \left(\sum_{t=1}^{n} z_t - b_1 \cdot \sum_{t=1}^{n} t \right) \tag{26}$$

By calculating b_1 and b_0 the residuals can be estimated as:

$$res(t) = z_t - b_1 \cdot t - b_0 \tag{27}$$

Then the standard deviation of the white noise is estimated

$$\sigma_a = \sqrt{\frac{1}{n-2} \sum_{t=1}^{n} res(t)^2} \tag{28}$$

The standard deviation of the estimated slope is calculated as

$$\sigma_{b1} = \sqrt{\frac{\sum_{t=1}^{n} res(t)^2}{(n-2) \cdot \sum_{t=1}^{n} (t - \bar{t})^2}} = \frac{\sigma_a}{\sqrt{\sum_{t=1}^{n} (t - \bar{t})^2}} \tag{29}$$

To decide if a window presents steady behaviour or not the t-value is calculated and compared to t-critical which follows a Student's distribution that depends on the significance level α and the degrees of freedom n-2.

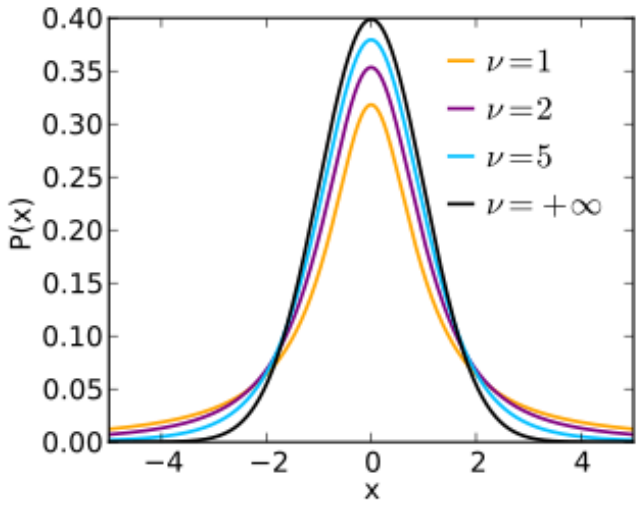

Figure 24: Impact of degrees of freedom (ν) on Shape of Student's t distribution probability density function. t-critical is derived from the inverse cumulative distribution function.

$$t = \frac{b_1}{\sigma_{b1}} \tag{30}$$

The null hypothesis is formulated as: The window displays steady state behaviour if

$$|t| < t_{a/2,n-2} \tag{31}$$

meaning that the slope component b_1 is not statistically significant. If the hypothesis is accepted the window is flagged as S_t=1, alternatively S_t=0. Each point is participating in more than one windows, due to nature of the rolling window algorithm. The total number of tests performed depends on the length of data and window and it is expressed as:

$$(\text{Number of Tests}) = (\text{Length of Data}) - (\text{Window Length}) + 1 \tag{32}$$

The probability of a point to have steady behaviour is calculated based on the times it participated in a window which displayed steady behaviour over the total times this point participated in windows. Then it is compared to the steady state probability threshold which was set as input arguments in the algorithm.

The modification which is implemented in the initial algorithm include a second steady-state test. Another threshold value for the slope b_1 is used to control missed and "false positive" steady state regions in the time series. If b_1 of a window is over the maximum allowable value, which is given as an input and is the same for all windows, then the window is categorized as un-steady.

After the end of the double check process for each critical variable steady state identification, a time-point is categorized steady if all critical variables are considered to show steady behaviour there.

4.1.4 Train-Test Data Split

In order to evaluate the performance of the ML anomaly detection models which are implemented in this study, the dataset should be split in Train and Test data. The previous data preparation steps have been applied to the total dataset. The split was decided to be 75-25% meaning that 75% of the rows are kept for training purposes of the ML models and the other 25% for testing. The operation selected randomly 75% of the dataset. The motive behind this

action is capturing as much as possible of the ME/vessel operational conditions in the training phase in order to eliminate false positive anomalies in the test period (Vanem & Brandsæter, 2021).

4.1.5 Data Normalization

When data are normalized the efficiency and accuracy of the ML algorithms increases (Masmoudi et al., 2021). Also, to employ other data preparation methods the data must be normalized. Two techniques were examined by Masmoudi et al. (2021), Min-Max Normalization and Z-score Normalization. In the Min-Max approach the data are scaled based on the minimum and maximum value of the feature. The scaled data range is [0, 1]. In Z-score normalization the data of the feature are scaled based on their mean and standard deviation. It is decided to follow Z-score normalization.

$$z = \frac{x - \mu}{\sigma} \tag{33}$$

4.1.6 Data Division into Sub-systems

Based on what has been previously presented and considering the application of the anomaly detection models which are to be created in this study, some systems of those may not be included in the analysis either due to not being relevant to this study or due to absence of data. It was found that researchers in anomaly detection consider different subsystems in their studies depending on their purpose, as seen in Table 1. Cai et al. (2017); Gharib & Kovács (2022) used these subsystems for fault diagnosis by implementing ML and data analysis techniques respectively, whereas Nahim et al. (2015) divided the engine in these subsystems to detect faults through a thermodynamic model.

Table 1: ME subsystems according to different studies.

Cai et al. (2017)	**Gharib & Kovács (2022)**	**Nahim et al. (2015)**
Fuel System	Fuel System	Combustion & Emissions
Lubrication System	Lubrication System	Lubrication System
Cooling System	Cooling System	Cooling System
Intake & Exhaust System	Air Supply System	Air System
	Exhaust System	Injection System

Due to the complexity and number of systems in a ME, it is believed that by examining each subsystem separately, the algorithms' overall performance will increase. The proposed methodology is to detect anomalies in each subsystem, then processing them in parallel and explore interactions between them. For example, it is expected that an anomaly in the Fuel system will trigger an anomaly in the Intake & Exhaust system and vise versa.

4.1.7 Dimensionality Reduction

This step is considered to be a second data reduction phase. To enhance the performance of the ML algorithms which will be used, the dataset's dimension must decrease. It should be noted that the anomaly detection methods are capable of running in big datasets without any issue.

The need to use a method of dimensionality reduction emerges when high correlation between the features exists. Vanem & Brandsæter (2021); Masmoudi et al. (2021) considered PCA to perform this task. Factor Analysis was also used by Masmoudi et al. (2021). PCA works by converting correlated features to linearly new uncorrelated principal components. This results

in new orthogonal linear combinations. The method tries to capture as much of the total system's variance in the first principal component. Each of the next components captures again the maximum possible percentage of variance. One can inspect variance captured by each component and the total variance explained by a certain number of components visually through a Scree plot. A Scree plot includes the variance explained by a principal component and the cumulative explained variance of the system over the number of principal components. The selection of number of components can be made through this plot.

PCA works by calculating the covariance matrix of the dataset. Then, the eigenvalues and eigenvectors of this matrix are found. Given a known number of required principal components, k, the eigenvectors which correspond to the top k eigenvalues multiplied by the data matrix give the reduced dataset. A simple example is given below.

Simple Example of PCA Implementation

A dummy dataset containing five students and their grades in Math, English, and Art is given. This dataset consists of three dimensions and there is a need to reduce them by one with the use of PCA. The covariance matrix is calculated as follows:

Table 2: Student grades dataset for PCA demonstration.

Student	Math	English	Art
1	90	60	90
2	90	90	30
3	60	60	60
4	60	60	90
5	30	30	30

$$\mathrm{cov}(\mathrm{X},\mathrm{Y}) = \frac{1}{\mathrm{n}} \cdot \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{x} - \overline{\mathrm{x}}) \cdot (\mathrm{y} - \overline{\mathrm{y}}) \tag{34}$$

where,

- x, y: members of the X, Y variables
- $\overline{\mathrm{x}}$, $\overline{\mathrm{y}}$: mean of X, Y variables
- n: number of members

$$\mathrm{cov} = \begin{bmatrix} 504 & 360 & 180 \\ 360 & 360 & 0 \\ 180 & 0 & 720 \end{bmatrix} \tag{35}$$

The eigenvalues are the roots of the characteristic equation of the covariance matrix.

$$\det(\mathrm{A} - \lambda \mathrm{I}) = 0 \implies \lambda_1 \approx 44.82,\ \lambda_2 \approx 629.11,\ \lambda_3 \approx 910.07 \tag{36}$$

The eigenvectors, v, are calculated for each eigenvalue as:

$$(\mathrm{cov} - \lambda \mathrm{I}) \cdot \mathrm{v} = 0 \implies \mathrm{v}_1 = \begin{pmatrix} -3.75 \\ 4.28 \\ 1.00 \end{pmatrix}, \mathrm{v}_2 = \begin{pmatrix} -0.50 \\ -0.68 \\ 1.00 \end{pmatrix}, \mathrm{v}_3 = \begin{pmatrix} 1.05 \\ 0.69 \\ 1.00 \end{pmatrix} \tag{37}$$

Finally, based on the two largest eigenvalues and their eigenvectors, the reduced dataset containing two principal components can be calculated as

$$y = A \cdot W = \begin{bmatrix} 90 & 60 & 90 \\ 90 & 90 & 30 \\ 60 & 60 & 60 \\ 60 & 60 & 90 \\ 30 & 30 & 30 \end{bmatrix} \cdot \begin{bmatrix} 1.06 & -0.50 \\ 0.69 & -0.68 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 226.8 & 4.2 \\ 187.5 & -76.2 \\ 165.0 & -10.8 \\ 195.0 & 19.2 \\ 82.5 & -5.4 \end{bmatrix} \tag{38}$$

Where,

- y: reduced dataset matrix
- A: raw dataset matrix
- W: matrix of the eigenvectors with the two largest eigenvalues

This matrix can now be used as input to conduct the needed analysis. The reduced dimensions will allow for better performance and increased efficiency of the ML algorithms while there is minimal information loss.

4.2 Anomaly Detection Models

In this study, unsupervised learning techniques are proposed to handle the anomaly detection task. This is done mainly due to the unsupervised nature of scenarios considered in the maritime sector (Velasco-Gallego & Lazakis, 2022c).

Furthermore, the examined methods utilize K-Means clustering, DBSCAN, GMM clustering, and SOMs. The three first models are categorized as clustering methods, whereas the fourth is an Artificial Neural Network ANN.

The concept followed in this study is to obtain clusters through the training process and then predict to which of the predefined clusters each test data-point belongs to. The goal is to identify if the new points belong to these clusters or not. If a point belongs to a cluster, it is treated as normal, and if not, it is considered to be an anomaly. The method used to determine if a test data point presents normal or anomalous behavior varies between the methods, due to their typology. The specific method used in each case is presented in the paragraphs that follow.

4.2.1 Anomaly Detection Framework for Data Points

Due to the split of the ME into four sub-systems, the task of classifying a point as anomaly is considered more complex than the case with no sub-systems due to the following reasons. First reason is the presence of four datasets containing the anomalies, rather than one. Secondly, is the requirement of handling these four datasets and finding common anomalous points between them.

To flag a point as an engine anomaly, it must have already been flagged in at least one system. If the same time-point is found to be present in more than one anomalies dataset, then the likelihood of that point to not be a false-positive anomaly increases.

4.2.2 Anomaly Detection with K-Means Clustering

In the case of K-Means, following the training phase valuable information about each cluster in the model is collected. This information includes all the points which are assigned to each cluster and each cluster's centroid. Based on these, the distance of every point to its assigned cluster center is calculated and stored in an array.

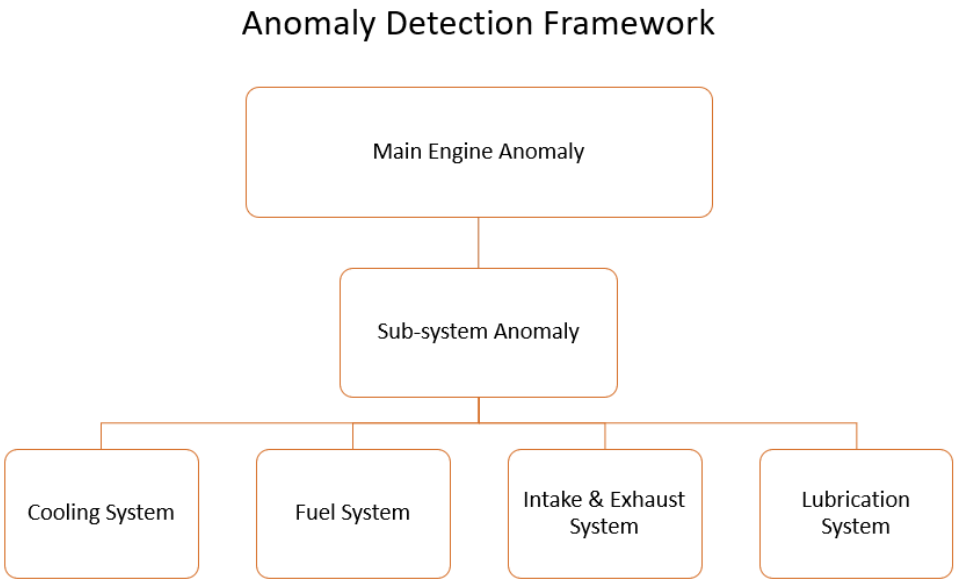

Figure 25: Anomaly detection framework for data points.

The principle under which anomaly detection is conducted is stated as: If a point's distance from its assigned cluster center is more than a predetermined value then it considered anomalous.

Considering that K-Means produces hyper-spherical cluster shapes, the first attempt is to assume the maximum distance of the training points from the center of the cluster as a form of radius. This distance is then used as a threshold value. By predicting to which cluster each test point is assigned to, the distances from the centers are calculated. The points where these the distances exceed the threshold are classed as anomalies.

By following this "cluster radius" approach underestimation of anomalies occurs since outliers, which have not been successfully removed from the test data after the preparation phase, affect the "radius". K-Means method must include all given points into clusters and, as a result, outliers are also part of clusters. Due to their lower frequency, these points lay far away from cluster centroids. Though, when searching for the maximum distance from a centroid it is probable that this refers to an outlier. This results in a non representative threshold value being used and outliers-anomalies within the test dataset being be regarded as normal.

To combat this issue a second approach is developed. Instead of using the maximum distance from the centroid as a threshold the 98th percentile distance is used. This approach aims to combat the effect of outliers present in the training set and give more realistic anomaly detection result.

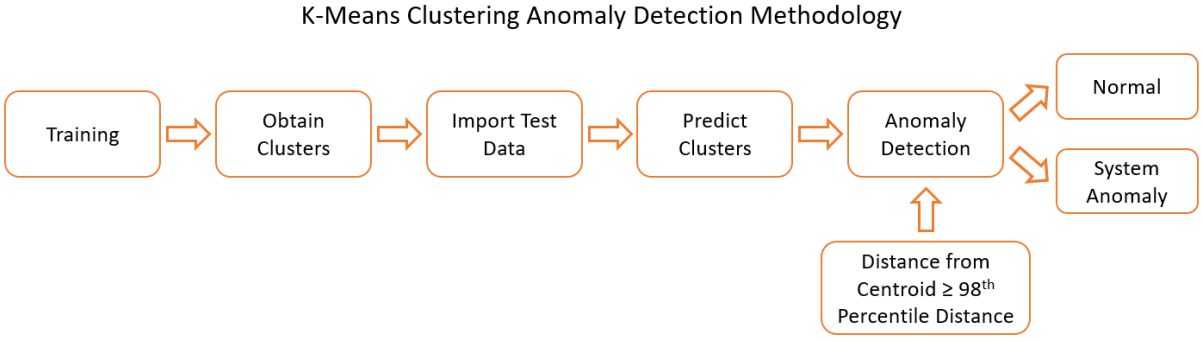

Figure 26: K-Means clustering anomaly detection methodology.

4.2.3 Anomaly Detection with GMM

GMM is a probabilistic clustering method. When training the model, cluster assignments are performed based on probabilistic method. A point is assigned to a certain cluster if the probability of that point being a member of this cluster is higher than the probabilities of being member of other clusters.

By applying the same rationale to the test data, a prediction is derived, indicating the clusters to which these points are affiliated. Subsequently, the array of probabilities is derived and scanned for anomalies. To classify a test point as normal, the probability corresponding to its assigned cluster should be higher than a predetermined value. If that is not true, the point is considered to be an anomaly.

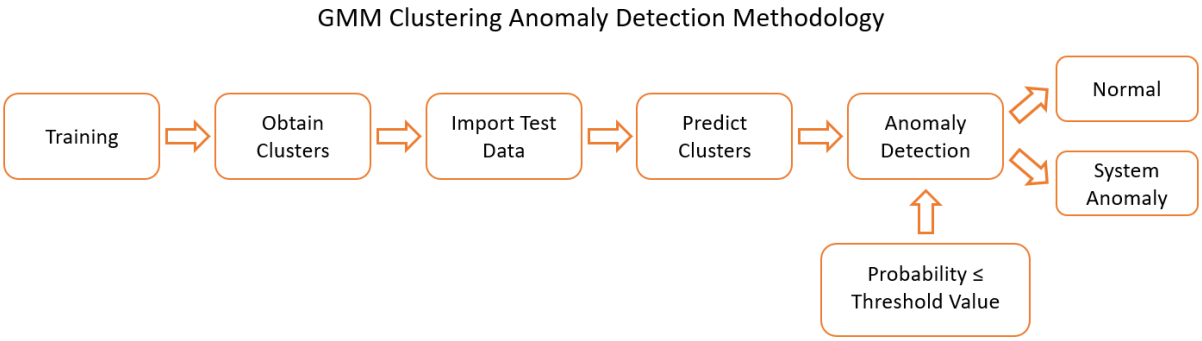

Figure 27: GMM anomaly detection methodology.

4.2.4 Anomaly Detection with DBSCAN

DBSCAN requires special treatment due to the ability of this method to identify noise or outlier points in the training phase. This is expected to assist in delivering superior clustering results compared to K-Means and GMM. This characteristic empowers the method to establish clusters primarily comprised of core points, which convey essential information, while border points play a role in delineating the shapes of these clusters.

In this study, due to the extent of data preparation, the percentage of noise points within total points in the training phase is expected to be insignificant and as a result, no further processing of these points takes place.

Although DBSCAN is a density based clustering method, the metric for detecting anomalies is distance based and depends on one of the method's hyper-parameters; size of neighborhood (ε). This variable is selected due to its importance in the method. It is assumed that normal points will lay inside the neighborhood or at least at its border, meaning that the distance between their closest point will be less or equal to ε ($d \leq \varepsilon$). Respectively, if a point's closest neighbor's distance is greater than ε, this point is considered as an anomaly.

4.2.5 Anomaly Detection with SOM

SOM is an unsupervised ANN which can be used for clustering purposes. In this case, the model is used to detect anomalies in the test dataset. Previous applications of SOMs for anomaly detection in the maritime field include Vanem & Brandsæter (2021) and Raptodimos & Lazakis (2018). The followed procedure is to train a 2D SOM of a certain size, for a number of epochs determined by a combination of the quantization and the topographic errors. Then, by having the trained map, clusters of data-points that are created by the neurons are obtained. The following step is to assign the test data-points to these clusters. By having a larger number

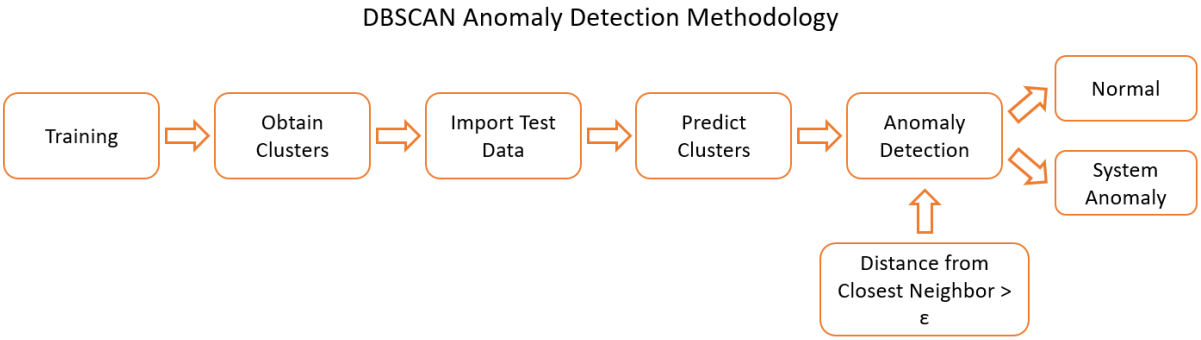

Figure 28: DBSCAN anomaly detection methodology.

of clusters, compared to the other methods, it is assumed that the average of all training points that are assigned to this node is an adequate reconstruction of the test point to be used for anomaly detection purposes (Vanem & Brandsæter, 2021). Hence, the residuals are obtained and the anomalies will be detected based on those.

Based on the above, two models which utilize the same trained SOM will be examined in this study. The aim is to explore different techniques when it comes to the anomaly detection part. As previously stated, both are based on residual analysis. The first method aims to detect anomalies through a threshold value in the residuals whereas, the second explores clustering of the residuals.

Anomaly Detection with SOM Based on Threshold Value

This approach seeks to classify data points as anomalies when their residual values exceed a predetermined threshold. The threshold value is chosen based on the distribution of these residuals, ensuring that it effectively captures deviations from the expected pattern in the data.

Anomaly Detection with SOM Based on Clustering of Residuals

In this case, instead of applying a simple threshold value, the K-Means clustering algorithm is employed. As previously explained in K-Means clustering theoretical part, the anomaly detection criterion is the point's distance from the cluster centroid. If it is greater than the 98th percentile of all distances of cluster points from the centroid, then it is considered an anomaly. This methodology is applied here, meaning that after obtaining the residuals clusters, the distances from the centroids are calculated and those points that their distance exceeds the 98th percentile are categorized as anomalies.

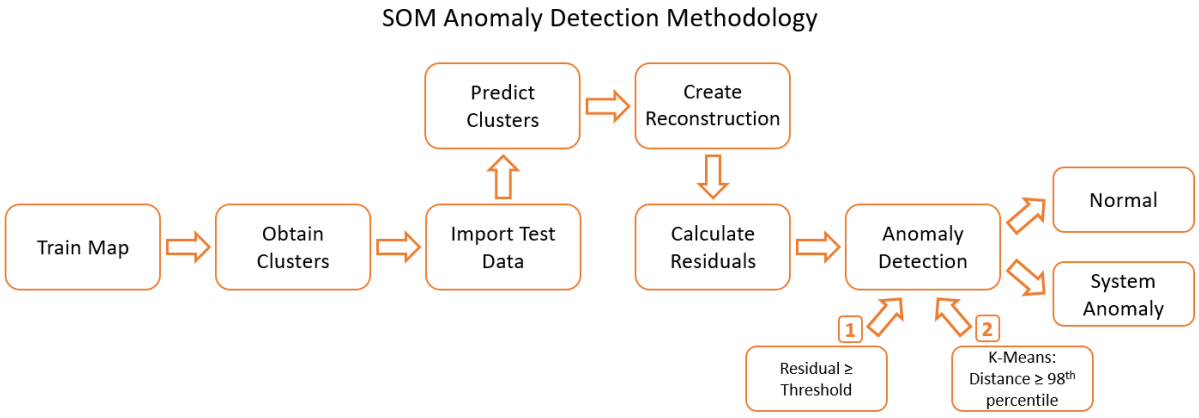

Figure 29: SOM anomaly detection methodologies (threshold and clustering based).

4.3 Simulation of Anomalies

The dataset to be used in the analysis may contain anomalies which are not known in advance. In case the anomaly detection models are trained with a dataset containing anomalies, they would not be capable of identifying them in the test phase. Hence, the steady state algorithm is deployed with a goal of eliminating the pre-existing anomalies in order to have an appropriate training dataset. As a result, anomalies must be simulated in the test data in order to validate the models.

4.3.1 Typology of Anomalies

According to Chandola et al. (2009), anomalies are categorized in three types. Point anomalies, contextual anomalies, and collective anomalies. In the case of an engineering application and especially in the marine anomaly detection domain, Velasco-Gallego & Lazakis (2022a) identified six types of anomalies.

- Single point anomalies.
- Two-point anomalies.
- Multiple point anomalies.
- Collective anomalies.
- Degradation.
- Transition occurrences between steady operational states.

An explanation of each anomaly type, except transitional occurrences which should have been filtered out, is given in the text that follows.

Point Anomalies

Point anomalies are the simplest form of anomalies in data analysis. They refer to individual data instances that deviate significantly from the overall data distribution. This category of anomalies is the primary focus of most research in the field of anomaly detection (Chandola et al., 2009). In the marine anomaly detection field, Velasco-Gallego & Lazakis (2022a) expanded the category of "point anomalies" into single, two-point, and multiple point anomalies. The explanation behind this sub-categorization has to do with the fact the two and multiple point anomalies are just repeated single point anomalies in the dataset. To illustrate point anomalies in an actual ship-related scenario, a propeller RPM time-series should be taken into consideration when a vessel is sailing through weather.

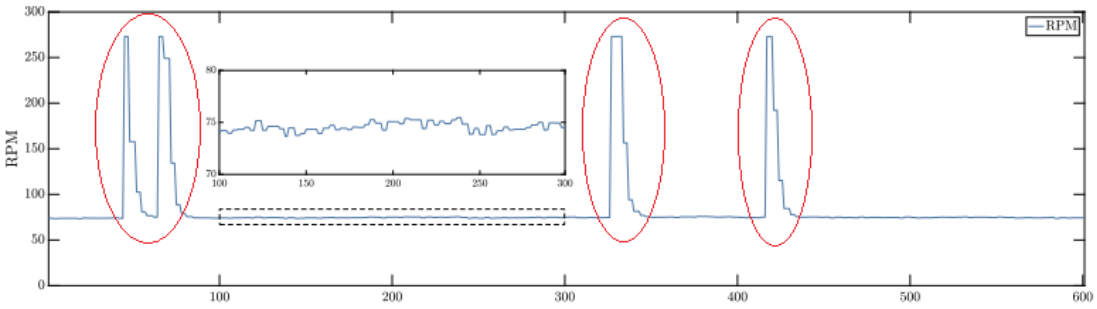

Figure 30: Point anomalies observed in a propeller RPM time-series graph. Source: Øyvind Øksnes Dalheim & Steen (2020b).

Collective Anomalies

Collective anomalies arise when a group of related data instances collectively deviates from the norm in the entire dataset. While each individual instance in the group might not be anomalous on its own, their combined occurrence as a collection is considered anomalous.

Collective anomalies have been studied in various types of data, including sequences, graphs, and spatial data according to Chandola et al. (2009). It is important to note that while point anomalies can occur in any dataset, collective anomalies only manifest in datasets where instances are interconnected.

As per Velasco-Gallego & Lazakis (2022a), in a marine engine example, the high variations in the exhaust gas outlet temperature of a turbocharger under steady operation can be considered as a collective anomaly or a sudden noise increase in the time-series of shaft power.

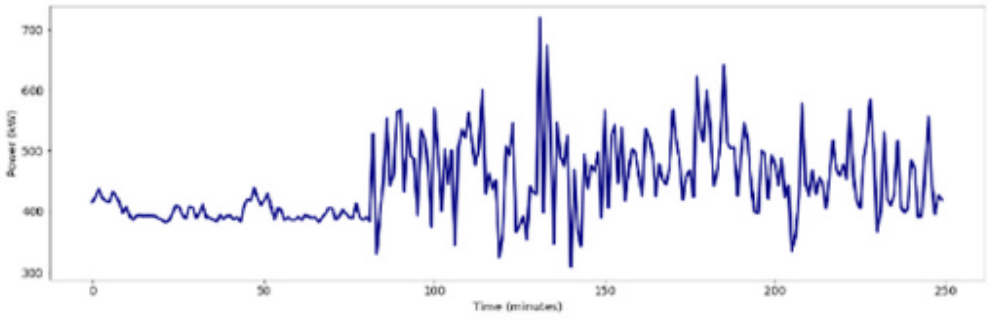

Figure 31: Collective anomalies observed in a shaft power time-series graph. Source: Velasco-Gallego & Lazakis (2022a).

Degradation Sequences

Attention is also paid to degradation patterns. These sequences are also considered as collective anomalies. In Velasco-Gallego & Lazakis (2022a) are studied as a separate anomaly category. This approach is also used in this study. This is done due to the importance of such phenomena in condition monitoring and prognosis applications.

4.3.2 Simulation of Point Anomalies

The first step that needs to be made in order to simulate point anomalies is to determine the time series on which they will be induced. The descriptive characteristics of the data must be known and based on them the value of the anomalous instance is selected. When referring to point anomalies, a spike is expected to be visible in the time-series plot. The last required element to fully specify such anomalies is their position in the time-series (Velasco-Gallego & Lazakis, 2022a).

When the aim is to simulate two or multiple point anomalies, spikes of similar magnitude should be placed in close distance to each other, relative to the time-series length.

4.3.3 Simulation of Collective Anomalies

When referring to collective anomalies, Velasco-Gallego & Lazakis (2022a) mentioned a high variability example in the turbocharger's exhaust gas temperature time-series. The presence of high variability can be easily be misinterpreted as noise. The underlying difference lies in that noise is a permanent phenomenon which may obscure true data patterns whereas a high variability is temporal issue which can be categorized as an anomaly. To simulate such

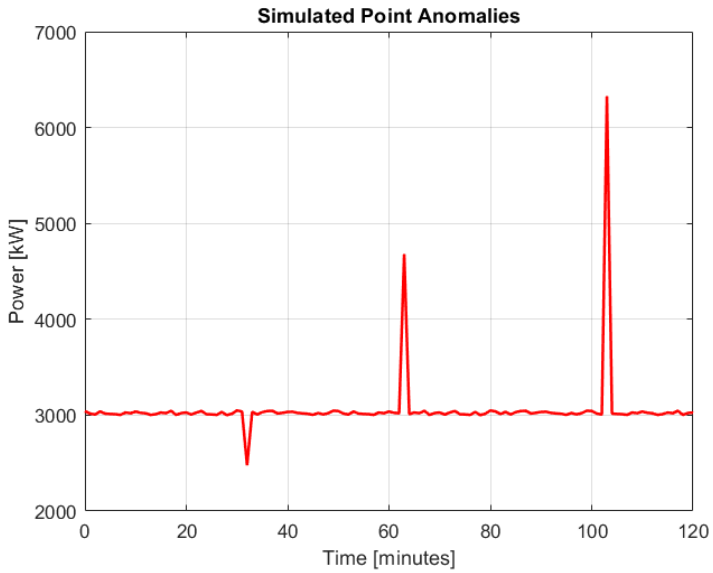

Figure 32: Example of simulated point anomalies in ME Power.

anomalies noise is injected through multiple Gaussian distribution with different means and standard deviations (Velasco-Gallego & Lazakis, 2022a; Zhao et al., 2019).

Other patterns which may be categorized as collective anomalies are extremely low or high values at certain time periods which are not found elsewhere in the dataset (Chandola et al., 2009). Similarly to what mentioned for point anomalies, the descriptive statistics of the time-series must be know along with the position within it where the anomalies will be placed.

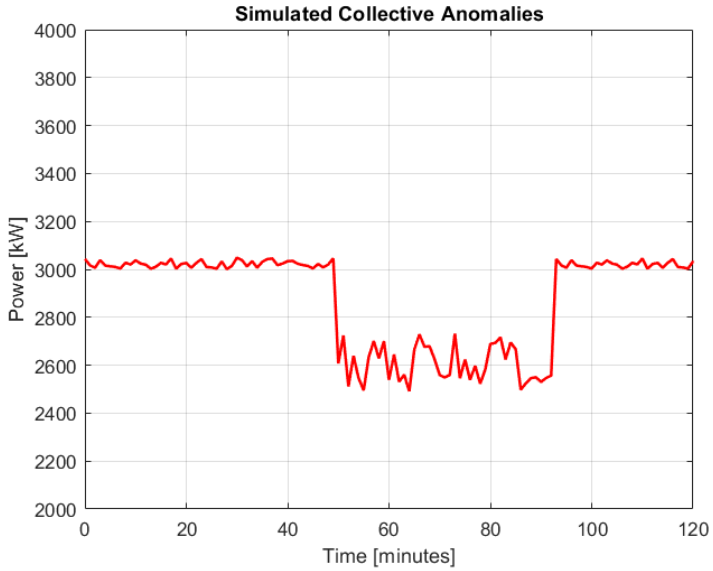

Figure 33: Example of simulated collective anomalies in ME Power. In a true dataset, the value of Power being around 2600 kW might not be an anomaly. In the figure's context since no other information is given about Power these points may be categorized as collective anomalies.

4.3.4 Simulation of Degradation Sequences

An exponential model with Brownian motion is chosen to simulate the degradation sequence. Similar models have been used in Velasco-Gallego & Lazakis (2022a). Its effectiveness

in representing patterns of accelerated degradation of engineering components has been proven by Li et al. (2021). The mathematical modeling of such pattern involves a stochastic process $X(t),\ t \geq 0$, with $X(t)$ representing a condition indicator of the system. The model is expressed as:

$$X(t) = \vartheta' \cdot \exp\left(\left(\beta' - \frac{\sigma^2}{2}\right) t + \sigma \cdot B(t)\right) \tag{39}$$

where, ϑ' and β' are random parameters representing the individual differences of components, σ is a deterministic parameter representing the increasing random error, and $B(t)$ is the standard Brownian Motion, responsible for the stochastic dynamics of the degradation.

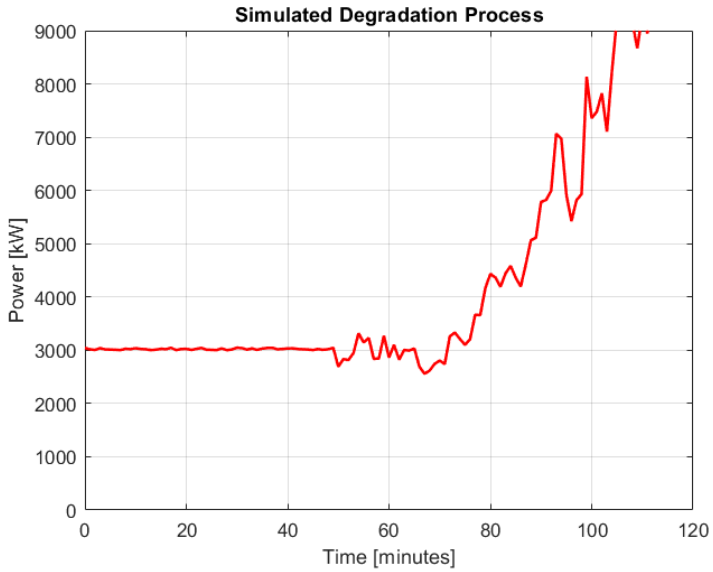

Figure 34: Example of simulated degradation process in ME Power.

Brownian Motion

The Brownian Motion $B(t)$ at time t is characterized by three key principles.

1. The change in the Brownian Motion over a time interval $[s, t]$ is independent of the values at times before s.

2. The distribution of the change in Brownian Motion over a time interval $[s, t]$ depends only on the length of the interval $t - s$, not on the specific values of s and t.

3. The increments of the Brownian Motion are normally distributed with mean 0 and variance $t - s$, resulting in the following: $B(t) - B(s) \sim \mathcal{N}(0, t - s)$.

The notation can be expressed as follows: $B(t) \sim \mathcal{N}(0, t)$.

To simulate the Brownian motion discrete time steps are used. The increments ΔB_i over small intervals Δt are sampled from a Gaussian distribution with mean 0 and variance Δt. The cumulative sum of these increments gives the Brownian Motion values at each time step: $B(t + \Delta t) = B(t) + \Delta B_i$

5 Case Study & Results

In this section, the case study and its outcomes are presented. The aim is that the order of operations and steps within the analysis is maintained in the report for better understanding of the reader.

Initially, a description of the data is presented. Then, the implementation of the data preparation steps follows. This phase is of critical importance since the transformed dataset is the input for the main part of the study, which is the implementation of the anomaly detection models. After this phase, the simulated anomalies, which are employed to test the models, are presented. An extensive analysis of all the examined methodologies is found later. Special attention is paid on each algorithm's tuning step. Then, a brief discussion concerning the main outcomes from the application of each model follows.

5.1 Data Description

Data from a bulk carrier vessel which are collected via a data acquisition system are used in this study. Since the aim is to detect anomalies in the operation of machinery systems, a two stroke ME has been selected as the case study. All ME related parameters that are monitored via the DAQ have been included in the original dataset. These involve mostly temperature and pressure time series across all cylinders focusing on cooling, lubrication, intake, and exhaust characteristics of the engine. Data relevant to the fuel system of the engine are also part of the dataset. The data have been collected from the vessel's sensors and then sent to a database from which they have been extracted from to be utilized in this study. Data from January 1st until 31st December of 2021 were collected with a sampling rate of 1 minute.

The initial dataset's features that are utilized in this study may be found in Table 3.

5.2 Data Preparation

Each of the following steps are used in order to improve data quality and integrity for the anomaly detection task. Dataset size decreases by implementing each of these preparation methodologies. The process complies with what has been presented previously in the methodology section and is summarized as follows.

- Data cleaning
- Data de-noising
- Steady state identification
- Data normalization
- Data split into subsystems
- Dimensionality reduction

In order to demonstrate the effectiveness of these steps, a histogram and a time-series plot of RPM data per step are presented respectively in Figure 36 and Figure 37. The distribution of raw RPM data are displayed in the first subplot. Idle state is dominant. In the second subplot, by removing idle state, a clearer representation of the data is exhibited. In the third subplot, after the smoothing step, a similar distribution is illustrated . Lastly, the data distribution after steady state identification is presented. It is visible that the data preparation step affects the distribution of RPM.

Table 3: Dataset feature names.

Variable Names	
Cyl 01 AFT main bearing temp_AMS	ME cyl 3 PCO outlet temp_AMS
Cyl 01 crank pin bearing temp_AMS	ME cyl 3 scav air temp_AMS
Cyl 01 fore main bearing temp_AMS	ME cyl 4 exh gas outlet temp_AMS
Cyl 02 AFT main bearing temp_AMS	ME cyl 4 JCW outlet temp_AMS
Cyl 02 crank pin bearing temp_AMS	ME cyl 4 PCO outlet temp_AMS
Cyl 03 AFT main bearing temp_AMS	ME cyl 4 scav air temp_AMS
Cyl 03 crank pin bearing temp_AMS	ME cyl 5 exh gas outlet temp_AMS
Cyl 04 AFT main bearing temp_AMS	ME cyl 5 JCW outlet temp_AMS
Cyl 04 crank pin bearing temp_AMS	ME cyl 5 PCO outlet temp_AMS
Cyl 05 AFT main bearing temp_AMS	ME cyl 5 scav air temp_AMS
Cyl 05 crank pin bearing temp_AMS	ME cyl 6 exh gas outlet temp_AMS
Cyl 06 AFT main bearing temp_AMS	ME cyl 6 JCW outlet temp_AMS
Cyl 06 crank pin bearing temp_AMS	ME cyl 6 PCO outlet temp_AMS
M/E Shaft RPM_TRQM	ME cyl 6 scav air temp_AMS
M/E T/C RPM_IND1	ME cyl lub oil temp_AMS
ME air cooler cool w inlet press_AMS	ME FO inlet press_AMS
ME air cooler cool w inlet temp_AMS	ME FO inlet temp_AMS
ME air cooler cool w outlet temp_AMS	ME fuel index_AMS
ME axial vibration_AMS	ME JCW inlet press_AMS
ME Consumption_TRQM	ME JCW inlet temp_AMS
ME control air inlet press_AMS	ME JCW outlet press_AMS
ME cyl 1 exh gas outlet temp_AMS	ME main LO inlet press_AMS
ME cyl 1 JCW outlet temp_AMS	ME main LO inlet temp_AMS
ME cyl 1 PCO outlet temp_AMS	ME scav air receiver inlet pres_AMS
ME cyl 1 scav air temp_AMS	ME scav air receiver temp_AMS
ME cyl 2 exh gas outlet temp_AMS	ME TC exh gas inlet temp_AMS
ME cyl 2 JCW outlet temp_AMS	ME TC exh gas outlet temp_AMS
ME cyl 2 PCO outlet temp_AMS	ME TC LO inlet press_AMS
ME cyl 2 scav air temp_AMS	ME TC LO outlet temp_AMS
ME cyl 3 exh gas outlet temp_AMS	Shaft Power_TRQM
ME cyl 3 JCW outlet temp_AMS	Shaft Torque_TRQM
ME thrust bearing pad temp_AMS	

5.2.1 Data Cleaning

Data were received in twelve files, each one containing the data corresponding to one month. All files were concatenated vertically resulting in the large raw data file. This file contained data from 104 sources. Many of those were duplicates, others contained processed data (ISO corrected etc.), data not directly sourced from sensors, and data with abnormal values.

The first action was the removal of these features. The aim was to only keep data directly sourced form sensors. This step was carried out manually, adhering to the instructions provided by the data source regarding variable naming-data sourcing combination. The result was a significantly more compact dataset of 63 variables, the list of those can be seen in Table 3.

Handling of missing and/or NaN values took place. First, the total number of such instances per variable were determined. As can be seen by Figure 38, an almost constant amount of NaN values is contained in each variable. The presence of these has been linked to the period before commissioning of the DAQ system onboard the vessel. More than average NaN values are found in Cylinder 3 crank pin bearing temperature and turbocharger RPM time series for unknown reasons. It is believed that a possible sensor fault may be responsible for those issues. Based

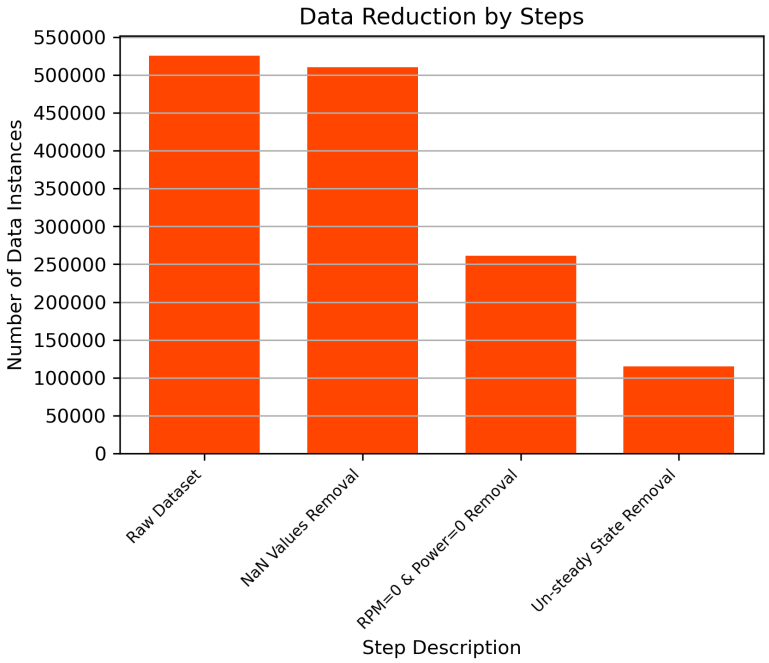

Figure 35: Data reduction by steps in the data preparation phase. Only the steps in which data-points are reduced are shown in this figure.

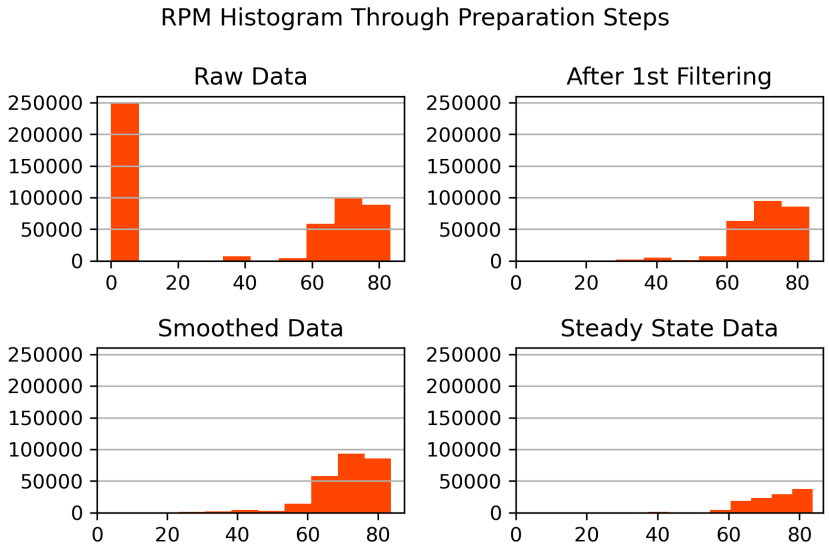

Figure 36: Histogram of ME RPM distribution per data preparation step.

on all the above, it was concluded that data imputation was not required.

5.2.2 Data De-noising

From the four algorithms tested for the de-noising of the time series data, it was decided to use the Savitzky-Golay (SG) filter. This method provides flexibility and the data were fitted better than the other examined methods. The decision to employ this filter was based on the property of SG, which enables it to preserve some high-frequency components while effectively removing noise. On the other hand, although frequently used, moving average presented significant lag and flattened certain effects. With respect to the Gaussian filter technique, it was not

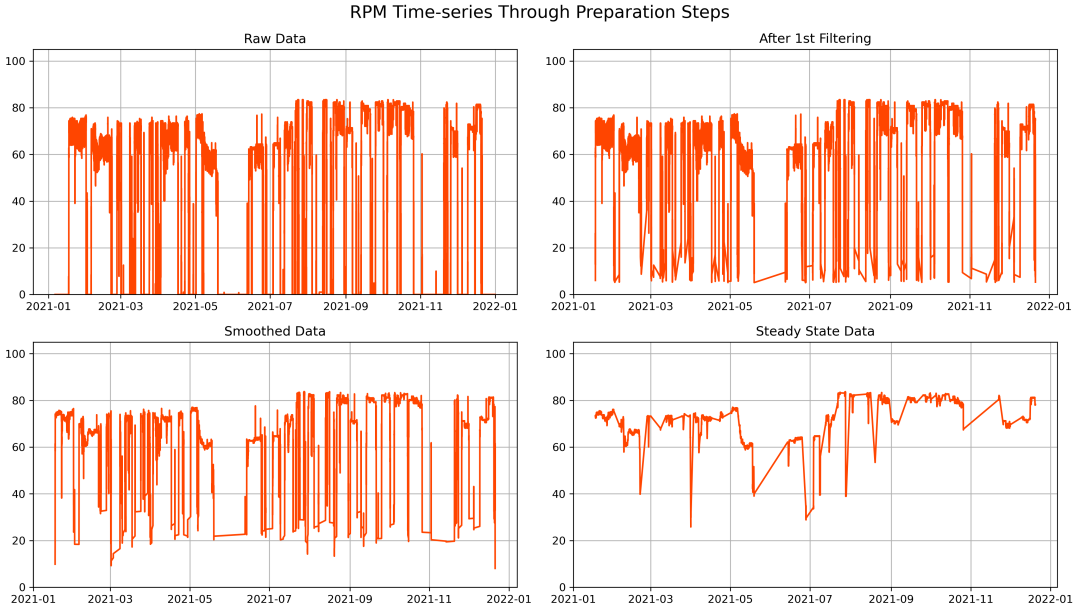

Figure 37: Time-series plot of ME RPM per data preparation step.

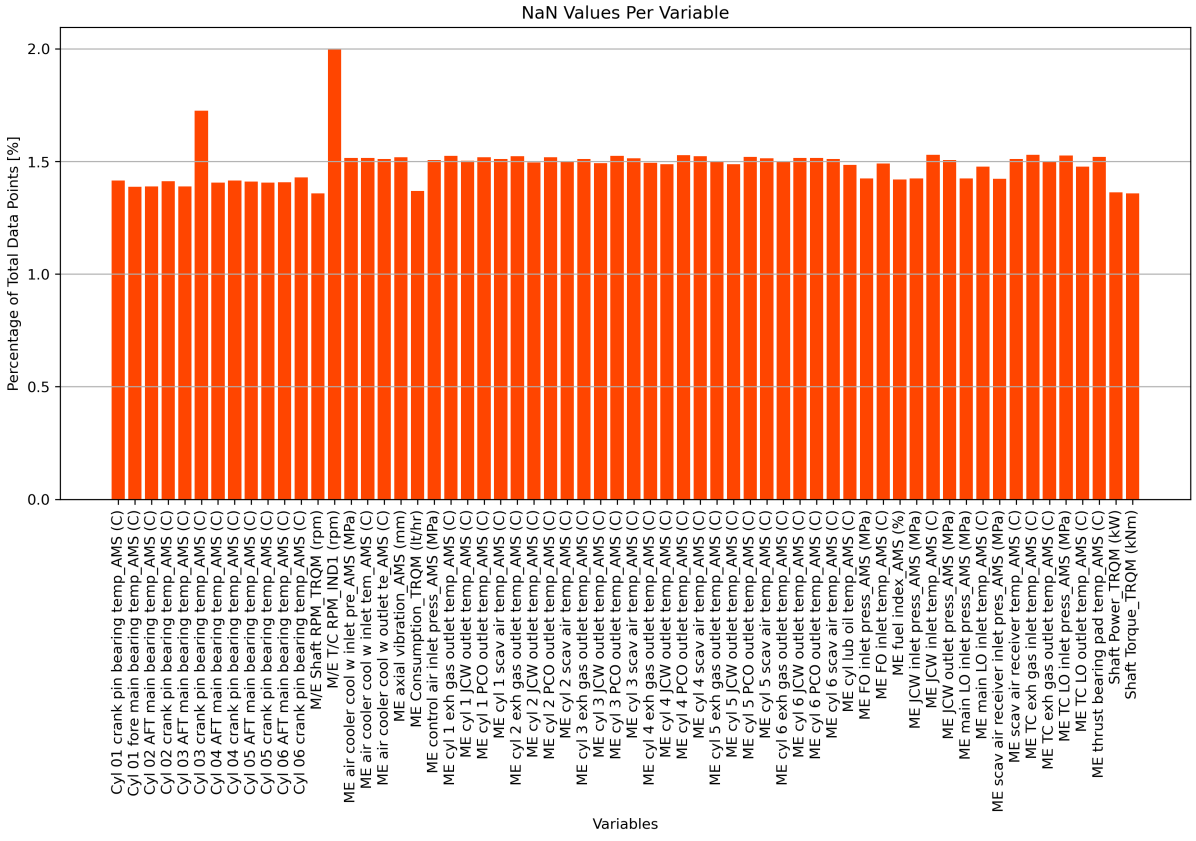

Figure 38: NaN values per variable in the raw dataset.

possible to fine tune the method resulting in either over or under-smoothed results. Between EWMA, which was previously used in a marine engine application (Velasco-Gallego & Lazakis, 2022d), and SG, the second delivered slightly better results.

The filter was applied to the total dataset using the same parameters (window size, polynomial degree) to determine the smoothing. Selected polynomial order for the filter was three.

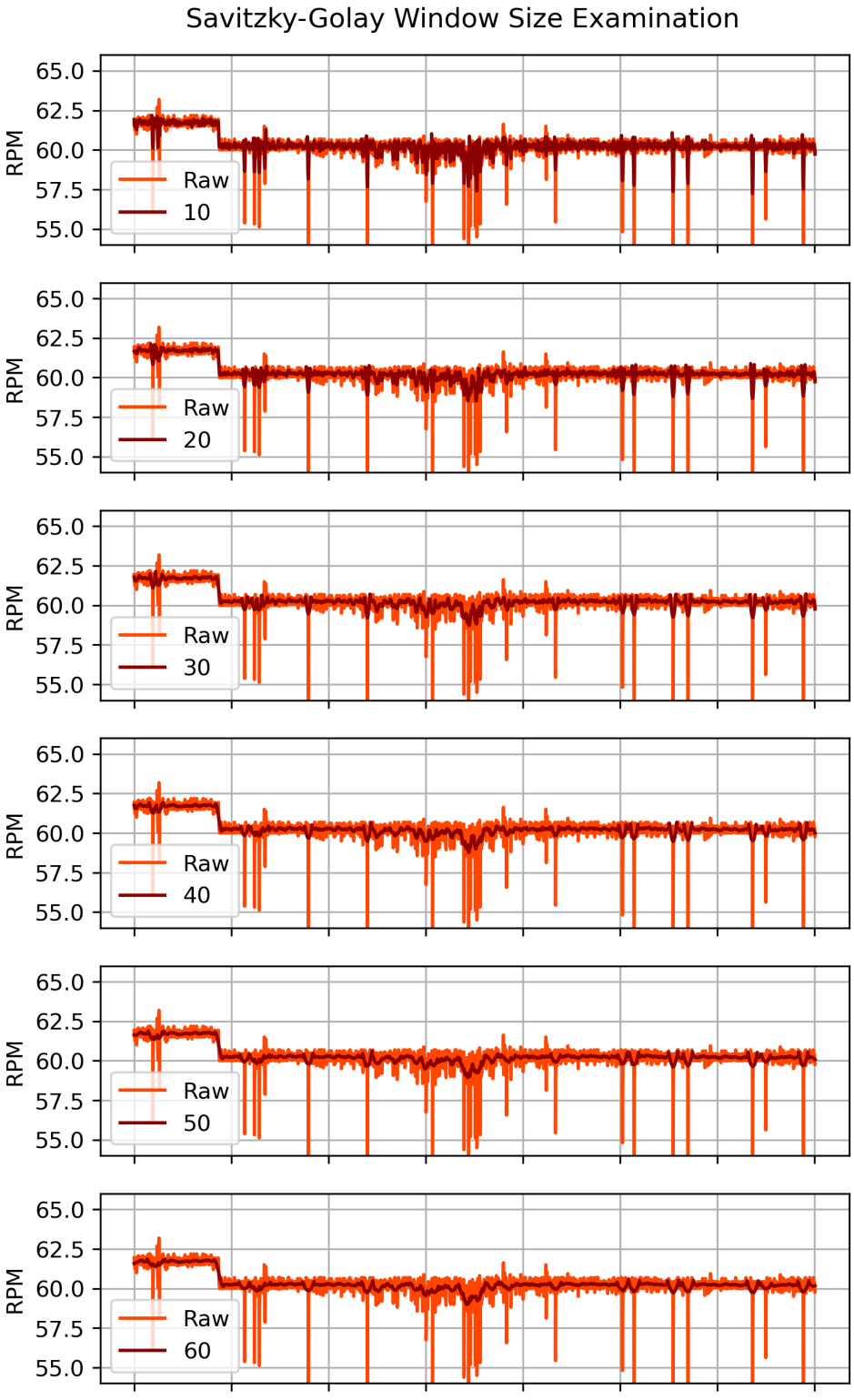

Figure 39: Optimum window size investigation of Savitzky-Golay filter.

Higher-degree polynomials introduced excessive oscillations and lower-degree oversimplified the time-series. A third degree curve smoothed the data in an adequate way while captured intricate patterns and variations. The tuning process of the algorithm consisted of investigation of the smoothing results with multiple window sizes in a sample dataset. Windows ranging from 10 to 60 data points (1 point corresponds to 1 minute frequency) were examined. The optimal

window size was found to be 40.

The effectiveness of smoothing can be observed through commonly used graphs for engines. In the Power vs RPM graph, the shape before and after remains the same while in both cases, a similar to propeller law curve is followed (Figure 40). Also, the distribution of shaft power data after smoothing is analogous to what is observed before. Changes appear at the areas with higher distribution of points, due to the noisy sensor signal being redistributed by the smoothing process (Figure 41).

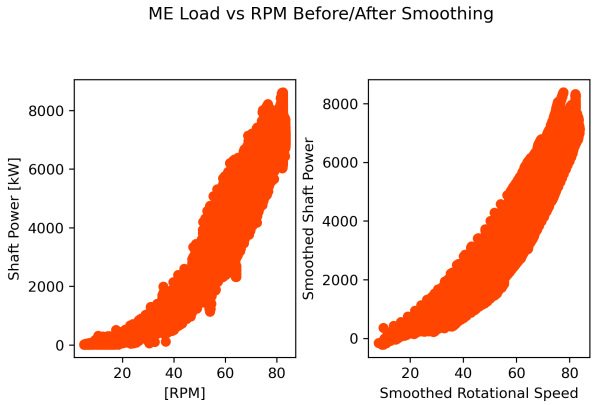

Figure 40: Shaft power vs RPM graph comparison between raw and smoothed datasets.

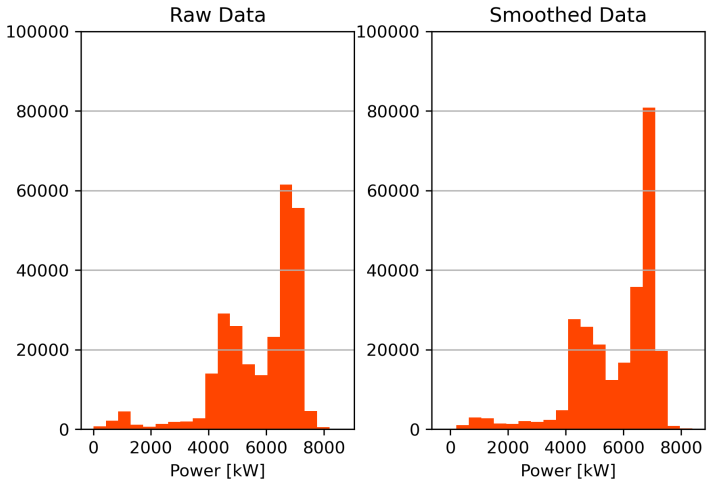

Figure 41: Histogram of Shaft power data before & after smoothing process.

5.2.3 Steady State Identification

Following the smoothing process, the steady state algorithm is deployed. The parameters of the ME system identified as critical are ME RPM and shaft power. These two parameters offer macroscopic description of the system's state when combined. The aim of such algorithm is to remove transient states and obtain a steady dataset to use in further steps of the analysis. This highly depends on algorithm parameters tuning. For example, longer windows may be more suitable for capturing sensor drift and long term trends whereas shorter windows are utilized to identify abrupt changes in the data (Øyvind Øksnes Dalheim & Steen, 2020a). Also, in the steady state testing section of the algorithm, the significance level of t-statistic

along with probability and slope thresholds affect the performance. Since it is deployed in each critical variable separately before combining the results, the selected parameters differ in terms of significance level. This dissimilarity enhanced the identification task by allowing each algorithm to perform better individually before combining the results.

The algorithms output is a vector containing the steady state probability of each point. To classify a time-point as steady, both RPM and power probabilities should be greater than the lowest accepted. A mask of these point indexes is used for filtering to obtain the steady dataset.

Table 4: Parameters used in steady state algorithm. The parameter testing process was based in multiple trial runs.

Parameter	ME RPM	Shaft Power
Window Size	60	60
Significance Level	3%	2%
Maximum Accepted Slope	0.3	0.3
Lower Accepted Probability	0.1	0.1

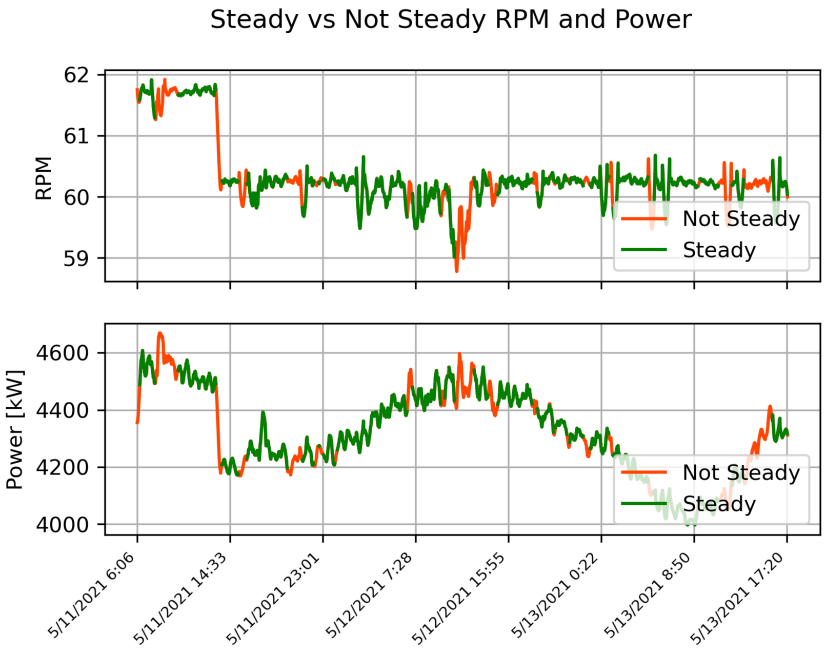

Figure 42: Steady state identification algorithm plot. Steady RPM and shaft power over not-steady.

The performance of the algorithm has been tested on multiple sample datasets. One of them is shown in Figure 42. "Steady" refers to points presenting this behavior on both parameters. The model allows for normal fluctuations in the data while big drops and spikes are classified as "not-steady".

5.2.4 Data Normalization

Z-score or Normal Scaler is employed for the normalization task. The difference between a variable's mean and each point's value over standard deviation scales the data while not affecting the variability. Before and after normalization plots of graphs commonly used for performance

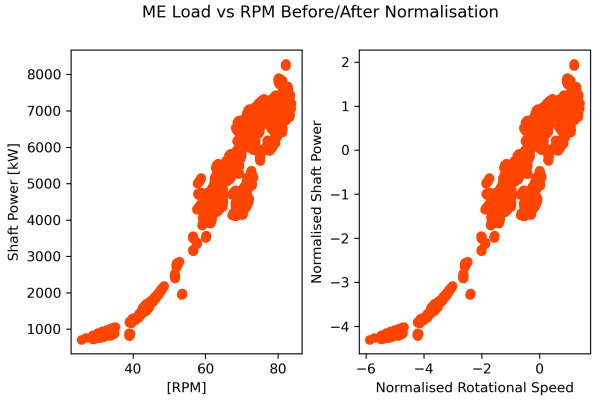

Figure 43: Power vs RPM graph before and after normalization.

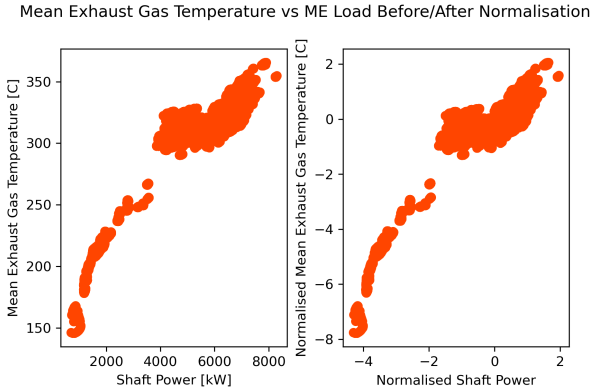

Figure 44: Mean exhaust gas temperature vs Power graph before and after normalization.

evaluation of engines are presented in Figure 43, Figure 44, and Figure 45 to demonstrate the process.

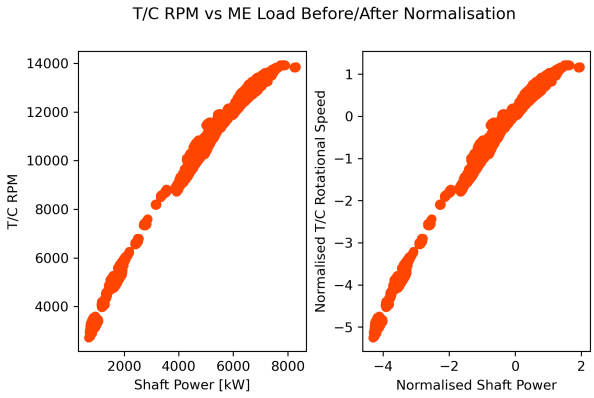

Figure 45: Turbocharger RPM vs Power graph before and after normalization.

5.2.5 Data Division Into Sub-systems

Following normalization, the dataset is divided into subsystems according to what Cai et al. (2017) presented, but with some additions and modifications. In that phase the dataset is almost fully prepared, with all steps being completed except for dimensionality reduction, which will be performed separately in each system. The lubrication system includes parameters referring

to Lubricant Oil (LO) temperatures, pressures, and also main and crank pin bearing temperatures. Contained in intake & exhaust there are the signals of turbocharger inlet and outlet temperatures. The Fuel System in this study also includes the Shaft RPM, Power, and Torque parameters, whereas the cooling system includes features related mostly to jacket and piston cooling. Complete tables of feature per sub-system can be found in Appendix A. Additional information regarding the systems is found in Appendix B.

5.2.6 Dimensionality Reduction

PCA is employed for dimensionality reduction separately on each sub-system. The need to perform such task arises from the presence of highly correlated variables within every system. Many of those parameters, such as exhaust gas temperature, lubricant oil pressure and others, are monitored across all cylinders, resulting in multiple sets of six (engine cylinder number) correlated variables. This can be observed in Figure 46 where a correlation heatmap of all parameters is presented. A solution would be averaging the between-cylinder columns at each time-point, (Kim et al., 2020b, 2021), but this option may lead to information loss from individual cylinders that might be meaningful for anomaly detection or lead to incorrect conclusions. By using PCA, dimensions are reduced with the cost of losing a small percentage of the initial dataset's variability. The desired combination between principal components and initial dataset's explained variance in the transformed dataset is defined by the user and can be determined with the help of a Scree plot.

The optimal number of Principal Components (PCs) is defined by a threshold value in the explained variance ratio by each component. The threshold value is set at 0.01, meaning that the optimal value of PCs is reached when the increase in explained variance ratio that the new PC offers is less than 1%. Another approach considered for such task is the elbow method, but the other approach delivered the required results. Based on PCA methodology, each PC tries to capture as much of the system's variance as possible. Thus, the slope of the explained variance ratio over number of PCs curve is expected to be declining as PC number increases. The reduction in components of each subsystem after PCA can be seen in Table 5. In the same table is visible the total explained variance of the initial dataset which is captured by the PCs. The Scree plot for the cooling and the intake & exhaust systems are presented below in Figure 47 and Figure 48. Additionally, the plots for the remaining sub-systems may be found in Appendix B.

Table 5: Number of components of each subsystem before and after PCA.

Subsystem	Cooling	Fuel	Intake & Exhaust	Lubrication
Before	18	7	17	18
After	8	5	4	6
Explained Variance Ratio	96.9%	99.8%	97.5%	97.9%

In all sub-systems except fuel the reduction of features is more or equal to 50% of the original. These results are expected due to the presence of same sensors across all cylinders. In fuel, the second dimension size is reduced from seven to six. This result is inconsistent with other subsystems' results and expectations regarding this particular subsystem. It is known that an increase in shaft RPM would result in increases in consumption, fuel index, power, torque suggesting that these features are highly correlated. Possible reasons for this outcome are that the dataset might inherently possess a complex structure, where each column represents crucial and distinct features that collectively contribute to the overall variance. Another possible explanation could be weak correlation between columns of the subsystem so PCA cannot identify

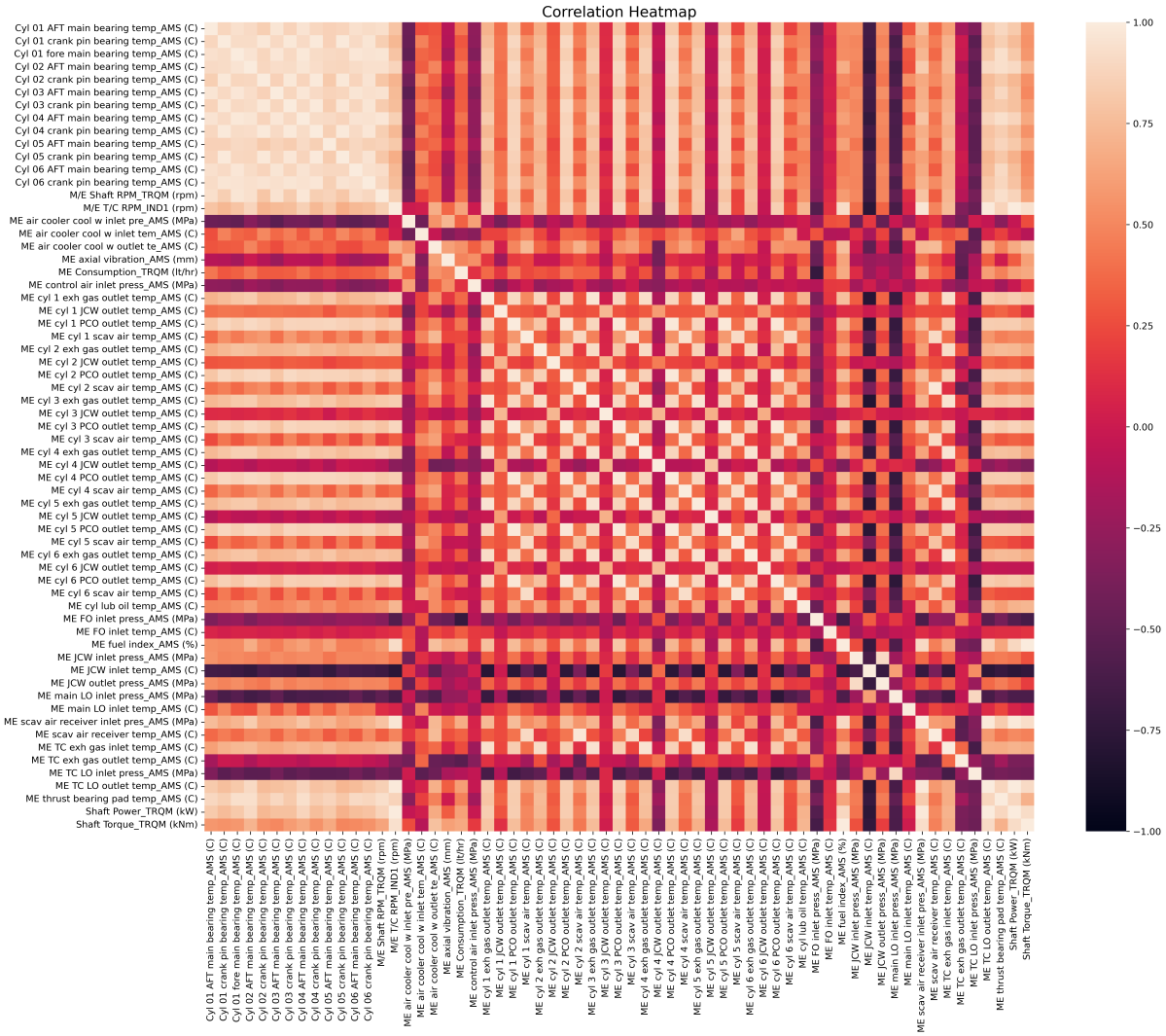

Figure 46: Correlation heatmap of variables in the dataset before division in systems and PCA.

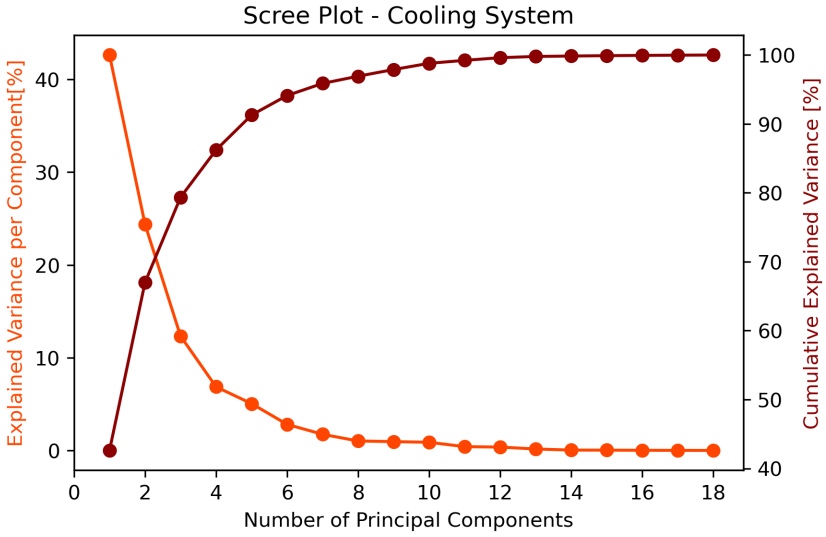

Figure 47: Scree plot to visualize PCA in the cooling system. Explained variance per principal component & cumulative explained variance over number of principal components.

sub-spaces that explains high percentage of variance.

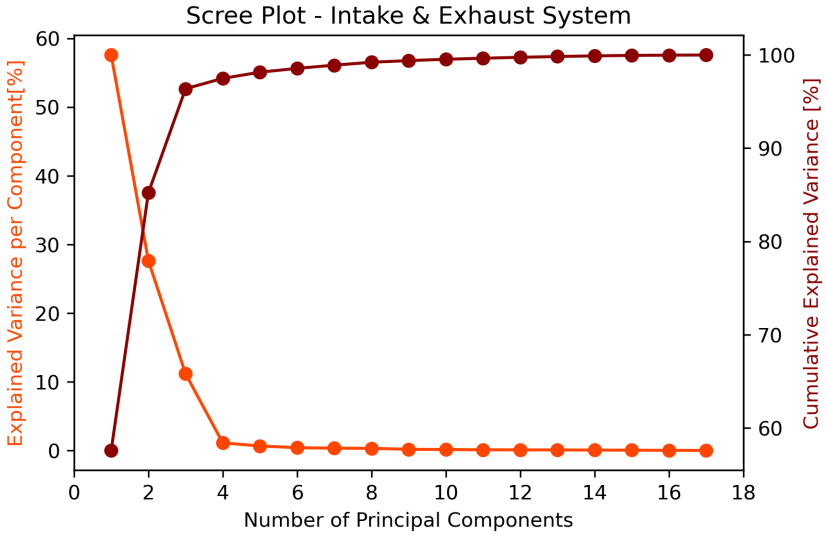

Figure 48: Scree plot to visualize PCA in the intake & exhaust system.

5.3 Anomaly Detection Models Implementation

In the pursuit of ensuring the seamless and reliable operation of marine engines, the application of clustering methods and machine learning models has unveiled a promising avenue for anomaly detection. This section presents a comprehensive analysis of the results obtained through extensive experimentation and implementation of these techniques in the context of marine engine systems operations.

Throughout this part, detailed descriptions of the methodologies employed, the datasets utilized, and the performance metrics considered for evaluating the efficacy of anomaly detection will be displayed. Furthermore, a comparative analysis of the various models applied will be presented, highlighting their strengths and weaknesses.

Additionally, the purpose of this study is to find representative models for the anomaly detection task by analyzing each method's strengths and weaknesses and comparing different performance metrics. To avoid losing this core purpose, the rest of this part focuses on two out of the four systems of the ME, namely cooling and intake & exhaust. Results analogous to what shown for the selected systems may be found in Appendix C.

As discussed, the dataset used for this anomaly detection task lacks labeled anomalies. Consequently, conventional performance metrics shaped as a ratio of True Positives, True Negatives, False Positives, and False Negatives cannot be applied. Hence, the metric used for evaluation of the algorithms at this stage is formulated as:

$$\text{Ratio of Anomalies} = \frac{\text{Anomalies Detected}}{\text{Dataset Points}} \tag{40}$$

This metric has been employed also in Vanem & Brandsæter (2021) and is carefully selected as it does not categorize points as True/False Positives/Negatives, since there is no definitive truth to classify data points as True Positives or Negatives. Consequently, traditional performance metrics that rely on such categorization are not applicable. Furthermore, this ratio metric reflects the relative performance of the employed anomaly detection methods. Also, it aids in the selection of the most appropriate model or parameter tuning for a specific task. In the sections that follow, Ratio of Anomalies will be used to identify and compare the different models.

5.3.1 K-Means Clustering Algorithm Results

The K-Means algorithm groups data points into K clusters, each centered around its respective centroid and determined using the squared Euclidean distance. K-Means can be adapted to the task of anomaly detection by examining the proximity of data points to cluster centroids. As previously discussed, K must be predefined in this algorithm. This results in the formulation of a problem: finding the optimal value of K in order to end up with a good clustering result. Five metrics are utilized in the search for K: the elbow method, the silhouette coefficient, the Calinski-Harabasz index, the last leap, and the last major leap.

In an ideal environment with well defined clusters all of the metrics values should agree. When no ideal cluster number if found, the silhouette coefficient and the Calinski-Harabasz index are preferred. This is done because they encapsulate both compactness and separation of clusters, that are requirements of K-Means, compared to other metrics which prioritize one of the two criteria.

While diving further into the intricacies of anomaly detection with K-Means clustering, it is essential to remember the differentiated approach to the anomaly detection criterion. The methodology prioritizes robustness by utilizing a criterion which considers the 98th percentile of distances in the training dataset. This choice, as opposed to the straightforward maximum distance or "radius", safeguards the model against categorizing outliers and anomalies as normal points.

Cooling System

In the case of the cooling system, the approach was to run the algorithm for $K = [2, 35]$. Since no known initial estimation method for K exists, a big enough number was chosen as the maximum potential cluster number. The optimal number of clusters has been found to be equal to 13. This number was selected by prioritizing the silhouette coefficient and the elbow plot as cluster number determination metrics.

An issue concerning the silhouette coefficient is its value. As previously discussed, the optimal number of clusters according to this metric is the one which has the maximum coefficient. A value of 1 means well separated and compact clusters, 0 means an indifferent and not significant result, whereas -1 is equivalent to wrong cluster assignment. One could expect a higher silhouette value for the optimal K, though, the complexity of the dataset is such that does not allow for very good separation and compactness.

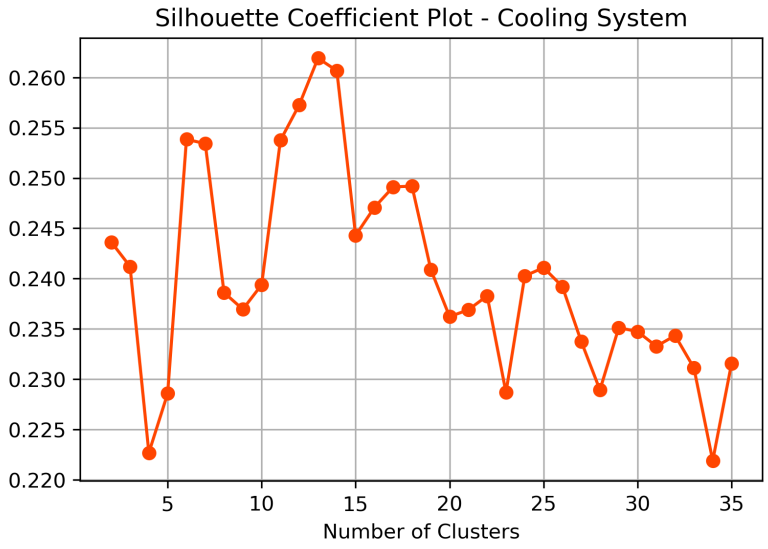

Figure 49: Silhouette coefficient in K-Means clustering plot over number of clusters for the cooling system.

Additionally when referring to the elbow plot, no clear elbow appears to be present in the first sight. When the figure is looked more carefully, two elbow points may be identified at K = 9 and K = 13. Based on the second elbow, it is safe to assume that 13 is a good number of clusters for the particular dataset.

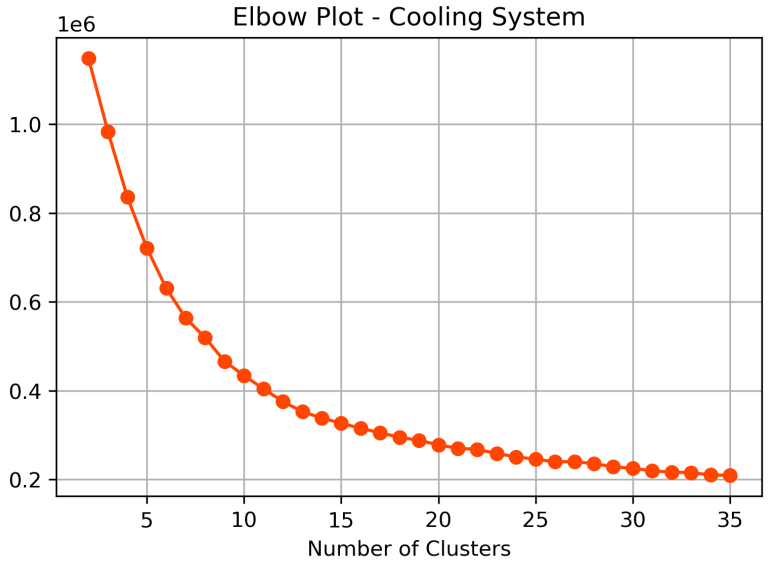

Figure 50: K-Means clustering elbow plot for the cooling system.

Table 6: Optimal number of clusters of cooling system according to different methods (K-Means clustering).

Metric	Elbow	Silhouette	CH index	LL	LML
Clusters	13	13	2	22	10

The anomaly detection algorithm is developed based on the K = 13 clusters. By applying the trained model to the test dataset, the resulting distribution of points in these clusters between the train and test phases is almost identical signaling good clustering performance. Slight differences are present due to the detected anomaly points. The algorithm detected 597 different points, corresponding to 2.08% value in Ratio of Anomalies.

Intake & Exhaust System

The initial potential values of K have been kept the same as in the cooling system. The optimal value of clusters has been found to be K = 5. This value has been identified from the elbow, the CH index, the silhouette coefficient, and the LML index. Although the elbow method suggested a second potential cluster value, K = 11, it is worth noting that the consensus among four different methods favors the choice of K = 5 as the most suitable number of clusters for this system. The alignment across multiple metrics reinforces the robustness and reliability of this particular number, enhancing confidence in its selection.

Based on what previously discussed, the anomaly detection model has been trained with 5 clusters. The test dataset's points distribution to clusters is highly comparable with the train dataset's. This alignment suggests that the model has successfully generalized its understanding of the underlying data structure during training, effectively extending its ability to classify

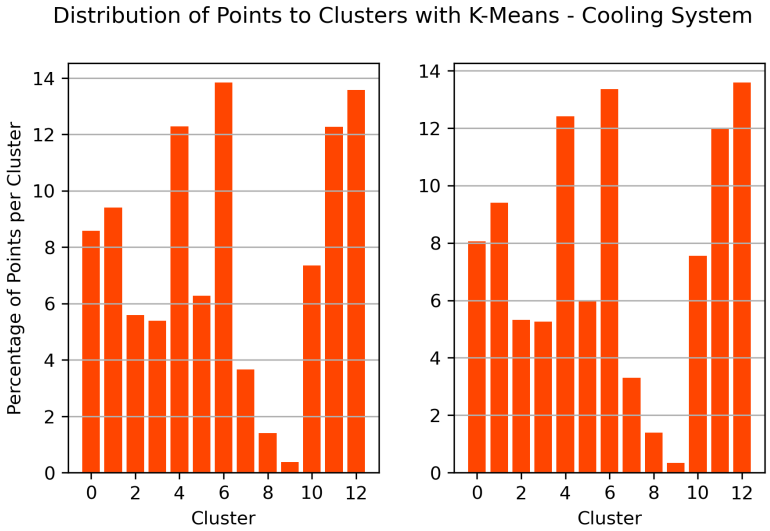

Figure 51: Distribution of points to clusters between train (left) and test (right) phases with K-Means in the cooling system.

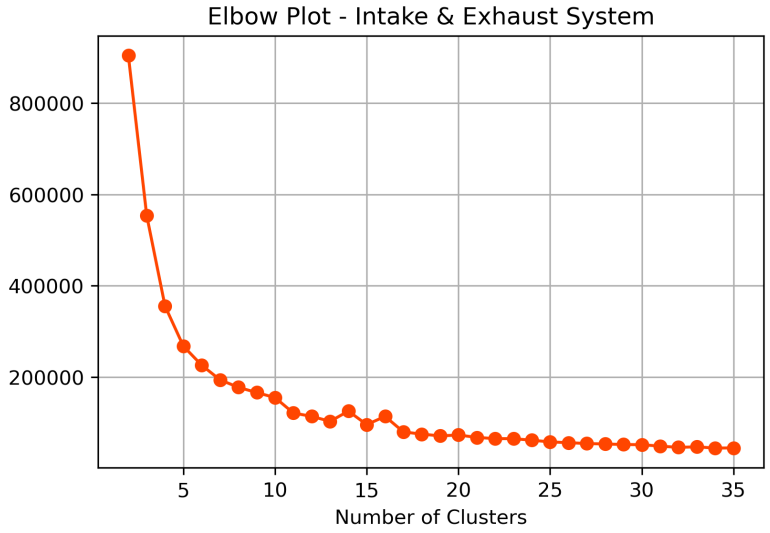

Figure 52: K-Means clustering elbow plot for the intake & exhaust system.

unseen data points into the same or similar clusters. Further to that, the model detected a total of 539 points as anomalies, and the equivalent percentage of those with regards to the test dataset is 1.87%.

Discussion Over K-Means Clustering Results

K-Means detected similar number of anomalies between the two systems. It was found that the optimal value of K was 13 in the cooling system and 5 in the intake & exhaust system. Although these numbers have been decided as optimal for the particular cases, the plethora of metrics suggesting different values of K may easily turn the selection process to a confusing operation. Furthermore, the manual character of this process requires plenty of time.

With regards to the anomaly criterion, the 98th percentile of distances between points and cluster centroids has been found, through experimentation, to be ideal for anomaly detection. In the case of considering the largest distance as the threshold, many anomalies are classified as normal points since the method must assign all train points to clusters, including the potential

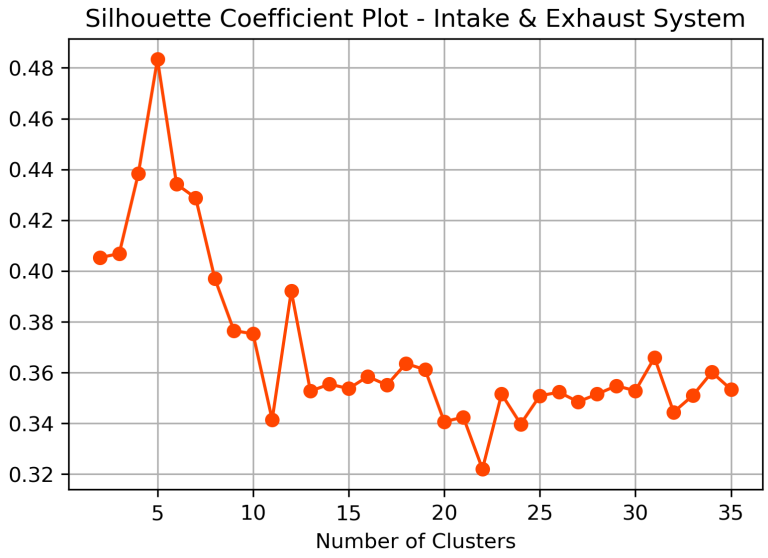

Figure 53: Silhouette coefficient in K-Means clustering plot over number of clusters for the intake & exhaust system.

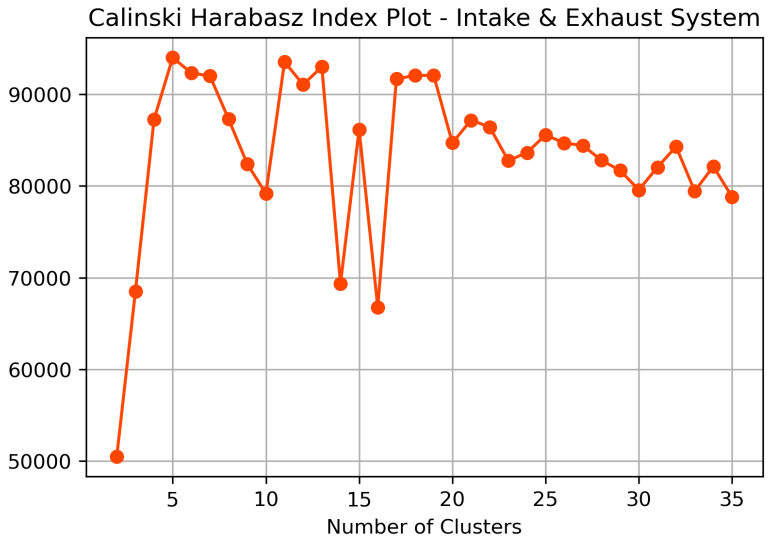

Figure 54: Calinski-Harabasz index plot over number of clusters for the intake & exhaust system (K-Means clustering).

anomalies. In the case of using a lower percentile value as threshold, many normal points, positioned at cluster boundaries, are classified as anomalies and overall the anomaly ratio increases dramatically.

Another drawback of K-Means is the random initialization of the K centroids. This results into slightly different outcome at each algorithm run. Alternative methods to this process have been developed but were not explored in this study. Finally, this algorithm requires increased run-time compared to the others. Nevertheless, it remains the most recognized clustering algorithm.

5.3.2 Clustering With GMM Algorithm Results

GMM algorithm's assumption is that each data are grouped in clusters generated by Gaussian distributions. Similarly to K-Means, the number of clusters (k) needs to be pre-determined,

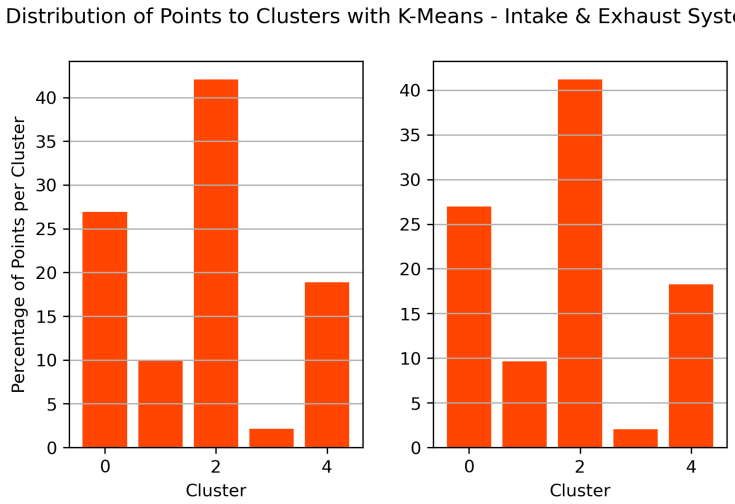

Figure 55: Distribution of points to clusters between train (left) and test (right) phases with K-Means in the intake & exhaust system.

and it should be established prior to the anomaly detection task. The available metrics to evaluate and find the optimal number of clusters are: the AIC, the BIC, and the silhouette coefficient.

AIC and BIC are the preferred metrics for finding the optimal cluster number due to their ability to penalize models with a lot of clusters in order to avoid over-fitting. Generally, lower values of AIC and BIC are preferred. An elbow point should be identified in the graph of the metric over k in order to reach an optimal value of k. Usually if found, the elbow point is the same for both metrics and is used as the optimal k value. Alternatively, if no elbow is found or $k_{AIC} \neq k_{BIC}$, the optimal k is considered to be the one that is derived from the silhouette coefficient.

The anomaly detection criterion in this algorithm is based on the cluster assignment probability. If a point's probability is below a predefined threshold, then it is considered to present anomalous behavior. The threshold is defined by the cluster assignment probabilities of the training data points.

Cooling System

The possible cluster numbers for GMM in the cooling system are around the same as with K-Means. It should be noted that this method uses K-Means to initialize the center positions of the k Gaussian distributions for faster converging time. In this particular system, the algorithm searched for the optimal k between 3 and 33. The optimal number was found to be equal to $k = 24$ clusters through the combination of BIC and AIC. $k = 6$ was proposed by the silhouette coefficient.

BIC and AIC penalize models with more clusters in order to avoid over-fitting. In the particular case, from $k = 22$ the curve's slope started to reduce so 24 is considered to deliver good results.

The distribution of the cluster assignment probabilities of the train dataset has been plotted in order to determine the threshold value for the anomaly detection criterion. It can be observed that the majority of points have a probability value of $P \geq 0.75$. Based on that observation, every test point with cluster assignment probability less than 0.75 will be considered to be an anomaly.

The distribution of points in clusters between the train and test phases showcased similar results, indicating a small number of detected anomalies. This number has been found to be

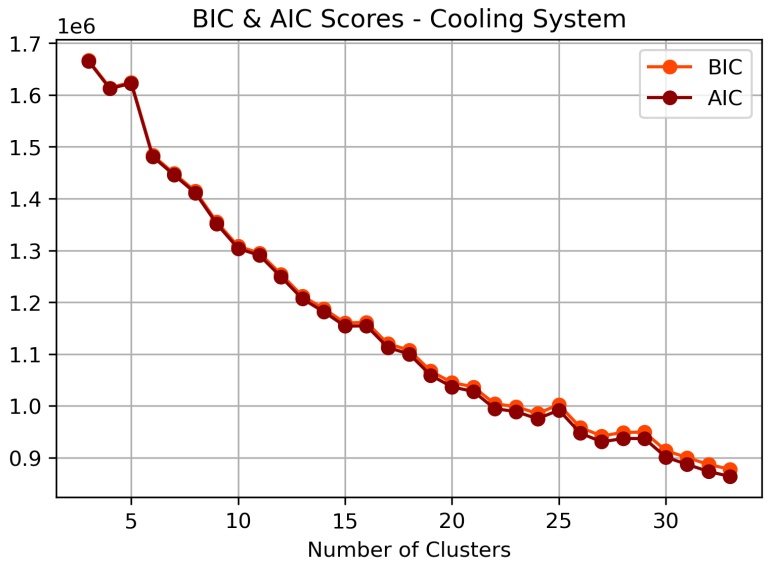

Figure 56: BIC and AIC plot for the cooling system.

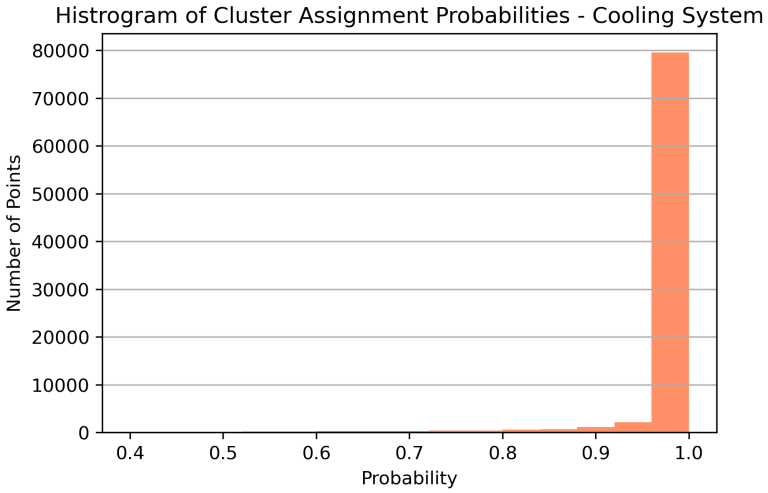

Figure 57: Distribution of cluster assignment probabilities for the cooling system (GMM).

equal to 582 distinct points, which approximately correspond to 2.02% of the tested points.

Intake & Exhaust System

The application of the anomaly detection algorithm with GMM in the intake & exhaust system follows the same approach as in the cooling system. The possible cluster values in this case are $[2, 29]$ and, as indicated by BIC and AIC the optimal is found to be k = 13. The silhouette score indicated a value of k = 6.

The cluster assignment probabilities are obtained after the training process of the GMM clustering algorithm. As seen in Figure 60, the distribution of probabilities has a longer tail compared to the cooling system. The first option would constitute of setting the threshold value equal to around 0.75 resulting in a ratio of anomaly between 7-10%. The second option is to assume that this distribution corresponds to anomaly-free data and set the threshold value accordingly. Following that option, the threshold value is set at P = 0.58, resulting in a total of 594 detected anomaly points. The percentage of those in the total test point is equal to 2.06%.

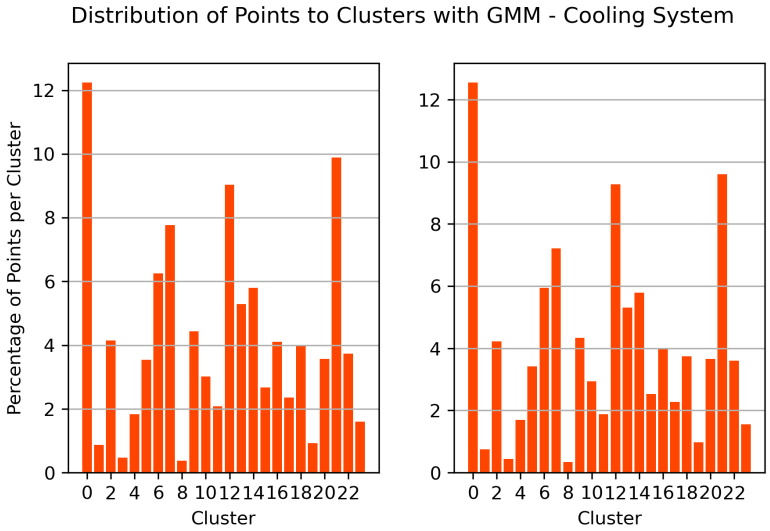

Figure 58: Distribution of points to clusters between train (left) and test (right) phases with GMM in the cooling system.

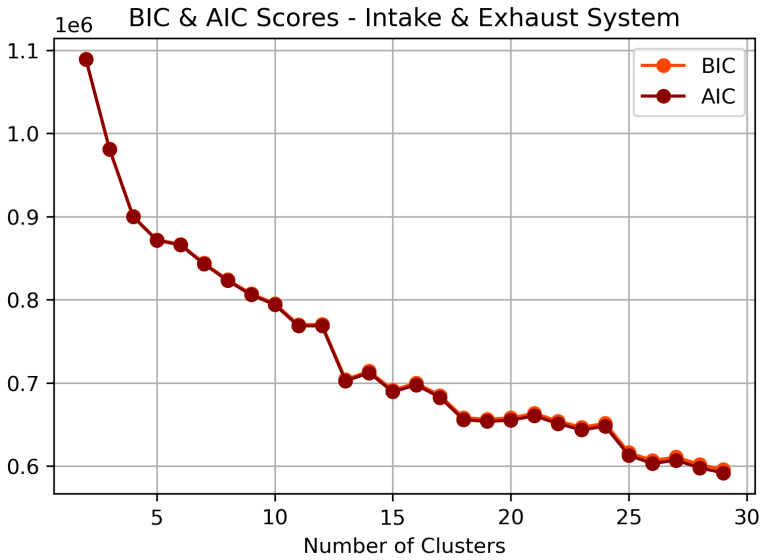

Figure 59: BIC and AIC plot for the intake & exhaust system.

This relatively small percentage of anomalies has no effect in the distribution of points to the 13 clusters between the train and test datasets, which is almost identical (Figure 61).

Discussion Over GMM Results

GMM offers a probabilistic clustering approach based on Gaussian distributions. They offer probability estimates for the assignment of every point in any cluster, which is found to be helpful with points that are between cluster borders. Furthermore, the probabilistic cluster assignment and the probabilistic anomaly detection criterion offer a diversified and more robust approach to the anomaly detection problem, when compared to the distance based criteria. On the other hand, this method requires the number of clusters to be predefined, which presents a challenge and requires computational time in order to find the optimal value. Additionally, the method relies on K-Means and its random initialization for the first iteration. The last

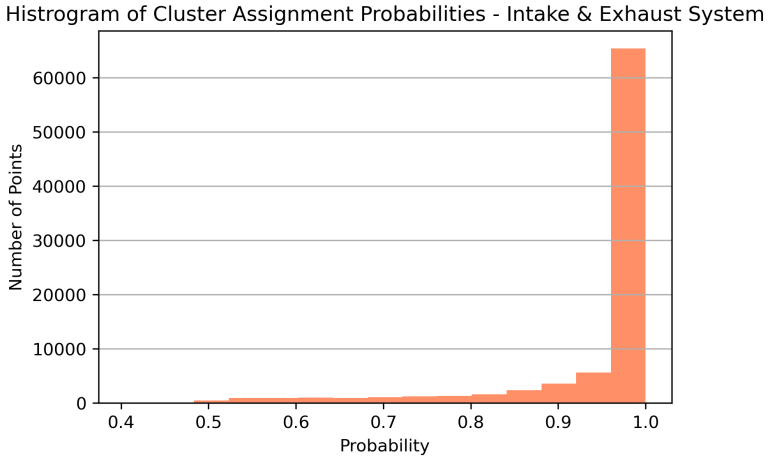

Figure 60: Distribution of cluster assignment probabilities for the intake & exhaust system (GMM).

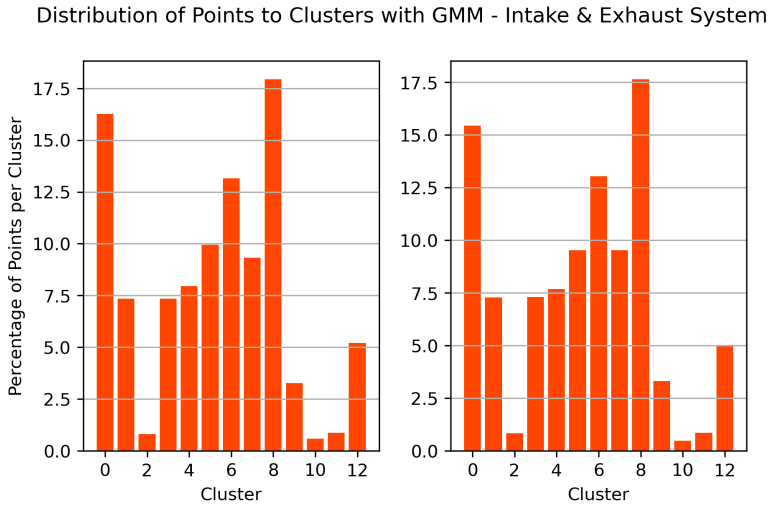

Figure 61: Distribution of points to clusters between train (left) and test (right) phases with GMM in the intake & exhaust system.

drawback is the method's poor performance when the data are not normally distributed.

5.3.3 DBSCAN Algorithm Results

This method possesses the distinct advantage to identify outliers or anomalies right from the training phase. This is an asset of this particular method due to the capability of eliminating possible remaining anomalies in the train dataset. In this scenario, the assumption is made that the training dataset does not contain anomalies, thus leading to the expectation of encountering only a small number of outliers points.

The algorithm's performance relies on two parameters which must be tuned. Through the procedure described in Theoretical Background section, the problem is deconstructed in a way that necessitates the tuning of only one of these parameters, namely MinPts. The process involves an iterative approach, where different values of MinPts are explored. For each iteration, the optimal value of ε is determined, and subsequently, the DBSCAN algorithm is applied. The clustering results are then assessed, and the iteration yielding the highest score is selected, ultimately leading to the identification of the optimal parameter combination.

The number of clusters in this method is not known a priori. Instead, the number of clusters is determined dynamically during the execution of the algorithm. Larger MinPts tends to create

less, but bigger, clusters, whereas smaller MinPTs creates a numerous compact clusters.

In the context of both the cooling and intake & exhaust systems, the process of determining the optimal MinPts value followed a two-step procedure. Initially, a broad array of candidate values was generated, with these values spaced at larger intervals. Subsequently, the winning parameter from this initial array served as the central value for another array, which was designed to explore values in close proximity to this central parameter. The final selection for MinPts was based on the best parameter identified within this more refined array.

As for the anomaly detection task, this is based on ε parameter. If a test point's distance to each closest core point is larger that ε, then this point is considered an anomaly.

Cooling System

The optimal parameters of the DBSCAN anomaly detection algorithm in the cooling system were found to be equal to MinPts = 216 and $\varepsilon = 1.568$. This result is far away from the initial estimation of MinPts $= 2 \cdot$ Columns, proposed by Ester et al. (1996). The optimal value of ε was found through the elbow in the average k-NN over number of points graph, with k = MinPts = 216.

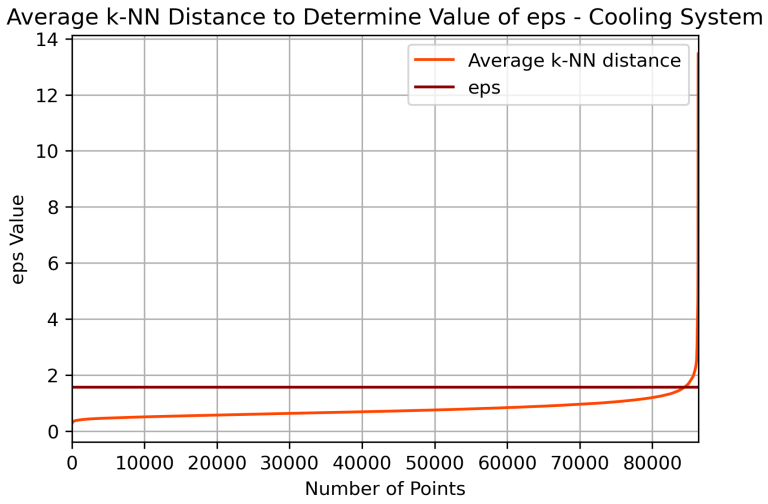

Figure 62: Average k-NN distance graph to specify optimal value of ε in the cooling system (DBSCAN).

As a density based method, DBSCAN clusters data points based on their proximity, which is determined by the distance parameter ε. In this particular case, the algorithm clustered the training dataset in four clusters. Notably, one of these clusters stands out as it encompasses nearly 90% of all the training data points, while the remaining three clusters collectively accommodate the remaining data points. In a similar manner, the distribution of points to clusters in the test phase results in the same image. This clustering pattern suggests the presence of a dominant cluster that captures a substantial portion of the dataset, while the other clusters represent more specialized or less prevalent patterns within the data. This results in 465 points identified as anomalies, corresponding to 1.62% of the test data.

Intake & Exhaust System

The optimal results of DBSCAN hyper-parameters in the intake & exhaust system were found to be MinPts = 133 and $\varepsilon = 0.469$. Similarly to the cooling system, MinPts value was very different from the initial estimation of Ester et al. (1996).

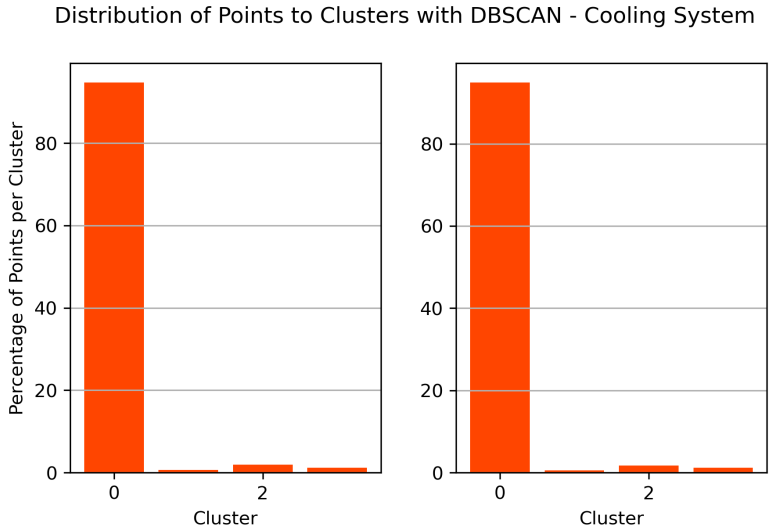

Figure 63: Distribution of points to clusters between train (left) and test (right) phases with DBSCAN in the cooling system.

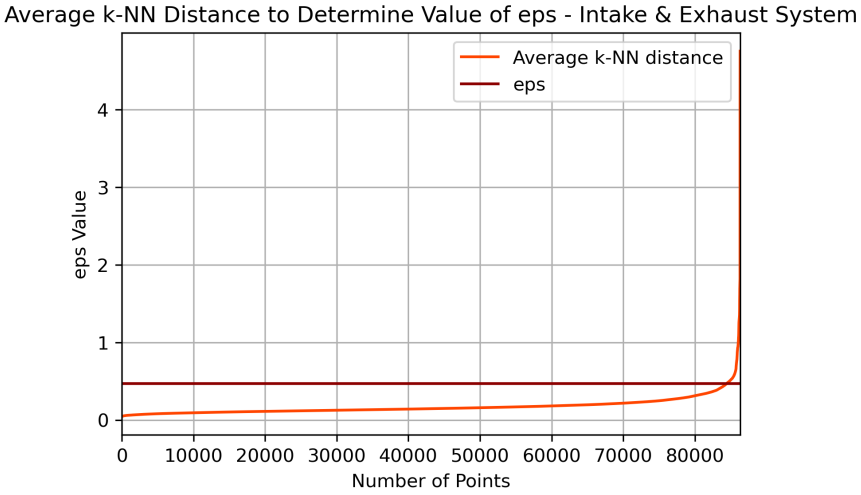

Figure 64: Average k-NN distance graph to specify optimal value of ε in the intake & exhaust system (DBSCAN).

In contrast to the previous scenario, the analysis indicated the presence of a significantly larger number of clusters in this case, 31 in number. Furthermore, during both the training and testing phases, it appears that data points are distributed more evenly among these clusters, suggesting a more balanced and comprehensive clustering outcome.

The shift towards smaller values for the MinPts parameter, which subsequently leads to a reduction in the ε parameter, is a significant contributing factor to this observed increase in cluster number. Additionally, it is crucial to note that this shift in hyper-parameters results in the emergence of numerous smaller clusters, which can capture more subtle data patterns. However, it also raises concerns about the increased sensitivity to anomalies during the test phase. The exact trade-offs between cluster degree of detail and anomaly detection performance require further investigation and validation. The model detected 1006 anomaly points, 3.31% of the total test data points.

Discussion Over DBSCAN Results

The application of DBSCAN in the cooling and intake & exhaust systems showcased another

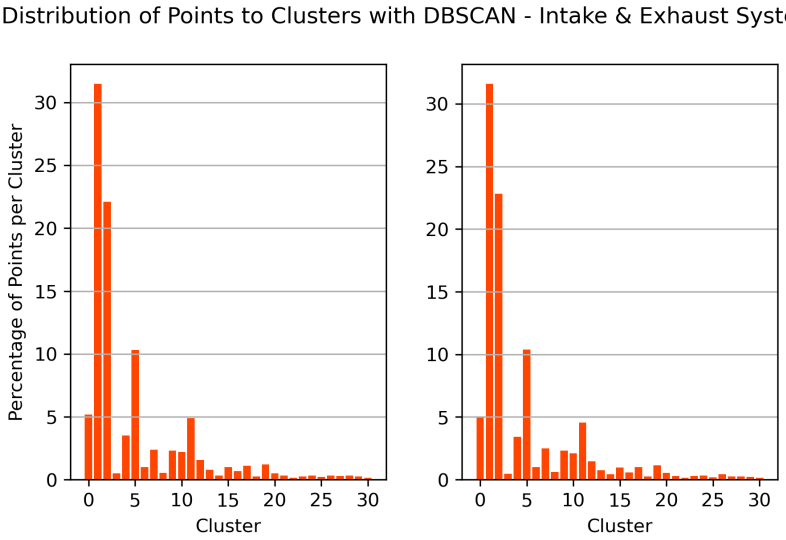

Figure 65: Distribution of points to clusters between train (left) and test (right) phases with DBSCAN in the intake & exhaust system.

characteristic of the algorithm that has not been taken into consideration. Fewer, big clusters lead to less anomalies, whereas more and compact clusters lead to more anomalies. This outcome can be observed in DBSCAN results and in the comparative results table of the next chapter (here). This clustering algorithm is density based and does not require the number of clusters to be given in advance. Instead, it is determined based on the algorithm hyper-parameters while it is executing. In parallel with the categorization of points as outliers in the training phase, this algorithm is considered to be more advanced when compared to K-Means or GMM. Hyper-parameter selection is of critical importance since the algorithm is very sensitive to changes in them. This could also be a potential reason for the observed anomaly detection results.

5.3.4 SOM Threshold Based Anomaly Detection Algorithm Results

The utilization of SOMs leads to a clustering result which is comparable to the other methods but also different. SOMs assign points to a predetermined number of nodes, which are either arranged in a 1D or a 2D pattern. Node positions are updated with every iteration of the algorithm, creating a more accurate clustering result. In the 2D arrangement, which is used in this study, more nodes are updated given a certain radius of influence, due to their more compact placement. The number of nodes is calculated according to the number of observations (n) in each dataset by the equation round($\sqrt{5 \cdot \sqrt{n}}$). The resulting map size is equal to 38×38. Furthermore, the rate of node weight update within the radius of influence of each node can be equal, or given by a Gaussian distribution, resulting in better tuned map. The second approach is employed in this study. Regarding the learning rate, it is reduced based on the number of iterations in order to avoid over-fitting of the algorithm. Radius of influence is also updated with a similar function.

The residuals of all test data points are calculated following the training of the map. The calculation involves calculating the average of all training points assigned to each node as a form of test point reconstruction. For each test point, the residual is equal to the point itself minus the reconstruction. The anomaly detection part of the algorithm is based on a threshold value on this residual. This value is unique for each system and it is found through the distribution of average residuals. The specifics of the operation and the algorithm parameters are described separately for each system.

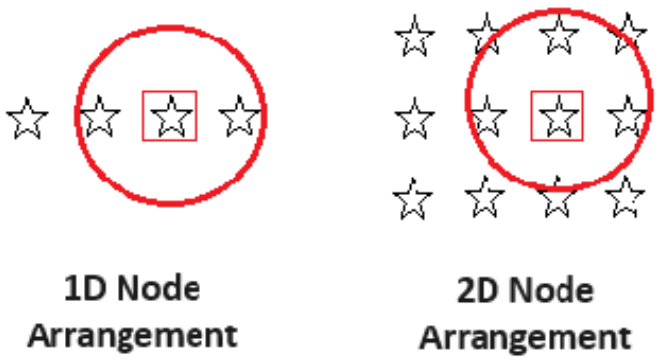

Figure 66: Difference in radius of influence between nodes in 1D vs 2D SOM node arrangement.

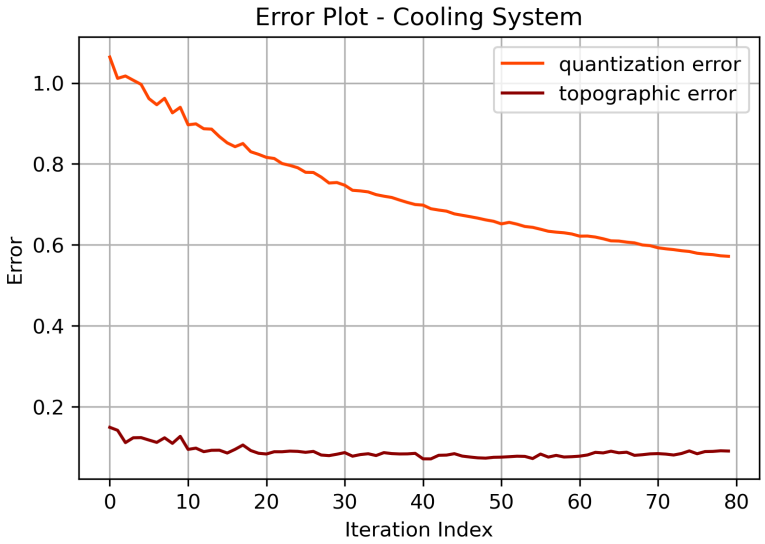

Figure 67: Quantization and Topographic errors plot of SOM in the cooling system.

Cooling System

The optimal values for the initial learning rate and radius of influence are found through experimentation with different values. For the particular system they are equal to 1.1 and 0.8 respectively. The SOM is trained for 80 iterations or epochs. As can be seen in Figure 67, this parameter combination provides a decreasing quantization error as iterations progress, along with a consistent topographic error from the 40th iteration and onwards.

A plot of point distribution to clusters between the train and test phase is required in order to evaluate the clustering performance. The number of clusters is significantly larger compared to the previous methods since each neuron and its assigned points shape a cluster. $38 \times 38 = 1444$ neurons are present. To make visualization possible, the nodes of the SOM are clustered with K-Means. The optimal number of node clusters is found to be 13. The cluster each node is assigned to may be seen in Figure 68. Each node is identified by the position it held in the original grid. As far as for the points distribution to these node clusters, it is similar between the train and test phases, indicating a good resulting map.

Regarding the residuals and their distribution, this is depicted in a plot, and the threshold for anomaly detection is established based on this visualization. In the particular system, the majority of residuals is between 0 and 0.3 and the distribution is tailed towards bigger values. This results in the threshold value being set at 0.65. Each point with residual $\geq$

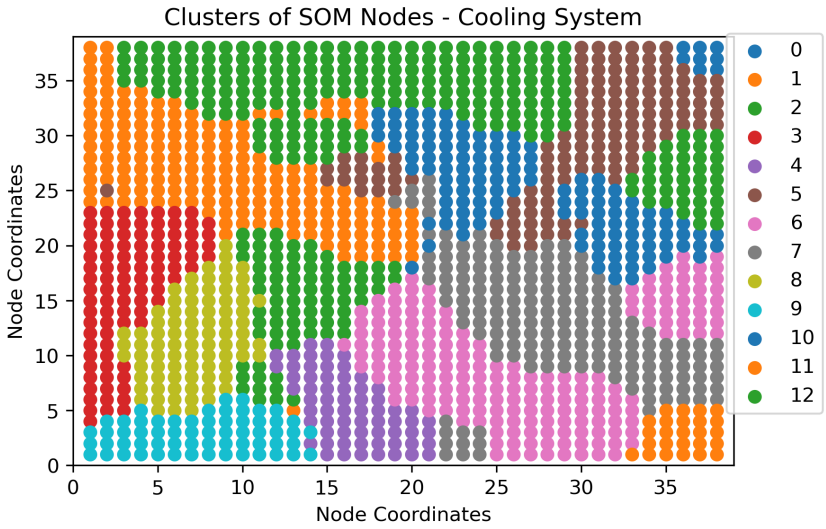

Figure 68: Clusters of SOM nodes plot in the cooling system.

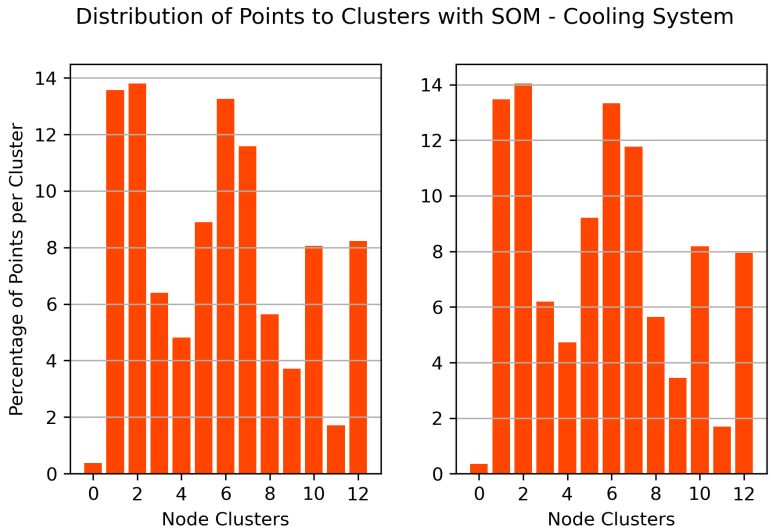

Figure 69: Distribution of points to SOM node clusters between the train (left) and (test) phases in the cooling system.

0.65 is categorized as an anomaly. 822 anomalies have been identified by this combination of parameters. This number is equivalent to 2.86% of the test dataset.

Intake & Exhaust System

Similarly to the cooling system, the SOM algorithm in this dataset is also trained for 80 iterations. The optimal initial values of learning rate and radius of influence are equal to 1.15 and 1.85 respectively. The plot which visualizes the decrease in quantization and topographic errors over the number of iterations shows a negative trend in the quantization error and a steady value of topographic error after decreasing for 50 iterations.

The optimal cluster number for the nodes is found to be 10. The cluster to which each node is assigned to is displayed in Figure 72. Based on that, the distribution of points in these ten node clusters between the train and test phases may be observed in Figure 73. Again, the distributions display high similarity, an indication of the capability of the model to effectively

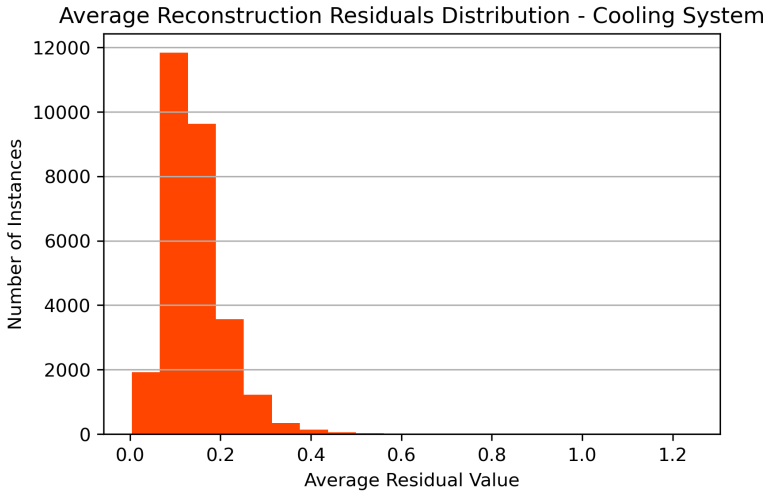

Figure 70: Distribution of reconstruction residuals in the cooling system.

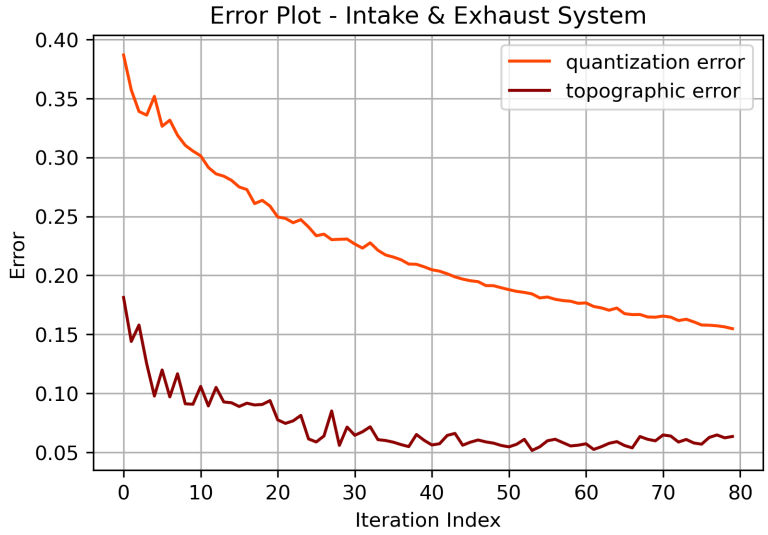

Figure 71: Quantization and Topographic errors plot of SOM in the intake & exhaust system.

cluster unseen data, resulting in a favorable outcome.

The reconstruction procedure in this dataset lead to a more concentrated distribution of residuals, with values ranging between 0.1 to 0.2. Based on that and the tail of the distribution towards the higher values, the threshold value is set at 0.3. The algorithm detected 452 anomalous points, a number that translates to a ratio of anomalies value equal to 1.57%.

Discussion Over SOM Threshold Based Anomaly Detection Results

The main drawback of SOMs according to this study is the plethora of parameters that must be tuned. For example, it is required to specify values for map size, number of iterations, learning rate, size of neighborhood and others. Additionally the algorithm requires a lot of computational time making the tuning process difficult.

The threshold based anomaly detection delivered good results due to the choice of parameters. The map was created to have enough nodes to allow for good reconstruction, which was based on the average of the points assigned to a node. As a result, this method is sensitive to

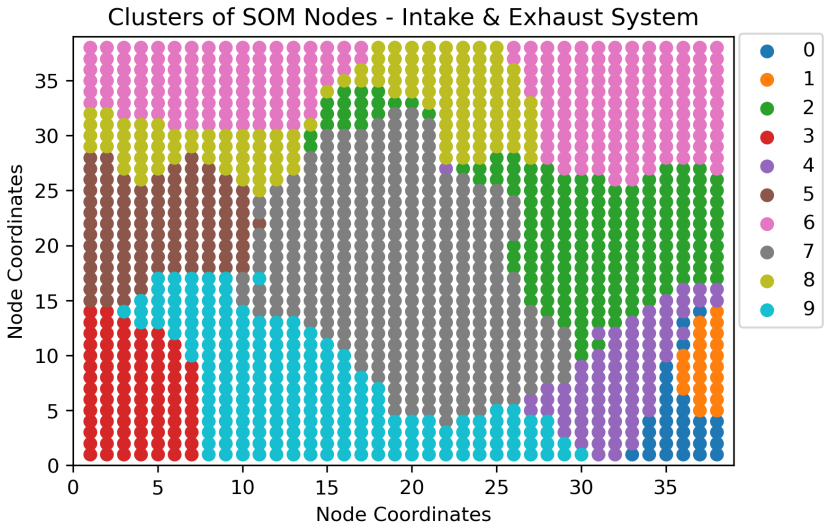

Figure 72: Clusters of SOM nodes plot in the intake & exhaust system.

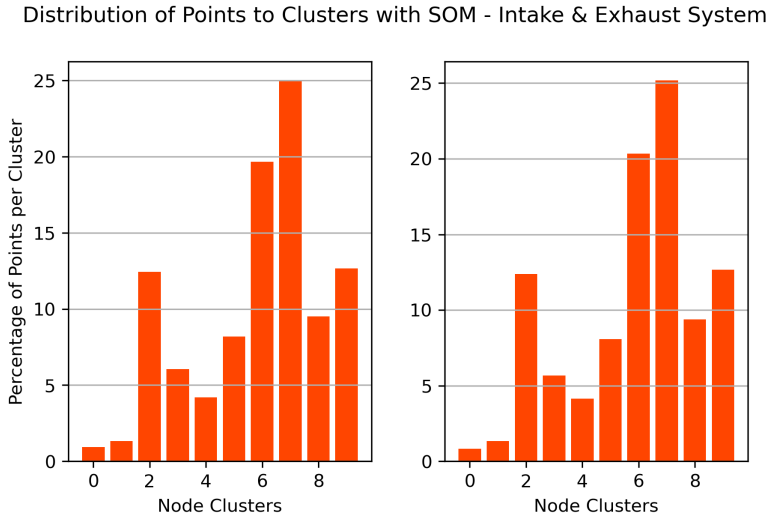

Figure 73: Distribution of points to SOM node clusters between the train (left) and (test) phases in the intake & exhaust system.

the quality of the map. The selection of the threshold values which are responsible for anomaly detection was performed by averaging the residuals, based on the specific characteristics of each system. More refined approaches regarding the determination of the threshold values may be developed in future research.

5.3.5 SOM Clustering Based Anomaly Detection Algorithm Results

In this version of the anomaly detection with Self Organising Maps, the anomaly detection part of the reconstruction residuals is handled by a clustering algorithm. Initially, the DBSCAN algorithm was employed, and the noise points it detected were classified as anomalies. The benefits of using DBSCAN for the task can be summarized in its ability to detect anomalies by itself and that the number of clusters must not be predetermined. This plan was later abandoned since almost no anomalies were detected. Instead, K-Means clustering has been employed. In contrast to DBSCAN, this algorithm requires prior specification of the cluster number. Additionally, it lacks autonomous anomaly detection capabilities since all points must initially be assigned to clusters. This implies that points situated at a considerable distance

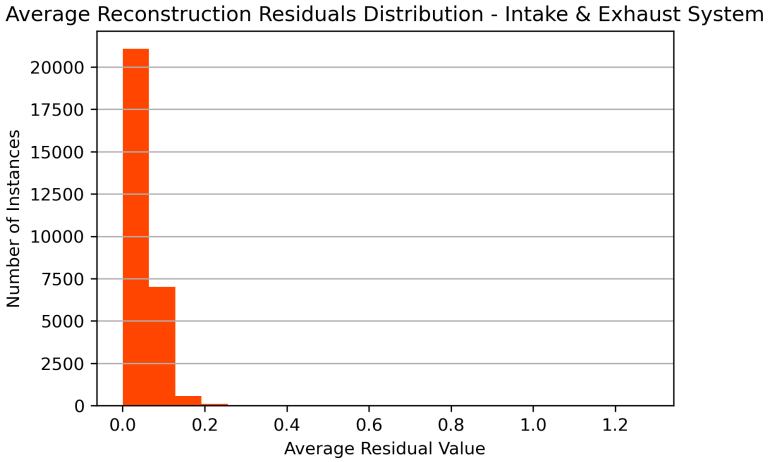

Figure 74: Distribution of reconstruction residuals in the intake & exhaust system.

from the centroid of a cluster could potentially be anomalies. Consequently, the approach of anomaly detection with K-Means is utilized, which is to classify points as anomalies if their distance from the cluster centroid exceeds the 98th percentile of all distances between cluster points and the cluster centroid (for the particular cluster).

The training and testing phases of the algorithm are identical between the clustering and threshold based anomaly detection methodologies with SOMs. Therefore, attention will be directed solely towards the outcomes of anomaly detection when employing this approach.

Cooling System

In the cooling system, the residuals have been clustered in two clusters. This number of clusters was suggested by the silhouette coefficient and the CH index. No elbow was identified in the Sum of Squared Error plot. By applying the anomaly detection criterion to these two clusters, 576 data instances were categorized as anomalies. As a percentage of total points this number is equivalent to 2.00% of them.

Intake & Exhaust System

The clustering of the residuals resulted in two clusters. Similarly to the cooling system the optimal number was found by prioritizing the silhouette coefficient and the CH index as no optimal number was identified by the elbow method. 576 points were detected as anomalies corresponding to 2.00% of the test dataset points.

Discussion Over SOM K-Means Based Anomaly Detection Results

The advantages and disadvantages of the SOM training process have been discussed in the threshold based version of the algorithm.

With regards to the anomaly detection scheme employed by this method, the clustering of residuals is proposed as an alternative to threshold based criteria. The method for categorizing points as anomalies was the same as in the K-Means algorithm, with the difference being that it has been applied directly to the obtained clusters and not in a testing phase. Overall, it produced results that are comparable to the threshold based method, something that is promising for future research.

6 Discussion

This section dives into the outcomes of the anomaly detection models, encompassing their evaluation and discussion. It centers on deviations from anticipated results, as well as the advantages and disadvantages associated with various approaches. Additionally, it investigates the detection of simulated anomalies within one system and analyzes the impact of dimensionality reduction within the same context.

6.1 Discussion Over Anomaly Detection Results

Upon the initial examination of the outcomes produced by the anomaly detection algorithms, it becomes apparent that the various methodologies under investigation delivered remarkably comparable results. They demonstrated similarities in both the percentage of detected anomalies within the test dataset and the manner in which data points were distributed among the clusters between the two phases. This convergence in results suggests consistent performance across the diverse approaches evaluated during the analysis.

Table 7: Anomalies detected by each algorithm in the cooling and intake & exhaust systems as percentage of total test data points.

Detected Anomalies as Percentage of Test Data					
System	K-Means	GMM	DBSCAN	SOM (Thresh.)	SOM (Clust.)
Cooling System	2.08%	2.02%	1.62%	2.86%	2.00%
Intake & Exhaust System	1.87%	2.06%	3.50%	1.57%	2.00%

In order to further analyze the results, it is important to examine the detected anomalies from the different methodologies in parallel. The expectation is to identify a common set of data points that are flagged as anomalies across all applied models. An additional expectation revolves around the discovery of shared anomalies between two or more of the systems.

Concerning the first expectation, all methods detected similar anomalies between June-2021 to November-2021. The selection of time window was chosen so that the detected anomalies are clearly visible. In the cooling system, K-Means and the two SOM based algorithms identified similar points as anomalies. GMM and DBSCAN, detected points in the same time series areas as the other methods, although scattered and not as many when compared to the others. In the intake & exhaust system, all methods except K-Means delivered comparable results. One may consider DBSCAN's detected anomalies to be more sparse compared to the other methods. In this time window, K-Means fails to identify the majority of anomalies that were detected by the other methods.

With respect to the second expectation, several of the detected points across the different methods seem to be common between the two systems. If the same data point is considered as an anomaly in more than one system, the likelihood of being a true anomaly is increased. In order to quantify this into a meaningful outcome, the number of common unique anomalies from all methods between the two systems will be used.

The number of total anomalies detected from all methods in the cooling system is 2216, whereas in the intake & exhaust system this number is 2142. As percentages of the test dataset, these numbers are transformed to 7.70%, and 7.45%. The number of common points between the systems is 562, approximately 26% of the average detected anomalies. This number showcases the performance of the employed methodology; detecting anomalies in ME sub-systems and then comparing and finding common points between those.

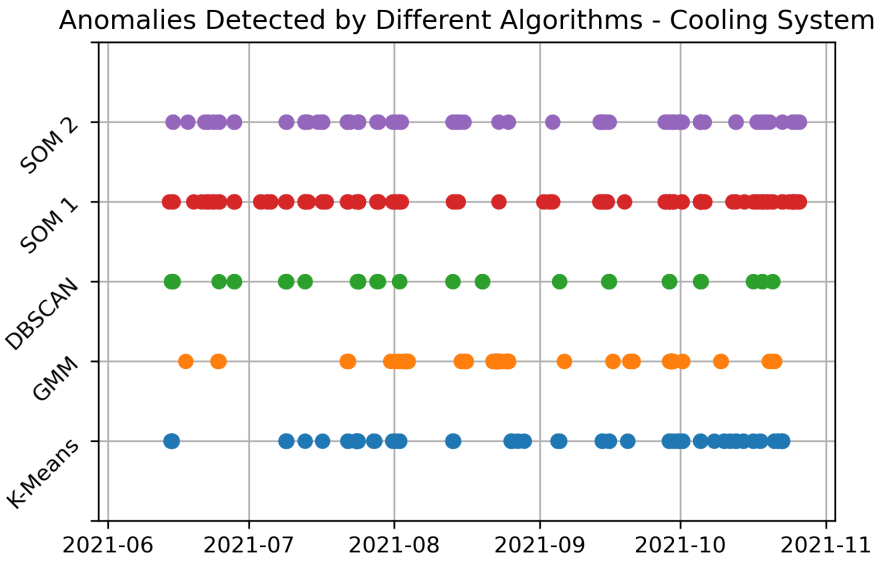

Figure 75: Comparison of detected anomalies by the different methods in the cooling system.

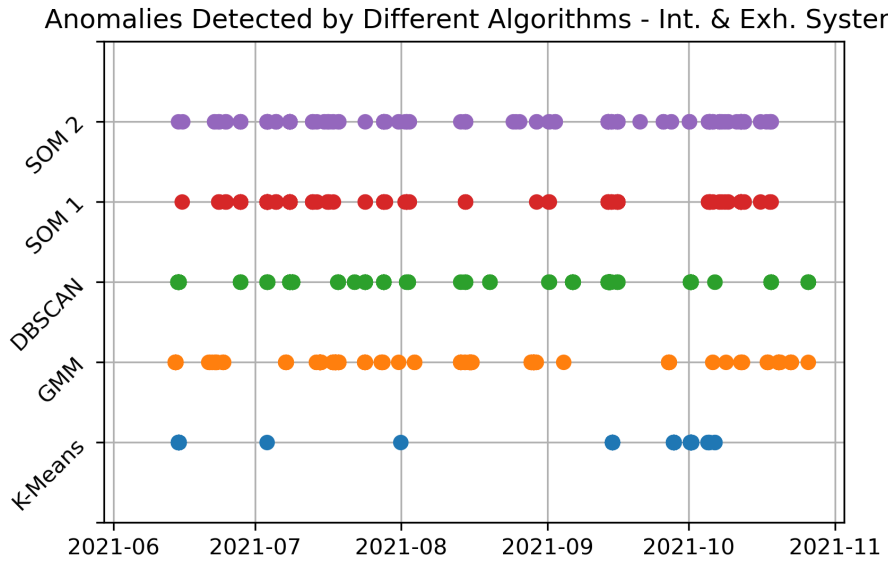

Figure 76: Comparison of detected anomalies by the different methods in the intake & exhaust system.

6.1.1 Proposal to Improve Results: Ensemble Anomaly Detection

When considering as anomalies all the unique detected points by the different methods, the process of anomaly detection in each ME system identified that 7.5-7.7% of the test dataset consists of anomalies. Even though the contents of the test data are not known, one could potentially argue that these percentages are high, especially compared to the anomaly percentages of the individual algorithms.

The proposal is to apply anomaly detection based on an ensemble of methods. By applying all algorithms to a particular system and then processing the results in parallel, the categorization of a point as anomaly may be done only if it is present in more than one anomaly dataset. This ensures a more robust detection process since false positives would be excluded.

6.1.2 Ensemble Anomaly Detection Results

The anomaly points are reduced from 2216 to 545 and from 2142 to 659 respectively, with anomaly ratios of 1.89% and 2.29%, when the proposal is applied to the cooling and the intake & exhaust systems. These numbers are close to the results that the methods delivered when they were applied separately. This is a promising result as a first indication.

Time-series plots of selected parameters from each system have been created in order to further analyze the results. The visualization is used to highlight the capabilities of the ensemble method, as well as those of the proposed algorithms in general. The figures are focused on a relatively small time window to allow for visual accuracy of the results.

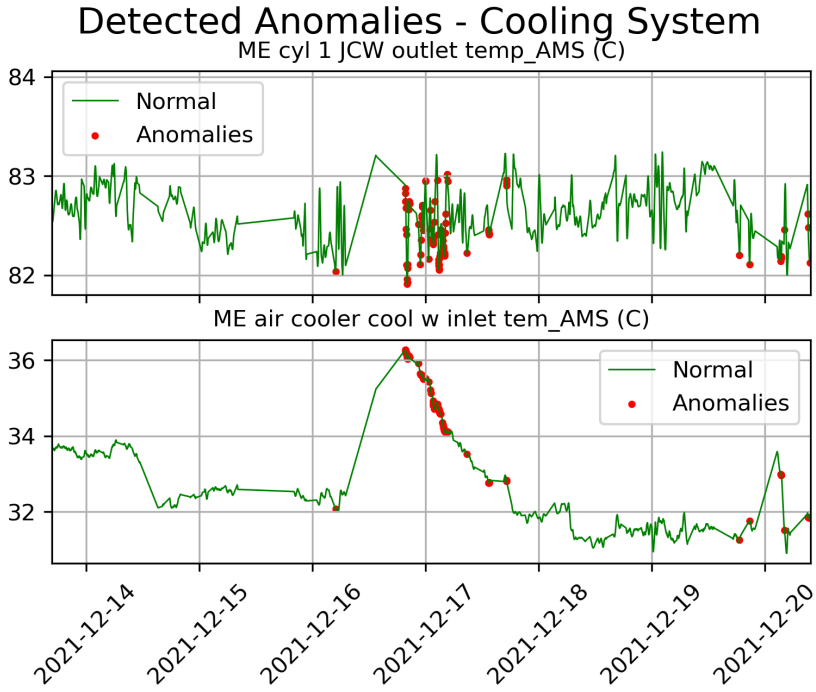

Figure 77: Time-series plot of normal points vs. detected anomalies in selected parameters of the cooling system.

Several outcomes may be derived from Figure 77 and Figure 78. First is the, now, proven capability of the methods to detect anomalies. The second concerns the detection of anomalies between the different systems of the ME. As observed, the same time points have been categorized as anomalies between both systems. However, at this stage of the research, it is not possible to determine which system or parameter was the initial cause of these anomalies. Furthermore, in both figures, anomaly points that might not be associated to anomalies from the visualized parameters may be observed. This is affiliated with anomalies caused by other parameters of the system.

6.1.3 Data Related Issues

In time series data, the order of observations holds significant meaning due to the presence of temporal dependencies which according to Vanem & Brandsæter (2021) contain information. This should be taken into consideration when splitting the dataset in parts. Despite the time-dependent nature of the data, they are randomly split it into training and test sets, overlooking the temporal sequence. While this is not ideal for time series data, it ensures that the training data adequately represents the test data.

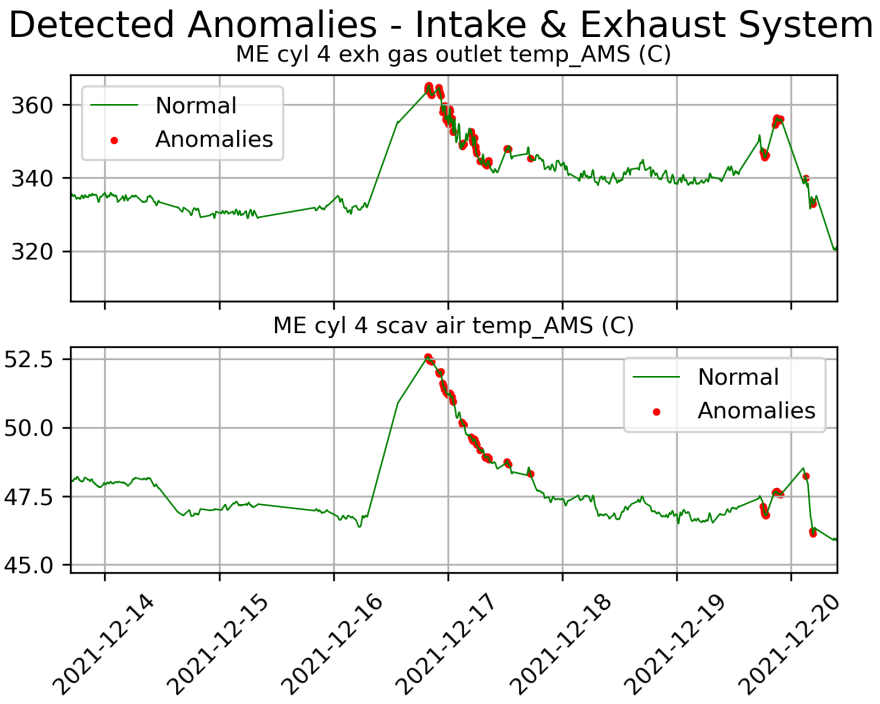

Figure 78: Time-series plot of normal points vs. detected anomalies in selected parameters of the intake & exhaust system.

To demonstrate the importance of representative training data, additional anomaly detection models have been trained based on datasets that were split by maintaining the time sequence. This means that the training dataset included the first 75% of observations, whereas the test contained the remaining 25%. The results showed more than 50% anomaly ratio in all ME systems. Moreover, there was no similarity observed in the distribution of points into clusters between the training and testing phases.

These issues are the main preventative factors from employing the methodology of this study for online anomaly detection. The training dataset should be representative of all future observations, which, in the case of splitting the dataset based on timestamp proved not to be true. To summarize, the training dataset should contain observations corresponding to all conditions that the vessel and the main engine may encounter.

A possible solution to combat the time dependent issues is to utilize a time series reconstruction model, apply it to the dataset and then obtain the residuals. Then, perform the anomaly detection algorithms on those.

Another data related issue is the quality of the training dataset. In unsupervised anomaly detection, it is assumed that the training data represent normal system conditions. If new observations significantly differ, they are identified as anomalies, potentially indicating system faults or deviations from standard operation. In this study, the dataset contained no known faults or anomalies though this may not be completely true. The thorough data preparation phase aimed to eliminate these anomalies, although it may not have been possible to achieve a 100% success rate in doing so. As a result, the detected anomalies may contain few false positive data points. Additionally, the training dataset may enclose some anomalies which are treated as normal, resulting in a few false negatives in the testing phase.

6.2 Effect of Dimensionality Reduction in Anomaly Detection Results

Dimensionality reduction offers a trade-off between decreased computational time, more efficient ML operations, and information loss. In this study, PCA has been utilized for the task.

This method transforms the dataset's correlated features to uncorrelated principal components (PCs). Each PC tries to captures as much of the dataset's variance as possible. As a result, every additional PC captures less of the system's variance compared to what the previous did. The methodology under which the optimal number of PCs is chosen may be found in PCA's methodology sub-section (see here).

The intake & exhaust system will be used as an example. The reason behind this selection is that this particular system had the biggest reduction in features by PCA, from 17 to 4, while maintaining 97.5% of the initial dataset's variance. The K-Means, GMM, and DBSCAN algorithms are applied in the non-PCA dataset and a comparative analysis of the results follows.

This dataset contains exactly the same data-points as the one used in the other steps of this study. The only difference between them is the number of features.

6.2.1 Application of K-Means in No-PCA Dataset

Through the iterations of the K-Means algorithm it has been found that the optimal number of clusters for the intake & exhaust system is 5. The algorithm searched in the range between 2 and 35. This number has been proposed by the CH index, the silhouette coefficient, and the Last Leap. The resulting distribution of points between the train and test datasets, after training the model and applying it to the test data, is comparable between the phases. This ensures a good clustering result. The model identified 541 anomalies which correspond to 1.88% of the test dataset.

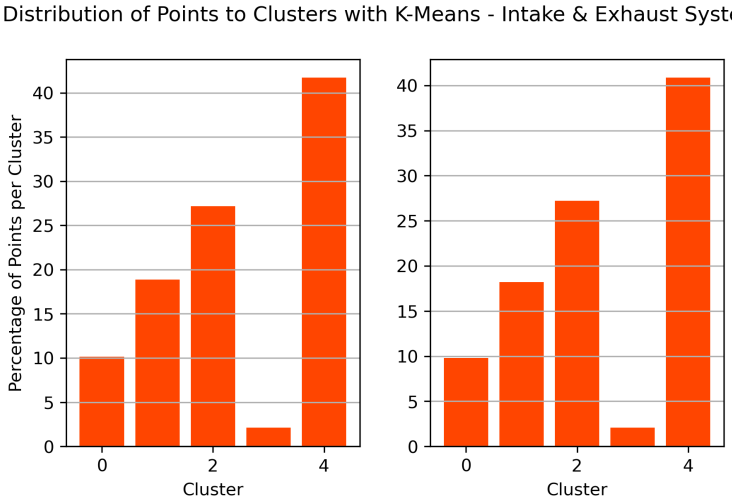

Figure 79: Distribution of points to clusters when using K-Means between the train (left) and (test) phases in the intake & exhaust system. The points of this dataset have not been transformed with PCA.

Figure 79 and the equivalent of the system where the original algorithm is applied to the data (Figure 55) share no similarities when it comes to distribution to clusters. It is uncertain if the equality in cluster number can be justified. Overall, the number of detected anomalies is comparable between the with and without PCA datasets.

6.2.2 Application of GMM in No-PCA Dataset

The potential number of clusters tested in the case of clustering with GMMs was $[12, 42]$. The optimal, according to BIC, AIC, and the silhouette coefficient was 40. The model has been trained based on that number. The distribution of the test points has been equivalent to what

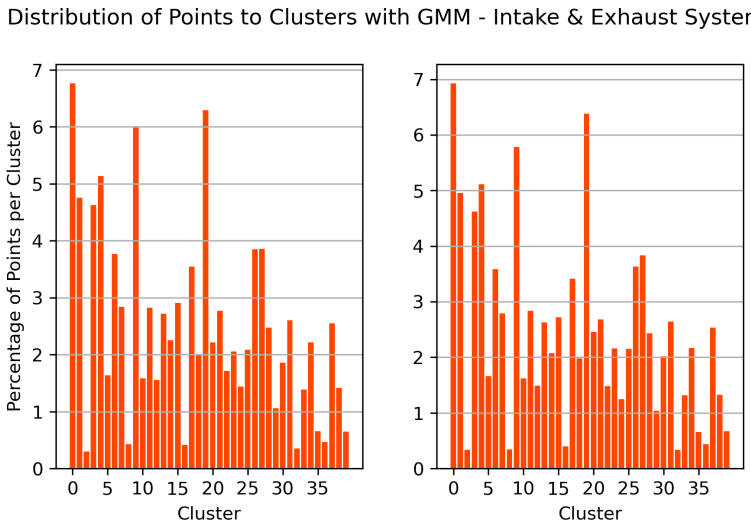

Figure 80: Distribution of points to clusters when using GMM between the train (left) and (test) phases in the intake & exhaust system. The points of this dataset have not been transformed with PCA.

found in the training phase. This point distribution has no similarities to the clustering of the same dataset with PCA.

The cluster assignment probability distribution of the training data is required in order to determine the anomaly detection threshold. Practically, the probabilities of all points exceed 0.9 and this is where the threshold is set. This result, when compared to the tailed distribution of the with PCA dataset, differs significantly. The procedure for setting the threshold has became more straightforward in this case. Only 324 anomalies have been detected by this model (corresponding to anomaly ratio 1.13%), which is less than the original model detected (594 anomalies).

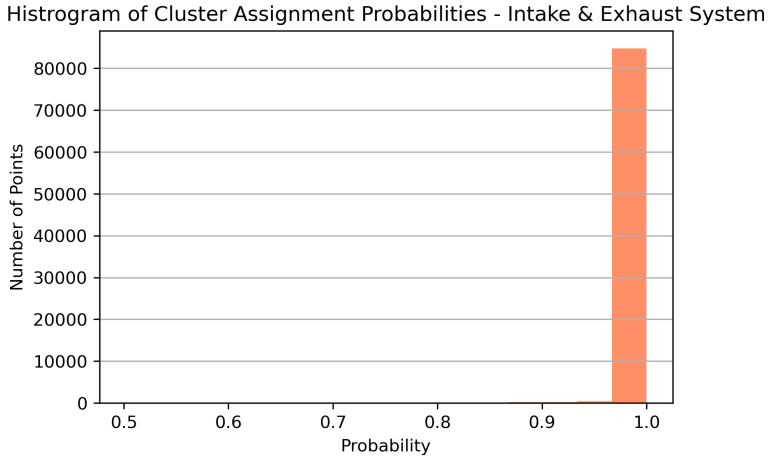

Figure 81: Distribution of cluster assignment probabilities of GMM for the intake & exhaust system in the dataset without PCA.

6.2.3 Application of DBSCAN in No-PCA Dataset

The optimal hyper-parameter values of DBSCAN have been found equal to MinPts = 63 and $\varepsilon = 0.5339$. The number of clusters developed through the training process with these parameters is equal to 64. Again, the number of clusters found in this example does not align

with the algorithm where PCA has been performed.

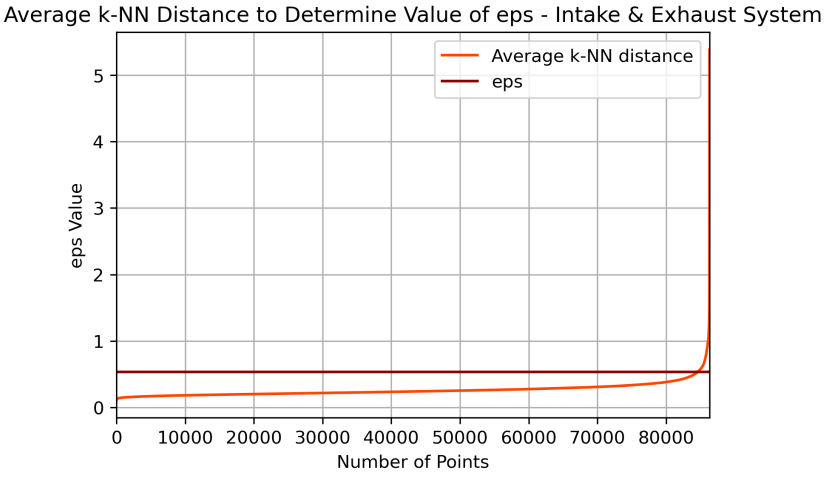

Figure 82: Average k-NN distance graph to specify optimal value of ε in the intake & exhaust system without PCA applied to the dataset (DBSCAN).

Using this method, the model detected a total of 798 anomalies. It is important to mention that the same method in the, transformed with PCA, dataset had also detected an increased number of anomalies compared to other methods. The ratio of anomalies value for DBSCAN is equal to 2.77%.

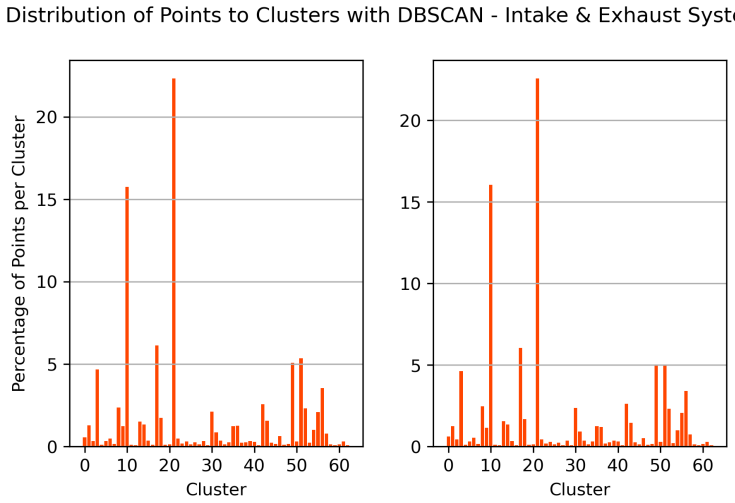

Figure 83: Distribution of points to clusters when using DBSCAN between the train (left) and (test) phases in the intake & exhaust system. The points of this dataset have not been transformed with PCA.

6.2.4 Comparative Analysis of Results Between With & Without PCA Datasets

Following the assumption that the transformed dataset contains only 97.5% of the variance of the original, the logical deduction is that the algorithms should detect fewer anomalies in it than in the original dataset. However, the initial significant observation which is derived when comparing the results is that more or equal anomalies are detected in the transformed dataset. Several reasons could have influenced this outcome which will be explained in the text that follows.

Table 8: Anomalies detected by each algorithm when applied to the intake & exhaust system with and without dimensionality reduction.

Detected Anomalies in The Intake & Exhaust System				
Algorithm / PCA	With	With [%]	Without	Without [%]
K-Means	539	1.87%	541	1.88%
GMM	594	2.06%	324	1.13%
DBSCAN	1006	3.50%	798	2.77%

1. There were multiple False Positive (FP) points categorized as anomalies in the detected anomalies datasets when dimensionality reduction was used. That should have been enough to influence the results.

2. Vanem & Brandsæter (2021) state that the first principal components are those carrying the transformed sensor signal, whereas the last carry mostly noise. Based on that, it is possible that by removing the remaining noise from the dataset (2.5% not explained variance), the dataset's quality has been enhanced along with the sensibility to true anomalies.

3. PCA may have identified combinations of the principal components that explain the anomalies better than the original parameters alone. By reducing dimensionality, PCA may emphasize certain patterns or anomalies that were not as evident in the original high-dimensional data.

Based on the analysis, it appears unlikely that all three points can hold simultaneously. In the case that point 1 is true, it implies that the anomaly detection has not been successful. The alternative scenario is that points 2 and 3 are valid. According to the findings so far, this is considered to be the case in this study.

Overall, the performance of the algorithms increased when utilizing PCA for dimensionality reduction. First, computational time was significantly reduced. Second, the dataset dimensions were decreased and the information was stored in a more compact format. Third, the algorithms delivered better, or at least equally good anomaly detection results.

6.3 Detection of Simulated Anomalies

The anomaly detection results highly depend on the dataset's quality. In this study the dataset contained no known faults. As a result, it has been difficult to evaluate the performance of the models. In order to do so, anomalies have been generated, according to the simulation methodology presented earlier (see here). The characteristics of these datasets are:

- The simulated anomalies are infused in a sequence of a parameter. Only some of the sequence points are altered.

- Altered parameter specifics: 80 point degradation sequence, 2 single point anomalies. The rest of the points have already been categorized as normal by the algorithms.

Additionally, the models have been tested with three datasets. The first contains one altered parameter and has total length of 250 points, 82 of which are the simulated anomalies. The second is similar to the first with the only difference being that the total length is 1500 points. The last simulated dataset has 1500 points length but contains simulated anomalies in two parameters.

Two questions arise from the way these anomalies have been simulated. The first refers to their placement within the time-series. The second question has to do with the possibility of encountering such data-points in real operational environments.

Regarding the anomalies placement, it is considered to not be important since time-sequence of data is not taken into consideration. Concerning the possibility of encountering the simulated instances in real life scenarios, they have been created in such way, based on the characteristics of the time-series that they simulate, in order to ensure a good result.

For the purpose of displaying the ability of the algorithms in detecting these anomalies, an analysis of the cooling system will be presented.

6.3.1 Simulated Anomalies Presentation

Dataset 1

The selected parameter of the cooling system on which the anomalies are simulated is the "ME Cylinder 3 Jacket Cooling Water Outlet Temperature". The parameter is sourced through the vessel's Alarm Monitoring System (AMS). Usually, this temperature is kept at almost constant levels and changes often occur due to faults. The time series plot for this parameter may be seen below. The two single point anomalies are on the left side of the figure, whereas the degradation pattern is on the right (Figure 84).

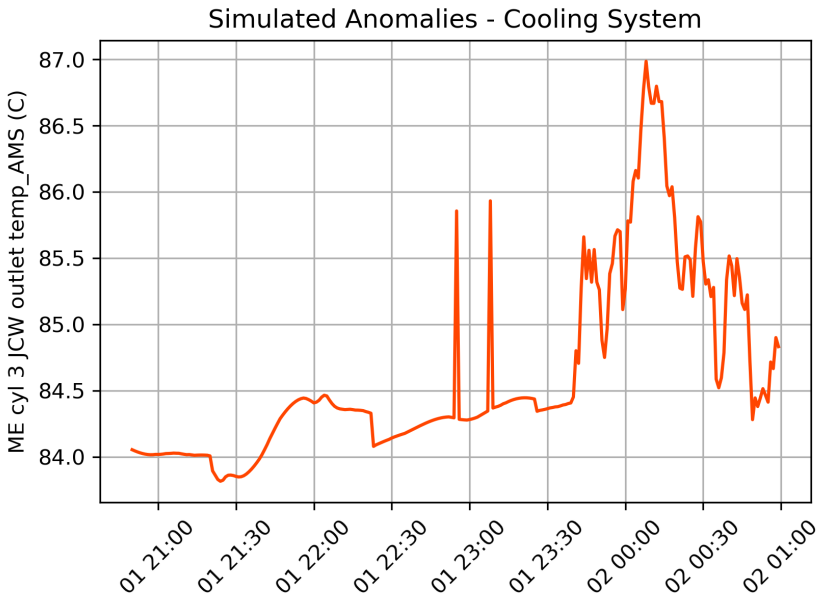

Figure 84: Time series plot of simulated anomalies in the ME Cylinder 4 Jacket Cooling Water Outlet Temperature of the cooling system (dataset 1).

Dataset 2

Similarly to the previous dataset, the simulated anomalies are infused in "ME Cylinder 3 Jacket Cooling Water Outlet Temperature" parameter.

Dataset 3

In this case, the simulated anomalies are infused in the jacket cooling water outlet temperature time-series of cylinders 3 and 4. The simulated anomalies in the second parameter have been placed at the same time-points where those of the first parameter are. By simulating anomalies in one more parameter of the system, it is expected that the scenario mimics better a real anomalous situation.

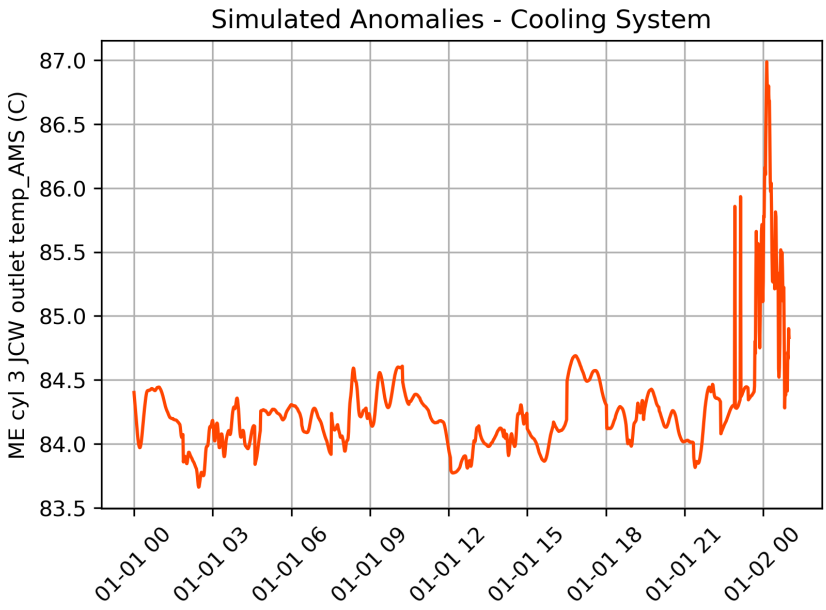

Figure 85: Time series plot of simulated anomalies in the ME Cylinder 4 Jacket Cooling Water Outlet Temperature of the cooling system (dataset 2).

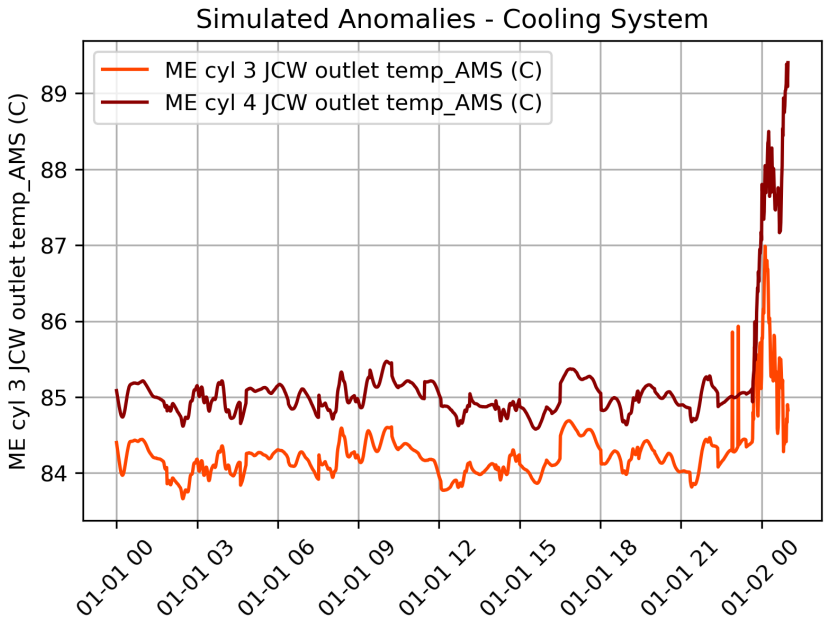

Figure 86: Time series plot of simulated anomalies in ME Cylinder 3 & 4 Jacket Cooling Water Outlet Temperature of the cooling system (dataset 3).

6.3.2 Results of Simulated Anomalies Detection

The models are trained with the same datasets as in the previous steps of the study. The main difference is that the test dataset is significantly more compact when compared to the 28310-point dataset that was originally used. Now, it contains the simulated anomalies and few normal points.

The accuracy metric, as proposed in Velasco-Gallego & Lazakis (2022a), is utilized in order to evaluate the results of anomaly detection. The mathematical formulation of this metric is

found below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{41}$$

where:

- TP: True Positive
- TN: True Negative
- FP: False Positive
- FN: False Negative

The results include the implementation of the anomaly detection algorithms with K-Means, GMM, and DBSCAN.

K-Means

In the K-Means implementation, the algorithm clustered the training data in 13 clusters, as this number has been found to be the optimal for this dataset. As expected, the distribution of test points to clusters displays no similarity with the train phase in any of the three cases that were tested. Furthermore, in the first dataset, this algorithm identified 41 anomalous datapoints, with no FP points. The algorithm managed to detected both single point anomalies. The performance of the algorithm did not change with the second dataset. Again, it managed to detect 41 anomalies in total. Both single point anomalies have been detected. When the third dataset was tested for anomaly detection, the algorithm managed to detect 75 of the 82 anomalous points.

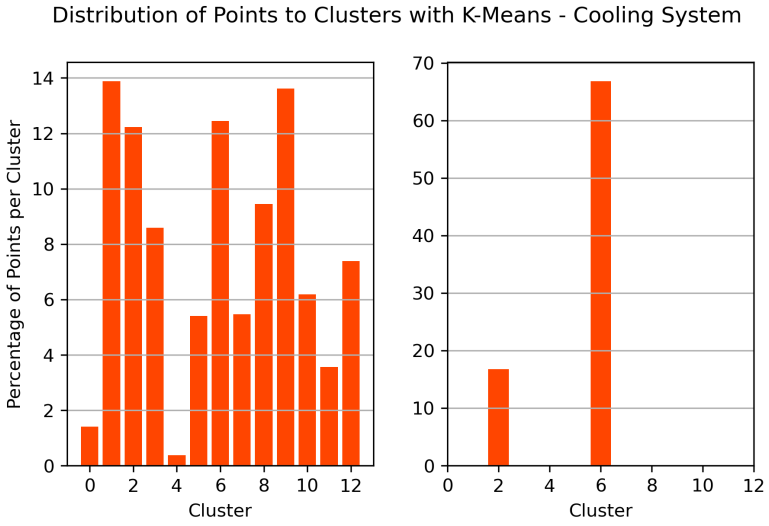

Figure 87: Point distribution to clusters between train (left) and test (right) phases with K-Means in the cooling system. The test dataset is "simulated anomalies Dataset 1". The distribution of points to clusters in the other 2 datasets is almost identical to the one presented here.

GMM

According to what presented previously, a good cluster number for the cooling system when using the GMM algorithm is 23. When the algorithm was tested with the first dataset, it identified only 9 TP points. One of the two simulated point anomalies has been detected. This result does not demonstrate the capabilities of GMM. In the anomaly detection implementation with the second dataset, the algorithm detected 126 points as anomalies. Out of those TP were

47 of them, with the remaining 79 being FP and 2 being FN. Similar results were obtained with the third dataset: 152 points detected, 68 TP, 84 FP, and 5 FN.

Overall, GMM underestimated the number of anomalies contained in the first dataset, whereas it overestimated the anomalies in the second and third.

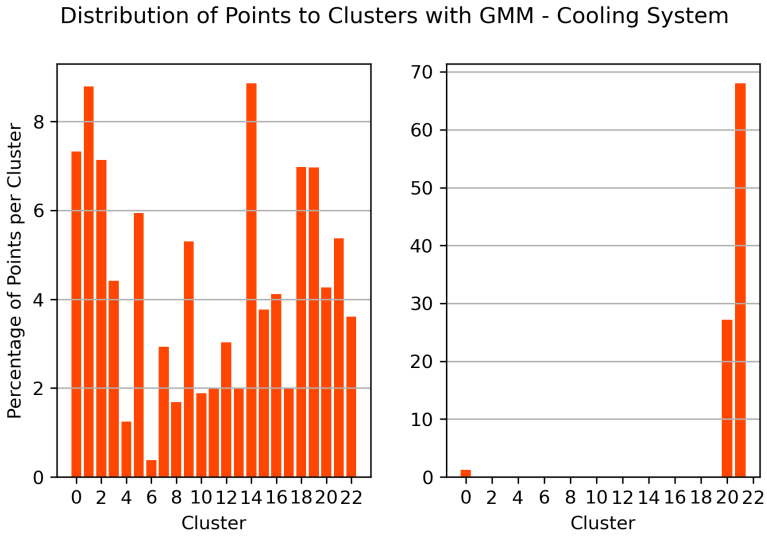

Figure 88: Point distribution to clusters between train (left) and test (right) phases with GMM in the cooling system. The test dataset is "simulated anomalies Dataset 1". The distribution of points to clusters in the other 2 datasets is almost identical to the one presented here.

DBSCAN

In the case of DBSCAN, as previously stated, the number of clusters is not defined in advance. In this case, it has been found to be 17. As far as the anomaly detection results are concerned, when dataset 1 was applied, the model detected 76 anomalies, all of them were TP. The model managed to detected both point anomalies. The anomalies detected from the second dataset were 79. The model detected both point anomalies. 76 of them were TP and 3 were FP. When the third dataset was applied, the model detected 81 anomalies. 79 of the 82 TP were detected, with 2 FP also being detected.

6.3.3 Comparative Evaluation of Simulated Anomaly Detection Results

The results indicated that the models are able to detect a respectable percentage of the simulated anomalies. K-Means and DBSCAN managed to achieve the results without identifying a lot of FP. On the contrary, GMM under-performed in the anomaly detection task with dataset 1 and performed averagely with datasets 2 and 3. Additionally, it detected significant amount of FP points.

Generally, the algorithms identified the point anomalies as well as lots of points of the degradation sequence. Again, the accuracy levels of the results should be enhanced by utilizing an ensemble approach. With the present results, DBSCAN's performance was superior when compared to the other methods.

Furthermore, the ability of the algorithms to capture the simulated anomalies is depended on their structure. In this example, the simulated time series was infused to only one or two

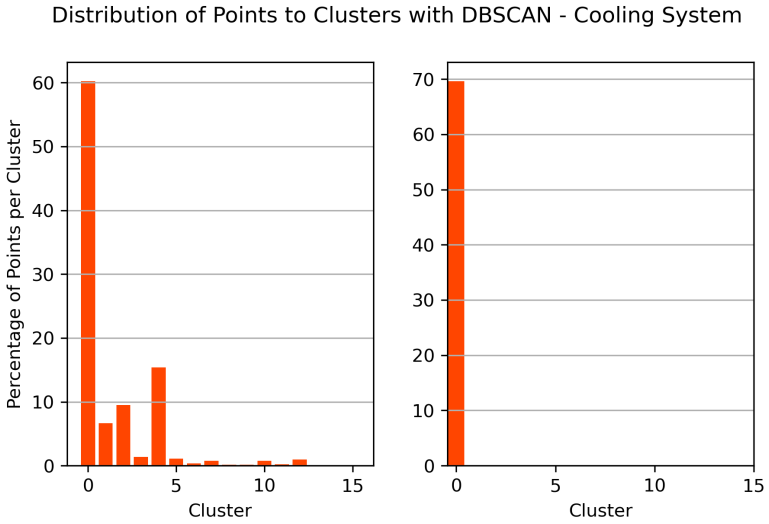

Figure 89: Point distribution to clusters between train (left) and test (right) phases with DBSCAN in the cooling system. The test dataset is "simulated anomalies Dataset 1". The distribution of points to clusters in the other 2 datasets is almost identical to the one presented here.

Table 9: Accuracy score for anomaly detection of simulated anomalies in the cooling system (dataset 1).

Accuracy Score Per Algorithm - Cooling System (Dataset 1)			
Algorithm	K-Means	GMM	DBSCAN
TP	41	9	76
TN	168	168	168
FP	0	15	0
FN	41	73	6
Accuracy	83.60%	70.80%	97.60%

Table 10: Accuracy score for anomaly detection of simulated anomalies in the cooling system (dataset 2).

Accuracy Score Per Algorithm - Cooling System (Dataset 2)			
Algorithm	K-Means	GMM	DBSCAN
TP	41	47	76
TN	1418	1372	1418
FP	0	79	3
FN	41	2	3
Accuracy	97.27%	94.60%	99.60%

Table 11: Accuracy score for anomaly detection of simulated anomalies in the cooling system (dataset 3).

Accuracy Score Per Algorithm - Cooling System (Dataset 3)			
Algorithm	K-Means	GMM	DBSCAN
TP	75	68	79
TN	1418	1343	1418
FP	0	84	2
FN	7	5	1
Accuracy	99.53%	94.07%	99.80%

parameters of the dataset. In a more realistic scenario, the affected parameters may have been more. The effect of PCA may have also influenced the results, due to the fact that it could have removed anomaly information explained by the lost percentage of variance.

A note should also be added concerning the "Accuracy" metric. This metric evaluates the anomaly detection process in general by taking into account TP, FP, TN, and FN and not just the percentage of detected anomalies. It is affected by the number of points in each category. For example, while having similar number of detected anomalies, DBSCAN's score increased from 97.6% in dataset 1 to 99.6% in dataset 2. This is caused from the increase in the number of normal points. As a result, the accuracy metric should be taken into account in parallel to the percentage of detected anomalies.

7 Conclusions

This study was involved in the anomaly detection of marine engine sensor data for condition monitoring. Several algorithms have been used, all of them utilizing unsupervised machine learning techniques. Each of the examined methods had different aspects where they excelled or they did not perform as well in. Nevertheless, through this research, several characteristics of the methods have been showcased. Robust methodologies have been developed in order to find optimal hyper-parameter values for the machine learning algorithms. Additional methods for finding optimal values used for anomaly detection purposes, such as thresholds, have also been established through extensive experimentation and optimization on multiple datasets.

Each of the utilized algorithms, namely K-Means, GMMs, DBSCAN, and SOMs had its own intricacies. It has been recognized that K-Means offered a simplistic and efficient approach, GMMs excelled in modeling more complex data distributions, DBSCAN exhibited robustness against noise, and SOMs provided valuable insights through the reconstruction process and the clustering of residuals. Nonetheless, each method had its own set of limitations, such as sensitivity to cluster numbers or assumptions about data distribution. Thus, it is suggested to use an ensemble of methodologies even if, as showed, all individual methods performed well, in order to have better anomaly detection results.

Other valuable outcomes included the potential positive effect of dimensionality reduction in complex marine engine system datasets and the ability of the models to detect simulated anomalies.

One of the main challenges associated with machine learning applications in the marine machinery environment in general is the quality of data. In order to construct robust algorithms, a plethora of data is required along with information about fault data. The latter may be used for classification purposes or they can be removed in order to create unsupervised models. Due to the absence of known faults or anomalies in this study, the approach used was to try and eliminate anomalies which may have been present in the data preparation phase and then assume that the remaining points displayed normal behavior. The algorithms have been trained with such datasets and the anomaly detection has been based on that assumption.

7.1 Future Work

Following the conclusion of this study and the valuable insights gained from it, it is important to highlight areas that could enhance the outcome of the present or that may hide potential for further advancements in the field of condition monitoring.

1. Exploration of Additional Clustering Techniques: The field of clustering offers multiple techniques, and further investigation into methods like hierarchical clustering or OPTICS can provide a broader perspective on anomaly detection. Techniques like K-Means++ and other emerging clustering algorithms may yield novel insights and improved anomaly detection capabilities.

2. Time Series Reconstruction & Clustering of Residuals: One of the main drawbacks of the present study is its inability to scale-up easily to an online anomaly detection methodology. Though clustering methods fit the anomaly detection task with their unsupervised character, proper representation of the test data in the training dataset is required to ensure robust anomaly detection. In the case where not many data are available for training purposes, combining time-series reconstruction techniques with clustering of residuals could enhance the ability to detect anomalies in a dynamic, online manner while preserving the temporal aspect of the data.

3. Anomaly Classification & Explainable AI: Developing a framework that can not only detect anomalies, but also classify them into specific fault categories would be valuable

for predictive maintenance and proactive decision-making tools. Future research can focus on developing models that, not only identify anomalies, but also provide explanations with the use of Explainable AI for why a particular data point is flagged as an anomaly. These tools could be great aid to operators and superintendent engineers.

4. Fault Prognosis: As a natural progression of the previous point, predicting faults before they occur is a significant goal. Future research can deal with fault prognosis techniques that utilize historical data, and anomaly detection models to predict future issues.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. (pp. 267–281). Akademiai Kiado.

Akinduko, A. A. & Mirkes, E. M. (2012). Initialization of self-organizing maps: Principal components versus random initialization. a case study.

Alia, S., Nasri, R., Meddour, I., & Younes, R. (2020). Comparison between sound perception and self-organizing maps in the monitoring of the bearing degradation. *International Journal of Advanced Manufacturing Technology*, *110*, 2003–2013.

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, *120*, 70–83.

Beşikçi, E. B., Arslan, O., Turan, O., & Ölçer, A. I. (2016). An artificial neural network based decision support system for energy efficient ship operations. *Computers & Operations Research*, *66*, 393–401.

Bindra, K. & Mishra, A. (2017). A detailed study of clustering algorithms. (pp. 371–376).

Brandsæter, A., Manno, G., Vanem, E., & Glad, I. K. (2016). An application of sensor-based anomaly detection in the maritime industry.

Brandsæter, A., Vanem, E., & Glad, I. K. (2017). Cluster based anomaly detection with applications in the maritime industry. volume 2017-December, (pp. 328–333). Institute of Electrical and Electronics Engineers Inc.

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.

Cabanes, G. & Bennani, Y. (2010). Learning topological constraints in self-organizing map. (pp. 367–374).

Cai, C., Weng, X., & Zhang, C. (2017). A novel approach for marine diesel engine fault diagnosis. *Cluster Computing*, *20*, 1691–1702.

Calinski, T. & Harabasz, J. (1974). A dendrite method foe cluster analysis. *Communications in Statistics*, *3*, 1–27.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey.

Cheliotis, M., Lazakis, I., & Cheliotis, A. (2022). Bayesian and machine learning-based fault detection and diagnostics for marine applications. *Ships and Offshore Structures*, *17*, 2686–2698.

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.

Economics, L. & International, N. (2021). Consultancy research into the uk maritime technology sector.

Ellis, C. (2023). When to use gaussian mixture models.

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

Farahnakian, F., Nicolas, F., Farahnakian, F., Nevalainen, P., Sheikh, J., Heikkonen, J., & Raduly-Baka, C. (2023). A comprehensive study of clustering-based techniques for detecting abnormal vessel behavior. *Remote Sensing*, *15*, 1477.

Gao, J. (2012). Clustering: Mixture model. University of Buffalo. Department of Computer Science and Engineering.

Gharib, H. & Kovács, G. (2022). Development of a new expert system for diagnosing marine diesel engines based on real-time diagnostic parameters. *Strojniski Vestnik/Journal of Mechanical Engineering*, *68*, 642–653.

Gkerekos, C. & Lazakis, I. (2020). A novel, data-driven heuristic framework for vessel weather routing. *Ocean Engineering*, *197*, 106887.

Gkerekos, C., Lazakis, I., & Theotokatos, G. (2019). Machine learning models for predicting ship main engine fuel oil consumption: A comparative study. *Ocean Engineering*, *188*, 106282.

Google (2022). K-means advantages and disadvantages. *Clustering in Machine Learning*.

Guanglei, L., Hong, Z., Dingyu, J., & Hao, W. (2019). Design of ship monitoring system based on unsupervised learning. volume 885, (pp. 270–276). Springer Verlag.

Gupta, A., Datta, S., & Das, S. (2018). Fast automatic estimation of the number of clusters from the minimum inter-center distance for k-means clustering. *Pattern Recognition Letters*, *116*, 72–79.

Gupta, R. (2023). How does the dbscan algorithm work: Pros and cons of dbscan.

Günter, S. & Bunke, H. (2002). Self-organizing map for clustering in the graph domain. *Pattern Recognition Letters*, *23*, 405–417.

Jimenez, V. J., Bouhmala, N., & Gausdal, A. H. (2020). Developing a predictive maintenance model for vessel machinery. *Journal of Ocean Engineering and Science*, *5*, 358–386.

Jin, X. & Han, J. (2010). *K-Means Clustering*, (pp. 563–564). Boston, MA: Springer US.

Karagiannidis, P. & Themelis, N. (2021). Data-driven modelling of ship propulsion and the effect of data pre-processing on the prediction of ship fuel consumption and speed loss. *Ocean Engineering*, *222*, 108616.

Kaski, S. (1997). Data exploration using self-organizing maps. *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series*, *82*, 57.

Kaufman, L. & Rousseeuw, P. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*.

Kim, D., Antariksa, G., Handayani, M. P., Lee, S., & Lee, J. (2021). Explainable anomaly detection framework for maritime main engine sensor data. *Sensors*, *21*.

Kim, D., Lee, S., & Lee, J. (2020a). Data-driven prediction of vessel propulsion power using support vector regression with onboard measurement and ocean data. *Sensors (Switzerland)*, *20*.

Kim, D., Lee, S., & Lee, J. (2020b). An ensemble-based approach to anomaly detection in marine engine sensor streams for efficient condition monitoring and analysis. *Sensors (Switzerland)*, *20*, 1–16.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps.

Kyrtatos, N. (1993). *Marine Diesel Engines, Design and Operation*. Simmetria.

Lamaris, V. T. & Hountalas, D. T. (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.

Lazakis, I., Raptodimos, Y., & Varelas, T. (2018). Predicting ship machinery system condition through analytical reliability tools and artificial neural networks. *Ocean Engineering*, *152*, 404–415.

Li, Y., Huang, X., Ding, P., & Zhao, C. (2021). Wiener-based remaining useful life prediction of rolling bearings using improved kalman filtering and adaptive modification. *Measurement: Journal of the International Measurement Confederation*, *182*.

Lloyd, S. P. (1957). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, *28*.

Makridis, G., Kyriazis, D., & Plitsos, S. (2020). Predictive maintenance leveraging machine learning for time-series forecasting in the maritime industry. Institute of Electrical and Electronics Engineers Inc.

Makwana, P., Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining of cluster in k-means clustering review on determining number of cluster in k-means clustering. *International Journal of Advance Research in Computer Science and Management Studies*, *1*.

Martinez-Luengo, M., Shafiee, M., & Kolios, A. (2019). Data management for structural integrity assessment of offshore wind turbine support structures: data cleansing and missing data imputation. *Ocean Engineering*, *173*, 867–883.

Masmoudi, O., Jaoua, M., Jaoua, A., & Yacout, S. (2021). Data preparation in machine learning for condition-based maintenance. *Journal of Computer Science*, *17*, 525–538.

Nahim, H. M., Younes, R., Nohra, C., & Ouladsine, M. (2015). Complete modeling for systems of a marine diesel engine. *Journal of Marine Science and Application*, *14*, 93–104.

NASA (2007). *NASA Systems Engineering Handbook*. National Aeronautics and Space Administration, Center for Aerospace Information.

Pawel, K. & Smyk, R. (2018). Review and comparison of smoothing algorithms for one-dimensional data noise reduction. (pp. 277–281).

Pyle, D. (1999). *Data Preparation For Data Mining*. Morgan Kaufmann Publishers.

Pölzlbauer, G. (2004). Survey and comparison of quality measures for self-organizing maps.

Qu, C., Zhou, Z., Liu, Z., & Jia, S. (2022). Predictive anomaly detection for marine diesel engine based on echo state network and autoencoder. *Energy Reports*, *8*, 998–1003.

Rahmah, N. & Sitanggang, I. S. (2016). Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra. volume 31. Institute of Physics Publishing.

Raj, R. (2023). Supervised, unsupervised and semi-supervised learning with real-life usecase.

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.

Richman, M. B., Trafalis, T. B., & Adrianto, I. (2009). *Missing Data Imputation Through Machine Learning Algorithms*, (pp. 153–169). Dordrecht: Springer Netherlands.

Riveiro, M., Pallotta, G., & Vespe, M. (2018). Maritime anomaly detection: A review. *WIREs Data Mining and Knowledge Discovery*, *8*(5), e1266.

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., da F. Costa, L., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PLoS ONE*, *14*.

Sander, J. & Ester, M. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications.

Savitzky, A. & Golay, M. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, *36*, 1627–1639.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Transactions on Database Systems*, *42*.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, *6*, 461–464.

Tian, J., Azarian, M. H., & Pecht, M. (2014). Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm.

Vanem, E. & Brandsæter, A. (2021). Unsupervised anomaly detection based on clustering methods and sensor data on a marine diesel engine. *Journal of Marine Engineering and Technology*, *20*, 217–234.

Velasco-Gallego, C. & Lazakis, I. (2022a). Development of a time series imaging approach for fault classification of marine systems. *Ocean Engineering*, *263*.

Velasco-Gallego, C. & Lazakis, I. (2022b). A novel framework for imputing large gaps of missing values from time series sensor data of marine machinery systems. *Ships and Offshore Structures*, *17*, 1802–1811.

Velasco-Gallego, C. & Lazakis, I. (2022c). Radis: A real-time anomaly detection intelligent system for fault diagnosis of marine machinery. *Expert Systems with Applications*, *204*.

Velasco-Gallego, C. & Lazakis, I. (2022d). A real-time data-driven framework for the identification of steady states of marine machinery. *Applied Ocean Research*, *121*.

Wen, S., Zhang, W., Sun, Y., Li, Z., Huang, B., Bian, S., Zhao, L., & Wang, Y. (2023). An enhanced principal component analysis method with savitzky–golay filter and clustering algorithm for sensor fault detection and diagnosis. *Applied Energy*, *337*, 120862.

Yuan, C. & Yang, H. (2019). Research on k-value selection method of k-means clustering algorithm. *J Multidisciplinary Scientific Journal*, *2*, 226–235.

Zhao, Z., Cerf, S., Birke, R., Robu, B., Bouchenak, S., Mokhtar, S. B., & Chen, L. Y. (2019). Robust anomaly detection on unreliable data. (pp. 630–637). Institute of Electrical and Electronics Engineers Inc.

Øyvind Øksnes Dalheim & Steen, S. (2020a). A computationally efficient method for identification of steady state in time series data from ship monitoring. *Journal of Ocean Engineering and Science*, *5*, 333–345.

Øyvind Øksnes Dalheim & Steen, S. (2020b). Preparation of in-service measurement data for ship operation and performance analysis. *Ocean Engineering*, *212*.

Appendix A: Parameters of Each Main Engine System

This appendix contains tables of the parameters in each sub-system of the main engine. The first table contains the parameters related to the cooling and fuel systems, whereas in the second are the parameters of the intake & exhaust and the lubrication systems.

Table 12: Parameters of cooling and fuel systems.

Cooling System	**Fuel System**
ME air cooler cool w inlet pre_AMS (MPa)	M/E Shaft RPM_TRQM (rpm)
ME air cooler cool w inlet tem_AMS (C)	ME Consumption_TRQM (lt/hr)
ME air cooler cool w outlet te_AMS (C)	ME FO inlet press_AMS (MPa)
ME cyl 1 JCW outlet temp_AMS (C)	ME FO inlet temp_AMS (C)
ME cyl 1 PCO outlet temp_AMS (C)	ME fuel index_AMS (%)
ME cyl 2 JCW outlet temp_AMS (C)	Shaft Power_TRQM (kW)
ME cyl 2 PCO outlet temp_AMS (C)	Shaft Torque_TRQM (kNm)
ME cyl 3 JCW outlet temp_AMS (C)	
ME cyl 3 PCO outlet temp_AMS (C)	
ME cyl 4 JCW outlet temp_AMS (C)	
ME cyl 4 PCO outlet temp_AMS (C)	
ME cyl 5 JCW outlet temp_AMS (C)	
ME cyl 5 PCO outlet temp_AMS (C)	
ME cyl 6 JCW outlet temp_AMS (C)	
ME cyl 6 PCO outlet temp_AMS (C)	
ME JCW inlet press_AMS (MPa)	
ME JCW inlet temp_AMS (C)	
ME JCW outlet press_AMS (MPa)	

Table 13: Parameters of intake & exhaust and lubrication systems.

Intake & Exhaust System	**Lubrication System**
M/E T/C RPM_IND1 (rpm)	Cyl 01 AFT main bearing temp_AMS (C)
ME cyl 1 exh gas outlet temp_AMS (C)	Cyl 01 crank pin bearing temp_AMS (C)
ME cyl 1 scav air temp_AMS (C)	Cyl 01 fore main bearing temp_AMS (C)
ME cyl 2 exh gas outlet temp_AMS (C)	Cyl 02 AFT main bearing temp_AMS (C)
ME cyl 2 scav air temp_AMS (C)	Cyl 02 crank pin bearing temp_AMS (C)
ME cyl 3 exh gas outlet temp_AMS (C)	Cyl 03 AFT main bearing temp_AMS (C)
ME cyl 3 scav air temp_AMS (C)	Cyl 03 crank pin bearing temp_AMS (C)
ME cyl 4 exh gas outlet temp_AMS (C)	Cyl 04 AFT main bearing temp_AMS (C)
ME cyl 4 scav air temp_AMS (C)	Cyl 04 crank pin bearing temp_AMS (C)
ME cyl 5 exh gas outlet temp_AMS (C)	Cyl 05 AFT main bearing temp_AMS (C)
ME cyl 5 scav air temp_AMS (C)	Cyl 05 crank pin bearing temp_AMS (C)
ME cyl 6 exh gas outlet temp_AMS (C)	Cyl 06 AFT main bearing temp_AMS (C)
ME cyl 6 scav air temp_AMS (C)	Cyl 06 crank pin bearing temp_AMS (C)
ME scav air receiver inlet pres_AMS (MPa)	ME cyl lub oil temp_AMS (C)
ME scav air receiver temp_AMS (C)	ME main LO inlet press_AMS (MPa)
ME TC exh gas inlet temp_AMS (C)	ME main LO inlet temp_AMS (C)
ME TC exh gas outlet temp_AMS (C)	ME TC LO inlet press_AMS (MPa)
	ME TC LO outlet temp_AMS (C)

Appendix B: Characteristics of each Main Engine System Data

This brief section has the purpose of showcasing the characteristics of the parameters that constitute the ME sub-systems. The following figures are presented for each system in order to do that:

1. Time-series plot of system parameters.
2. Correlation heatmap of system parameters.
3. Scree plot of system parameters (plot that visualizes the selection of optimal number of components in PCA)
4. Scatter-plot of principal components for each system, also containing distribution of each component's data-points.

Cooling System

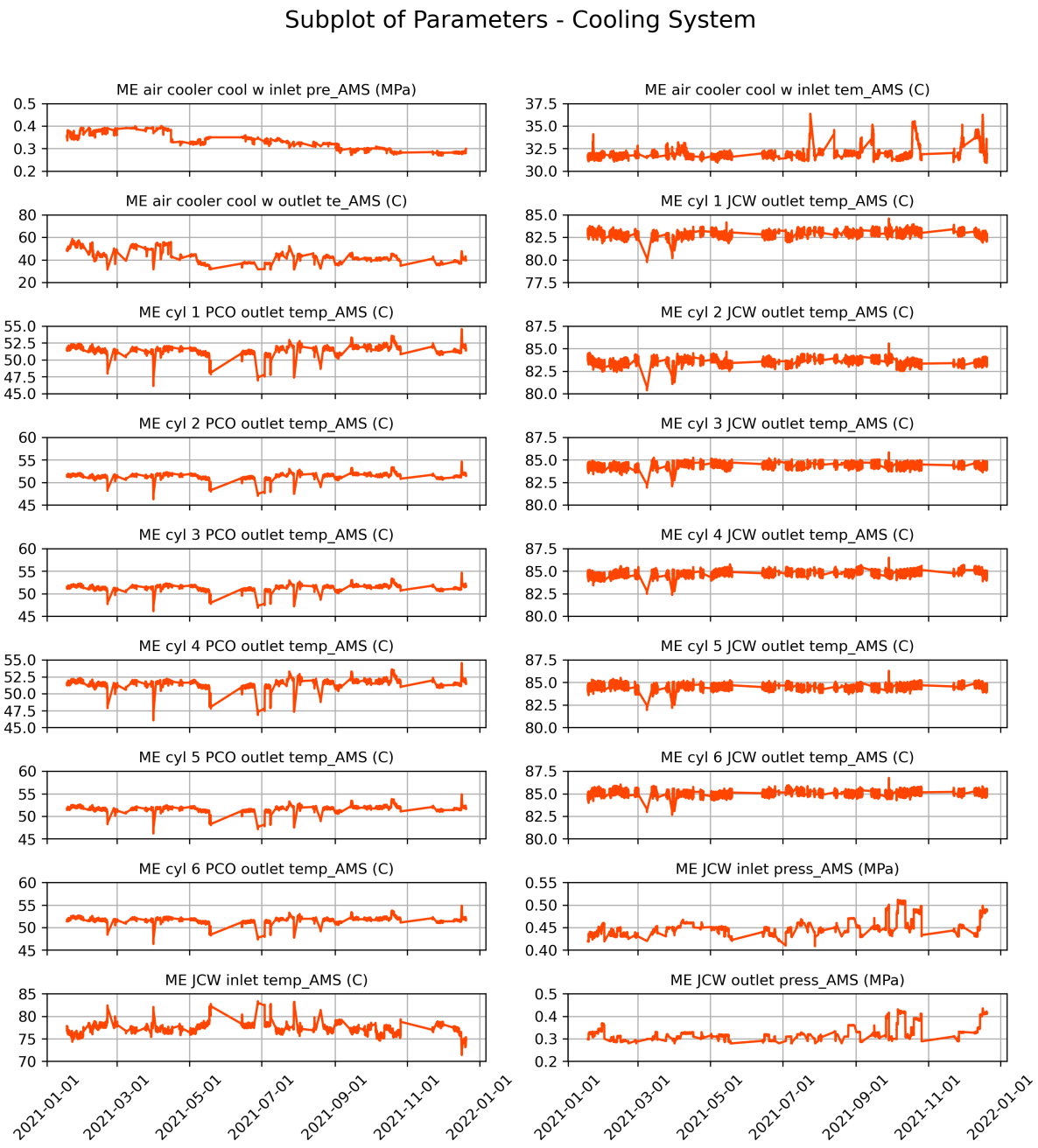

Figure 90: Time-series plot of parameters in the cooling system.

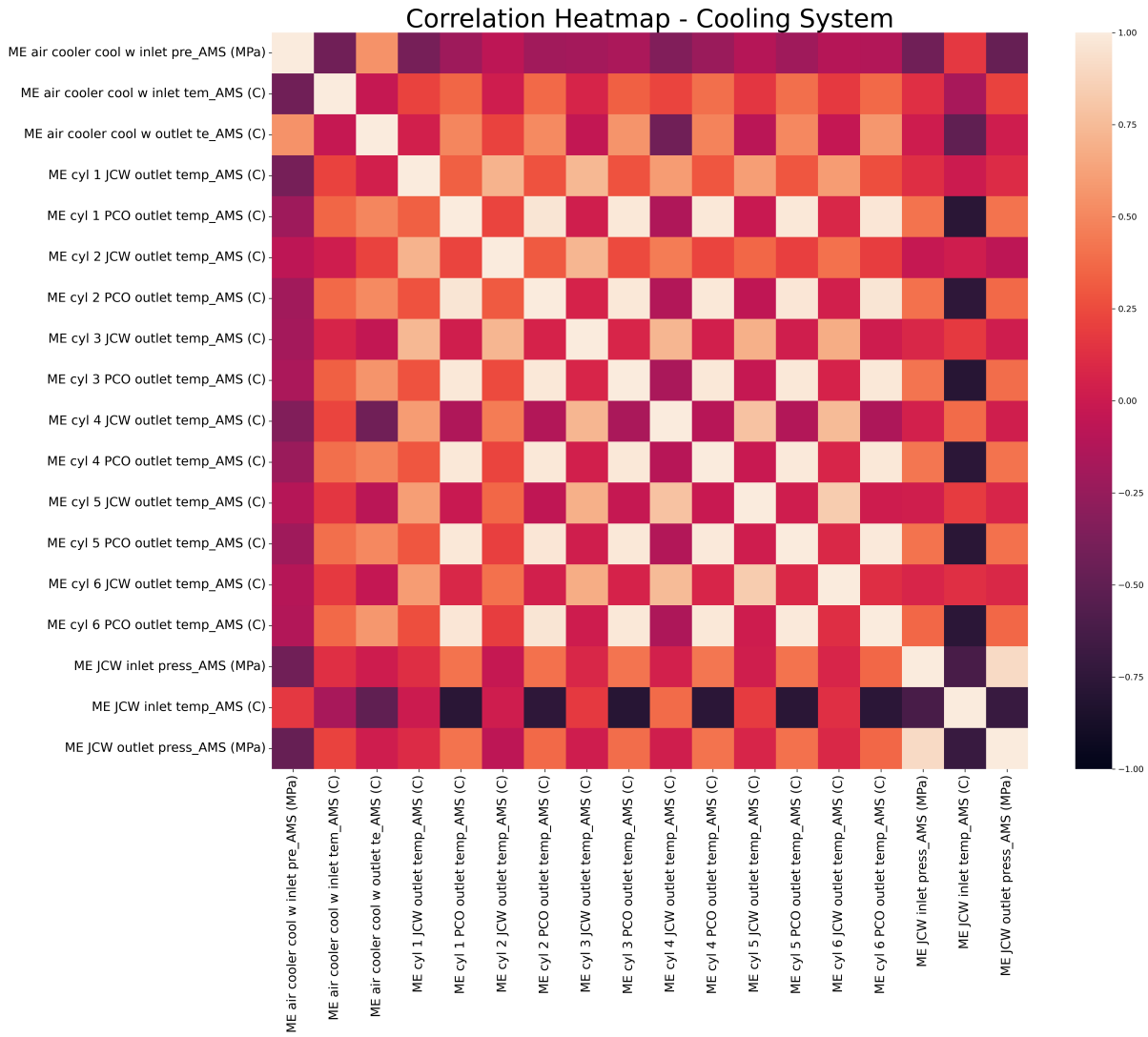

Figure 91: Heatmap of parameters in the cooling system.

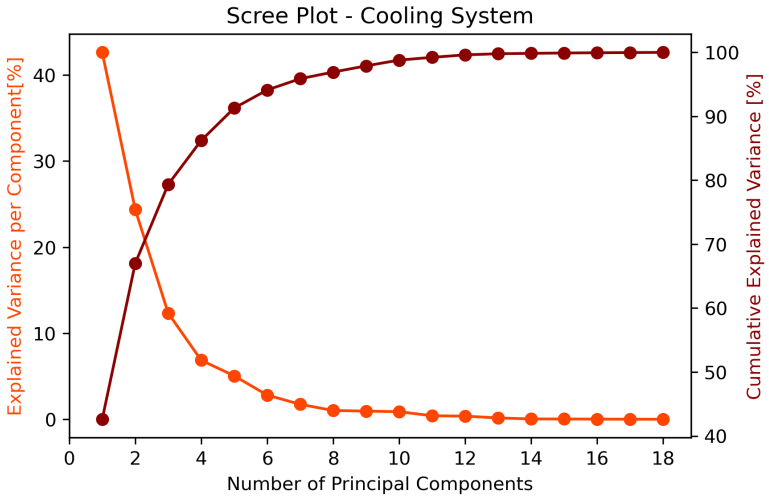

Figure 92: Scree plot of the cooling system. Used to determined optimal value of principal components.

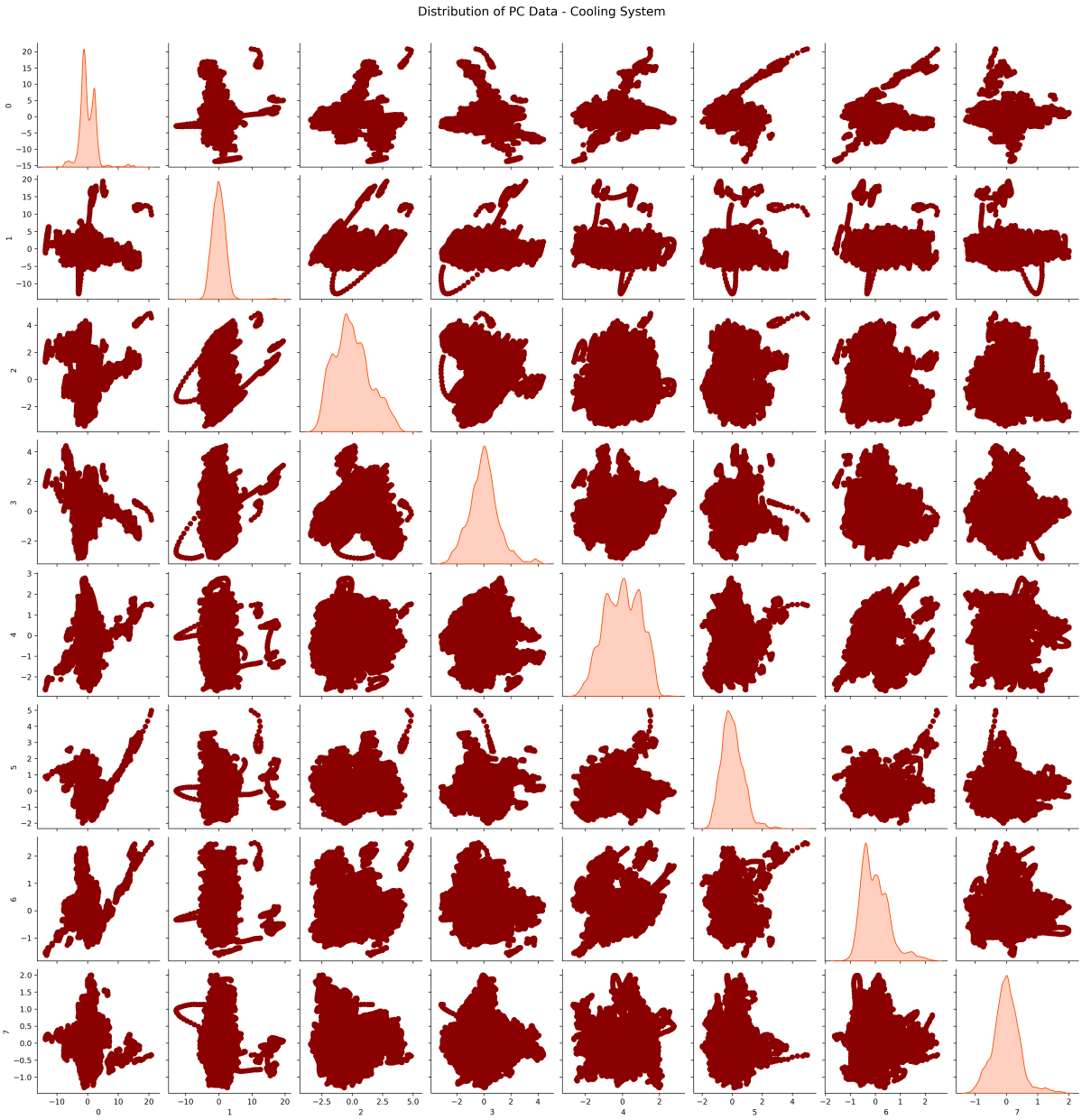

Figure 93: Scatter-plots and distribution of cooling system PCs.

Fuel System

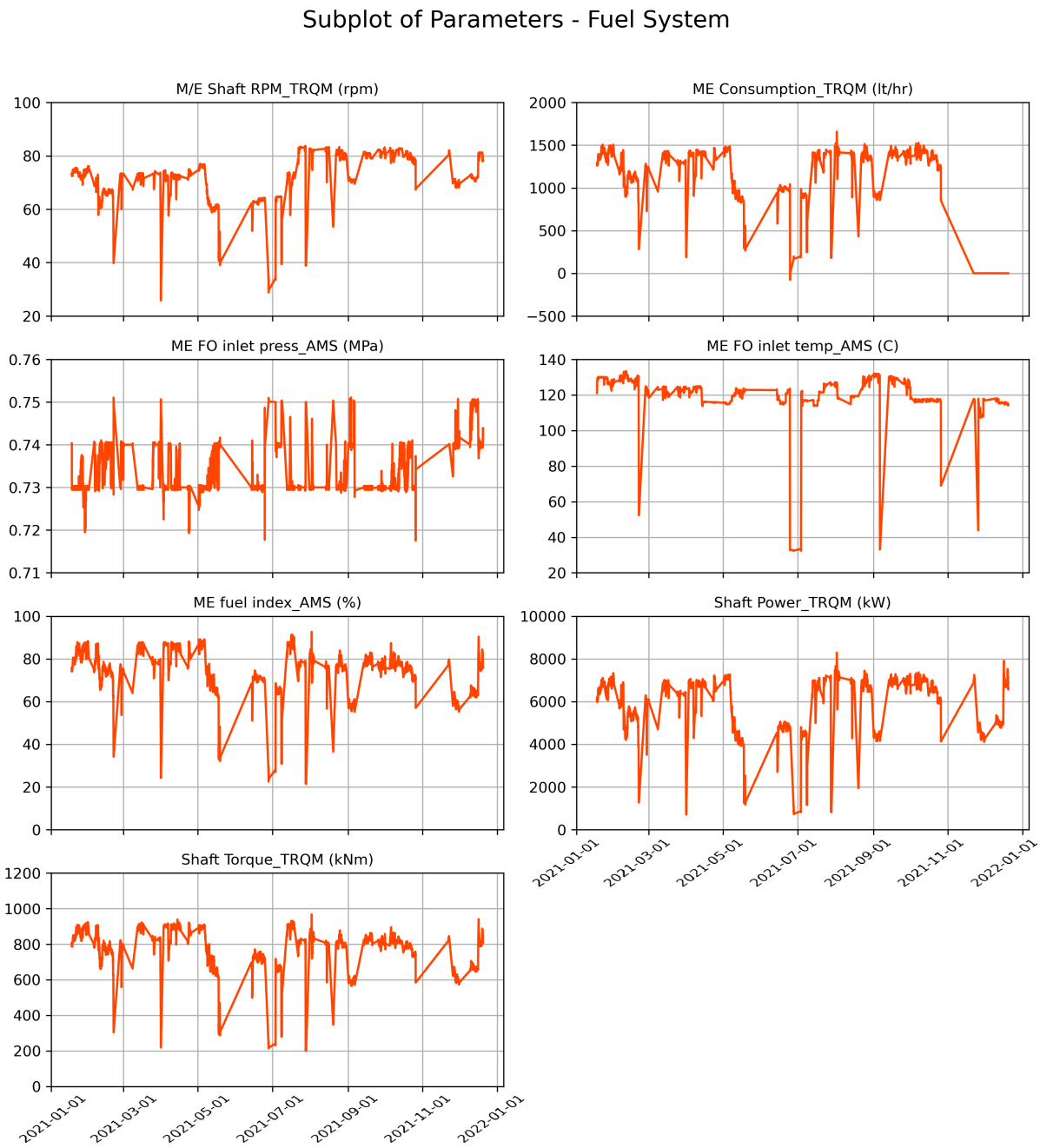

Figure 94: Time-series plot of parameters in the fuel system.

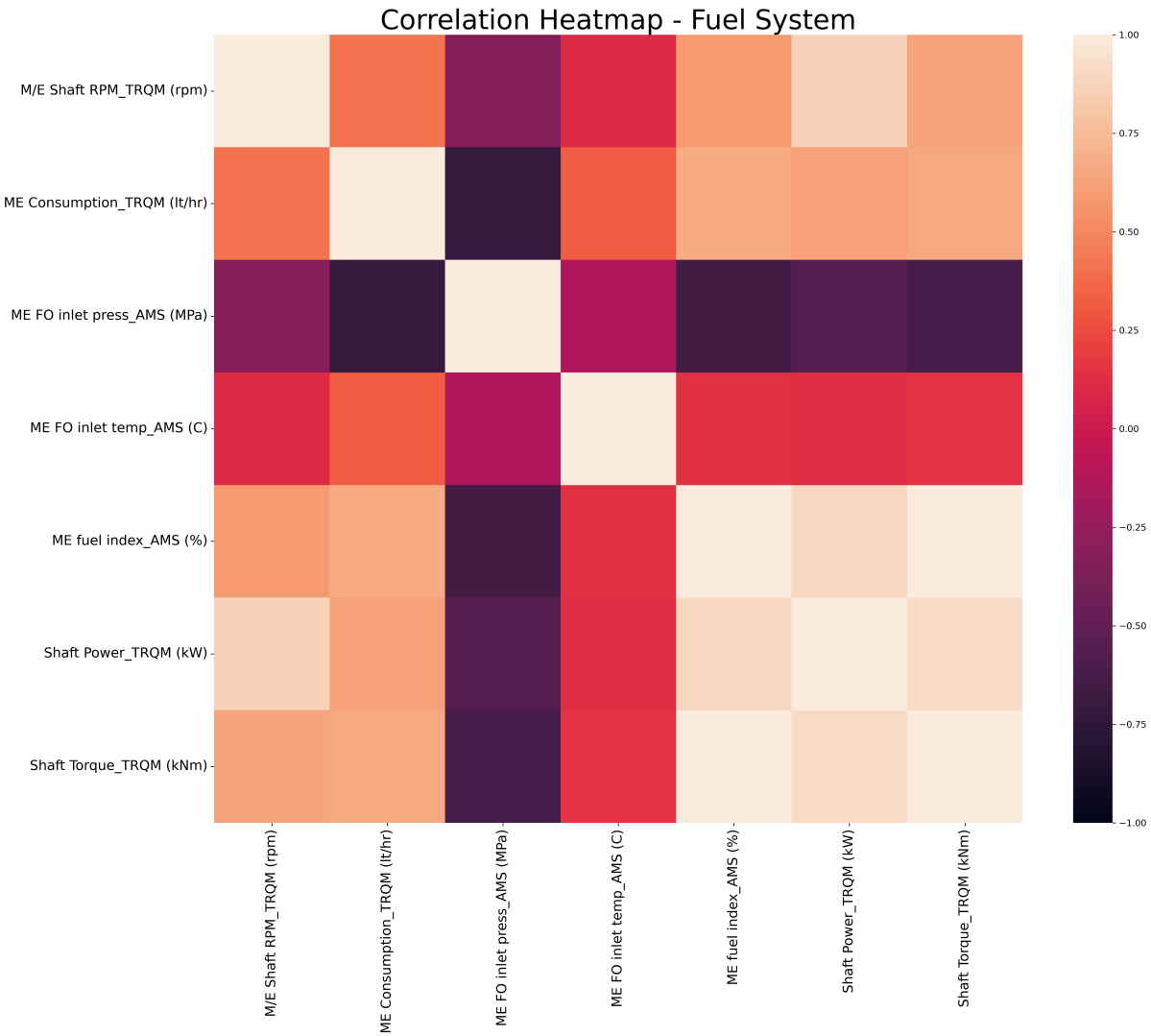

Figure 95: Heatmap of parameters in the fuel system.

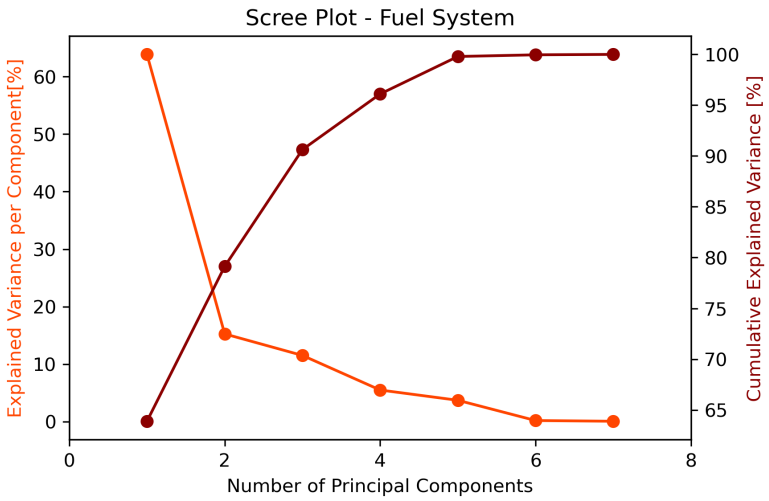

Figure 96: Scree plot of the fuel system. Used to determined optimal value of principal components.

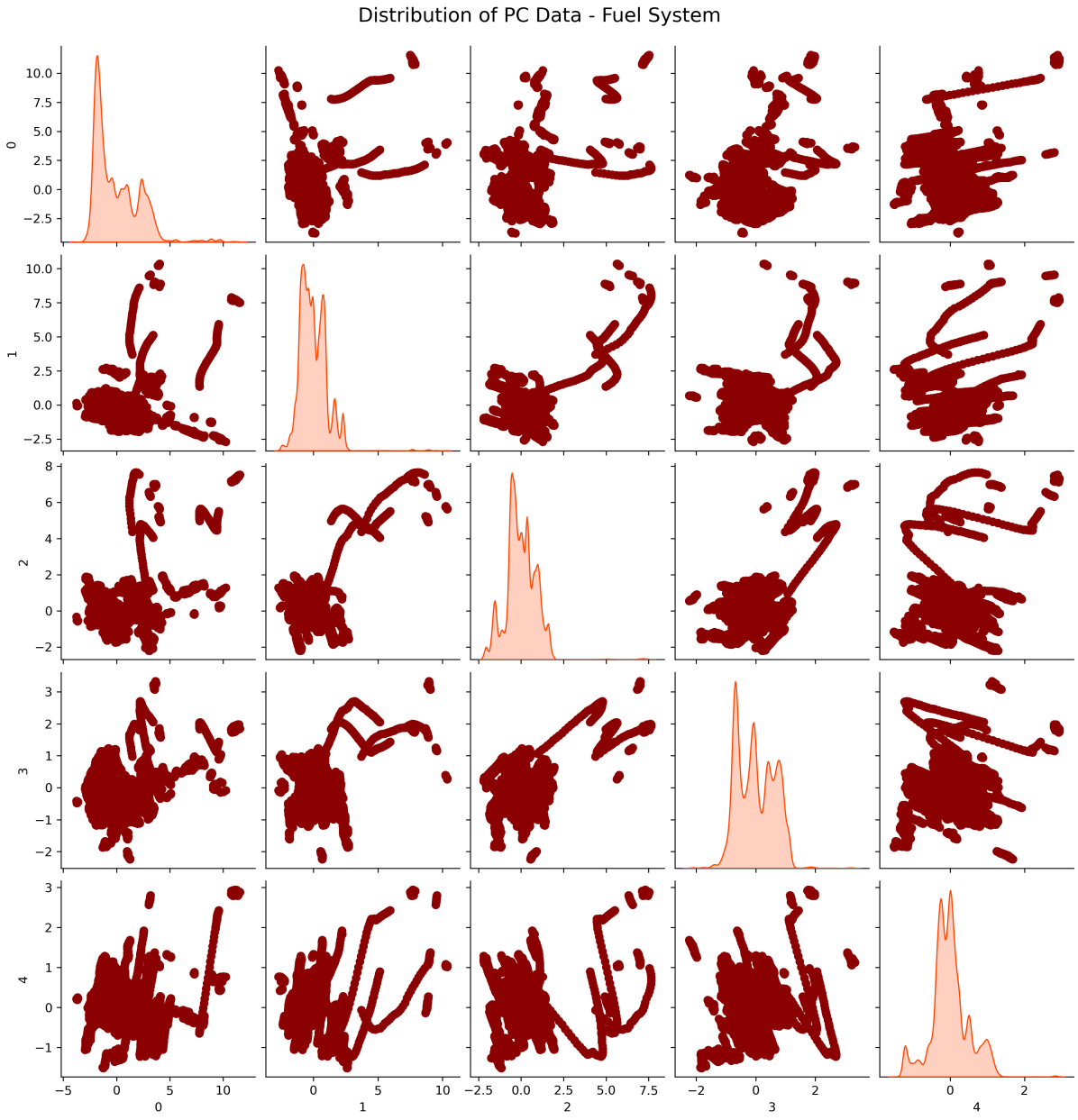

Figure 97: Scatter-plots and distribution of fuel system PCs.

Intake & Exhaust System

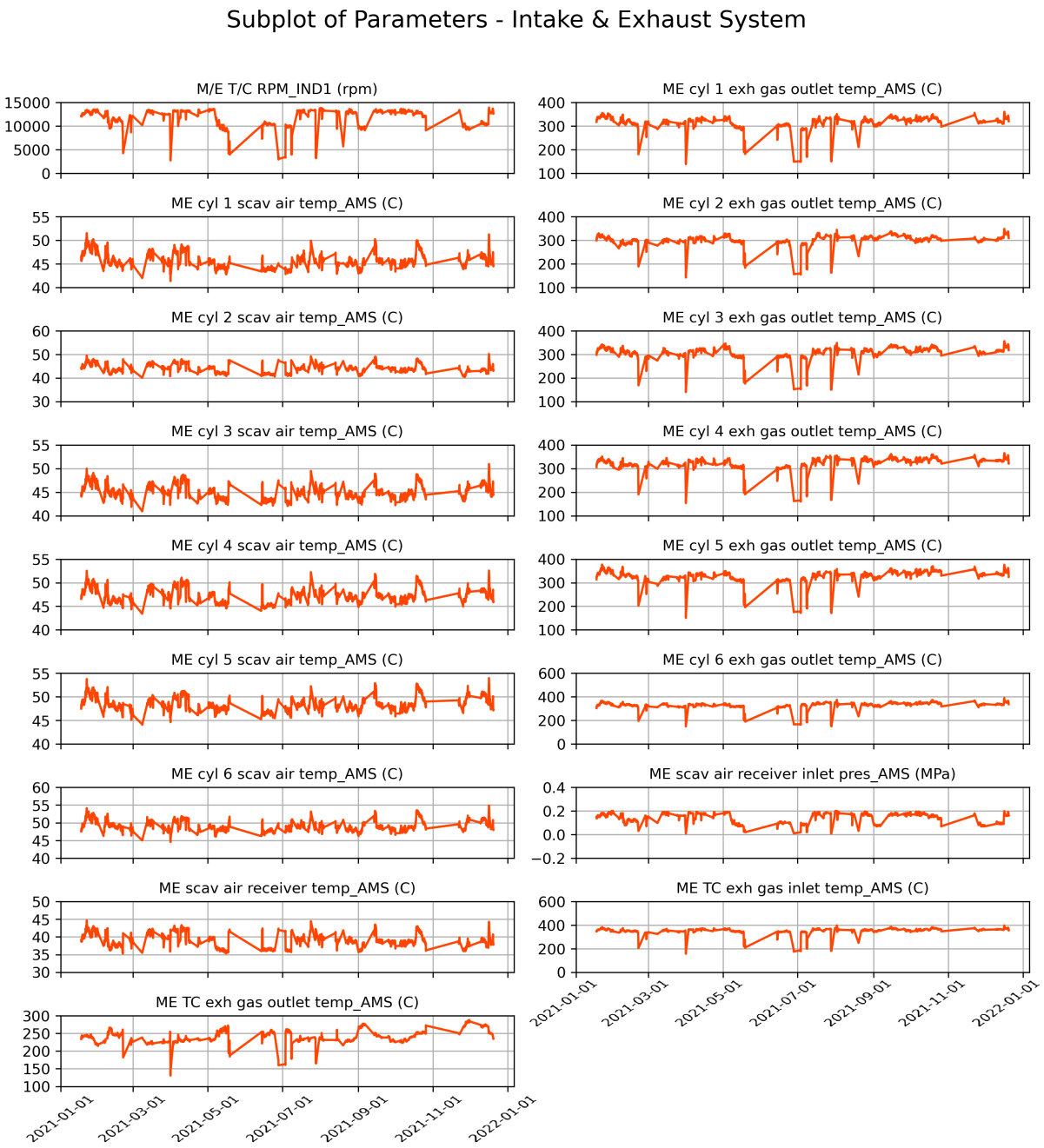

Figure 98: Time-series plot of parameters in the intake & exhaust system.

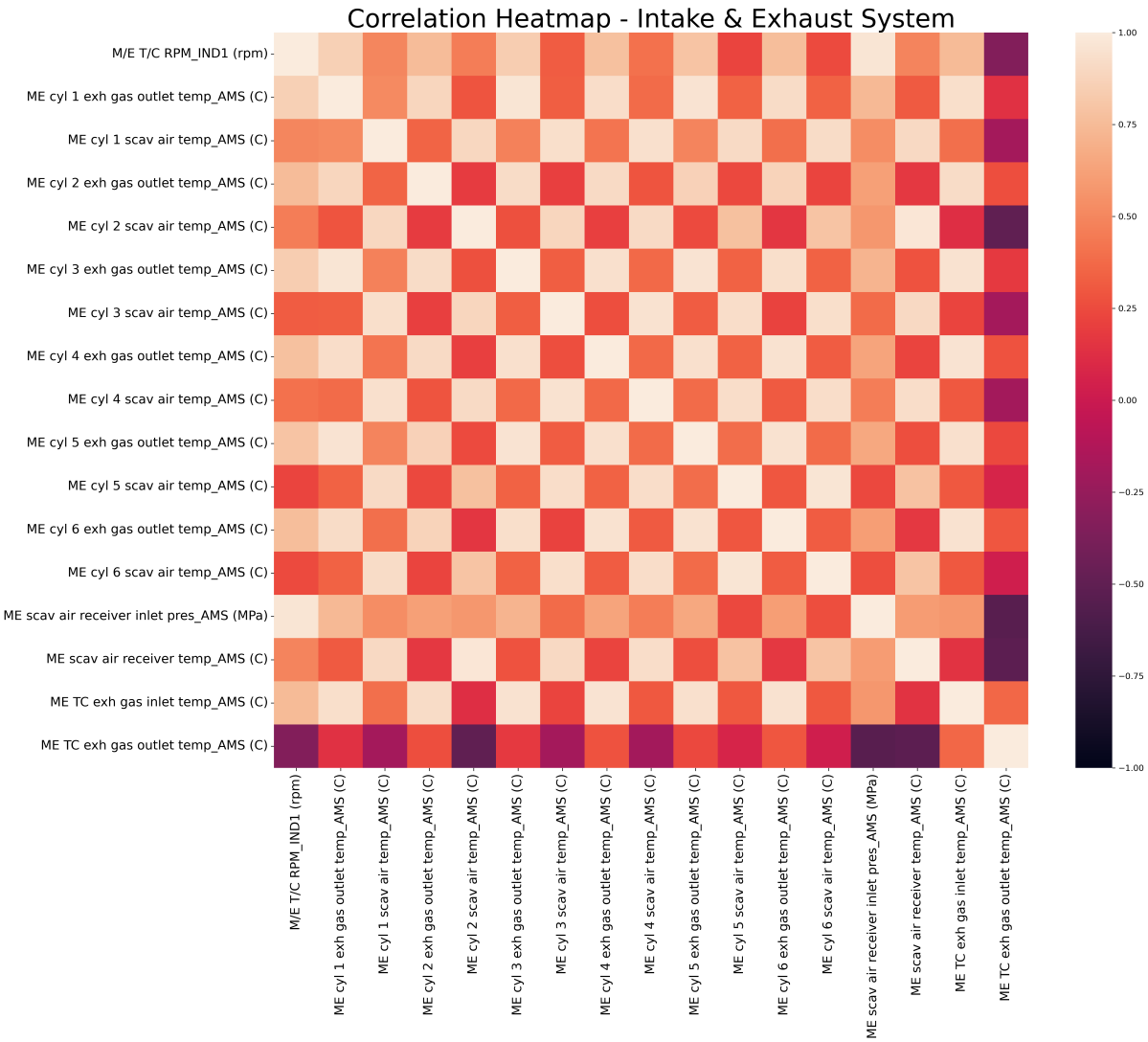

Figure 99: Heatmap of parameters in the intake & exhaust system.

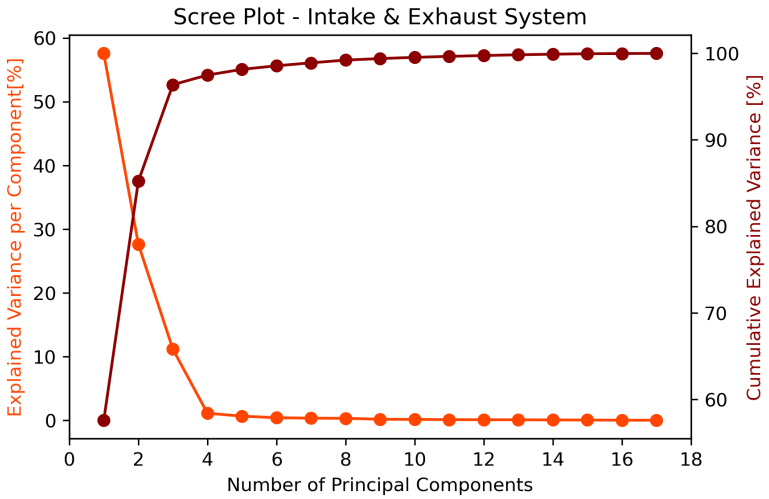

Figure 100: Scree plot of the intake & exhaust system. Used to determined optimal value of principal components.

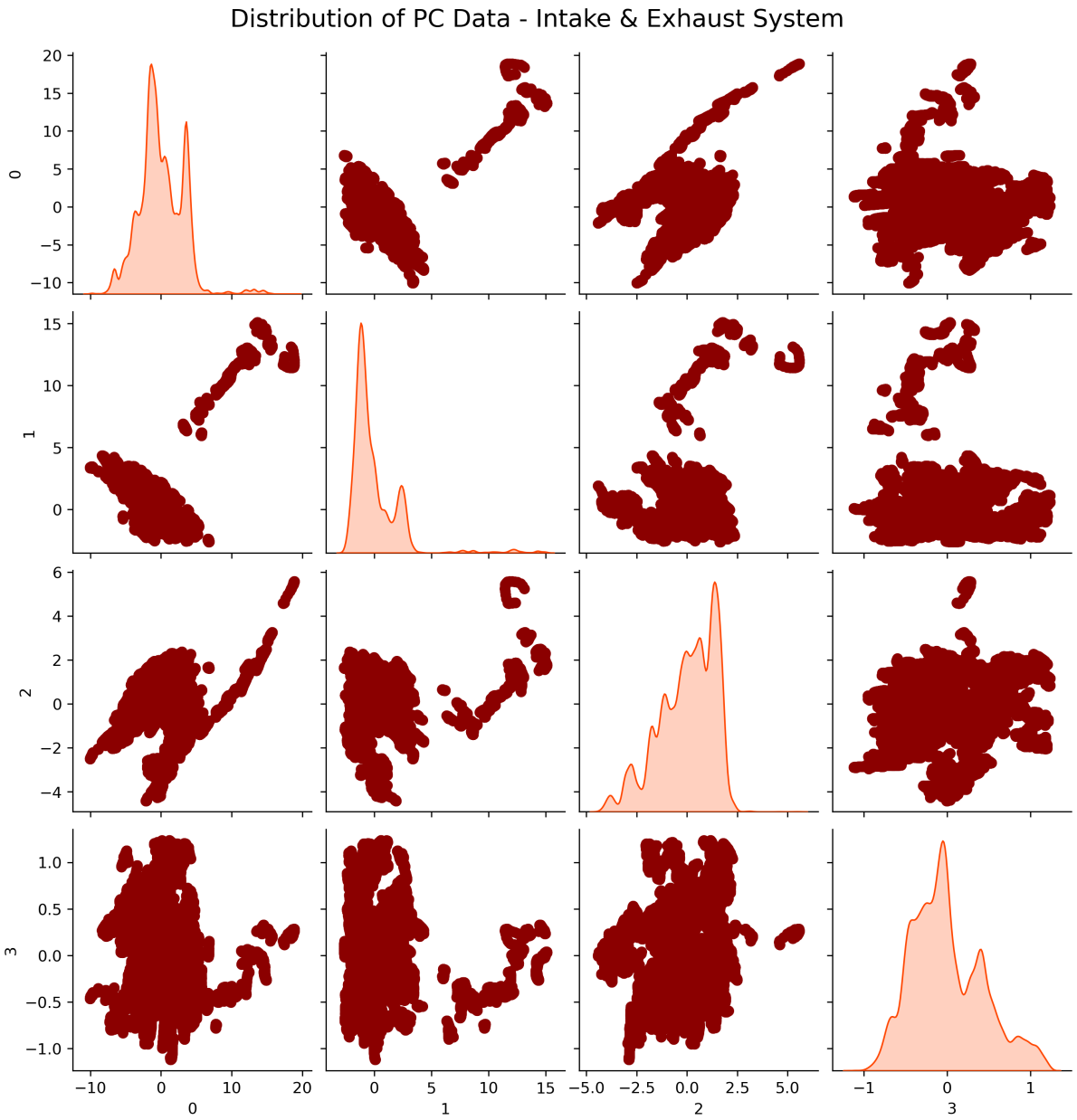

Figure 101: Scatter-plots and distribution of intake & exhaust system PCs.

Lubrication System

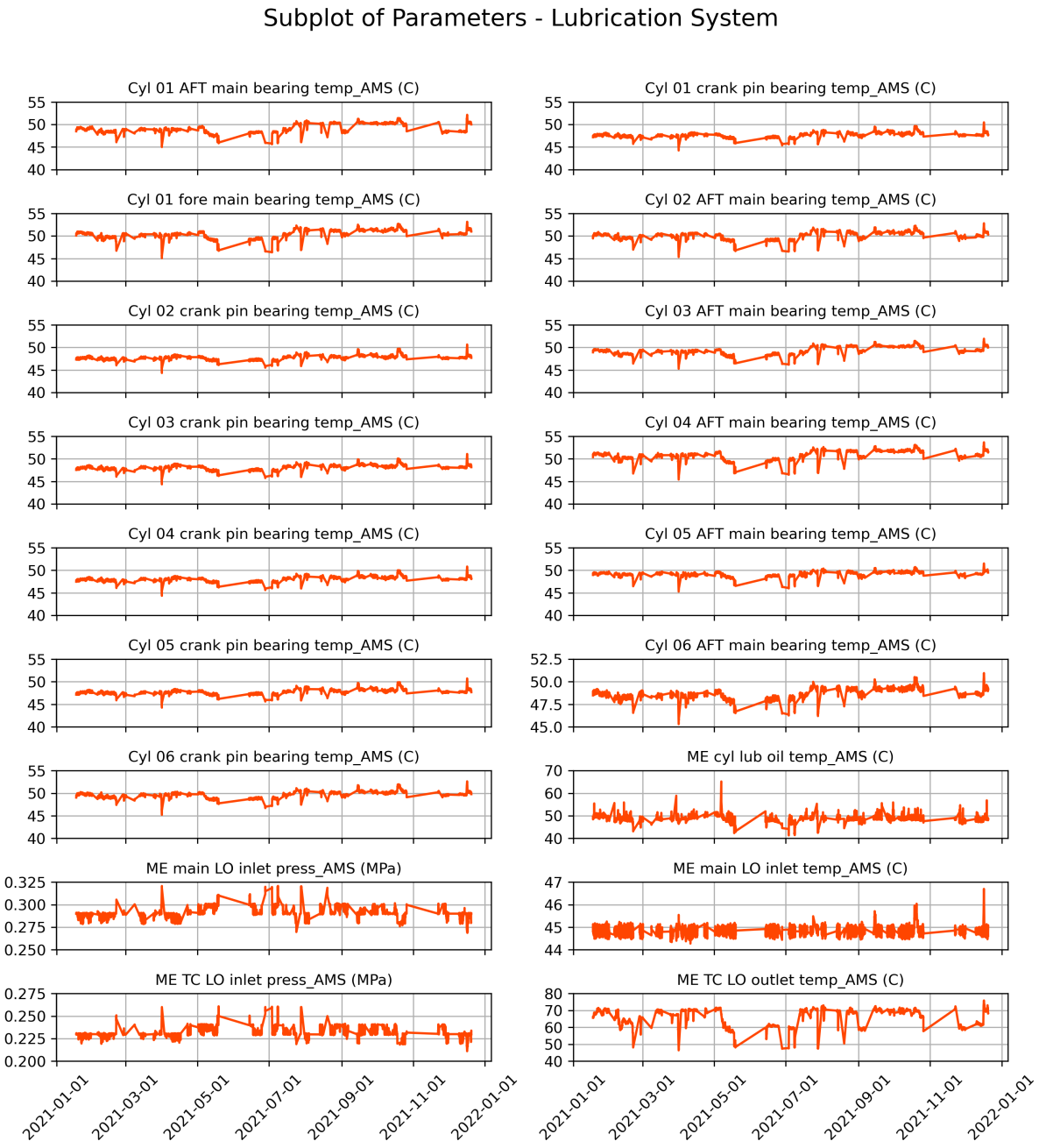

Figure 102: Time-series plot of parameters in the lubrication system.

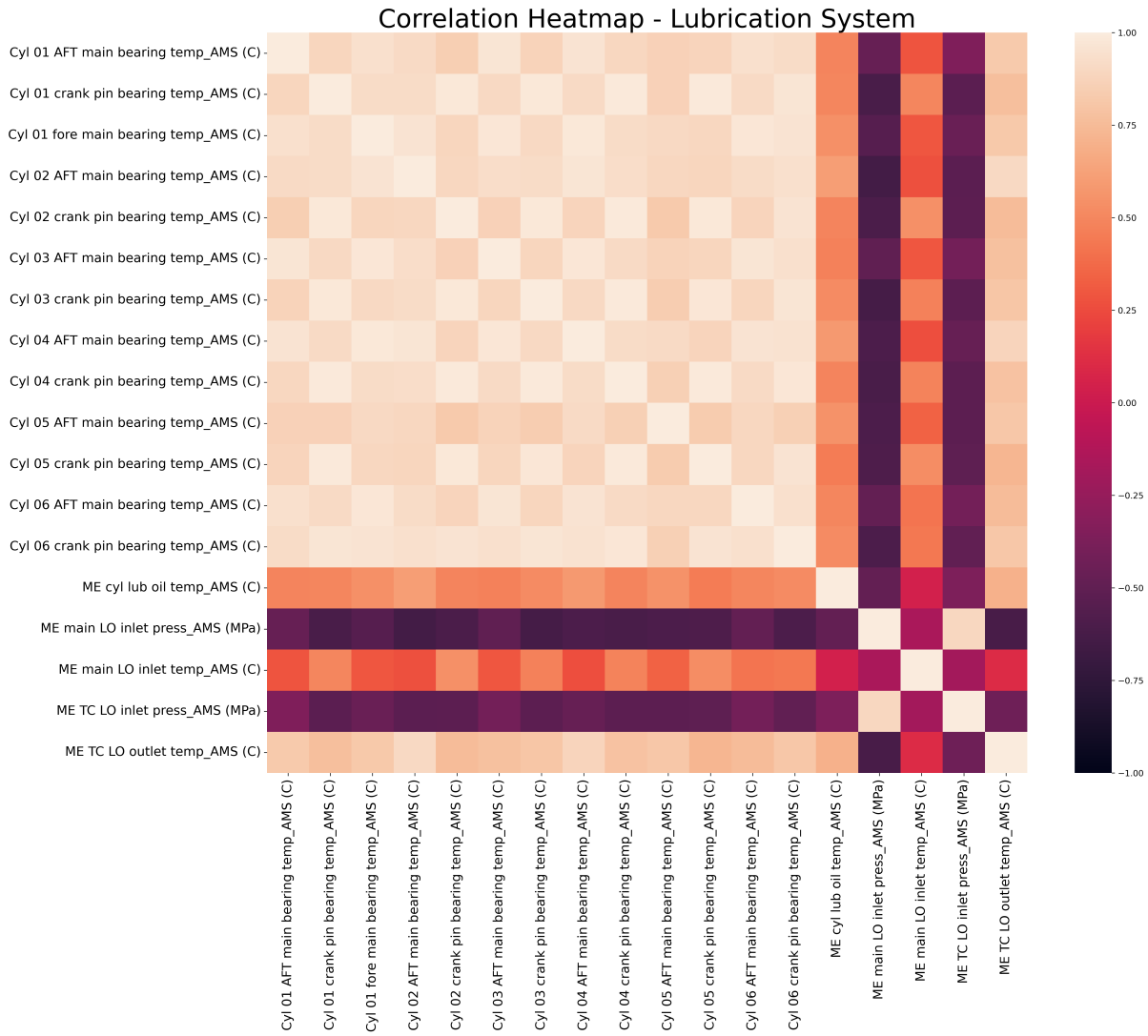

Figure 103: Heatmap of parameters in the lubrication system.

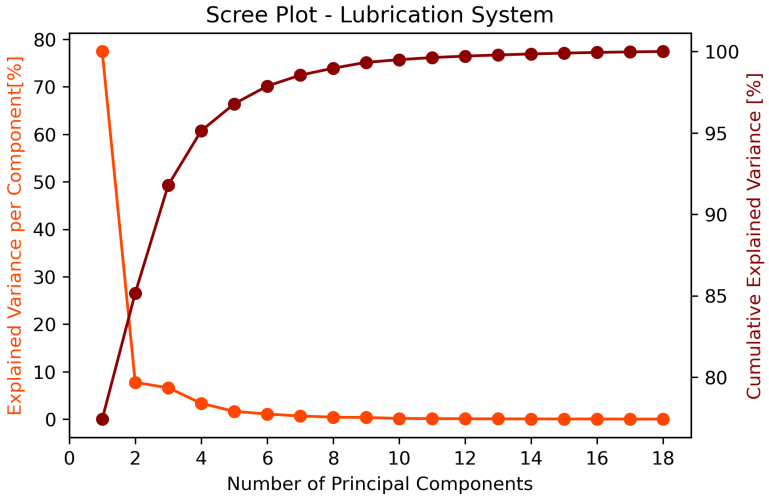

Figure 104: Scree plot of the lubrication system. Used to determined optimal value of principal components.

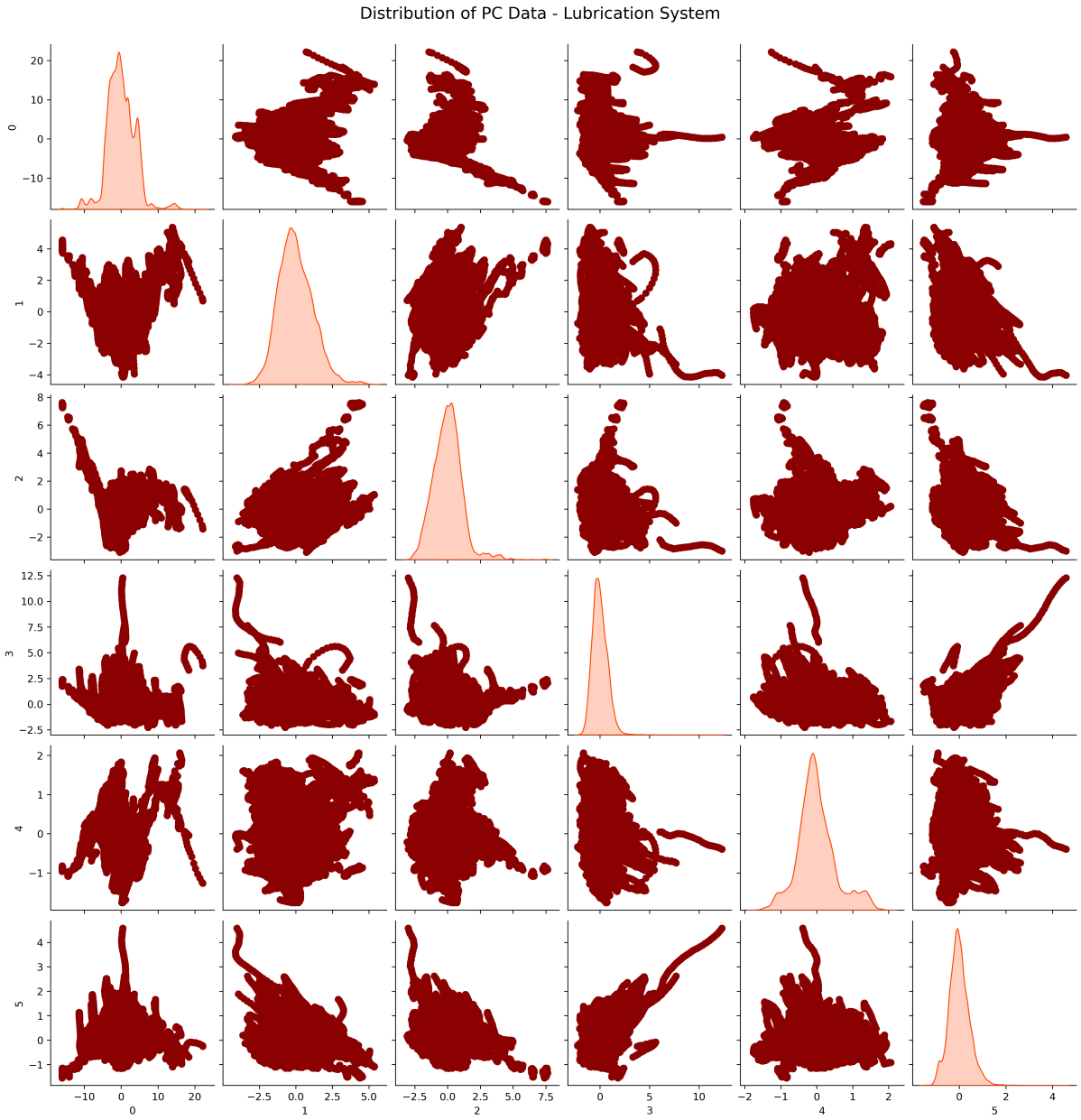

Figure 105: Scatter-plots and distribution of lubrication system PCs.

Appendix C: Anomaly Detection Results for the Fuel & Lubrication Systems

This appendix offers a brief presentation of the anomaly detection results in the other two sub-systems that have not been included in the results section of the thesis.

This results presentation offers a table of cluster number, detected anomalies, ratio of anomalies and distribution plots of points to clusters for each of the examined methods.

Fuel System

Table 14: Presentation of fuel system anomaly detection results.

Fuel System					
	K-Means	GMM	DBSCAN	SOM (Thresh.)	SOM (Clust.)
Cluster Number	23	25	28	11	
Anomalies	591	555	756	613	578
Anomalies [%]	2.05&	1.93%	2.63%	2.13%	2.01%

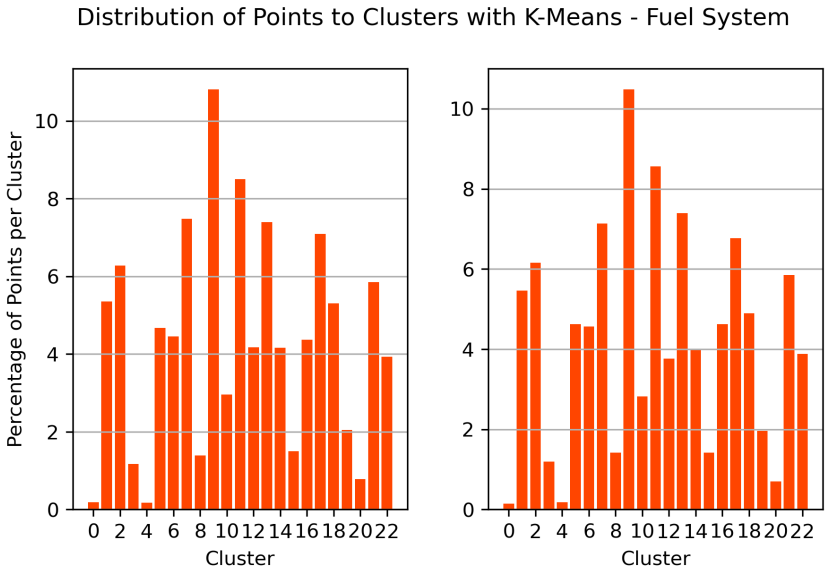

Figure 106: Distribution of points to clusters between train (left) and test (right) phases with K-Means in the fuel system.

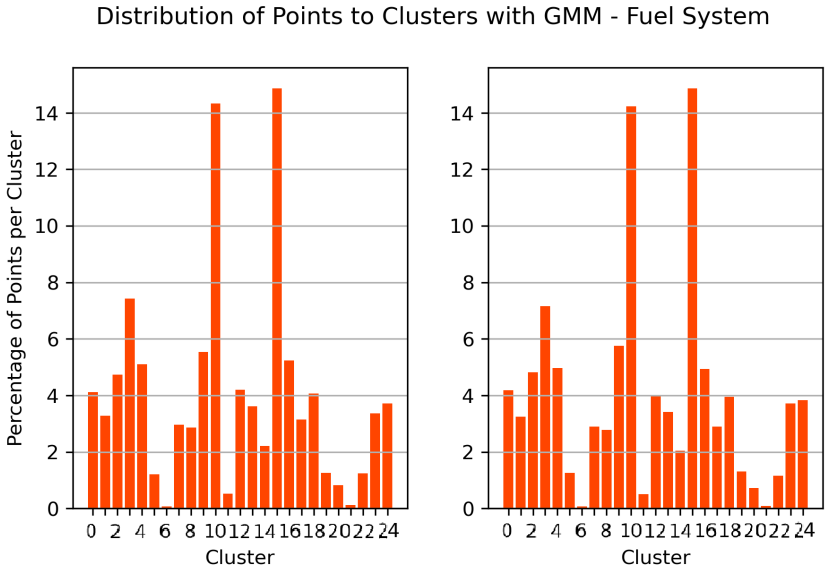

Figure 107: Distribution of points to clusters between train (left) and test (right) phases with GMM in the fuel system.

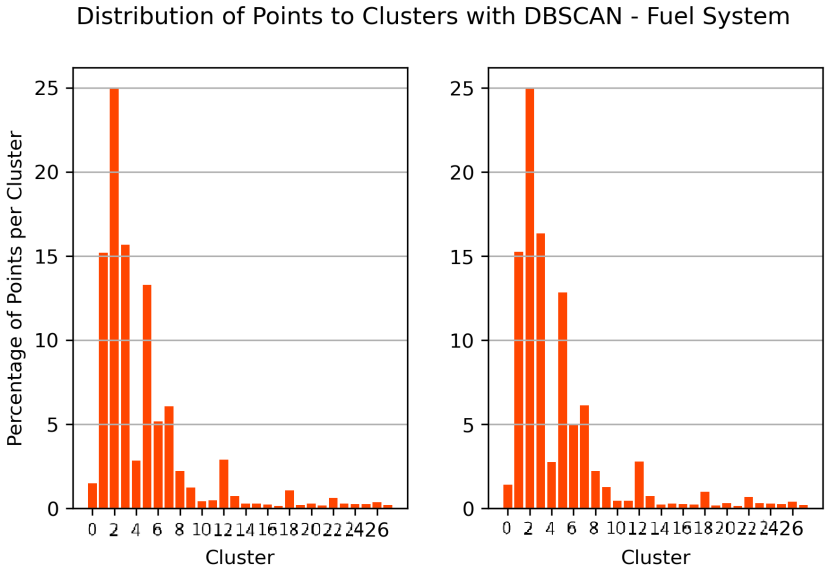

Figure 108: Distribution of points to clusters between train (left) and test (right) phases with DBSCAN in the fuel system.

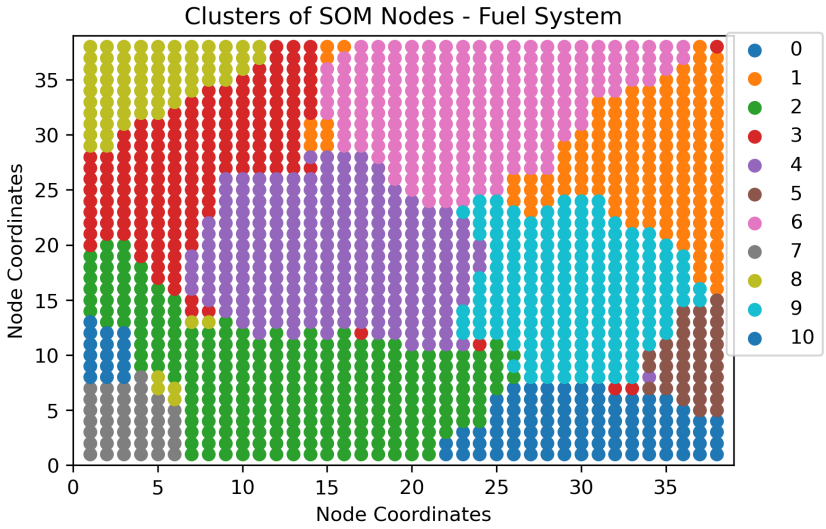

Figure 109: Clusters of SOM nodes plot in the fuel system.

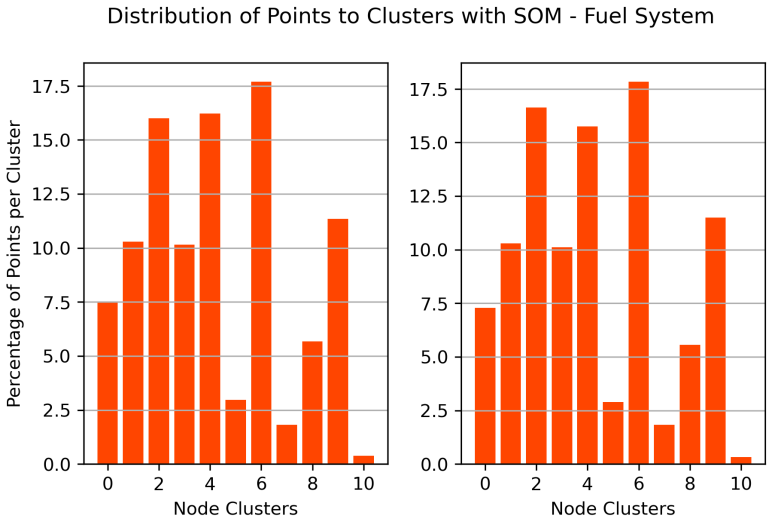

Figure 110: Distribution of points to SOM node clusters between the train (left) and (test) phases in the fuel system.

Lubrication System

Table 15: Presentation of lubrication system anomaly detection results.

Lubrication System					
	K-Means	GMM	DBSCAN	SOM (Thresh.)	SOM (Clust.)
Cluster Number	5	10	2	7	
Anomalies	572	613	951	770	576
Anomalies [%]	1.99%	2.13%	3.31%	2.68%	2.00%

A comment should be made here, about the results shown in Table 15. All methods identified smaller number of clusters as optimal when compared to the other systems. This is not considered to be a negative result, except in the case of DBSCAN where it managed to cluster the points in only two groups. This outcome can be considered negative from an anomaly detection perspective because it reflects a limited ability to capture the underlying structure of the dataset. When DBSCAN forms only one significant and one small clusters, the algorithm may be oversimplifying the data complexity. True anomalies often exhibit unique patterns that might not be detected by such simplified cluster structures.

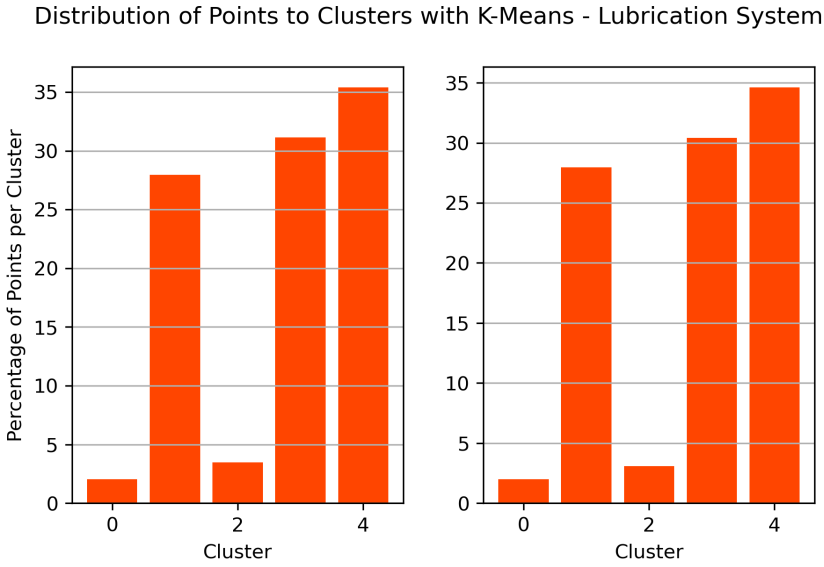

Figure 111: Distribution of points to clusters between train (left) and test (right) phases with K-Means in the lubrication system.

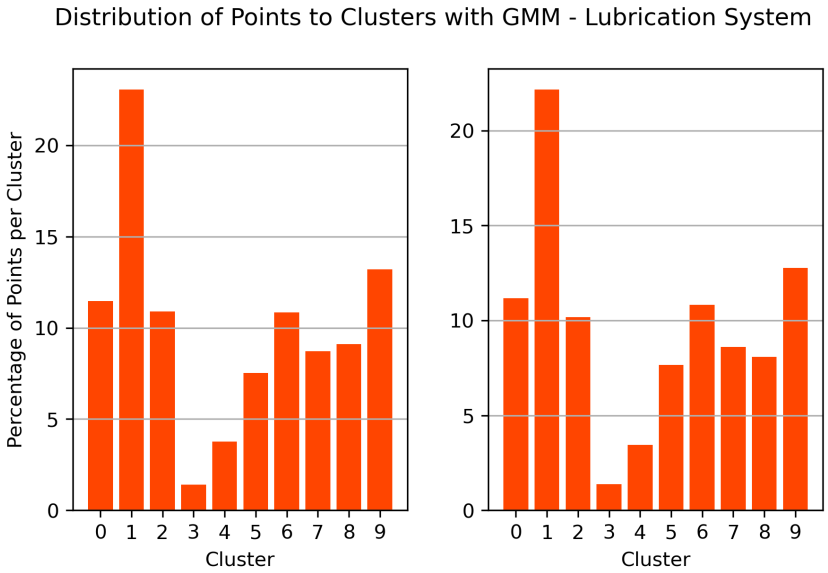

Figure 112: Distribution of points to clusters between train (left) and test (right) phases with GMM in the lubrication system.

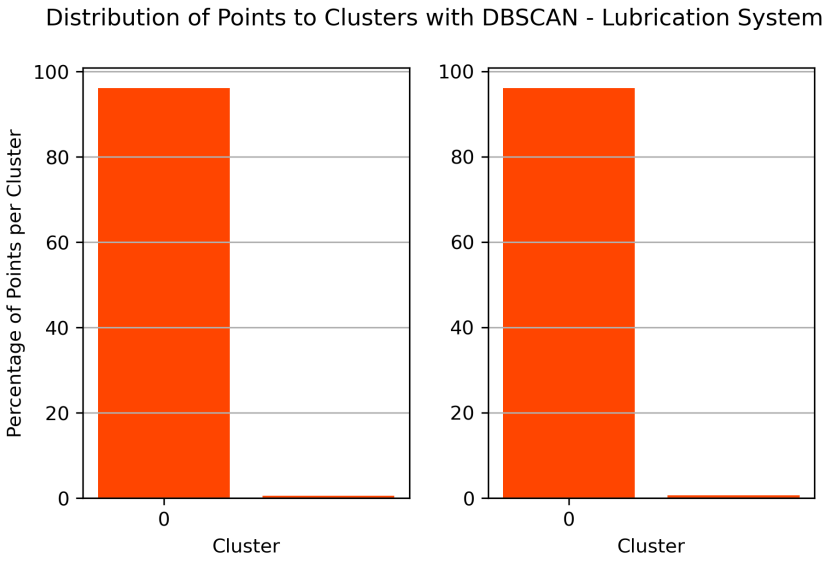

Figure 113: Distribution of points to clusters between train (left) and test (right) phases with DBSCAN in the lubrication system.

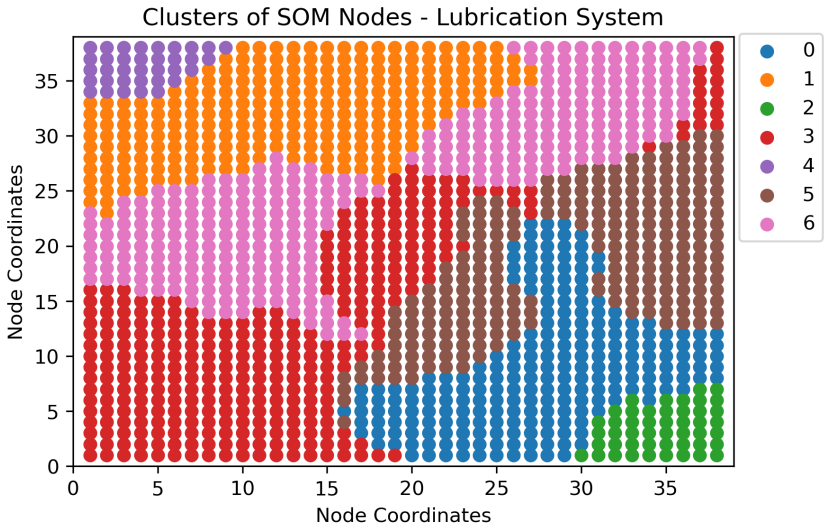

Figure 114: Clusters of SOM nodes plot in the lubrication system.

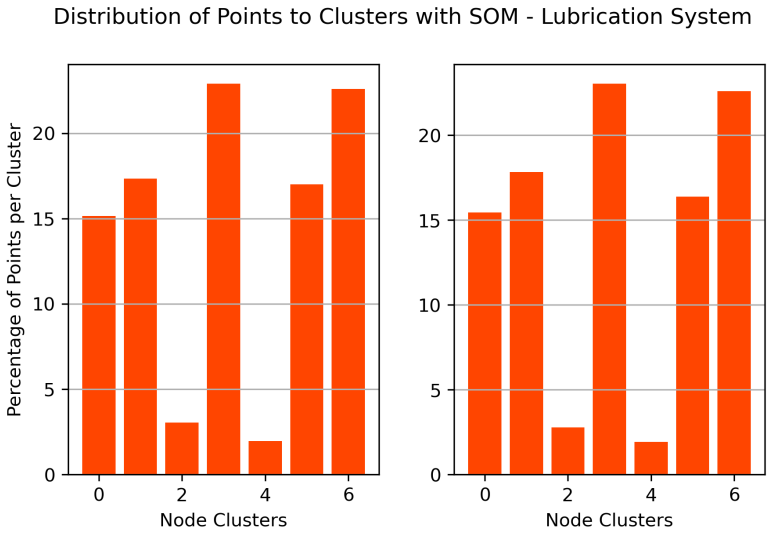

Figure 115: Distribution of points to SOM node clusters between the train (left) and (test) phases in the lubrication system.