DEGREE PROJECT IN MATHEMATICS,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2020

Anomaly Detection in Time Series Data using Unsupervised Machine Learning Methods

A Clustering-Based Approach

PETER HANNA

ERIK SWARTLING

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Anomaly Detection in Time Series Data using Unsupervised Machine Learning Methods

A Clustering-Based Approach

PETER HANNA

ERIK SWARTLING

Degree Projects in Mathematical Statistics (30 ECTS credits)
Master's Programme in Applied and Computational Mathematics
KTH Royal Institute of Technology year 2020
Supervisor at Xylem: Staffan Aldenfalk Jansson
Supervisor at KTH: Jimmy Olsson
Examiner at KTH: Jimmy Olsson

TRITA-SCI-GRU 2020:063
MAT-E 2020:026

Royal Institute of Technology
School of Engineering Sciences
KTH SCI
SE-100 44 Stockholm, Sweden
URL: www.kth.se/sci

Abstract

For many companies in the manufacturing industry, attempts to find damages in their products is a vital process, especially during the production phase. Since applying different machine learning techniques can further aid the process of damage identification, it becomes a popular choice among companies to make use of these methods to enhance the production process even further. For some industries, damage identification can be heavily linked with anomaly detection of different measurements. In this thesis, the aim is to construct unsupervised machine learning models to identify anomalies on unlabeled measurements of pumps using high frequency sampled current and voltage time series data. The measurement can be split up into five different phases, namely the startup phase, three duty point phases and lastly the shutdown phase. The approach is based on clustering methods, where the main algorithms of use are the density-based algorithms DBSCAN and LOF. Dimensionality reduction techniques, such as feature extraction and feature selection, are applied to the data and after constructing the five models of each phase, it can be seen that the models identifies anomalies in the data set given.

Keywords: anomaly detection,unsupervised machine learning, high frequency sampled, time series, clustering, dimensionality reduction, DBSCAN, LOF

Sammanfattning

Anomalidetektering av Tidsseriedata med hjälp av Oövervakad Maskininlärningsmetoder: En Klusterbaserad Tillvägagångssätt

För flera företag i tillverkningsindustrin är felsökningar av produkter en fundamental uppgift i produktionsprocessen. Då användningen av olika maskininlärningsmetoder visar sig innehålla användbara tekniker för att hitta fel i produkter är dessa metoder ett populärt val bland företag som ytterligare vill förbättra produktionprocessen. För vissa industrier är feldetektering starkt kopplat till anomalidetektering av olika mätningar. I detta examensarbete är syftet att konstruera oövervakad maskininlärningsmodeller för att identifiera anomalier i tidsseriedata. Mer specifikt består datan av högfrekvent mätdata av pumpar via ström och spänningsmätningar. Mätningarna består av fem olika faser, nämligen uppstartsfasen, tre last-faser och fasen för avstängning. Maskinilärningsmetoderna är baserade på olika klustertekniker, och de metoderna som användes är DBSCAN och LOF algoritmerna. Dessutom tillämpades olika dimensionsreduktionstekniker och efter att ha konstruerat 5 olika modeller, alltså en för varje fas, kan det konstateras att modellerna lyckats identifiera anomalier i det givna datasetet.

Nyckelord: anomalidetektering, öovervakad maskininlärningsmodeller, tidsseriedata, högfrekvent mätdata, klustertekniker, dimensionsreduktionstekniker, DBSCAN, LOF

Acknowledgements

We would like to thank our supervisor for the thesis Jimmy Olsson, the course responsible Anja Janssen and Staffan Aldenfalk Jansson who was our supervisor at the company that provided the project Xylem. Staffan helped us considerably throughout the whole work by always providing assistance when it was needed. We would also like to thank the team in the production center at Emmaboda, Conny Larsson, Bo Milesson, Jonas Elmgren and Johan Hasslestad, for kindly receiving us during the visit and helping us with necessary information about the measurements. Moreover, we would like to give a special thanks to Jürgen Mökander for constantly supporting us and providing relevant information throughout the whole project, even though he was not intended to do so.

Contents

List of Figures

List of Tables

Abbreviations

DBSCAN - Density Based Spatial Clustering of Applications with Noise

MinPts - Minimum Points

LOF - Local Outlier Factor

t-SNE - t-distributed Stochastic Neighbor Embedding

PCA - Principal Component Analysis

FFT - Fast Fourier Transform

PSD - Power Spectral Density

STL - Seasonal and Trend decomposition using Loss

LSTM - Long Short Term Memory

Chapter 1

Introduction

1.1 Background and Problem formulation

Each day, manufactured pumps are being tested in order to determine if they fulfil the necessary performance requirements for delivery. These performance requirements consist of intervals for values of parameters such as flow, head and power. Although the pumps are ready for delivery, one can not assure that the pumps are in an ideal state since they might contain internal damages that are not possible to see in the performance measurements alone. However, through the current, voltage and power, where the latter can be calculated using the current and voltage, one could potentially find relevant patterns in order to detect pumps that are deviating from the normal. These deviating pumps can be seen as anomalies in the data and in order to effectively detect the anomalies, statistical modelling and machine learning methods can be highly valuable. This thesis aims to identify these anomalies using unsupervised machine learning methods.

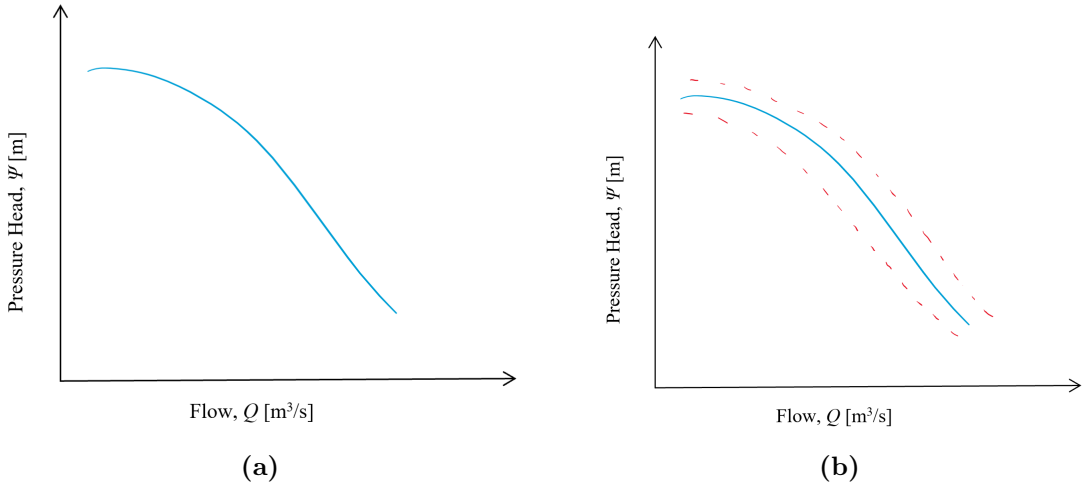
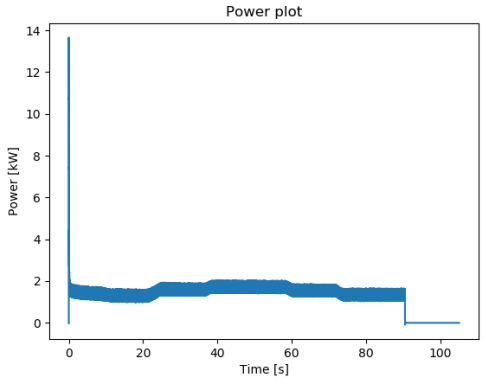
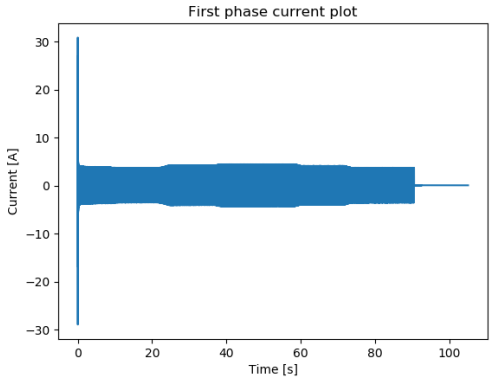


Figure 1.1: Example of a performance curve measuring pressure head Ψ $[m]$ against the flow Q $[m^3/s]$. The pump should operate within the red boundary which is seen in b) to be ready for delivery.

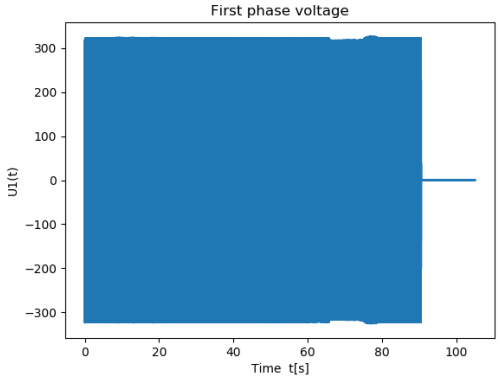
The data consist of high frequency (20 000 Hz) measurements of 3-phase voltage and current being fed to a pump in a test station. A test consists of a pump cycle around 1-2 minutes, and there is data available for more than 20 000 pump tests. The data is timestamped and stored by an external company and can be acquired through an API. The data will initially be observed in both time domain and in frequency domain using Fourier transform. Interesting features will be extracted from the data in time or/and frequency domain. Since the data acquired is unlabeled, it requires the use of an unsupervised machine learning framework to detect anomalies.



(a) Power plot of an arbitrary pump.



(b) Full cycle of first phase current.



(c) Voltage first phase

Figure 1.2: Full cycle of power, current and voltage in one phase of normal behavior. Because of the high sampling frequency, it is hard to visualize the actual waves of the current and voltage when visualizing the complete cycle. Note that the power is the three-phased power being calculated by the current and voltage.

1.2 Purpose

Manufactured Xylem pumps are being tested in order to, through the operating performance, determine if the pumps are ready for delivery. Pumps that pass the performance test might still contain damages but by looking for anomalies in the current, voltage and power data, one can further examine if the pumps are damaged. Damaged pumps may include imbalance, bearing damages, eccentricity etc. These types of damages will worsen the quality of the pump, resulting in a reduction of operating lifetime. This will in turn eventually decrease the reliability of the company since the customers are not receiving the expected operating lifetime. Thus, fulfilling the aim of this project will lead to an improved

manufacturing process resulting in an improved product quality.

In general, anomaly detection is of high importance in many lines of business. These lines include the health care-, security- and manufacturing industry for where this thesis can be useful for the latter. Due to the substantial growth in size and accessibility of Industrial Internet of Things (IoT) type data, the research on prediction and identification methods for time series data will become more and more important. Moreover, this thesis is believed to be useful for future papers regarding using proximity models with time series data.

1.3 Research questions

This thesis aims to answer the following questions:

- *Is it possible using high frequency current and voltage measurements to identify anomalies in a pump?*
- *Is it possible with the help of unsupervised clustering methods to correctly identify the anomalies in the data?*

1.4 Limitations

The database in which Xylem stores data contains measurements of many different pump models and configurations. Due to the sheer size of a measurement sample, the data that will be used is from a period between May 2019 to January 2020 and consists of only a single pump model. Also, the data set used will be unlabeled, which limits the usage of machine learning methods to unsupervised learning.

Since the aim of this work is to find suitable machine learning methods to identify anomalies in the data, it is not of the highest priority to construct final models containing many different machine learning methods, but rather a few functioning ones that are suitable for solving the task given. Therefore, it is worth to emphasize that this thesis is not a comparison of the performance of different unsupervised learning methods. Neither is it an attempt to identify what physical phenomena the found anomalies correspond to. This would be another thesis all together.

1.5 Related Work

Anomaly detection using unsupervised learning methods have shown to be a feasible task in several areas of application. Some areas include cyber-intrusion, medical anomaly detection and industrial damage detection, where the latter is the anomaly detection application used in this thesis [1].

There are several anomaly detection studies comparing different techniques. Clustering based methods, which this thesis will rely on, seems to be an achievable approach. Moreover, clustering based methods for anomaly detection have shown not only to be achievable, but also an adequate approach for different applications of anomaly detection [1], [2], [3], [4].

Hodge and Austin [2] state that there are three fundamental types of approaches within anomaly detection, where one of the approaches is unsupervised clustering. The authors further note two sub-techniques that are commonly employed in the usage of outlier detection, namely *accommodation* and *diagnostic*.

A recent study from Vizcaíno et al. [5] used a clustering based approach for finding anomalies in network data. The study used a Self-Organizing-Map since it allows to perform clustering as well as dimensionality reduction. The study eventually concluded that the technique could successfully be applied to different sources of the network data to find anomalies.

Since the data set used for this thesis will contain high-dimensional inputs, one needs to consider how it will impact on solving the task. From [6] various techniques are introduced and tested of high dimensional data sets, i.e. input variables with a lot of features. Density based methods, such as the Local Outlier Factor gave a good result on the performed experiment.

1.6 Outline

The structure of this paper will be as follows. Following this introductory part, Chapter 2 consists of all necessary mathematical theory of which this work have been relied upon. The mathematical theory section will consist of theory on anomaly detection, unsupervised learning, cluster analysis and dimension-reduction tools. The theory will lay a foundation of the methodology of this work. Chapter 3, the Data overview section, explains the data

more thoroughly and how it was gathered. Chapter 4, Method, presents how the features are extracted from necessary parts of the time series and how the machine learning models are constructed. Chapter 5, Results and Discussion then presents the output of the models and the results of the work, as well as an analysis and discussion of the results. The last chapter, Chapter 6, is the concluding part of the thesis which includes how other approaches could be applied and future work.

Chapter 2

Mathematical Theory

2.1 Anomaly Detection

Anomaly detection is an extensive and active research field concerned with finding patterns in data that do not coincide with expected behavior. These patterns are often referred to as anomalies, discrepancies, outliers, abberations or contaminants depending on the field of usage. Some of the more common fields of usage includes image processing, insurance and healthcare systems, fraud detection, cyber-intrusion detection, surveillance etc [7]. Anomaly detection is applicable to almost any system that deals with data and give valuable information about it. However, it is important to understand that not all anomalies are unexpected. What is to be classified as an anomaly is highly contextual and can therefore not be solved in all contexts by a single method. To this end, there exists numerous different techniques that can be used for different anomaly-types. Anomalies can be further sub-categorized which will be done in the next section.

2.1.1 Anomaly Types

To apply an anomaly detection technique one has to consider the different natures of anomalies. Anomalies can be divided into three different categories, namely *point anomalies*, *contextual anomalies* and *collective anomalies* [1]. For this work, different techniques based on the different categories will be used.

- **Point anomaly:** The point anomaly, is the simplest form of anomaly. A data instance is a point anomaly if the individual instance is deviating from the normal pattern.

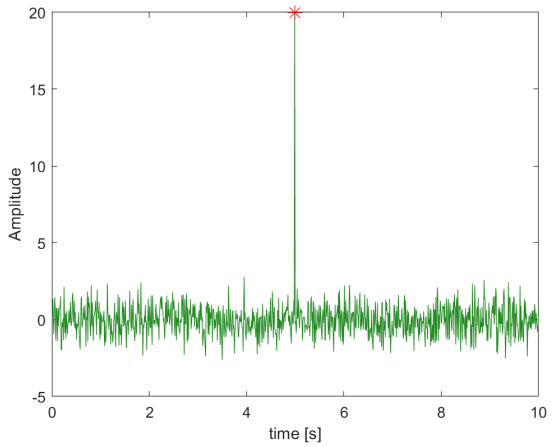


Figure 2.1: Synthetic time series data containing a point anomaly, which is marked in red.

- **Contextual anomaly:** A contextual anomaly occurs when a data instance is anomalous in a specific context. How the context is determined depends on the structure of the data. There are two attributes which need to be taken into consideration when analyzing the contextual anomalies. These are *contextual attributes*, which for example is the time parameter in time series data and *behavioral attributes* which describes the non-contextual characterstics of a data instance. The values of the behavioral attributes within a context is then used to determine the anomalous behavior. It should be stressed that a data instance might be a contextual anomaly in a certain context and normal in another. Moreover, the availability of contextual attributes is of importance when applying such techniques, as some context are straightforward and other cases not.

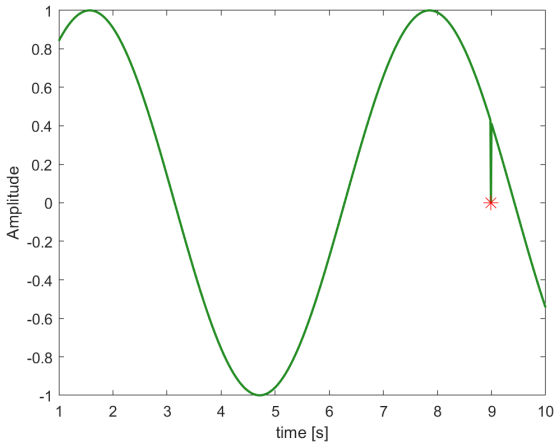


Figure 2.2: Synthetic time series data containing a contextual anomaly, which is marked in red.

- **Collective anomaly:** A collective anomaly occurs when a group of related data instances deviates from the rest of the data set. It is not necessarily the case that the individual data instances within the collective anomaly are anomalies themself, however the instances coupled together as a collection is anomalous. Unlike point anomalies, which can occur in any data set, collective anomalies only exists in data sets where the data instances are related. In contrary, the contextual anomalies depend, as mentioned above, on the availability of context attributes in the data. However, a point anomaly or a collective anomaly can also be a contextual anomaly, as long as context information is integrated within the data in question.

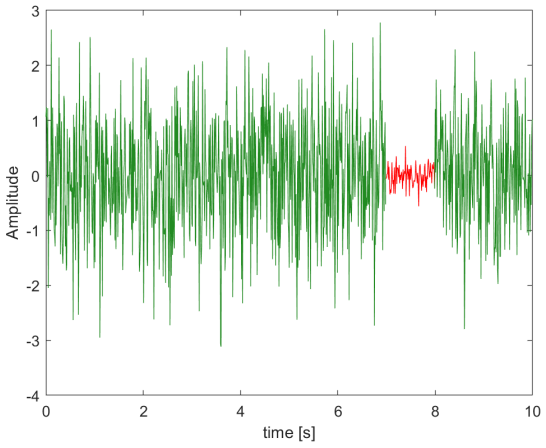


Figure 2.3: Synthetic time series data containing a collective anomaly, which is marked in red.

2.1.2 Anomaly detection techniques

There are multiple techniques that can be used for anomaly detection. The usage of these methods depend on the data structure and should be chosen accordingly. They can be divided into the following classes [8]:

- **Statistical methods** approximates a model of correct behavior based on past measurements. A new measurement would then be marked as an anomaly if it is statistically inconsistent with the model. An easy example is to use a running average as the model. If a new reading is far different from the running average it would indicate an anomaly. But it also means that samples with correct behavior are needed to form a good approximation of our model.

- **Probabilistic methods** revolve around fitting a parametric or non-parametric probabilistic model or distribution to the data. A new measurement is then marked as an

anomaly if the probability of the measurements with respect to the model lies beyond some predetermined threshold or confidence bound.

- **Proximity-based methods** use distance metrics between data points to identify anomalous and correct data. For example, classification can be made for a data point by comparing the density of measurements around the point and the density of measurements around its nearest neighbors.
- **Clustering-based methods** can be seen as a subset of Proximity-based methods. With these methods, clusters are first determined from the data and new measurements are then classified as anomalous if they do not belong to any cluster or contribute to a much smaller cluster of their own.
- **Prediction-based methods** use past measurements to train a predictive model. This model is then used to predict future values, and if these predictions differ too much from the corresponding measurement they are marked as anomalous. They are usually combined with a probabilistic method where a distribution is fitted to the residuals of the prediction. Residuals can then be compared to this distribution and identified as anomalous if they are significantly different from what is expected.

2.2 Unsupervised Machine Learning

In *supervised learning*, one is concerned with predicting the values of one or more outputs or response variables $Y = (Y_1, \ldots, Y_m)$ for a given set of input or predictor variables $X^T = (X_1, \ldots, X_d)$. Let $x_i^T = (x_{i1}, \ldots, x_{id})$ denote the inputs for the ith training case and let y_i be a response measurement. The predictions are based on the training sample $(x_1, y_1), \ldots, (x_N, y_N)$ of previously solved cases, where the joint values of all variables are known. The learning is usually characterized by some loss function $L(y, \hat{y})$, for example the squared loss $L(y, \hat{y}) = (y - \hat{y})^2$, where $\hat{y}$ denotes the estimated response [9].

Suppose that (X, Y) are random variables represented by some joint probability density $p(X, Y)$, then supervised learning can be formally characterized as a density estimation problem where the aim is to determine the conditional probability $p(Y|X)$. Usually, the properties of interest are the "location" parameters μ that minimize the expected error at each x

$$\mu(x) = \underset{\theta}{\operatorname{argmin}} E_{Y|X} L(Y, \theta). \tag{2.1}$$

When conditioning, one receives

$$p(X, Y) = p(Y|X) \cdot p(X), \tag{2.2}$$

where $p(X)$ denotes the joint marginal density of the X values alone. In the supervised learning case, $p(X)$ is typically of no direct concern. Instead, one is interested mainly in the properties of the conditional density $p(Y|X)$. Since Y is often of low dimension, and only its location $\mu(x)$ is of interest, the problem is simplified significantly.

In the *unsupervised learning* case, it is different. In this case, one has a set of N observations $(x_1, x_2, \ldots, x_N)$ of a random d-vector X having joint density $p(X)$. The goal is to directly infer the properties of this probability density without some sort of loss function that is learning the model. The dimension of X is occasionally much higher than in the supervised learning case, as well as the properties of interest are often more complicated than simple location estimates. These factors are somewhat mitigated by the fact that X represents all of the variables under consideration; one is not required to infer how the properties of $p(X)$ change, conditioned on the changing values of another set of variables.

For low dimensional problems, when $d \leq 3$, there are a variety of effective nonparametric methods for directly estimating the density $p(X)$ itself at every X value, and representing it graphically [10]. Due to the curse of dimensionality, which is further discussed below, these methods fail in higher dimensions. One must instead for estimating rather harsh global models, such as Gaussian mixtures or various simple descriptive statistics attempt to characterize $p(X)$.

In general, these descriptive statistics tries to characterize X-values, or collections of such values, where $p(X)$ is relatively large. Fore example, principal components, multidimensional scaling, self-organizing maps and principal curves attempt to identify low-dimensional manifolds within the X-space that represent high data density. This provides information about the associations among the variables whether or not they can be considered as functions of a smaller set of latent, or hidden, variables. This refers to dimensionality reduction and is discussed more thoroughly in section 2.3. Cluster analysis attempts to find several convex regions of the X-space that contain modes of $p(X)$. This can tell whether or not $p(X)$ can be represented by a mixture of simpler densities representing distinct types or classes of observations. The goal for mixture modelling is similar. Assocation rules attempt to construct simple descriptions, or conjunctive rules, that describe regions of high density in the special case of very high dimensional data that are binary valued.

With supervised learning, the measure of success, or lack thereof, is clear. Therefore, it is possible to judge adequacy in particular situations and comparing the effectiveness of different methods over various situations. Lack of success is directly measured by expected loss over the joint probability distribution $p(X, Y)$. This can be estimated in a variety of ways, including cross-validation. In the context of unsupervised learning however, there is no such direct measure of success. The difficulty arises to ascertain the validity of inferences drawn from the output of most unsupervised learning algorithms. One must resort to heuristic arguments not only for motivating the algorithms, as this is often the case with supervised learning as well, but also for judgement as to the quality of the results. This situation has led to heavy proliferation of proposed methods, since effectiveness is a matter of opinion and cannot be directly verified.

In this section, various unsupervised learning techniques that are useful for the model implementation are presented. The two main techniques that will be examined in detail are *cluster analysis* and *dimensionality reduction*.

2.2.1 Cluster Analysis

Cluster analysis, also called clustering or data segmentation, refers to a set of techniques for partitioning a collection of objects into subgroups or *clusters* so that objects within the same cluster are more similar than those assigned to other clusters in some aspect. Clustering is in itself not a single algorithm, but rather the task that is to be solved. There are various algorithms that aims at solving this task. These algorithms can be divided into groups of methods whose definitions of what constitutes a cluster as well as how to efficiently locate them can differ significantly. Common amongst all clustering techniques is the attempt to group the objects based on the supplied definition of similarity that it uses. Therefore, central to cluster analysis is the notion of similarity between objects. Clustering can in general be defined as an optimization problem where the similarity measure within subgroups is to be maximized. *Connectivity-based* clustering is a group of methods based on a core idea: Objects are more related to nearby objects than to objects farther away. The method, more frequently denoted as *hierarchical clustering*, initially treats each observation as a separate cluster. It then starts merging points together based on their distance from eachother. The procedure can be visualized using a dendrogram which shows the hierarchical relationship between the clusters.

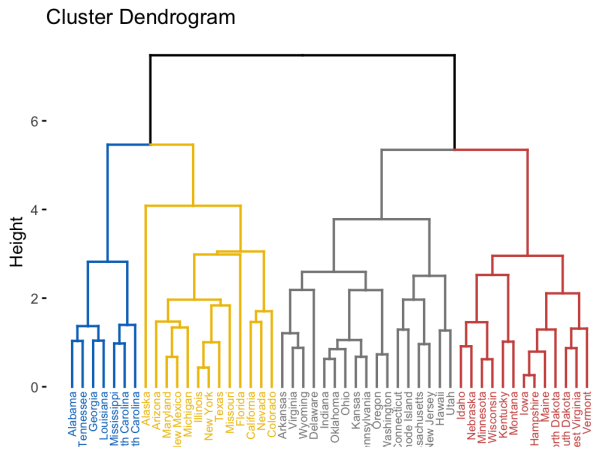


Figure 2.4: Hierarchical cluster dendrogram of height vs. US state. The image was obtained from [11].

Density-based clustering instead defines clusters as contiguous regions of high point density separated by regions of low point density. A big advantage of these methods is that there is no restriction on cluster-shape. One of the more used density-based method is DBSCAN which will be explained in more detail in section 2.2.1.1. *Centroid-based* clustering techniques, such as k-means-clustering, are based on finding the center of gravity of each cluster. These centroids are initiated arbitrarly and then iteratively updated according to the minimization of the distance from each point to its respective cluster centroid [9]. These methods suffer when a cluster shape is not well explained by its centroid as can be shown in figure 2.5.

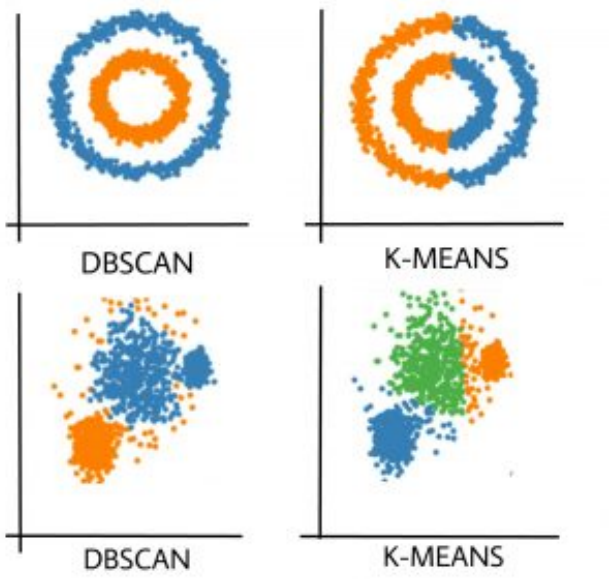


Figure 2.5: Visualization of how k-means struggles to correctly cluster the data when the centroid position does not give valuable information. Image taken from [12].

The last collection of methods that will be overviewed are the *Distribution-based* clustering methods. These methods, closely related to statistics and probability theory, attempt to find groups within the data which can be explained by the same distribution. It can be viewed as trying to find the distributions that each cluster is generated by. However, these methods usually require some constraint on the model complexity to avoid overfitting. The choice of distribution effectively determines the expected shape of clusters.

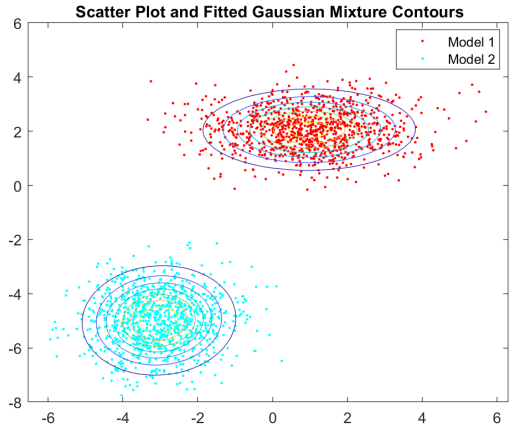


Figure 2.6: Two clusters each assigned a Gaussian distribution.

2.2.1.1 The DBSCAN algorithm

DBSCAN or Density-Based Spatial Clustering of Applications with Noise is a density based clustering method for identifying clusters in spatial data. It does so by looking at the local density of data elements. DBSCAN can also determine if information should be classified as outliers or noise. In short terms, the algorithm places regions with similar in-region neighbor density into separate clusters. Below, the formal definitions will follow [14]:

Definition 1: (ε-neighborhood of a point) The ε-neighborhood of a point p, denoted by $N_\varepsilon(p)$, is defined by

$$N_\varepsilon(p) = \{q \in D | dist(p, q) \leq \varepsilon\}. \tag{2.3}$$

That is, all points q with a distance to point p less than ε, where ε denotes the neighboring size, belong to the neighborhood of the point p.

Definition 2: (Directly density reachable) In a cluster there are two types of points, border points and core points. The border points make up the hull around the core points. There tends to be significantly less points within a border points ε-neighborhood then that of a core point. The border points will still be part of the cluster if they belong to the ε-neighborhood of a core point

- $p \in N_\varepsilon(q)$.

For q to be a core point it's ε-neighborhood is required to contain a minimum number of points $MinPts$

- $|N_\varepsilon(q)| \geq MinPts$ - (Core point condition).

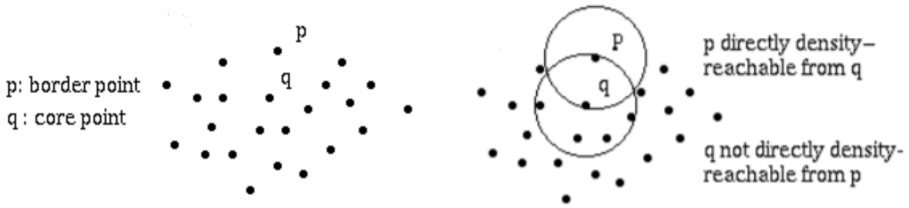


Figure 2.7: Core and border points. Figure taken from [14].

Definition 3: (Density reachable) A point p is density-reachable from a point q with respect to ε and $MinPts$ if there is a chain of points $p_1 \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1}

is directly density-reachable from p_i.

Definition 4: (Density-connected) A point p is density-connected to a point q with respect to ε and $MinPts$ if there is a point o such that both, p and q are density-reachable from o with respect to ε and $MinPts$.

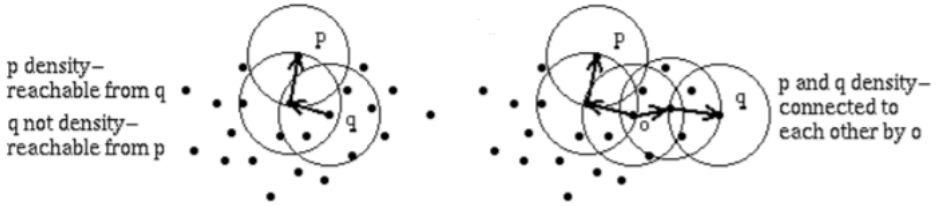


Figure 2.8: Density reachability and connectivity. Figure taken from [14].

Definition 5: (Cluster) If a point p is in a cluster A, a point q is also in cluster A if it is density-reachable from point p with respect to a distance and a minimum number of points within that distance,

- $\forall p, q$: if $p \in C$ and q is density-reachable from p with respect to ε and $MinPts$, then $q \in C$.

Two points p and q belonging to the same cluster C is analogous to point p and q being density-connected w.r.t. ε and $MinPts$

- $\forall p, q \in C$: p is density-connected to q with respect to ε and $MinPts$.

Definition 6: (Noise) Let $C_1, \ldots, C_k$ be the clusters of the database D with respect to parameters ε_i and $MinPts_i$, $i = 1, \ldots, k$. The *noise* is the set of points that are not assigned to any cluster C_i, i.e., noise $= \{p \in D | \forall i : p \notin C_i\}$.

2.2.2 The LOF algorithm

Local Outlier Factor (LOF) is a technique that uses nearest neighbor search. The method gives each data point a score depending on the average density of the points closest neighbors divided by the density of the point itself. It takes the number of neighbors to use as an argument. The formal definitions are as follows [7]:

Definition 1: (Hawkins-Outlier) An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.

Definition 2: (DB($pct, dmin$)-Outlier) An object p in a data set D is a $DB(pct, dmin)$-outlier if at least percentage pct of the objects in D lies greater than distance $dmin$ from p, i.e, the cardinality of the set $q \in D | d(p,q) \leq dmin$ is less than or equal to $(100 - pct)$ % of the size of D.

Definition 3: (k-distance of an object p) For any positive integer k, the k-distance of object p, denoted $k - distance(p)$ is defined as the distance $d(p,o)$ between p and an object $o \in D$ such that

1. For at least k objects $o' \in D \backslash p$ it holds that $d(p,o') \leq d(p,o)$, and
2. For at most $k-1$ objects $o' \in D \backslash p$ it holds that $d(p,o') \leq d(p,o)$.

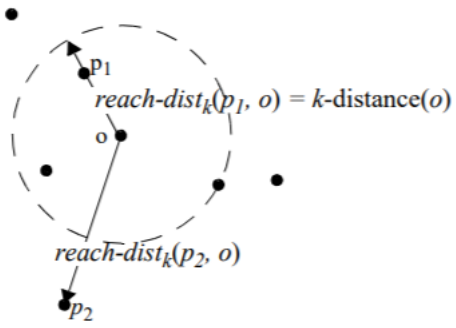


Figure 2.9: reach-dist($p1, o$) and reach-dist($p2, o$), for $k = 4$. Figure taken from [7].

Definition 4: (k-distance neighborhood of an object p) Given the k-distance of p, the *k-distance neighbourhood of p* contains every object whose distance from p is not greater than the k-distance, i.e. $N_{k-distance(p)}(p) = \{q \in D \backslash \{p\} | d(p,q) \leq k - distance(p)\}$. These objects q are called the *k-distance neighborhood of p*.

Definition 5: (Reachability distance of an object p with respect to object o) Let $k \in \mathbb{N}$. The *reachability distance* of object p with respect to object o is defined as

$$\text{reach-dist}_{\ k}(p,o) = \max\{k\text{ -distance }(o), d(p,o)\}. \tag{2.4}$$

Definition 6: (Local reachability density of an object p) The Local reachability density of an object p is defined as

$$\text{lrd}_{\text{Min }Pts}(p) = 1/\left(\frac{\sum_{o \in N_{MinPts}(p)} \text{reach-dist}_{\text{Min }Pts}(p,o)}{|N_{\text{Min }Pts}(p)|}\right). \tag{2.5}$$

Definition 7: (Local outlier factor of an object p) The (local) outlier factor of p is defined as

$$LOF_{\text{MinPts}}(p) = \frac{\sum_{o \in N_{\text{MinPts}}(p)} \frac{\text{lrd}_{\text{MinPts}}(o)}{\text{lrd}_{\text{MinPts}}(p)}}{|N_{\text{MinPts}}(p)|}. \tag{2.6}$$

The outlier factor of object p captures the degree to which p is an outlier. It is the average of the ratio of local reachabality density of p and those p's $MinPts$-nearest neighbors. It is easy to see that the lower p's local reachability density is, and the larger the local reachability densities of p's $MinPts$-nearest neighbors are, the higher is the LOF value of p.

2.2.3 Hyperparameter estimation

Considering the case of anomaly detection and the structure of the data and since the dynamics of the measurements are the same, one cluster is expected to contain all points excluding outliers. This makes it so that the $MinPts$ parameter can be set to be around half the data size in order to create only one cluster. The value for the parameter ε can be chosen by plotting the $MinPts - 1$ nearest neighbor distance ordered in an ascending manner. A good value for ε is then where the gradient of this plot increases the most drastic and creates an "elbow" shape in the nearest neighbor graph. For a too large ε value, clusters will merge into each other and create larger but fewer clusters. Whereas if ε is too small, much of the data will not be clustered [14].

2.3 Dimensionality Reduction

Statistical and machine learning methods face a great problem when dealing with high-dimensional data, and normally the number of input variables, i.e. the dimensionality, is reduced in order to apply a functioning algorithm. This process of reduction is called *dimensionality reduction* and it can be made in two different ways. The first way is to keep the most relevant variables from the original data set. This technique is called *feature selection*. The second way is to exploit the redundancy of the input data by finding a smaller set of new variables, each being a combination of the input variables, ideally containing the

same information as the input variables. This process is called *feature extraction*. These two approaches are further explained below [15].

2.3.1 Curse of Dimensionality

First introduced by Bellman [16], the course of dimensionality indicates that the sample size N needed to estimate an arbitrary function with a certain accuracy grows with the dimensionality d of the input variable $\mathbf{x_i}$.

Consider a nearest-neighbor approach for uniformly distributed inputs in a p dimensional unit hypercube. Suppose a hybercubical neighborhood about a target point is sent out to capture a fraction r of the observations. Since this corresponds to a fraction r of the unit volume, $e_p(r) = r^{1/p}$ will therefore be the expected edge length. In ten dimensions $e_{10}(0.01) = 0.63$ and $e_{10}(0.1) = 0.80$, while the entire range for each input is only 1.0. Hence, to capture 1 % or 10 % of the data to form a local average, 63 % or 80 % of the range of each input variable needs to be covered, resulting in neighborhoods that are no longer local [9].

One way to overcome the curse is applying dimensionality-reduction tools on the data. Below are some techniques used for this work [17].

2.3.2 Feature extraction

Feature extraction is a process aimed at reducing the number of features, or dimensions, of a data set. The dimensionality of the data set is reduced by encoding the original m-dimensional data into d number of features, creating instead a d-dimensional data set where $d \leq m$ [9]. The new features should aim to summarize the information given by the original data. There are a number of different reasons why this is done. Firstly, a data set with fewer dimensions will in most cases have a faster training time and it will also reduce the risk of overfitting resulting in a more generalized model. Secondly, a lot of distance-based machine learning algorithms struggle when the number of dimensions in the data is very large. This is related to *the curse of dimensionality*. In order to make these algorithms more effective it is important to reduce the number of features in the data. Generally feature extraction should follow the occam's razor principle, that one should opt for the simpler solution. Feature extraction can be performed in both supervised and unsupervised manners. The most basic supervised feature extraction technique would be

the use of expert knowledge in selecting what features to extract. Since this is not always reasonable, another way is to calculate a plethora of features and perform *feature selection* on this set. A group of common unsupervised feature extraction or dimensionality reduction methods are *Autoencoders*. Autoencoders are neural networks that are trained to encode the input into a lower dimensional plane and then to recreate the input signal from this lower dimensional representation. The encoded data can then be used as features for other algorithms. However the issue lies in that one usually need data samples with normal behavior in order to train the model.

2.3.3 Feature selection

Feature selection is another technique aimed at reducing the number of features in a data set. Feature selection, unlike *feature extraction*, does not find any new features to describe the data by, it instead selects a subset of the inherent features to save for modelling. This subset can be decided in various ways. In supervised learning this subset is often selected as the subset that gives the best test-performance for the classifier that is being trained [18]. In unsupervised learning, due to the lack of accuracy measures, one usually opts for the subset that holds the most variation of some describing statistic. This selection can be made using *variable ranking*. This is the process of sorting the features according to some scoring function S. Feature selection using variable ranking is then done by selecting the k highest ranked features according to the scoring function S [18]. Some common choices of S are variance, min-max spread or the density around the mean or median.

2.3.4 t-SNE

t-SNE, or t-distributed Stochastic Embedding, is a nonlinear dimensionality reduction technique that is intended to visualize high-dimensional data in a low dimensional space of two or three dimensions. The similarity of data point x_j to data point x_i is the conditional probability $p_{j|i}$, that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian distribution centered at x_i. For nearby points, $p_{j|i}$ is relatively high and for widely separated points, $p_{j|i}$ will almost be infinitesimal. The conditional probability $p_{j|i}$ can be defined as [19]:

$$p_{j|i} = \frac{\exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\left\|x_i - x_k\right\|^2 / 2\sigma_i^2\right)}, \tag{2.7}$$

where σ_i denotes the variance of the Gaussian centered on data point x_i. The variance σ_i needs to be selected and it is not probable that there is a single value of σ_i that is optimal

for all data points in the data set, since the density of the data might vary. In dense regions, smaller values of σ_i is usually more appropriate than in more sparse regions. Any choice of σ_i induces a probability distribution P_i over all data points. P_i has an entropy which increases as σ_i increases. This algorithm then performs a binary search for the value of σ_i that produces a P_i with a fixed user-defined perplexity. The perplexity is defines as

$$\text{Perp}\left(P_i\right) = 2^{H(P_i)}, \tag{2.8}$$

where $H(P_i)$ is the Shannon entropy of P_i measured in bits

$$H\left(P_i\right) = -\sum_j p_{j|i} \log_2 p_{j|i}. \tag{2.9}$$

The perplexity can be seen as a smooth measure of the effective number of neighbors. The performance of the algorithm is reasonably robust to changes in perplexity, and typical values lays between 5 and 50.

Considering equation (2.7), $p_{i|i}$ is set to zero as only the pairwise similarity is of interest. Moreover, when a high dimensional point x_i is an outlier, then the values of $p_{j|i}$ will be extremely small. Consequently, the position of the map point y_i is not well determined by the positions of the other map points. To circumvent this problem, the joint probability is instead set to be the symmetrized conditional probabilities

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \tag{2.10}$$

which ensures that $\sum_j p_{ij} > \frac{1}{2n}$ for all data points x_i.

Let y_i and y_j denote the low-dimensional counterparts of the high-dimensional data points x_i and x_j, then it is possible to construct the pairwise similarities in the low-dimensional map q_{ij}

$$q_{ij} = \frac{\left(1 + \left\|y_i - y_j\right\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \left\|y_k - y_l\right\|^2\right)^{-1}}. \tag{2.11}$$

The maps p_{ij} and q_{ij} will be equal if the map points y_i and y_j correctly models the similarity between the high-dimensional points x_i and x_j. Therefore, the aim is to minimize the single Kullback-Leiber divergence between a high-dimensional joint probability distribution P and

a low-dimensional joint probability distribution Q. The Kullback-Leiber divergence, which becomes the cost function in this case, is defined as

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{2.12}$$

where, similarily as before, $p_{ii} = q_{ii} = 0$. Moreover, it is assumed that $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$. Minimizing (2.12) is performed using a gradient descent method which ultimately yields a map that well reflects the high dimensional inputs. The gradient is defined as

$$\frac{\delta C}{\delta y_i} = 4 \sum_i (p_{ij} - q_{ij}) (y_i - y_j) \left(1 + \|y_i - y_j\|^2\right)^{-1}. \tag{2.13}$$

The pseudo code, where $\mathcal{Y}^{(t)}$ indicates the solution at iteration t, η indicates the learning rate, and $\alpha(t)$ represents the momentum at iteration t, can be written as:

Algorithm 1 t-distributed Stochastic Neighbor Embedding

INPUT: Data set $X = \{x_1, x_2, \ldots, x_n\}$
Cost function parameters: Perplexity $Perp$
Optimization parameters: Number of iterations T, learning rate η, momentum $\alpha(t)$.
OUTPUT: Low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$

BEGIN
Compute pairwise affinities $p_j|i$ with perplexity $Perp$
Set $p_{ij} = \frac{p_{j|i+} + p_{i|j}}{2n}$
Sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
for t = 1 : T **do**
  Compute low dimensional affinities q_{ij}
  Compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$
  Set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
end for
END

Chapter 3

Data overview

3.1 Data collection

The data used in this work is provided by Xylem. The measurement data of each pump being tested is stored in a database. The raw data can afterwards be downloaded through an API using stored timestamps. The measurements are taken with the help of equipment provided by Semiotic Labs.

3.2 Data description

The available data can be divided into two parts. The first part is time series of the voltage and current being fed to the 3-phase electric pump motor. This data set contains 6 columns: $I_1, U_1, I_2, U_2, I_3, U_3$, for each test cycle. This data is being measured with a sample rate of 20 kHz. In the case of a voltage drop, the voltage will quickly be regulated back to its normal operating value with the help of a voltage regulator. As a result, it will not be relevant to attempt to find anomalies in the voltage data alone. The voltage data will only be necessary to, along with the current, compute the three-phased power $Po(t)$. Regarding the current data, only the first-phase current will be used in the analysis. Due to the immense data size and the required computer RAM capacity, this is one way to instead include three times more data samples N using the same data size. Since the aim is to construct a model that can identify anomalies, limiting the data to only include one phase but being able to use three times as many samples seems rational since anomalies are expected to occur in all phases. A test cycle consists of the following four distinct parts. Please note that the figures below show the normal measurement pattern.

1. **Startup:** The startup is a very short process of about 100 milliseconds where the

current spikes and then settles.

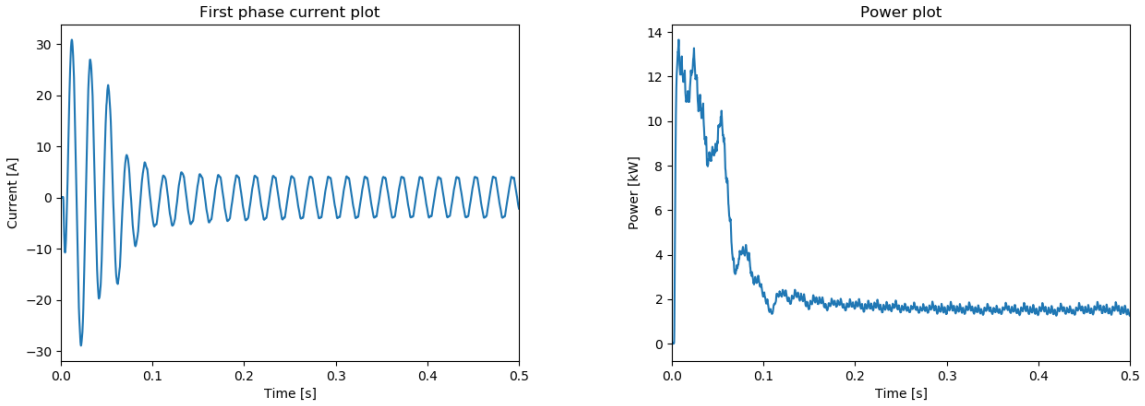


Figure 3.1: The first 500 milliseconds of the first phase current and power. The startup, or the initial power spike, can be seen as the first 100 milliseconds of the measurement.

2. **Venting:** The venting is a process the pump goes through to get all the air out of its housing.

3. **Duty points:** The duty points, or stationarity, are the parts of the cycle where the measurements of flow Q and pressure head Ψ are being done. The pumps regulatory system tries to find the predefined levels which are about 15 %, 50 %, 85 % of the maximum flow Q_{max}, yielding three duty points. Once the system is stable the measurements are made- and averaged over a 3 second window before moving on to the next duty point.

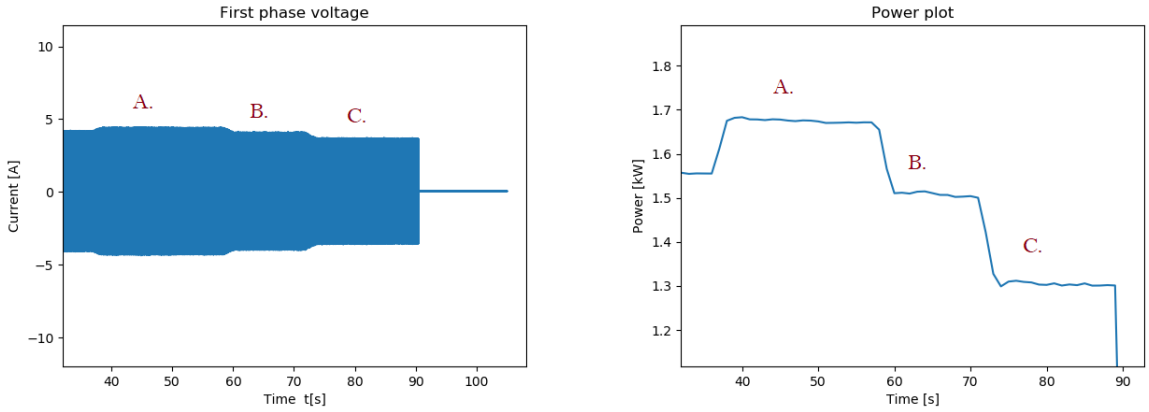


Figure 3.2: The three duty points shown in the first phase current and the power. A. First duty point. B. Second duty point. C. Third duty point. Note that the power measurement has been downsampled for easier extraction and visualization.

4. **Shutdown:** After all three duty points have been measured, the pump turns off which causes a very short oscillation before all power shuts off.

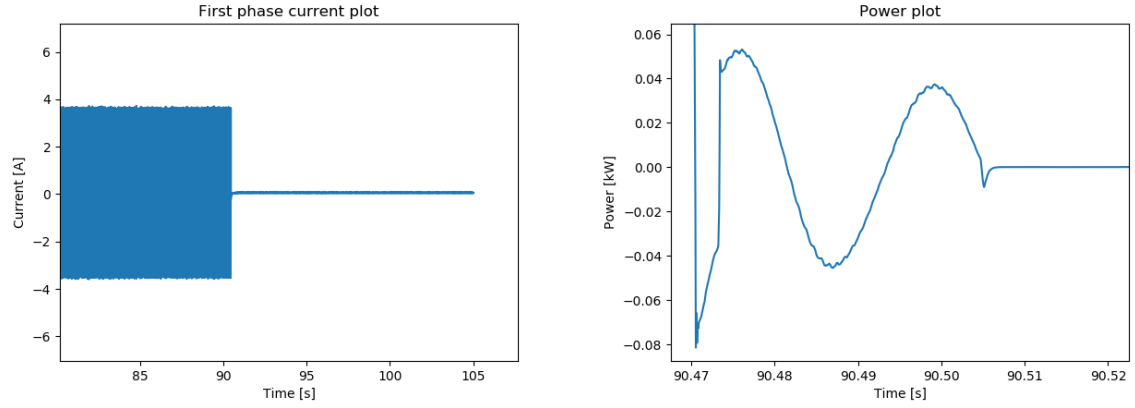


Figure 3.3: The shutdown part shown in the first phase current and the power.

A full cycle usually range between 1-2 minutes depending on how long it takes for the regulatory system to stabilize. This measurement data is available through a 3rd-party API and comes in the form of WAV audio files. Each WAV file is 15 seconds long and contain data of size 6x300042. Making a full pump cycle contain about 4-8 times more data. The data set used for modeling is from the time period 2019-05-13 to 2020-01-28 and consists of only one specific pump model with constant configurations. There are $N = 2100$ pumps

in the data set. The second part of the available data is raw information about each pump being tested. The information from this data set that will be used is the reference power Po_{ref} for each duty point of each pump.

The marked areas in figure 3.4 show the parts of a full test cycle that will be used for modeling. These include the startup, the duty points and the shutdown.

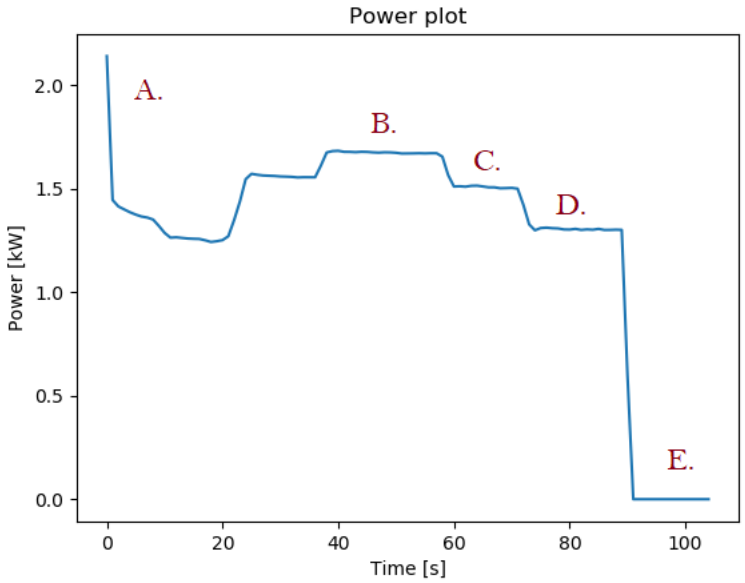


Figure 3.4: The full cycle of a pump's power measurement. A. Startup, B. First duty point, C. Second duty point, D. Third duty point, E. Shutdown. The measurement has been downsampled for visualization purposes.

Chapter 4

Method

4.1 Method Description

In this section, the full method section will be explained. Five different models, one for each measurement phase, is constructed. The phases consist of the startup, the first duty point, the second duty point, the third duty point and the shutdown. This is done mainly because of further reducing the dimensions of the input variable $\mathbf{x_i}$. Moreover, Fast Fourier Transform, or FFT, is only performed on the stationary data points and not on the startup and shutdown parts since the startup and shutdown parts are so small, yielding mostly high frequency information. It is desired to perform an FFT rather than feature extraction due to the information some frequencies give about physical phenomena.

4.1.1 Data extraction

Once the data has been collected, the data pre-processing can be initiated. Below presents how the time series data from the relevant part of the measurements, namely the startup, the three duty points and the shutdown part are being extracted. For the startup and shutdown phases, the power data $Po(t)$ will be used whereas for the duty points the first phase current I_1 will be used. The power is used for the startup and shutdown phase because it includes information from all three phases without too much memory usage. A downsampled version of the power will be used to locate the duty points as outlined in 4.1.1.2. This is done to reduce memory usage.

4.1.1.1 Startup extraction

The extraction of the startup is done using a fixed time window. The data can be extracted directly from the Po_t using a fixed window length. The window covers the first 2500 data points, which with a samplerate of 20 kHz translates to 125 ms. This window covers the whole power spike until the signal settles down before the venting begins.

4.1.1.2 Stationarity extraction

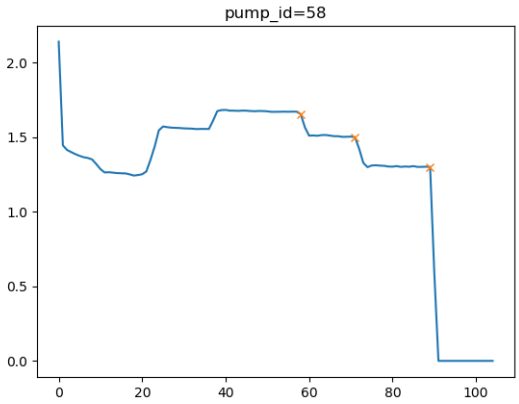
One part of the data for a test cycle that will be used for clustering is the stationary areas during which the flow and head measurements have been taken. These areas can be located using the reference power measurements Po_{ref} for each duty point. The window during which these measurements are made are between 3-5 seconds. The algorithm for extracting the stationary parts is as follows:

1. Calculate the gradient z_t of the power data Po_t for the complete pump cycle $0 \leq t \leq T_n$.

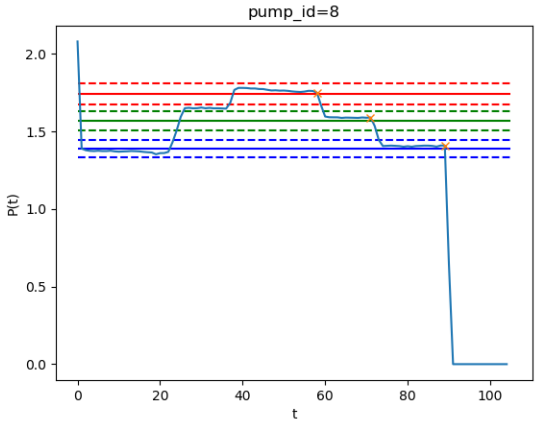
2. Locate all peaks Z_t of the gradient and remove all peaks smaller than a predetermined threshold r.

3. Iterate backwards over the peaks Z_t of the gradient z_t starting at $t = T_N$. If for a peak Z_t^i each data point Po_t in the time span $t_i - 5 < t < t_i - 2$ fulfill $|Po_t - Po_{ref}| < r \cdot Po_{ref}$, extract data I_t in this region. r can be viewed as a dimensionless scaling parameter and is set to be 0.04. Other inconsistencies in the data will also be filtered out by the algorithm. For example, instances such as measurements using more than three duty points and abnormal measurement pattern. This will result in a decreased sample size N for the stationary section of that pump.

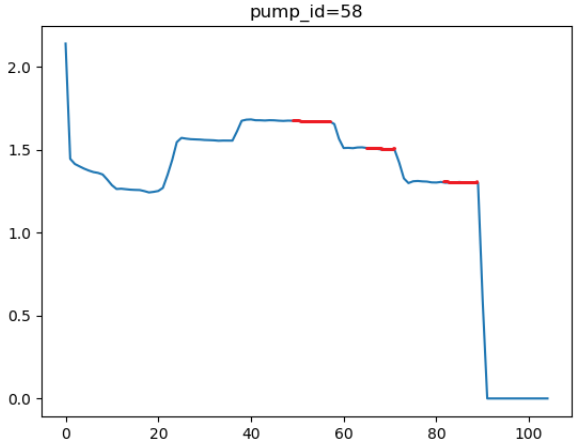
The peaks of the gradient are found using *find_peaks* from the python package *scipy.signal*. An illustration of the process elements is found in figure 4.1.



(a) Orange crosses marks where Q_{t_i}.

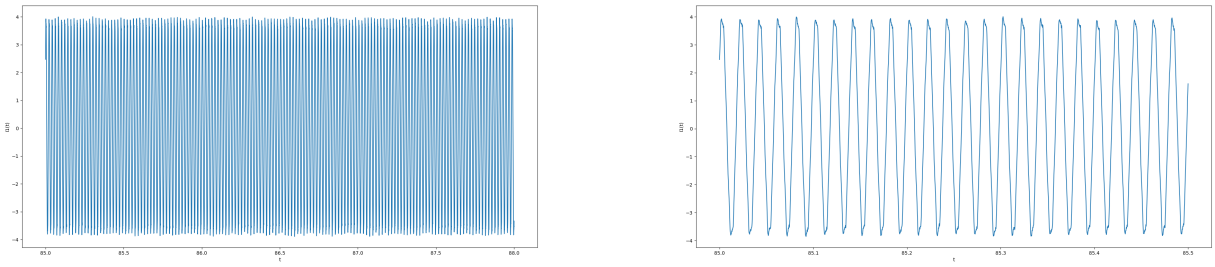


(b) Control if peaks within limit.



(c) Extraction of $t_i - 5 < t < t_i - 2$.

Figure 4.1: Figure of power curve $Po(t)$ stationarity extraction. In b) Po_{ref} marked with red, green and blue horizontal lines. $Po_{ref} \pm 0.04 Po_{ref}$ marked with red, green and blue dotted horizontal lines.



(a) 3 seconds

(b) 0.5 seconds

Figure 4.2: Example of extracted first phase current data I_1 for duty point 3.

4.1.1.3 Shutdown extraction

The shutdown of the cycle is the point where the power goes from $Po > 0$ (load) to $Po = 0$ (no load). The time $t_{shutdown}$ for which the shutdown begins is not consistent between cycles since each cycle is not of the same length. Because of this, $t_{shutdown}$ is found in a similar way as the stationarity extraction.

1. $t_{shutdown}$ is found by locating the time t of the last peak Z_t^i of the gradient z_t that lie above the threshold r. $t_{shutdown}$ is then equal to t. Samples are filtered out if the corresponding gradient does not lie above the threshold r.

2. Extract data Po_t for $t \in [t_{shutdown}, t_{shutdown} + t_{sd}]$, where t_{sd} is the total duration of the shutdown sequence.

4.1.2 Feature extraction and selection

The extracted data for the startup and shutdown will undergo a general feature extraction using the programming language *python* with the *tsfresh* package. This package calculates hundreds of features and automatically filters out features with missing data. These features will first undergo normalization and then feature selection. The normalization technique used is called Min-Max Feature scaling and is performed so that all features are on a common scale in the range. Data normalization is a crucial step before any clustering algorithm. In this case the data will be normalized into the range [0,1]. The scaling is done via:

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)}. \tag{4.1}$$

This normalization retains the relative distance between data points which is a positive property for outlier detection.

When selecting what features to keep for clustering one has to think about what the normalized data is expected to look like, within each feature. The normal data samples are expected to be clustered together, while anomalous samples are expected to be close to the edges of the proposed range since $x = 0$ and $x = 1$ are now the smallest and largest points within the range. One of the more common dimensionality reduction tools is Principal Component Analysis, or PCA. However, since a single outlier affects variance less the more data that is available, this tool will not be reasonable to implement. A common scoring function for outlier detection is Min-Max spread. However after the proposed normalization, Min.Max spread is not applicable. The choice of scoring function in this case of anomaly detection will instead be the distance between the 90th and 10th quantile of data for each feature

$$S = quantile(0.90) - quantile(0.10). \tag{4.2}$$

This scoring function is connected to the density of the main cluster of data. If this cluster is very dense it means that the points at the far ends of the range [0,1] are further away from the cluster and are more likely to be outliers. The choice of quantile is connected to the expected contamination rate of outliers. A comparison of a different quantile can be found in the appendix figure A.11 and shows that the choice does not seem to have a large impact. Note, that the selected features may not necessarily be the same for each measurement phase.

4.1.3 FFT and frequency selection

The stationary part corresponding to duty point 1, 2 and 3 goes through a FFT. With the one-sided FFT coefficients, one can determine the power spectral density $s(f)$ as

$$s(f) = \frac{|a(f)|^2}{N_f}. \tag{4.3}$$

Where $a(f)$ is the FFT coefficients of the one-sided spectrum and N_f is the number of data points fed to the FFT. The Power Spectral Density, or PSD, coefficients $s(f)$ are then normalized to have values between 0 and 1 using Min-Max Feature scaling shown in equation 4.1. With the normalized coefficients the desired frequencies can be selected using the same quantile-based scoring function proposed in the previous section. The PSD of a pump sample can be found in figure 4.3 showing frequencies between 0 and 300. For more

theory about Fourier transforms, the reader is referred to section A.2 in the appendix.

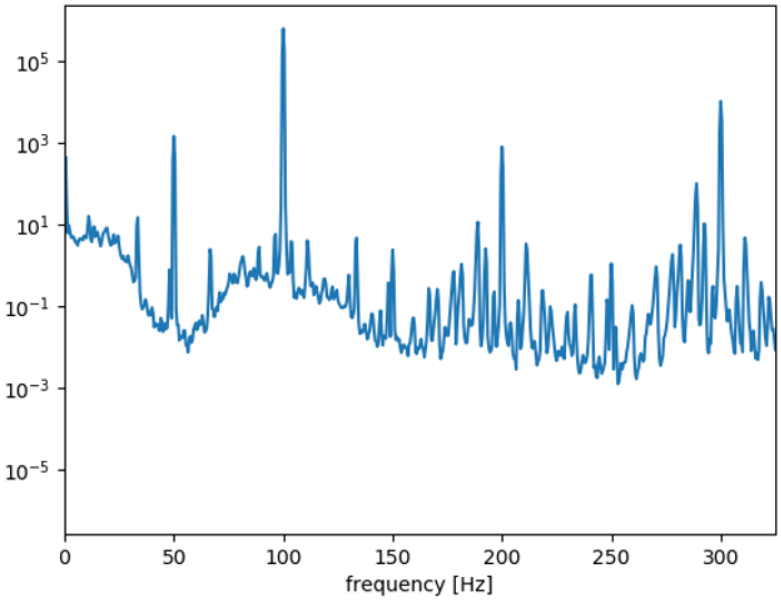


Figure 4.3: Frequency spectra for the first phase current of duty point 3 of a pump measurement.

4.1.4 DBSCAN

After the startup, stationary and shutdown parts of the time series have been extracted and feature- and frequency selection have been performed, the clustering can be initiated. The DBSCAN algorithm is applied through the function *DBSCAN* in the python package *sklearn.clustering*. The algorithm procedure is based of the definitions presented in section 2.2.1.1, and how the hyperparameter ε is determined can be seen in section 2.2.3. The parameter $MinPts$ is chosen as $N/2$. As explained earlier, choosing a larger value of N increases the number of points needed to form a cluster. Using $MinPts = N/2$ technically disables the formation of more than one cluster. Thus, points that do not belong to the main cluster are marked as outliers.

4.1.5 LOF

The procedure for applying the LOF is based of the definitions in section 2.2.2. It is applied using the *LOF* function from *scikit.cluster* in python. A LOF-score threshold parameter γ is used to infer which points are to be marked as outliers based on their LOF-score. The threshold is found using a visualization of ordered LOF-scores as can be seen in the appendix figures A.1-A.5. The common choice that will also be implemented here is to look for an

elbow shape in the graph. The $MinPts$ parameter setting is set to the same value as for DBSCAN.

4.2 Model evaluation

4.2.1 Result visualization

The final part of the model construction is visualization of the established clusters. Visualizing the clusters of all features will not be possible considering the high dimensionality of the data set. Thus, it is necessary to use a dimensionality reduction tool to permit the visualization of clusters through a low dimensional map. The visualization part also permits the model to be evaluated. The implementation of t-SNE is done through python using the function $TSNE$ from $sklearn.manifold$. A certain perplexity $Perp$ will be set for all models in the main analysis. However, to show that t-SNE gives robust results, three different values of the perplexity parameter will be used for all models, where one is used for the main analysis. The figures using the other two perplexities can be seen in the appendix figures A.12-A.21.

4.3 Rationale for Method Choice

The methods for this thesis involve using clustering methods, more specifically density based clustering, and applying different dimensionality techniques in order to identify anomalies in the data set given.

Clustering approaches to anomaly detection have been used before [1], [2], [3], [4]. Moreover, the study from Guggilam et al. [6] compares different outlier detection techniques, such as k-means-clustering, angle-based-outlier detection and LOF. In the experiment, the number of features of the data set ranged between $9 \leq d \leq 57$. Considering that the number of features will have an upper limit of $d = 40$ for this work, the results from [6] can therefore be a good justification why the LOF algorithm was implemented in this thesis. To ensure that the choice of number of features d limits the effect of curse of dimensionality, the choice of the upper limit d is set to be slightly lower than the upper limit in [6].

Density-based methods, such as the DBSCAN algorithm are used over centroid based methods, such as the popular choice k-means clustering. In DBSCAN, one is not forced to specify

the number of clusters beforehand. This is however necessary in the k-means-clustering [20]. As mentioned in 2.2.1, by also not having a restriction of the cluster shape, the choice of density based methods are further justified.

The Gaussian mixture model approach was also taken into consideration. Applying the Expectation-Maximization algorithm in a Gaussian mixture model framework for finding a latent variable is indeed a feasible unsupervised method for these types of problems [21]. However, it was kept out with the intention of keeping the method more generalized and not having to imply a Gaussian distribution on the data set.

Regarding the dimensionality reduction tool, the use of t-SNE was due to the following. Even though the distance between two neighboring points is altered after the t-SNE mapping, one of the goals of the t-SNE algorithm is to represent high dimensional data correctly in lower dimensions. This includes keeping neighboring points in high dimensions also neighbors in the lower dimensional map [19].

4.4 Method Evaluation

As mentioned in section 2.2, the evaluation of unsupervised learning is a challenge since there are no direct measures of success as in the supervised learning case. This makes the evaluation a bit dubious. There are, as in every method choice, some drawbacks of the methods proposed for this work.

The disadvantage of using the LOF algorithm lays in determining which data point are outliers and which are not. As mentioned in [22] there is no clear rule regarding what LOF-score that determines an outlier. However, as explained earlier, the common method of choosing a threshold, γ is to look for an elbow shape in the ordered LOF-score graph.

One evaluation can be to look at the number of joint outliers, which refers to outliers found in both the DBSCAN algorithm and LOF with a specific threshold γ. If the number of joint outliers is large then one can arguably increase the confidence that the outliers detected in both methods are relevant.

Lastly, the numbers of outliers will be analyzed with increasing number of feature d. Due to the curse of dimensionality, the sample size N needs to be increased with increasing number

of features d. As a result, the model will be less reliable and more outliers will be expected for both the DBSCAN and LOF algorithm.

Chapter 5

Results and Discussion

The outline of this chapter will be as follows: First showing the results of the variable ranking and thereof how many features to include. Following will be the clustering model results for all the measurement phases. Lastly, there will be a summary discussion comparing the results of each phase.

5.1 Variable ranking

The results of the variable ranking are shown in figure 5.1. The result sought after is that there exists a clear optimal cutoff limit for the number of features to use for clustering. As explained earlier, the variables are ranked in an increasing order of quantile-density (the distance between the 90th and the 10th quantile). An ideal result would then be a number of features having very low distance, whereas the remainder of variables would have a quick increase in distance.

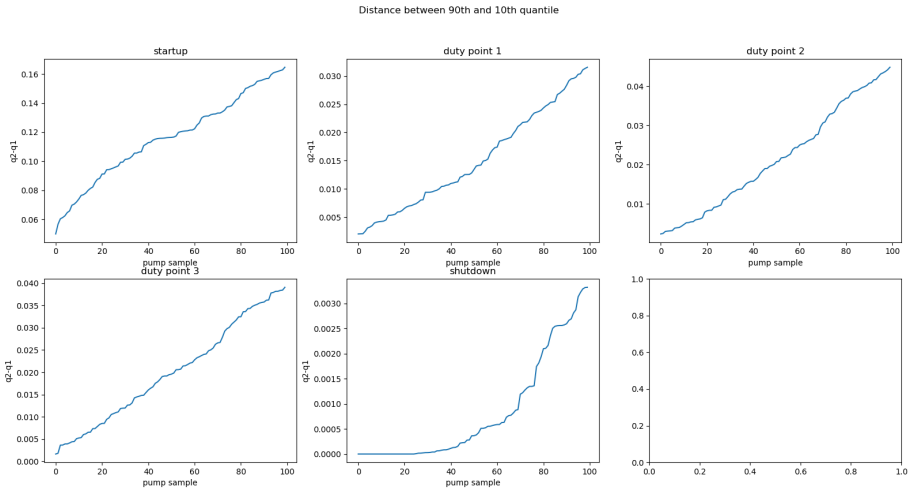


Figure 5.1: Figure showing the quantile-distance measured proposed in section 4.1.2. The distance between the 90th and 10th quantile is plotted in an ascending order as a function of pump sample.

What can be seen is that there exists no clear cutoff limit of features for any of the sections, except for the cases of the shutdown where the values initially are roughly zero and rapidly increases to approximately 0.003. This creates a problem when deciding how many features to include in the clustering process. Since arguably the most common dimensionality reduction tool, PCA, is rather unaffected by a single outlier, the quantile-density selection criteria was developed in hope of having strong explanatory power of the features that include outliers. The reasoning was that it also scales with how far away an outlier is located. The results however does not make it clear how many features to include. Due to this, four different number of features will be included in order to also see the effect that an increase of dimensions has on the clustering algorithms.

5.2 Model results

This section visualizes the clustering results and gives a brief comment about each section. After all sections have been covered a more extensive discussion will be done. In the appendix, one can find the LOF-score figures as well as the DBSCAN ε-estimation figures. The complete data set of the pump model was used in the analysis which resulted in a sample size of $N = 2100$.

5.2.1 Startup

Startup					
$N = 2100$	DBSCAN		LOF		Joint
d	ε	Outliers	γ	Outliers	Outliers
10	0.10	37	2.5	18	18
20	0.15	29	2.0	25	25
30	0.25	35	2.0	20	20
40	0.30	53	1.8	28	28

Table 5.1: Table of DBSCAN ε setting and LOF-score threshold γ in the startup phase. The number of outliers for each number of features d is also displayed in the table.

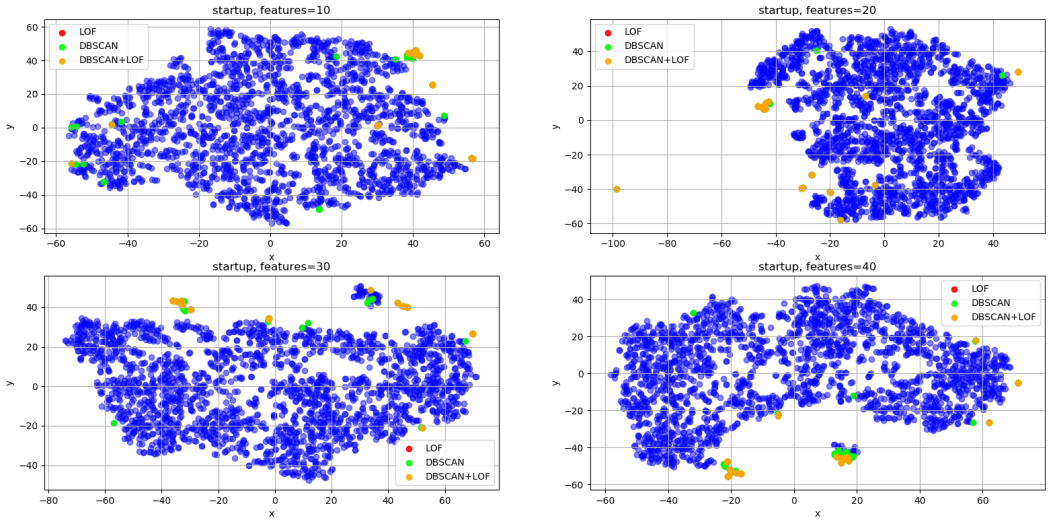


Figure 5.2: t-SNE visualization of clustering results for varying amount of features d in the startup phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the startup phase using $Perp = 25$.

It can be seen that DBSCAN has identified more outliers than LOF for all number of features and that all points identified by LOF are also jointly identified. For 10 and 20

features there are a lot of spread out points that does not look like outliers but also one smaller tightly knitted cluster where all points seem to have been tagged as outliers. For 30 and 40 features there is a different behavior, there are far less spread out points and now two smaller clusters with many points as outliers. For $d = 40$ almost all points in the two smaller clusters have been labeled as outliers. The number of outliers stay relatively constant during the increase of features. The points tagged as outliers make up about 1.4-2.5 % of the data for this section. There is one point that is of extra interest, located at $(x, y) = (-100, -40)$ in $d = 20$ it differs significantly from the rest.

5.2.2 Duty point 1

Duty point 1					
$N = 1232$	DBSCAN		LOF		Joint
d	ε	Outliers	γ	Outliers	Outliers
10	0.10	41	25	53	41
20	0.125	76	20	79	74
30	0.15	101	10	136	100
40	0.20	118	7.5	151	117

Table 5.2: Table of DBSCAN ε setting and LOF-score threshold γ in the first duty point phase. The number of outliers for each number of features d is also displayed in the table. Note that N has decreased due to filtered out inconsistencies in some samples.

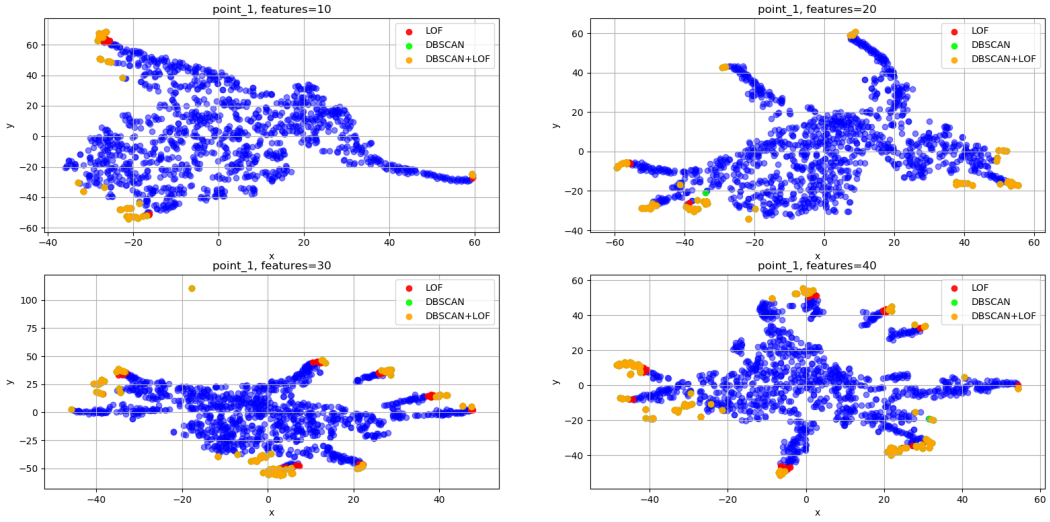


Figure 5.3: t-SNE visualization of clustering results for varying amount of features d in the first duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the first duty point phase using $Perp = 25$.

For the first duty point it is straight away seen that LOF identifies more outliers than DBSCAN and almost all points tagged by DBSCAN are also tagged by LOF. The outliers are mostly located in areas of high density and there are few outliers with dissimilar neighbors. It can be seen however that the number of outliers increase rather rapidly as the number of features increase. The potential outliers amount for roughly 4.3-12.2 % which is a much larger floor and span than seen in the previous startup section. There are a few noticeably separate clusters that have been partly tagged as outliers in $d = 30$ and $d = 40$.

5.2.3 Duty point 2

Duty point 2					
$N = 1232$	DBSCAN		LOF		Joint
d	ε	Outliers	γ	Outliers	Outliers
10	0.10	30	25	39	30
20	0.125	82	15	107	82
30	0.15	96	10	114	96
40	0.20	105	7.5	132	105

Table 5.3: Table of DBSCAN ε setting and LOF-score threshold γ in the second duty point phase. The number of outliers for each number of features d is also displayed in the table. Note that N has decreased due to filtered out inconsistencies in some samples.

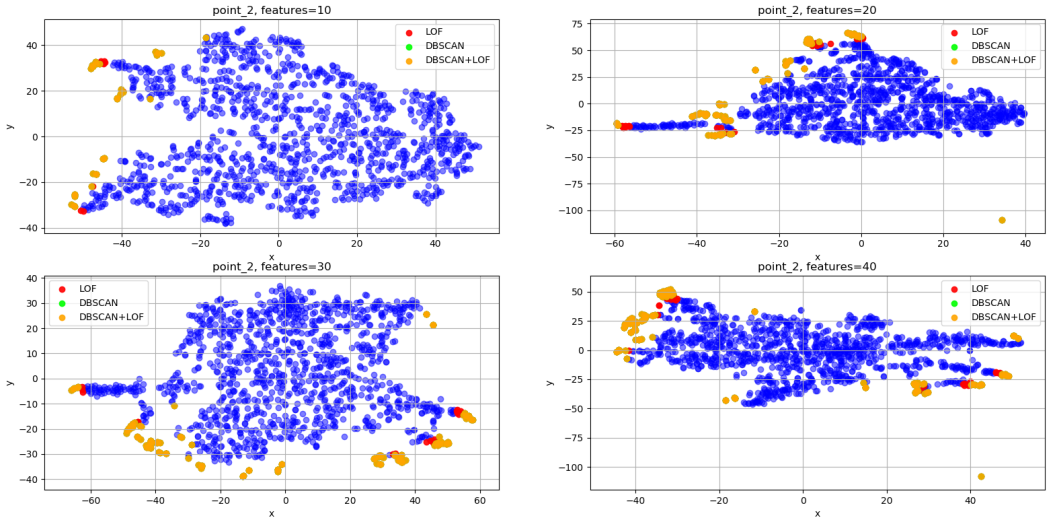


Figure 5.4: t-SNE visualization of clustering results for varying amount of features d in the second duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the second duty point phase using $Perp = 25$.

Moving on to duty point 2 it can instantly be seen by observing figure 5.4 that LOF, similarly as in the case of the first duty points, finds more outliers than DBSCAN and there is no

outlier using DBSCAN alone. The number of outliers seems to, in this case as well, increase in relation to growing number of features.

5.2.4 Duty point 3

Duty point 3					
$N = 1232$	DBSCAN		LOF		*Joint*
d	ε	Outliers	γ	Outliers	Outliers
10	0.10	42	25	57	42
20	0.125	100	15	109	100
30	0.15	117	10	121	114
40	0.20	162	5.0	228	162

Table 5.4: Table of DBSCAN ε setting and LOF-score threshold γ in the third duty point phase. The number of outliers for each number of features d is also displayed in the table. Note that N has decreased due to filtered out inconsistencies in some samples.

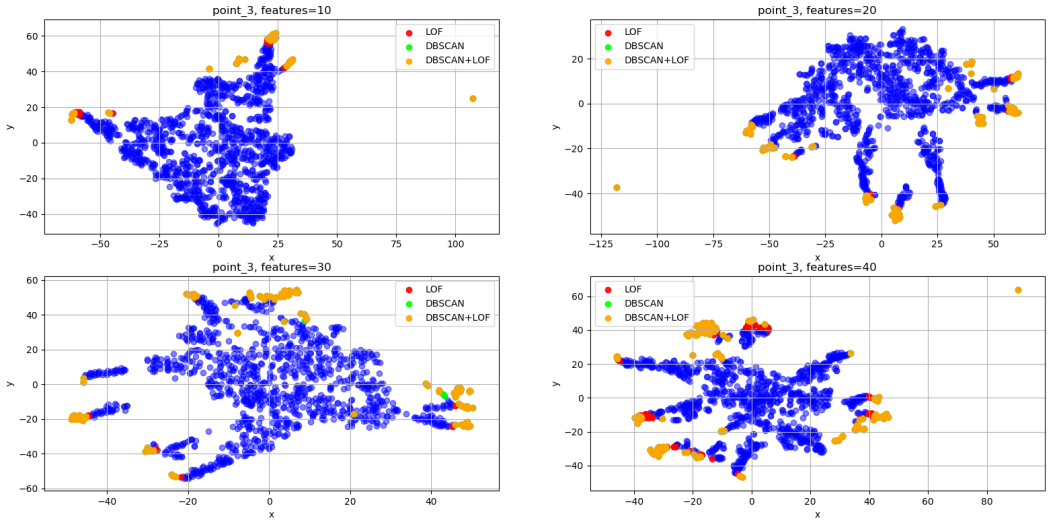


Figure 5.5: t-SNE visualization of clustering results for varying amount of features d in the third duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the third duty point phase using $Perp = 25$.

In the second duty point, there was no instance when the DBSCAN algorithm did find an anomaly where LOF did not. For the third duty point, when the number of features equals 30, this is not the case as there seems to be a data point where the first mentioned method did find by itself, just like the first duty point case. Otherwise, similarities from the other duty points still remains, such as increasing number of outliers with growing number of features d. This phase also contributed to the most outliers as $d = 40$. The outliers lays on high density areas on the border of the main cluster.

5.2.5 Shutdown

Shutdown					
$N = 2094$	DBSCAN		LOF		Joint
d	ε	Outliers	γ	Outliers	Outliers
10	0.15	38	9000	38	38
20	0.20	38	2000	38	38
30	0.25	38	500	38	38
40	0.30	43	200	43	43

Table 5.5: Table of DBSCAN ε setting and LOF-score threshold γ in the shutdown phase. The number of outliers for each number of features d is also displayed in the table. Note that N has slightly decreased due to filtered out inconsistencies in some samples.

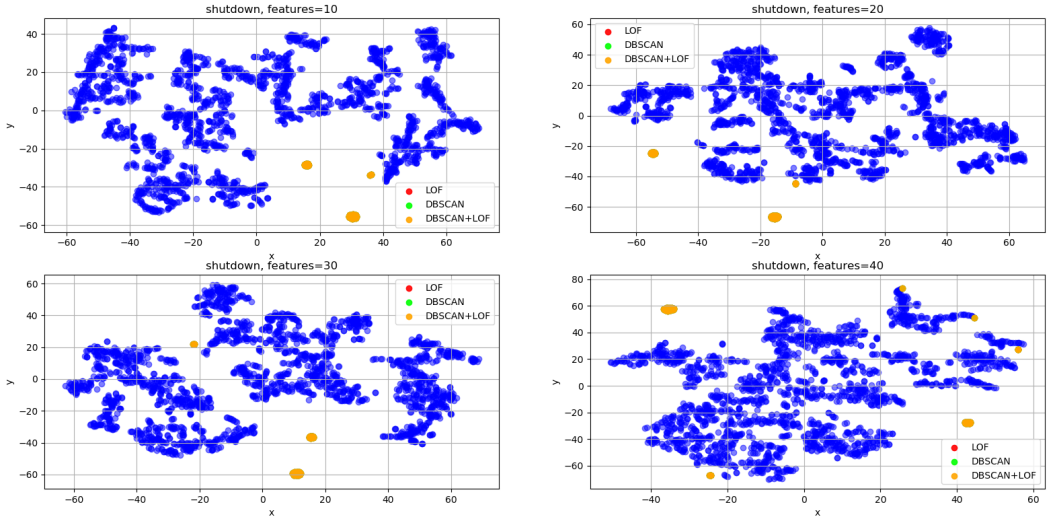


Figure 5.6: t-SNE visualization of clustering results for varying amount of features d in the shutdown phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the shutdown phase using $Perp = 25$.

The outliers found in the shutdown phase seems to be located in a more distinctive fashion than the other phases. Here the outliers, which all are found by both DBSCAN and LOF, seems to be very distinct from the main cluster. This makes it intuitively easier to validate the result. Unlike the case with the duty points, the number of outliers does not seem to depend on increasing, or decreasing, number of features.

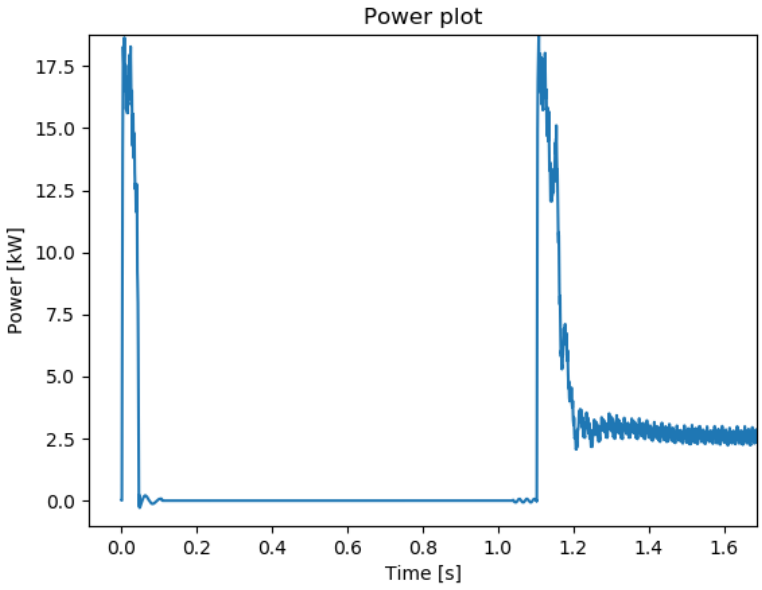


Figure 5.7: The startup of a certain pump's power measurement that has been identified as an outlier from both DBSCAN and LOF for features $d = 10$, $d = 20$, $d = 30$ and $d = 40$. As shown, the startup is not following the normal pattern as illustrated in figure 3.1 which indicates a correctly identified outlier.

5.3 Complete model discussion

Common amongst the models, with the exception of the shutdown model, is that the number of outliers found increases rapidly with the number of dimensions. This is true for both DBSCAN and LOF. The effects that increasing dimensions have on LOF and DBSCAN are rather noticeable and can be observed in the LOF-score and ε plots found in the appendix. In figures A.1-A.5, the bigger values of the LOF-score decreases meaning the algorithms are less confident. From this, the separability between outliers and normal points decreases which can be seen as the curves getting flatter, which is especially visible for the duty point phases. The ε values in figure A.6 are seen to grow larger with the number of dimensions. This indicates that the neighbor distance increases in high dimensions. As with the LOF-score, the range at which one can make the cutoff increases with the number of dimensions as can be seen as the graph getting flatter. Once again this is especially visible for the duty points. The shutdown model exerts a different behavior than the other models. The outliers are tightly knitted together in a few distinct clusters as can be seen in the t-SNE visualization, with only a very few points not following this behavior. This can also be observed in the LOF-score figure A.5 and the ε figure A.10, where there are a few large

LOF-scores and ε values followed by the baseline values.

From the t-SNE visualizations it can be observed that, however expected, the choice of features clearly has an impact on the clustering results. The duty point models, which uses only frequencies as features, exhibit a rather different behavior than the startup and shutdown models which uses a wide range of features. However, without any accuracy measure it is hard to conclude which set of features based of the scoring function is better. In a general note, it can possible be that the frequencies that have been selected by the choice of variable-scoring function are not explanatory enough in terms of outlier detection. The same goes for the selection of features for the startup and shutdown models. In terms of reliability, the outlier points that are surrounded only by points that are not tagged as outliers are those expected to be least reliable. Outlier points that form clusters together with other outlier points are expected to be more reliable due to them being points with similar properties. Outliers that are present far away from any neighboring points in the t-SNE space are also expected to hold more reliability. By comparing the t-SNE visualizations using different perplexities figure A.12-A.21 it can be concluded that the choice of perplexity does not have a significant impact on the formation of outlier clusters.

Chapter 6

Conclusion

The aim of this thesis was to research whether the data provided could be used in order to find anomalies using unsupervised clustering techniques. It can be concluded that the proposed methods do in fact identify unexpected data points within the set. However the problem lies in validating these points, which is expected in unsupervised learning methods. The validation must come in two parts: First validating whether or not the points found are in fact outliers when comparing to the rest of the data. Secondly, to conclude if those outliers validated are in fact portraying physical anomalous behavior. The first part is what is being done in this report. However the dimensionality and choice of method make the validation slightly harder. The approach to tackle the problem of validating the model was through a visual inspection using the dimension-reduction tool t-SNE, which turned out to be useful in this case.

6.1 Alternative approaches

The approach used in [23] based on time series forecasting could be applicable to this problem. The method consists of first forecasting future values of the time series using Holt-Winters method. Then the residual between the predicted value and true value is compared against a Gaussian distribution. From this distribution one can conclude with confidence intervals if a data point is to be considered an outlier based on some confidence level. The assumption that the residuals follow a Gaussian distribution would have to be investigated or else another distribution would need to be imposed.

A similar approach is to decompose the time series using STL ("Seasonal and Trend decomposition using Loss") as outlined in [24]. With the residuals detached from the series,

a threshold can be set to identify point anomalies in the residuals. The strength of these methods is that they are not hindered by the curse of dimensionality. Another angle of approach is to instead do projected clustering in lower dimensions, the technique proposed in [25] works by finding lower dimensional projections which are locally sparse and applying outlier detection in these projections.

An alternative method could be the use of a LSTM (Long Short-Term Memory) recurrent neural network. For this problem the LSTM proposed in [26] would work similar to that of the Holt-Winters method in [23] where future values of the time series are predicted using the LSTM. These predictions are then used to form residuals with the true values of the series. A multivariate Gaussian distribution is then fit to the residuals by maximum likelihood estimation. Data points located far out in the tails of this distribution can then potentially be anomalies. LSTM is one of the most well used method within speech recognition in later years by for example Google Voice Transcription [27].

Since the data now is pre-processed, it becomes feasible to use the prediction-based methods such as the Holt-Winters method in [23] or especially research the use of LSTM [26]. An advantage of using prediction of the time series data as in [23] and [26] rather than using a feature mapping is that there is no information loss in calculating features. Since the data is then a raw time series, the notion of an outlier is arguably easier to validate than if the data was first mapped into features.

6.2 Future Work

A dilemma is that anomalies might go undetected without a sufficient number of frequencies present within the analysis. This is hindered by the choice of method since problems arises when there are many features included in clustering. There are a few ways that this could be handled with that has not been explored within this analysis. First, the selection of frequencies used for clustering might instead be done using expert domain knowledge. Selecting frequencies that have a known physical correlation could provide better results at a lower dimensional cost. Another way of handling this problem is to analyze different measures for sorting frequencies and features by relevance. The proposed quantile-density previously explained might yield insufficient results compared to some other statistic that has not been explored.

The second problem is the reliability and validity of the results. As discussed earlier the validation of a method in unsupervised machine learning is very complicated since there isn't a way to calculate accuracy without true labels. There are however different validation methods that one can apply to density-based clustering. A good comparison of some indices and the introduction of a novel index of validation, *DBCV*, can be found in [13]. This area was being researched towards the end this thesis, however due to the late discovery it was not implemented.

Moreover, a future work could be to physically examine the anomalies in more detail. The anomalies that are identified in this work does not exactly tell which type of anomaly it is. By the help of expert knowledge, one could further inspect the pumps that are identified as anomalies and eventually point out damages and the types of damages the pumps might contain. If a pattern is recognized from a specific type of damage, say pump cavitation, then the outliers could be labeled with that type of damage and likewise for other outliers with other types of damages.

Bibliography

[1] V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection: A survey", *ACM Computing Surveys*, September 2009, http://cucis.ece.northwestern.edu/projects/DMS/publications/AnomalyDetection.pdf

[2] V. Hodge and J. Austin. "A survey of outlier detection methodologies". *Artif. Intel. Rev. 22, 2*, October 2004, pp. 85–126.

[3] M. Agyemang, K. Barker AND R. Alhajj, 2006. "A comprehensive survey of numeric and symbolic outlier mining techniques". *Intel. Data Anal. 10, 6*, December 2006, pp. 521–538.

[4] A. Patcha AND J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends." *Comput. Netw. 51, 12*, August 2007. pp. 3448–3470.

[5] M. L. Vizcaíno, C. Dafonte, J. F. J. Nóvoa, D. Garabato, M. A. Álvarez, "Network Data Unsupervised Clustering to Anomaly Detection" *Proceedings*, Vol.2(18), September 2018.

[6] X. Xu, H. Liu, L. Li, M. Yao, "A Comparison of Outlier Detection Techniques for High-Dimensional Data", *International Journal of Computational Intelligence Systems*, Vol. 11, 2018, pp.652–662, https://www.researchgate.net/publication/324500225_A_Comparison_of_Outlier_Detection_Techniques_for_High-Dimensional_Data

[7] M. M. Breunig, H. Kriegel, R. Ng and J. Sander, "LOF: Identifying Density-Based Local Outliers", *ACM SIGMOD 2000 Int. Conf* https://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf

[8] F. Giannoni, M. Mancini and F. Marinelli. "Anomaly detection models for IoT time series data", November 2018.

[9] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*. 2nd ed. New York, NY, USA: Springer New York Inc., 2001.

[10] B. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, 1986.

[11] `https://www.datanovia.com/en/lessons/divisive-hierarchical-clustering/`. Accessed May 2020.

[12] `https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering`. Accessed May 2020.

[13] D. Moulavi, P. Jaskowiak, R. Campello, A. Zimek, J. Sander, "Density-Based Clustering Validation", *Proceedings of the 14th SIAM International Conference on Data Mining (SDM)*, Philadelphia, PA, 2014. `https://www.dbs.ifi.lmu.de/~zimek/publications/SDM2014/DBCV.pdf`

[14] M. Ester, H. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.", *Proceedings of 2nd International Conference*, `http://www2.cs.uh.edu/~ceick/7363/Papers/dbscan.pdf`

[15] C.O.S. Sorzano, J. Vargas, A. Pascual Montano, "A survey of dimensionality reduction techniques", *Natl. Centre for Biotechnology (CSIC)*, March 2014. **arXiv:1403.2877** `https://arxiv.org/ftp/arxiv/papers/1403/1403.2877.pdf`

[16] R. Bellman, *Adaptive control processes*, NJ: Princeton University Press, Princeton, 1961.

[17] P. Pudil and J. Hovovicova, "Novel methods for subset selection with respect to problem knowledge," in *IEEE Intelligent Systems and their Applications*, vol. 13, no. 2, March-April 1998, pp. 66-74.

[18] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection". `http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf`

[19] L. van der Maaten and G. Hinton, "Visualizing Data Using t-SNE", *Journal of Machine Learning Research*, vol.9, November 2008.

[20] B. Anderson, *Pattern Recognition: An introduction*, ED-Tech Press, 2019, pp.127.

[21] L. Li, R. J. Hansman, R. Palacios, R. Welsch, "Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring", *Transportation Research Part C*, March 2016, Vol.64, pp.45-57.

[22] P. Mathur, "Outliers", *Science & Technology*, April 2015, 1(2), pp.63-66, http://www.discoveryjournals.org/sciencetech/Current_Issue/2015/A10.pdf

[23] A. Aboode, "Anomaly Detection in Time series Data Based on Holt-Winters Method", March 2018, https://kth.diva-portal.org/smash/get/diva2:1198551/FULLTEXT02.pdf

[24] R. Cleveland, W. Cleveland, J. McRae, I. Terpenning, "STL: A Seasonal-Trend Decomposition Based on Loess", *Journal of Official Statistics*, Vol. 6. No. 1, 1990, pp.3-73, https://www.wessa.net/download/stl.pdf

[25] C. Aggerwal, P. Yu, "Outlier Detection for High Dimensional Data", February 2002, https://www.researchgate.net/publication/2401320_Outlier_Detection_for_High_Dimensional_Data

[26] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, "Long Short Term Memory Networks for Anomaly Detection in Time Series", April 2015, https://www.researchgate.net/publication/304782562_Long_Short_Term_Memory_Networks_for_Anomaly_Detection_in_Time_Series

[27] F. Beaufays, "The neural network behind Google Voice transcription", August 2015, https://ai.googleblog.com/2015/08/the-neural-networks-behind-google-voice.html

[28] G. Arfken, *Mathematical Methods for Physicists*, 3rd ed. Orlando, FL: Academic Press, 1985 pp. 787-792.

[29] G. D. Bergland, "A Guided Tour of the Fast Fourier Transform", *IEEE Spec- trum*, July 1969, pp. 41-52.

Appendix

A.1 Python

For the implementation of this work, Python version 3.7 was utilized.

A.2 Fourier Transform

The continuous Fourier transform is defined as [28]

$$f(v) = \mathcal{F}_t[f(t)](v) = \int_{-\infty}^{\infty} f(t)e^{-2\pi ivt}dt. \tag{A.1}$$

Consider the generalization of a discrete function $f(t) \to f(t_k)$ by letting $f_k \equiv f(t_k)$, where $t_k \equiv k\Delta$, with $k = 0, \ldots, N-1$, where N is the number of measurements or time values. By writing this out, one yields the discrete Fourier transform $F_n = \mathcal{F}_k\left[\{f_k\}_{k=0}^{N-1}\right](n)$ as

$$F_n \equiv \sum_{k=0}^{N-1} f_k e^{-2\pi ink/N}. \tag{A.2}$$

The inverse transform $F_n \equiv \mathcal{F}_k\left[\{f_k\}_{k=0}^{N-1}\right](n)$ is then

$$f_k = \frac{1}{N}\sum_{n=0}^{N-1} F_n e^{2\pi ikn/N}. \tag{A.3}$$

To increase the transform speed, Fast-Fourier-Transform, or FFT, is often used. FFT which is a particular way of rearranging and factoring terms in the sums of the Fourier transform. Using the technique of Cooley and, it cuts the number of multiplications to $(N/2)\log_2 N$. To see further detail on the internal operations, the reader is referred to [29].

A.3 Figures

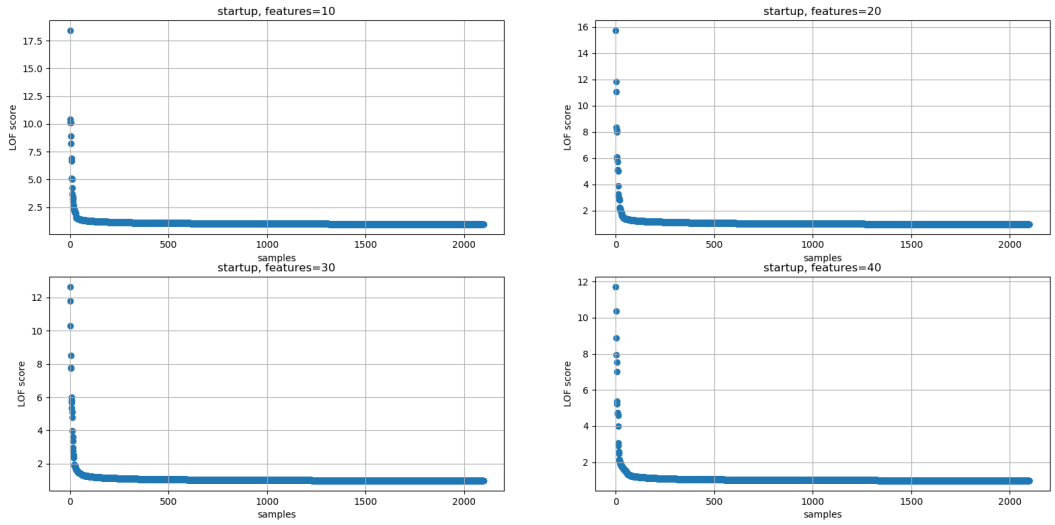


Figure A.1: LOF-score of the startup phase for dimensions ranging from 10 to 40.

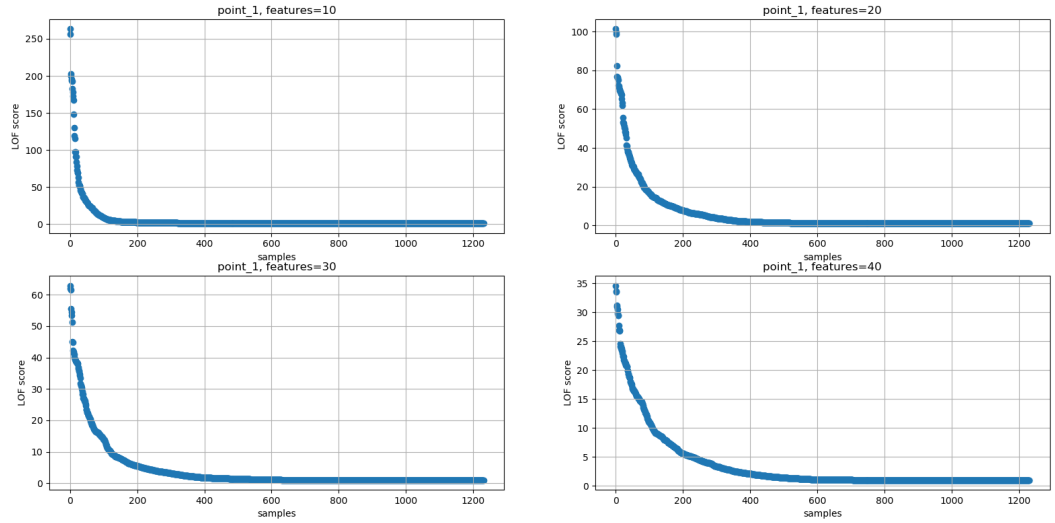


Figure A.2: LOF-score of the first duty point phase for dimensions ranging from 10 to 40.

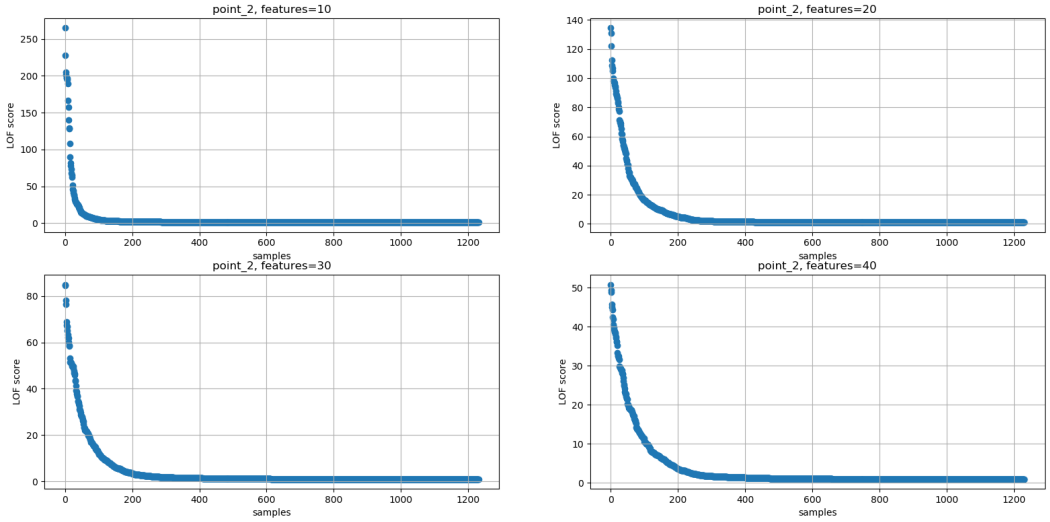


Figure A.3: LOF-score of the second duty point phase for dimensions ranging from 10 to 40.

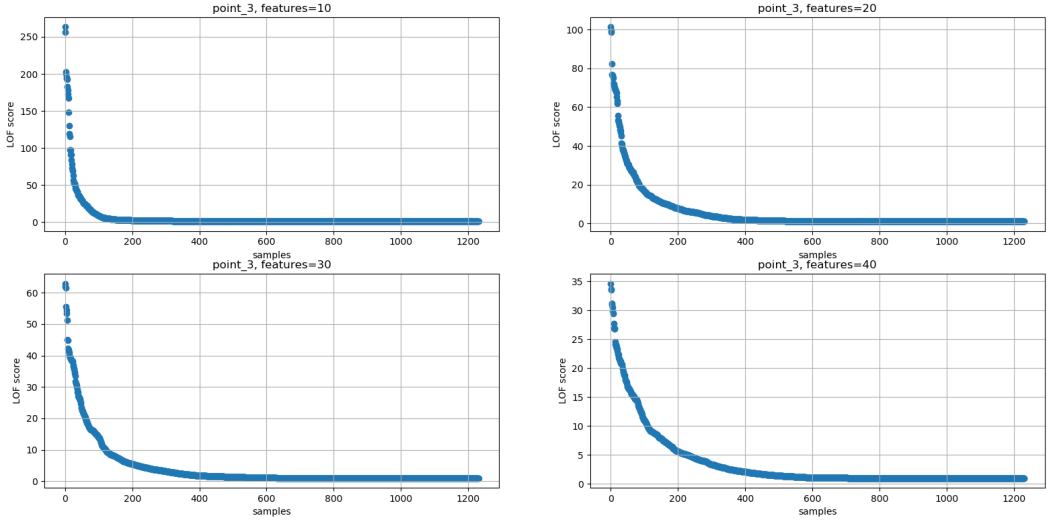


Figure A.4: LOF-score of the third duty point phase for dimensions ranging from 10 to 40.

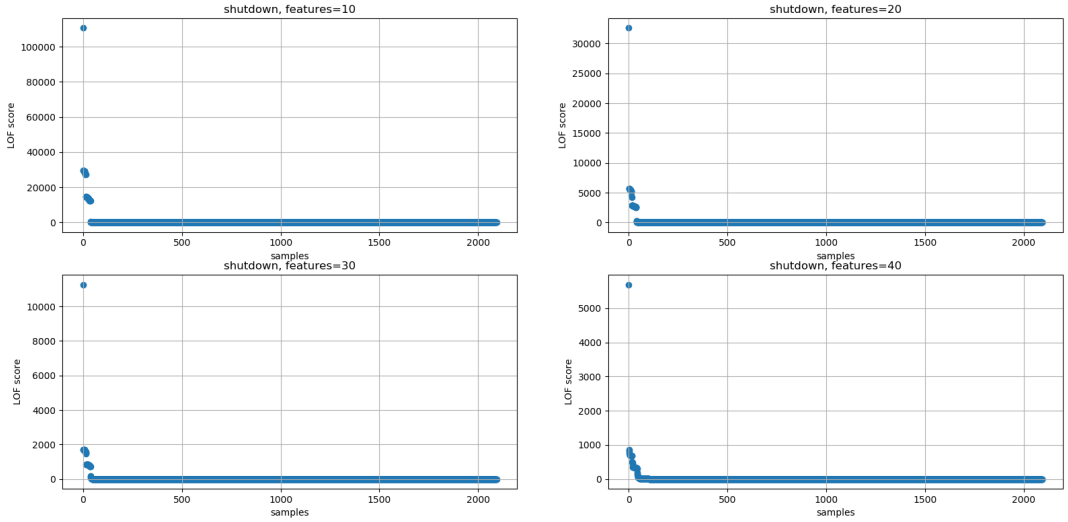


Figure A.5: LOF-score of the shutdown phase for dimensions ranging from 10 to 40.

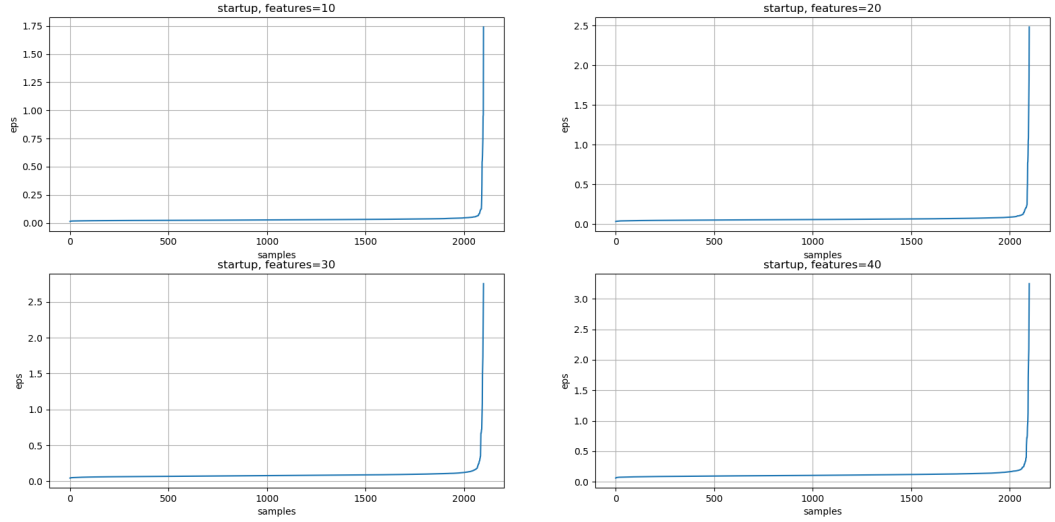


Figure A.6: ε-estimation graph of the startup phase for dimensions ranging from 10 to 40. Here eps, ε, is the distance from a point to its $MinPts$ nearest neighbor which is used in the DBSCAN algorithm.

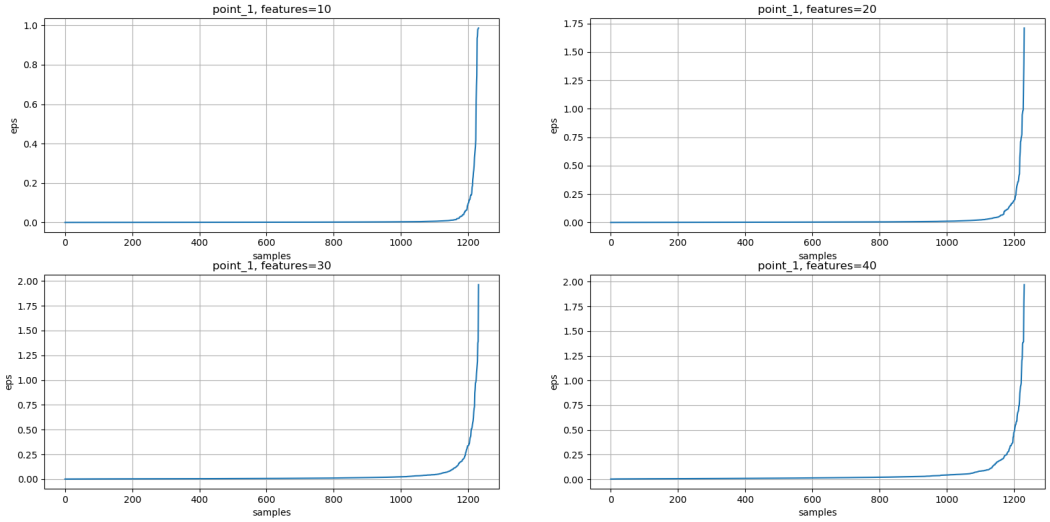


Figure A.7: ε-estimation graph of the first duty point phase for dimensions ranging from 10 to 40. Here eps, ε, is the distance from a point to its $MinPts$ nearest neighbor which is used in the DBSCAN algorithm.

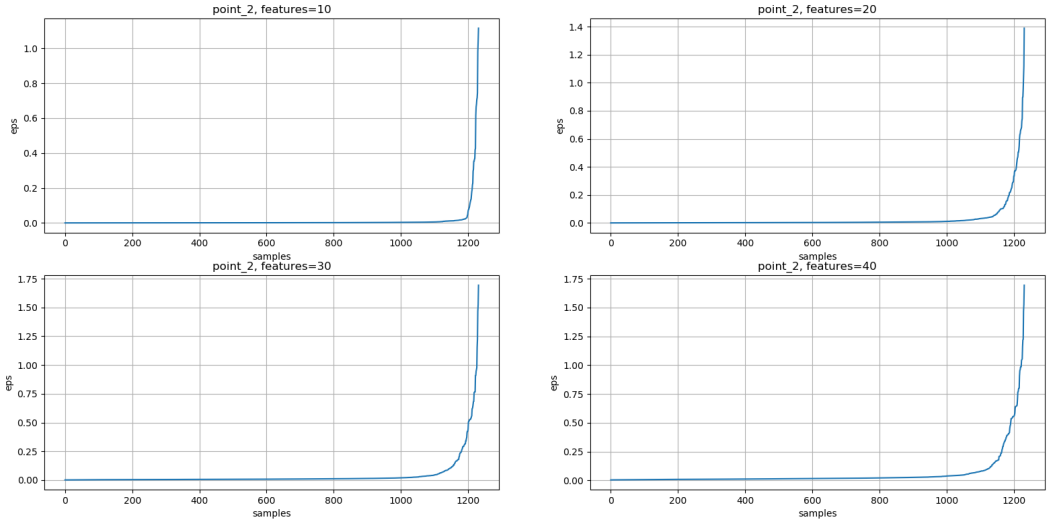


Figure A.8: ε-estimation graph of the second duty point phase for dimensions ranging from 10 to 40. Here eps, ε, is the distance from a point to its $MinPts$ nearest neighbor which is used in the DBSCAN algorithm.

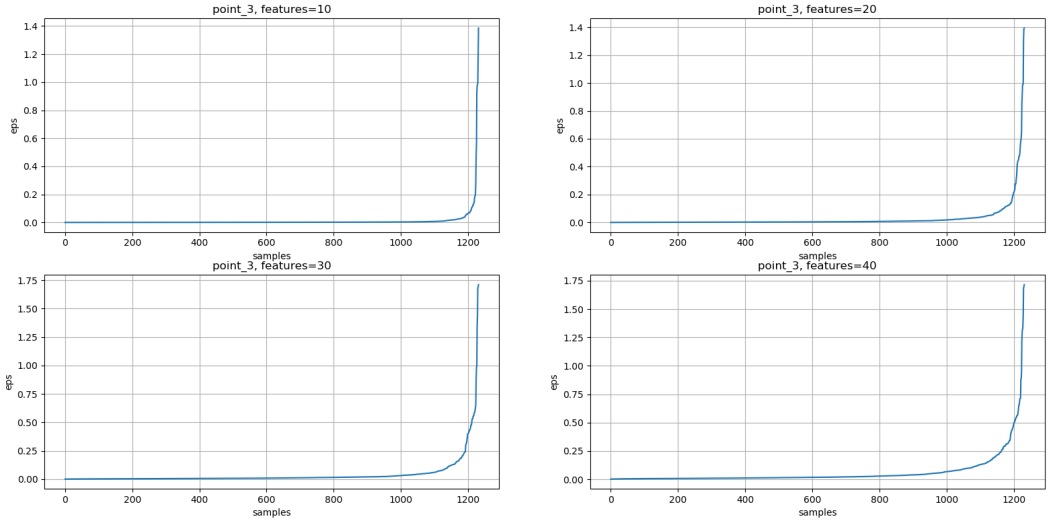


Figure A.9: ε-estimation graph of the third duty point phase for dimensions ranging from 10 to 40. Here eps, ε, is the distance from a point to its $MinPts$ nearest neighbor which is used in the DBSCAN algorithm.

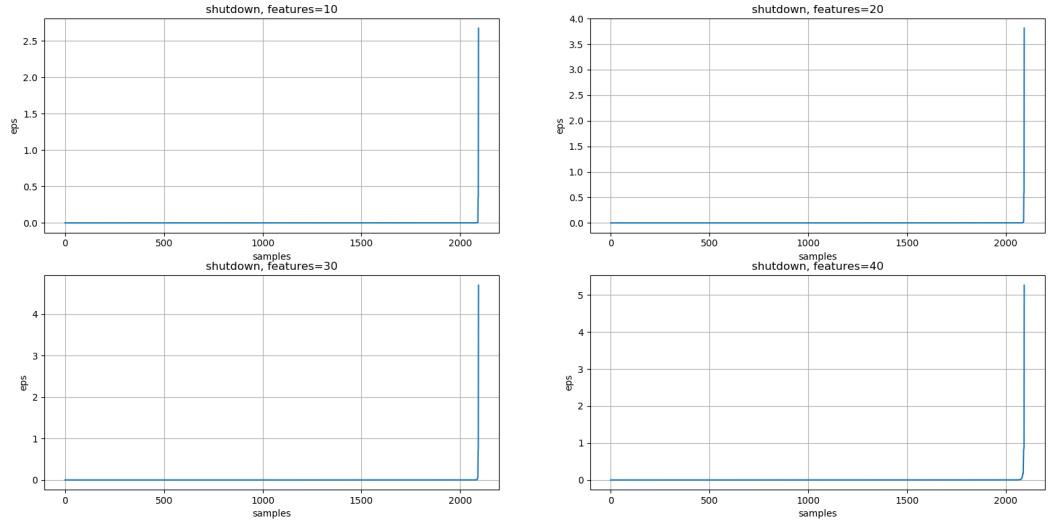


Figure A.10: ε-estimation graph of the shutdown phase for dimensions ranging from 10 to 40. Here eps, ε, is the distance from a point to its $MinPts$ nearest neighbor which is used in the DBSCAN algorithm.

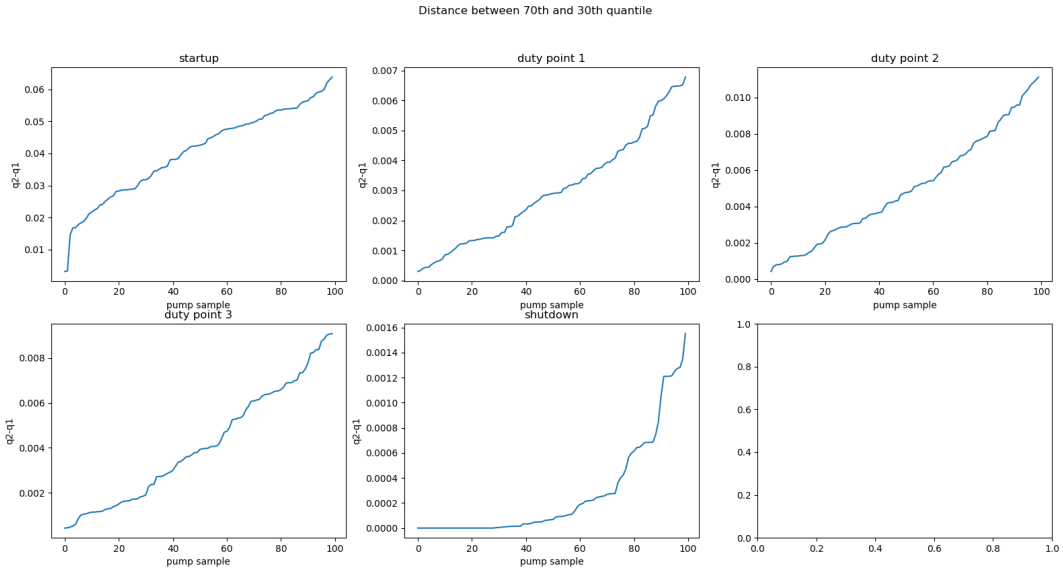


Figure A.11: The distance between the 70th and 30th quantile is plotted in an ascending order as a function of pump sample.

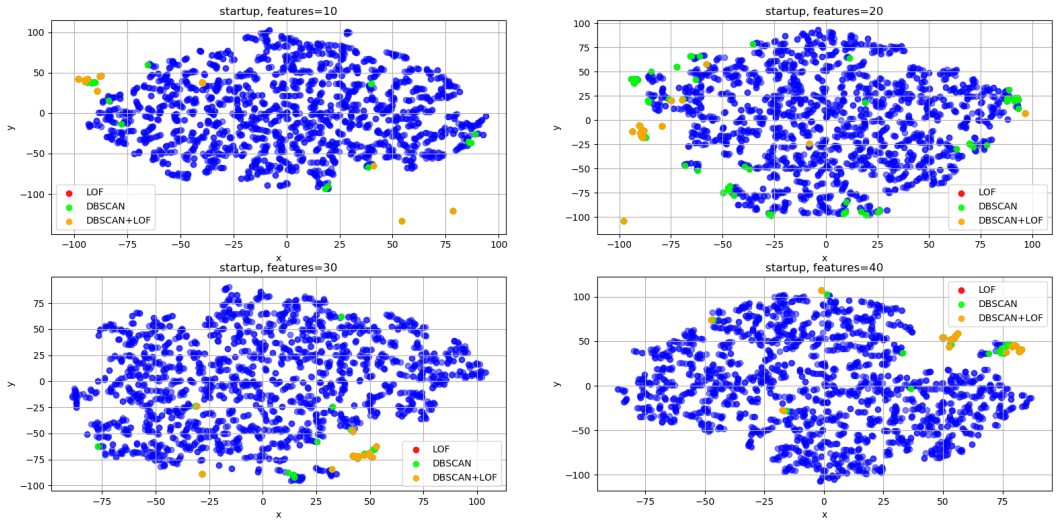


Figure A.12: t-SNE visualization of clustering results for varying amount of features d in the startup phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the startup phase using $Perp = 5$.

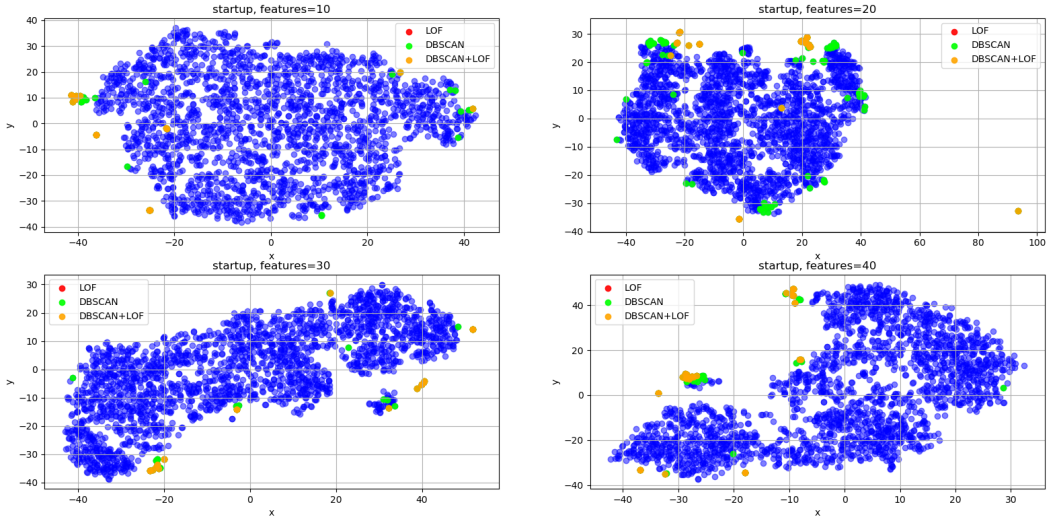


Figure A.13: t-SNE visualization of clustering results for varying amount of features d in the startup phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the shutdown phase using $Perp = 50$.

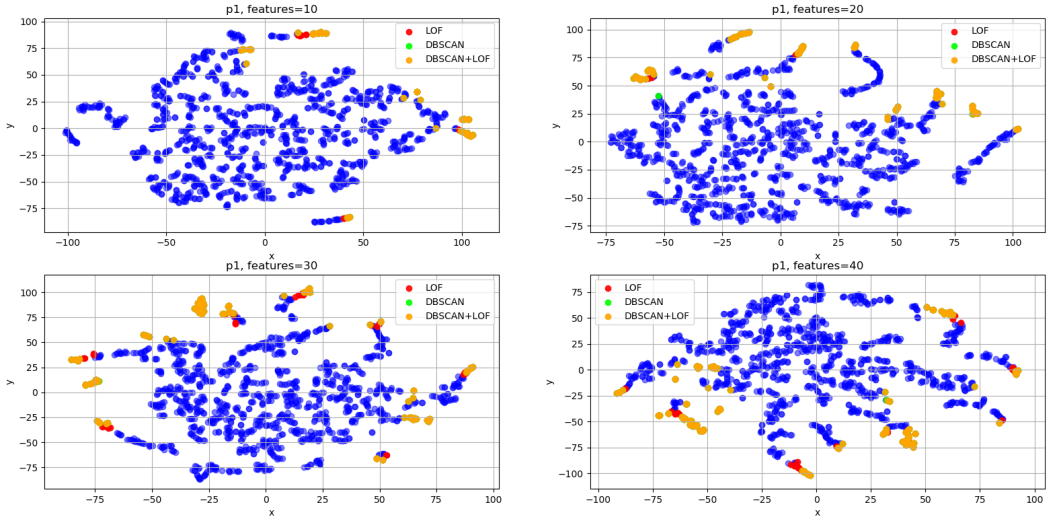


Figure A.14: t-SNE visualization of clustering results for varying amount of features d in the first duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the first duty point phase using $Perp = 5$.

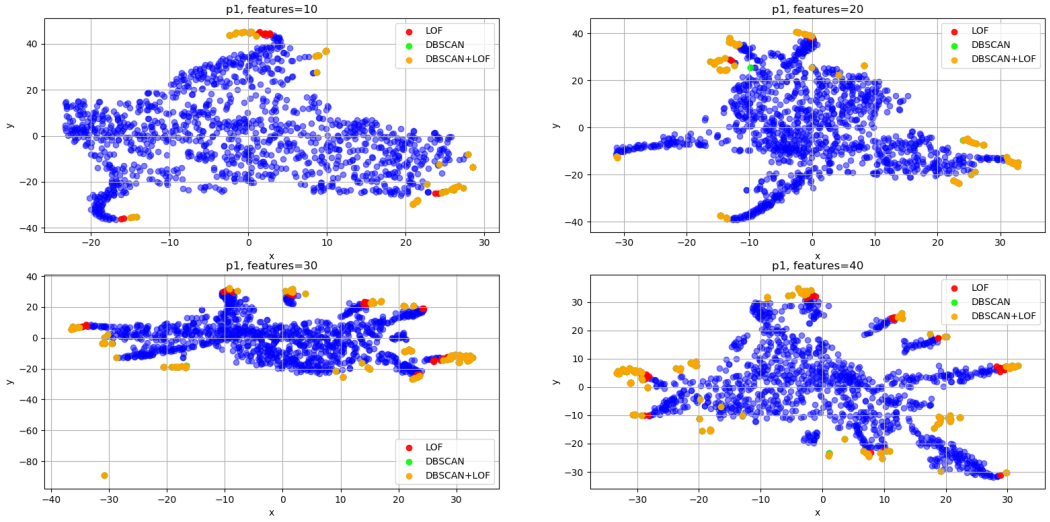


Figure A.15: t-SNE visualization of clustering results for varying amount of features d in the first duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the first duty point phase using $Perp = 50$.

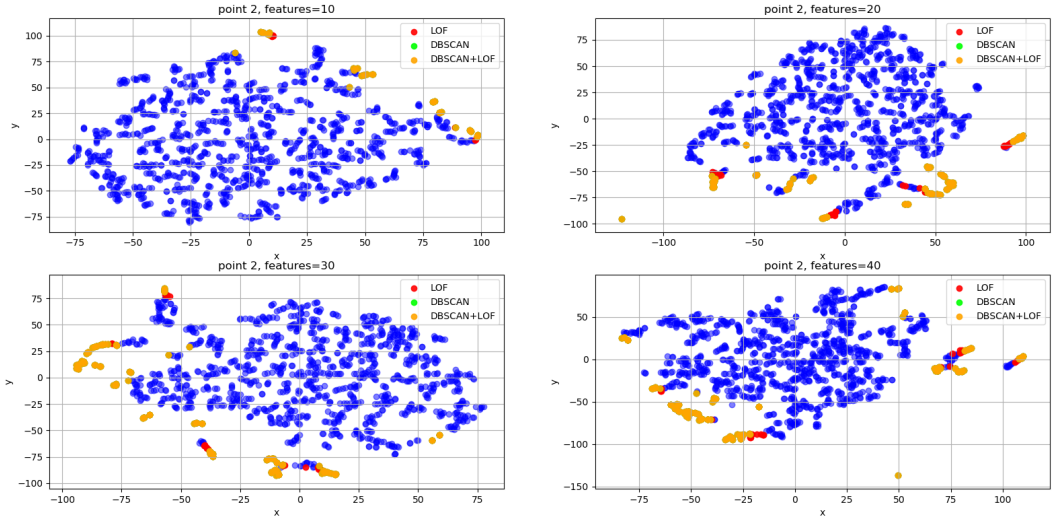


Figure A.16: t-SNE visualization of clustering results for varying amount of features d in the second duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the second duty point phase using $Perp = 5$.

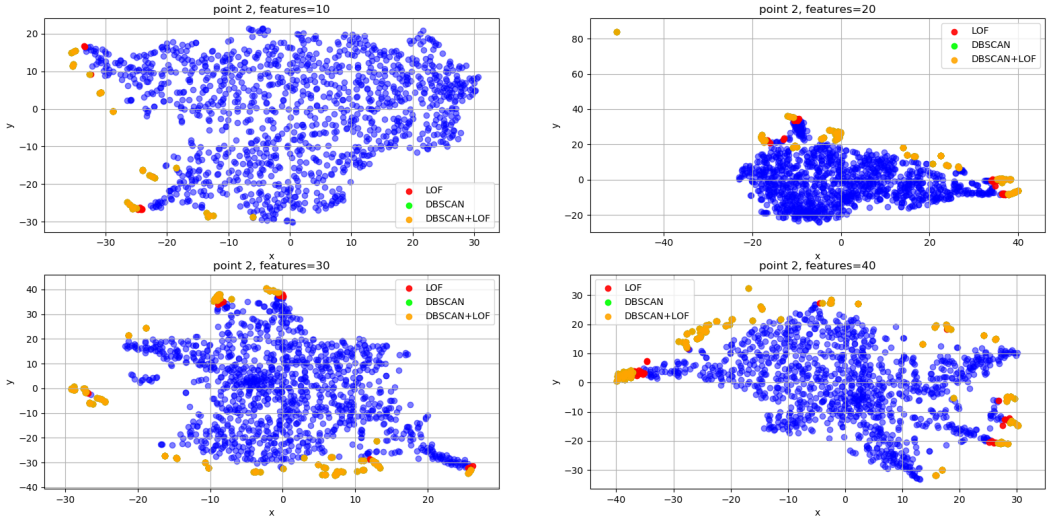


Figure A.17: t-SNE visualization of clustering results for varying amount of features d in the second duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the second duty point phase using $Perp = 50$.

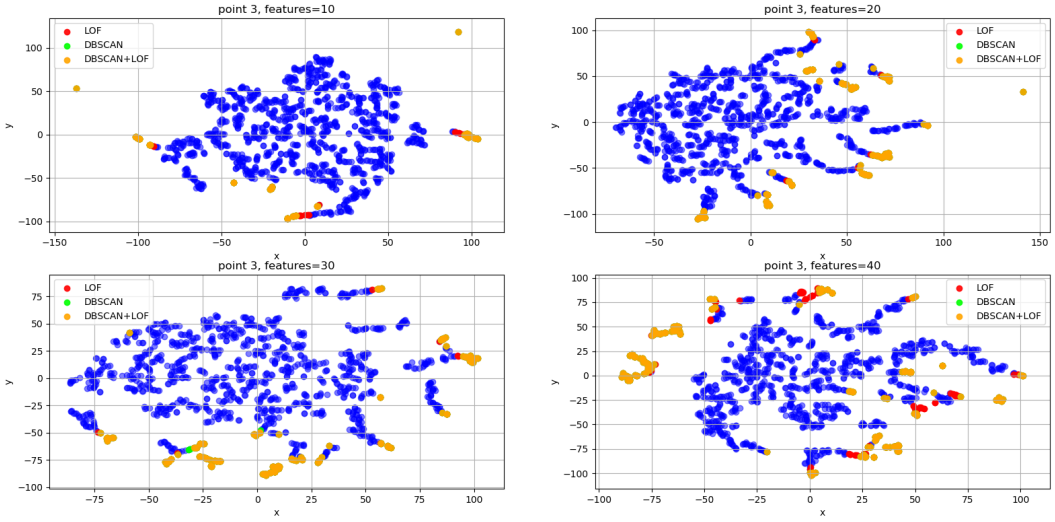


Figure A.18: t-SNE visualization of clustering results for varying amount of features d in the third duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the third duty point phase using $Perp = 5$.

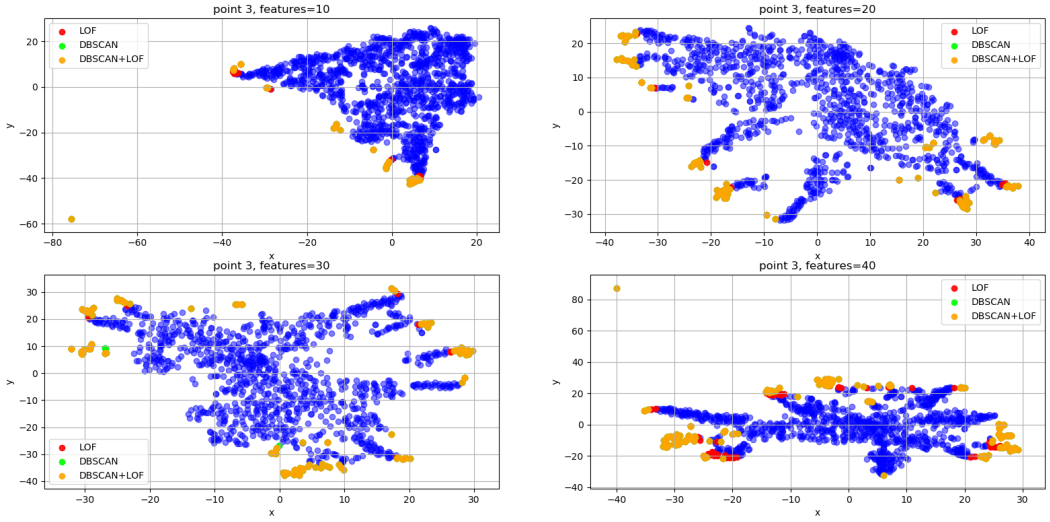


Figure A.19: t-SNE visualization of clustering results for varying amount of features d in the third duty point phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the third duty point phase using $Perp = 50$.

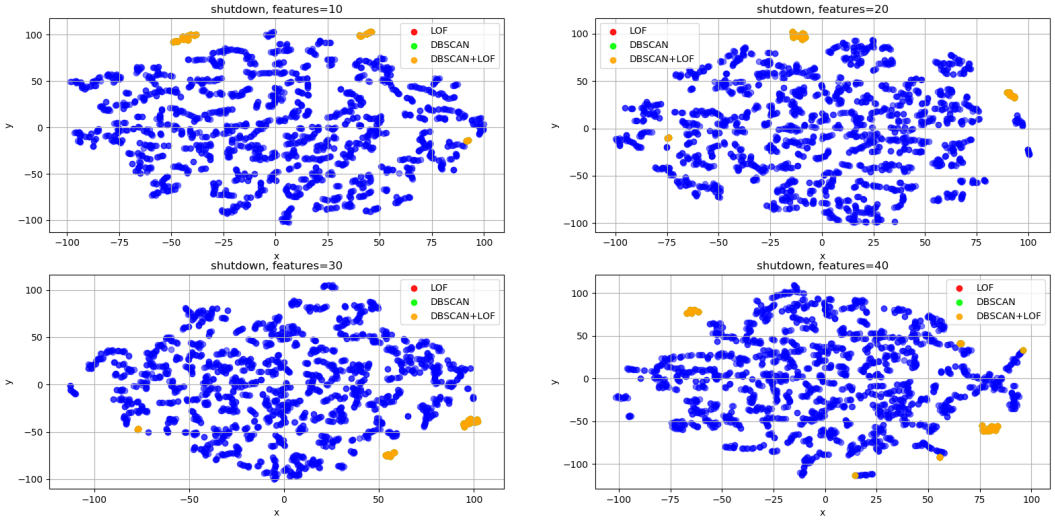


Figure A.20: t-SNE visualization of clustering results for varying amount of features d in the shutdown phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the shutdown phase using $Perp = 5$.

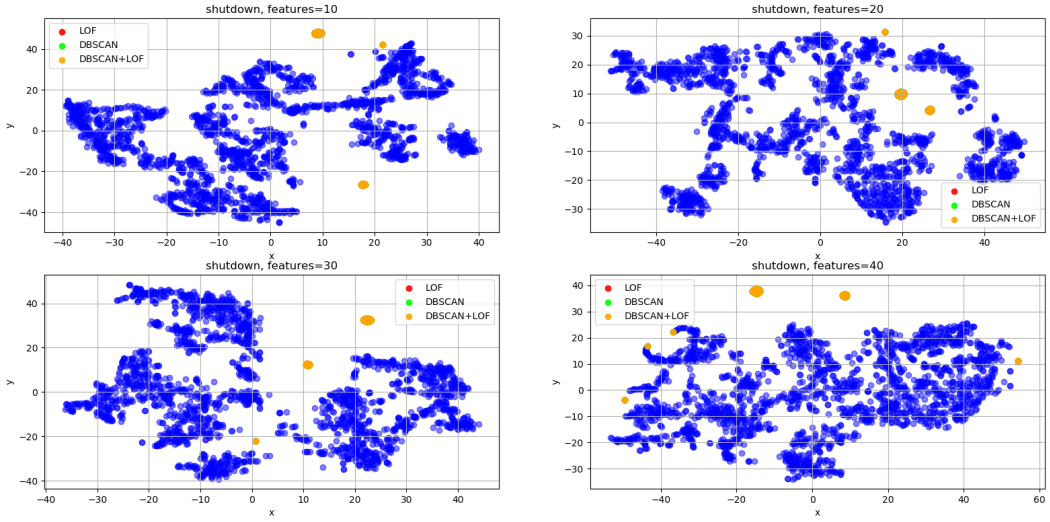


Figure A.21: t-SNE visualization of clustering results for varying amount of features d in the shutdown phase. Outliers identified by DBSCAN and LOF are marked in green and red respectively. Outliers identified by both methods are marked in orange. This figure shows the result of the shutdown phase using $Perp = 50$.

TRITA SCI-GRU 2020:063