VRIJE UNIVERSITEIT AMSTERDAM

MASTER THESIS

# Predictive Maintenance for Sewage Pumping Stations using Machine Learning

A Comparative Study


*Author*

–

*Supervisor*

Prof. Dr. R.D. van der MEI

*Second Reader*

Prof. Dr. A.E. EIBEN

*External Supervisors*

Thomas Berends

Theo Mosch


July 01, 2022


# Abstract

This thesis investigates how well Machine Learning techniques work in predicting when a failure will occur at sewage pumping station Zuidbroek. We also investigate if the vibrations or power consumption (or both) from the two pumps in the station form a better predictor. Around 15,000 observations were used together with 17 pump failures. Two machine learning models are compared, a supervised and unsupervised model. The first model is a Random Forest were we used different sampling techniques to overcome the heavily unbalanced data. The second model is k-means clustering. The Random Forest models show poor performance where a large number of false positives are predicted and 0% precision. The k-means clustering models show much better performance and are able to distinguish two different system phases. In combination with a decision rule we were able to achieve a precision of 63% and a sensitivity 94%. This model also results in an average time to failure of two hours, meaning that it gives, on average, a two hour window after the prediction until the actual failure occurs.

# Acknowledgements

This thesis is written as part of the Master's program in Business Analytics at the Vrije Universiteit Amsterdam and is written during a six month graduation internship. The internship took place at the water board: Hoogheemraadschap Hollands Noorderkwartier(HHNK) in association with the consultancy bureau Neelen & Schuurmans.

I would like to thank everyone in the Predictive Maintenance project group for all their help during the weekly meet-ups. Many thanks to Thomas Berends and Theo Mosch for their guidance, Anna Keune for writing the scripts to download the necessary data, Cees Hus for explaining the technical side of the machinery and Casper van der Wel for brainstorming with me.

I would also like to thank my supervisor from the VU, Rob van der Mei, for his insights, feedback and patience during a period of writing a thesis from home.

# Contents

# Chapter 1

# Introduction

Water boards in The Netherlands are government agencies that are responsible for water management in specific regions. Hoogheemraadschap Hollands Noorderkwartier (HHNK) is one of 21 water boards and operates in Noord-Holland, above the Noordzeekanaal. There they work on safety and social issues such as preventing floods, providing clean and healthy water and safe waterways. As such, they manage hundreds of water pumping stations that require regular maintenance in order to function properly. When it comes to maintenance, there are different strategies available. Susto et al. [20] define the following main categories:

1. *Run-to-Failure*: maintenance is performed after a failure has occurred. This is the simplest strategy but also the least cost-effective because of the high costs of interventions and the down-time of the asset after a failure.

2. *Preventive Maintenance*: maintenance is performed according to a schedule that is based on time or process iterations. This generally works well to prevent failures but also results in unnecessary maintenance being performed and leads to an increase in operating costs.

3. *Predictive Maintenance*: maintenance is performed by continuously monitoring a machine or process and predicting when a failure will occur. By using analytical tools based on historical data, this strategy allows detection of failures before they happen. This makes it possible to intervene in time which then leads to less down-time and operational costs.

Predictive maintenance shows to be a good maintenance strategy because it maximizes operational hours while reducing the required maintenance and associated costs [16]. And since technological advances have made it easier to collect large amounts of data of industrial processes, Machine Learning (ML) has become a popular choice when applied to performing Predictive Maintenance. The major distinction in Machine Learning is made between supervised and unsupervised techniques. With supervised learning, the data contains a label that is to be predicted. Regression techniques are used for numerical labels while categorical labels require classification techniques. In the case of Predictive Maintenance, supervised learning techniques are used to classify the condition of an asset based on the historical data of that asset during different conditions.

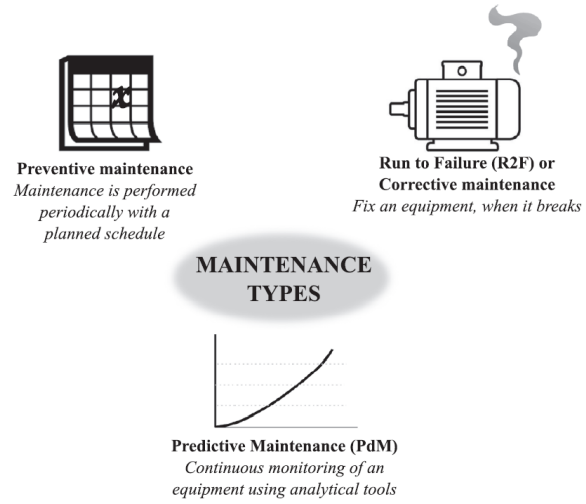



**Figure 1.1:** Maintenance Strategies [8]

The labels in the data correspond to the condition, for example healthy, deterioration, failure.

With unsupervised learning, there is no label to classify a certain condition of an asset. The main focus here is to determine groups in the available data that can be related to the conditions of an asset. A certain group could indicate whether the machine is running as normal while another group could indicate an upcoming failure. Unsupervised learning can also work as outlier detection algorithms, where data points that show large deviations from the majority can be interpreted as irregular behaviour from the asset.

HHNK currently has a vision to become a more data-driven organization. So, in collaboration with consultancy firm Neelen & Schuurmans, they have started a number of data science projects. One of these projects focuses on performing predictive maintenance for sewage pumping stations which transport sewage water for purification. Maintenance on these pumping stations currently happens according to a fixed maintenance plan or when a pump malfunctions. However, by using monitoring data, the water board might be able to predict when maintenance is required before a malfunction occurs. Using a predictive maintenance strategy can result in preventing downtime of the pumping stations, irregular work hours for maintenance engineers and less costs when compared to regular maintenance.

Domain experts at HHNK believe that failures at a pumping station can be determined in two ways, namely by using data of the vibrations of the pump or the power consumption of the pumps. In order to determine the effectiveness of Predictive Maintenance for the pumping stations of HHNK, different measurements are being taken at a station in Zuidbroek. These measurements include, among others, vibrations, temperature and power consumption from the two pumps. There is also data available that

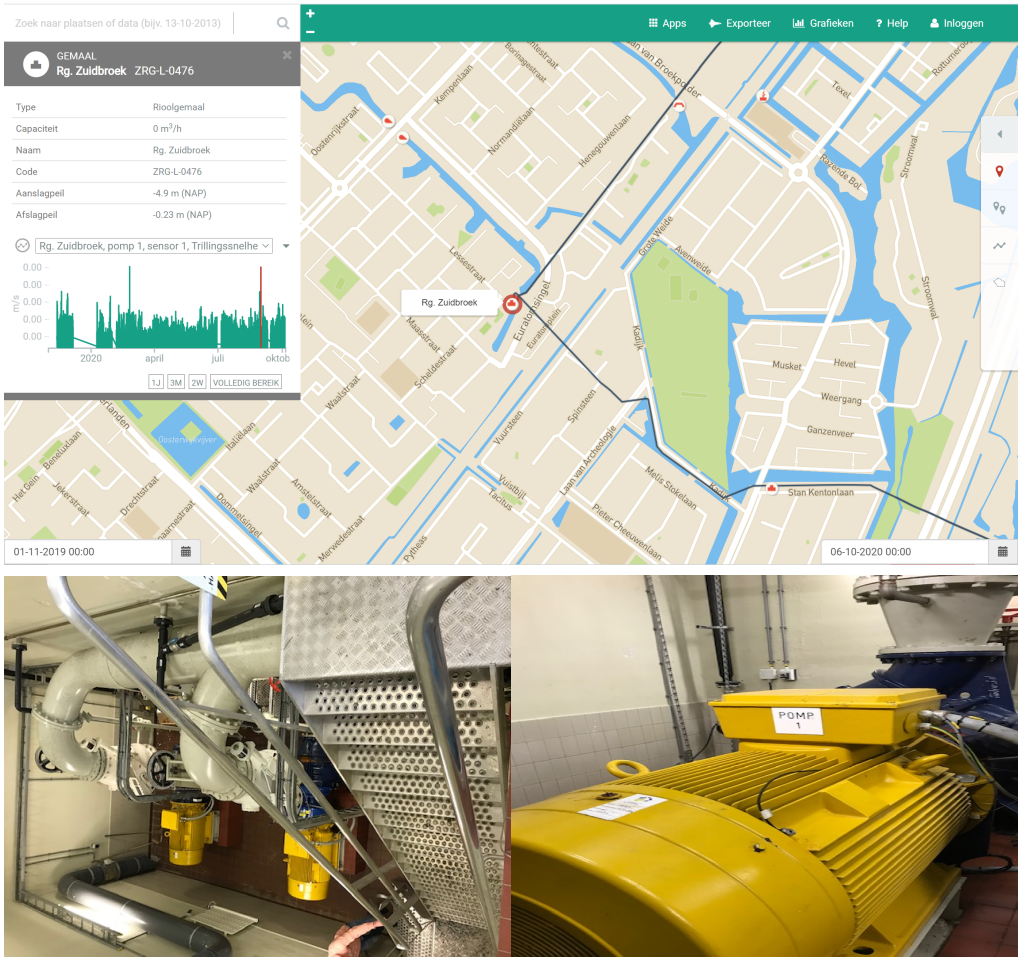



**Figure 1.2:** Sewage pump station Zuidbroek with its two centrifugal pumps. These pumps pump the water that flows into the basement to higher ground so that it can be transported to the water purification site.

contains information on when and what type of malfunctions occurred, meaning that supervised Machine Learning can be applied.

The aim of this research is to develop a model which can be used for performing predictive maintenance on station Zuidbroek by using the available data. The main research question of this thesis is therefore:

*"How well do certain Machine Learning models perform when predicting the failure of components of the sewage pump station Zuidbroek?"*

We will also determine whether or not there is a difference in predictions when using either the vibrations or the power consumption of the pumps. If the same predictions can be achieved with just the power consumption, there would be no need to place the expensive vibration sensors. The sub-question of this thesis is therefore:

*"Can the same predictions be achieved without the vibration data?"*

We will apply one supervised classification model, namely Random Forest, and one unsupervised model, namely k-means clustering. We expect the Random Forest model to perform the best since tree-based model seem to work very well in Predictive Maintenance [8]. A Random Forest is also explainable, meaning that we can determine which attributes of the data influence the classification of the model. This also makes it useful in determining the influence of the vibration data when compared to the power consumption data. The unsupervised model, k-means, was included due to the limited number of malfunctions which make the data set highly imbalanced. So this model can be useful in determining groups of data, independent of the malfunctions and the temporal aspect of the data.

The structure of this thesis is as follows: Chapter 3 discusses the available data, the preparation process, which attributes were created and finally which type of malfunctions were selected for early detection. Chapter 4 explains the concepts of Random Forest and k-means clustering and Chapter 5 shows the results of both models. Finally, Chapter 6 gives a final conclusion and an answer to the research questions.

# Chapter 2

# Literature Review

In regards to technology, it is becoming increasingly easier to collect and analyze large amounts of data of industrial processes. As such, Machine Learning (ML) methods have shown to be a popular choice when it comes to utilizing these data for maintenance approaches. Carvalho et al. [8] concluded that, among papers published between 2009 and 2018, Random Forest is the most frequented ML algorithm when performing predictive maintenance. This is followed by Artificial Neural Networks, Support Vector Machines and k-means clustering. The first three techniques are examples of supervised learning models and, in the case of predictive maintenance, are used to classify the condition of an asset based on the historical data of that asset during different conditions. Methods such as k-means are unsupervised techniques and can be used to determine certain groups in the data that relate to operating conditions.

While there does not seem to be a preference for a certain type of asset on which to apply these techniques, Carvalho et al. [8] does mention a preference for vibration data to train these ML models on. For example, Amihai et al. [3] applied a Random Forest model on vibration data collected from industrial pumps in order to predict certain metrics about the health of the pumps up to 7 days ahead. Another example is Biswal and Sabareesh [5], who used a neural network to classify the operational conditions of wind turbines. They also trained this model using vibration data that was collected when the turbine was in a healthy vs non-healthy condition.

In regards to unsupervised learning techniques, Uhlmann et al. [21] was successful in determining operational groups of a Selective Melting Machine by using data from multiple sensors.

Bruyn [7] also applied ML methods in order to perform Predictive Maintenance for several pumping stations in the municipality of Utrecht. They used a number of models, namely Long-Short Term Memory, X-Gradient Boasting, Random Forest and Support Vector Machine. Based on these models they attempted to predict major defects three days in advance and decide which model is best suited for this task. As for the data that was used, they ended up using only the flow rate and water level from their database, which were aggregated to a few different values. They then engineered a number of features such as day of the week, 5 minute difference in flow rate and the percentage

of time that the station was in defect during the past 3 days and added open source weather data. The flow rate and water level are also attributes that we will be using. It's also interesting to see that Bruyn [7] used different aggregation levels for different methods. For instance, for the non-temporal methods the data was aggregated per day whereas for the temporal methods the data was aggregated per minute. Their final conclusions were that the X-Gradient Boasting performed the best of the four algorithms, where 45% of the predicted failures were actually failures.

More recent papers published after 2020 often make the distinction in predictive maintenance between failure prediction and remaining useful life (RUL) [10] [22] [18]. Failure prediction is the generic use of predictive maintenance, which we will be utilizing in this thesis, where the main goal is to predict when a failure will occur [10]. Whereas RUL is related to prognostics which gives the remaining operational time of machinery before it requires replacement or repair. But when it comes to applying machine learning models for predictive maintenance, these papers also mention the more traditional ML methods that were also mentioned by Carvalho et al. [8]. These vary from different kinds of neural networks such as traditional, convolutional and recurrent networks to hidden Markov models, auto-encoders and transfer learning.

Davari et al. [10] summarizes works on general data-driven solutions for predictive maintenance. Here, most techniques are different types of neural networks but clustering is also mentioned as a method to identify types of faults. These methods are applied to different data sources from different types of sensors and machinery. Their summary also shows a preference for vibration data.

As for evaluating the performance of these failure prediction methods, the proposed metrics can be calculated form the confusion matrix. Measures such as recall, specificity and precision give an understanding of the number of accurately predicted failures and/or the number of wrongfully predicted failures. The AUC-ROC is also used to evaluate the performance of the prediction and for error analysis.

# Chapter 3

# Data

This chapter discusses the available data. Paragraph 3.1 elaborates on the different measurements taken at sewage pump station Zuidbroek and paragraph 3.2 explains what type of malfunctions occur and which of these are suitable for early detection. Paragraph 3.3 describes how the data was processed and which new features were created. Finally, Paragraph 3.4 discusses which attributes were chosen to create a number of data sets that are suitable to use with the chosen models mentioned in Chapter 4.

## 3.1 Zuidbroek Measurements

Sewage pump station Zuidbroek consists of two wastewater pumps that pump water from the basement to higher ground so that the water can travel to the water purification site. There are three vibration sensors on each pump and one temperature sensor on pump 1. The vibration sensors are located on the Non-Drive End (sensors 1 + 4), Drive End (sensors 2 + 5) and Coupling Side of the pumps (sensors 3 + 6). For each of these vibration sensors we have the following data: *the root mean square of the vibration speed (v-RMS), the root mean square of the acceleration (a-RMS)* and *the maximum acceleration (a-Peak)*. The v-RMS gives an indication of mechanical fatigue, the a-RMS indicates mechanical friction and the a-Peak indicates mechanical impact. Aside from the vibrations and temperature, there are also measurements such as the power consumption of the pumps and the amount of water that flows into the station. In total, there are 14 attributes spanning from January 2019 to October 2020. An overview of these attributes, along with the frequency of the measurements, is shown in Table 3.1. Each attribute also includes the timestamp of when the measurement was taken.

There is also data available of when certain malfunctions have occurred at Zuidbroek. This will be the target attribute that will be predicted with the supervised learning model, Random Forest. Chapter 3.2 explains which of these malfunctions were suitable for early detection.

[1] In Dutch: Debiet
[2] In Dutch: Waterhoogte
[3] In Dutch: Vullingsgraad

| Attribute | Description | Unit | Frequency |
|---|---|---|---|
| Flow rate [1] | The $m^3$ of water that has been pumped | $m^3/h$ | 2 minutes |
| Water level [2] | The water level in the basement. | M (NAP) | 2 minutes |
| Fill rate [3] | | % | 2 minutes |
| Pressure [4] | | Bar | 2 minutes |
| Power consumption pump 1 | The amount of electricity used by pump 1. | A | 2 minutes |
| Power consumption pump 2 | The amount of electricity used by pump 2. | A | 2 minutes |
| Pump temperature | The temperature of pump 1. | °C | $\sim$ 1 minute |
| v-RMS sensor | Root Mean Square of the vibration speed. | m/s | $\sim$ 1 minute |
| a-RMS sensor | Root Mean Square of the acceleration. | $m/s^2$ | $\sim$ 1 minute |
| a-Peak sensor | The maximum acceleration. | $m/s^2$ | $\sim$ 1 minute |
| Malfunction | Registered malfunctions | 1 or 0 | - |

**Table 3.1:** Available Data attributes measured at different frequencies.

## 3.2 Malfunctions

In regards to malfunctions that can be predicted, domain experts are especially interested in blockages [5] where it is expected that the flow rate and fill rate remain stable but the vibrations are high.

In the next chapter we will distinguish between malfunctions and failures as follows: malfunctions are every logged event from Zuidbroek but don't necessarily require intervention, while failures do require intervention. We will use failures for early detection. The available data contains information about the malfunctions that have occurred at Zuidbroek, the most important characteristics being the type of malfunction and the duration. Not all malfunctions are suitable for early failure detection. Based on domain expert suggestions, we selected the following malfunctions to be used as failures for early detection:

- Dry-run: When, during the set time of two minutes, less than the set dry-run (300 $m^3/h$) flow rate is being pumped.

[4] In Dutch: Persleidingdruk
[5] In Dutch: Pruikvorming

- Failure F.O.: can have multiple causes but is usually caused by an overload where too much power needs to be delivered to the motor in order to pump the necessary amount of water.
- Failure General: caused by a malfunction with the main power.

As mentioned before, the failure data is very skewed because failures do not occur frequently since station Zuidbroek is relatively stable. Starting off with eighty failures which can be used with the available data, five of these are dry-runs, five are failure F.O. and seventy are failure general. A large number of these failures are automatically generated and logged due to other, previously occurred malfunctions. For example, the logged dry-run failures are a result of a broken flow rate meter. And because the flow rate was not being measured longer than the set time of two minutes, an automatic dry-run failure was generated.

| | Before Selection | After Selection |
|---|---|---|
| Total over period January 2019 - October 2020 | 80:<br>• dry-runs: 5<br>• failure F.O.: 5<br>• failure general: 70 | 42:<br>• failure general: 42 |
| Total excluding gaps over period January 2019 - October 2020 | 34:<br>• dry-runs: 2<br>• failure F.O.: 1<br>• failure general: 31 | 17:<br>• failure general: 17 |

**Table 3.2:** The number of malfunctions after selecting those suitable for early prediction. The models that use the vibration data will have less malfunctions due to the gaps in this data.

Another example of this is shown in Figure 3.1. This graphs shows the mean 2-minute values for the power consumption, two sensors, the outgoing pressure, flow rate, fill rate and water level. The dotted lines represent when a malfunction was reported ((1)) and when it was signed off ((0)). We can see here that a mechanic was present at the sewage pump station around 07:30 am (the blue dotted line) and that they very likely turned off pump 1 in order to work. This caused the flat line in the first two graphs for the power consumption and vibrations from pump 1. During this off-period for pump 1, the third and fourth graphs show that pump 2 is now pumping more in order to pick up the slack from pump 1. See the shorter cycles in the power consumption and vibrations for pump 2. After a few hours an automatic malfunction "Failure General Pump 1" is logged (the red dotted line) due to pump 1 being off for longer than the set time. About an hour later, the mechanic is done working and pump 1 is turned back on (the green dotted line).

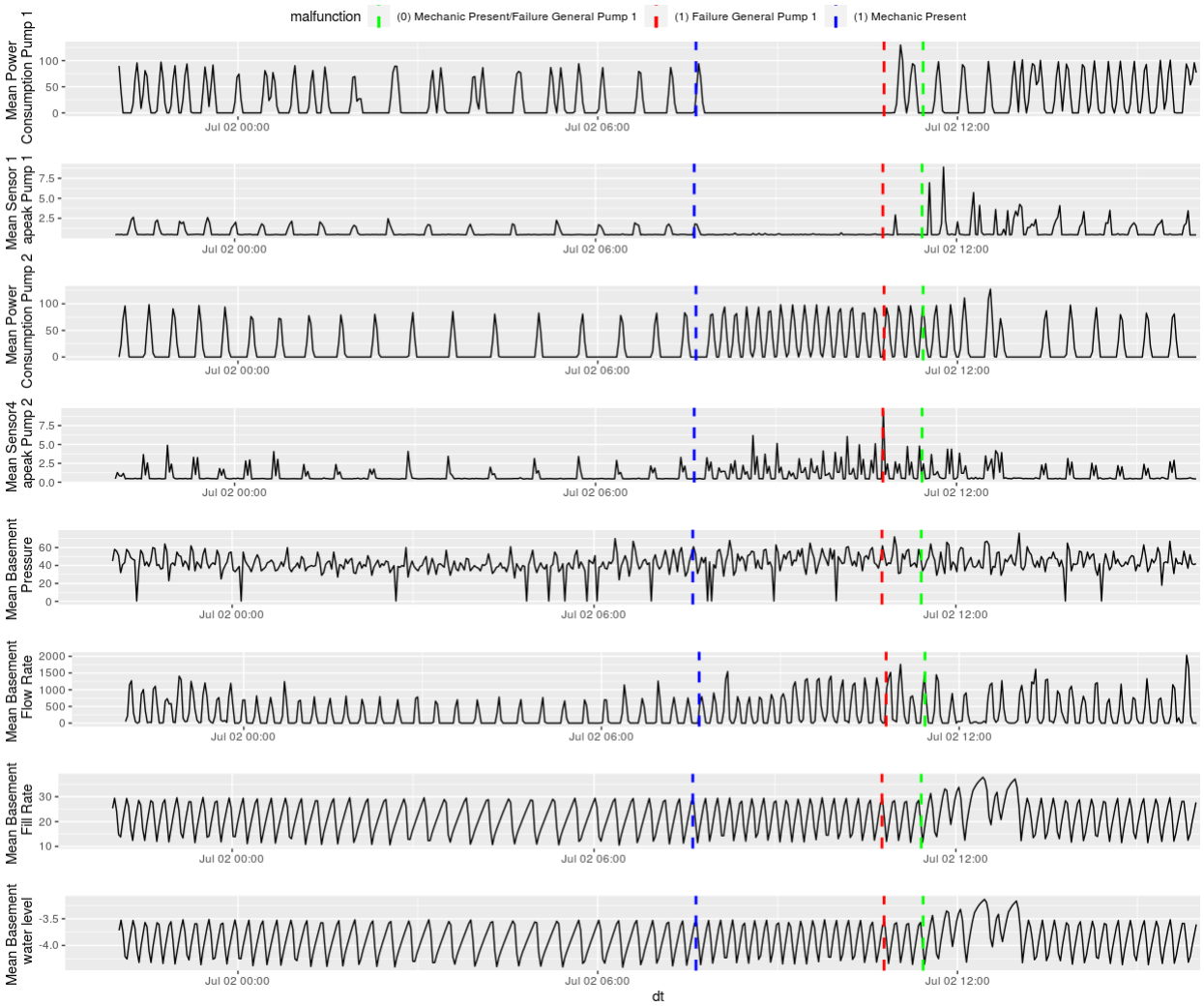



**Figure 3.1:** Malfunction effect: Failure General Pump 1 logged in July 2019 is caused by a mechanic switching pump 1 off. Because pump 1 was off longer than the set time-out, an automatic failure was generated.

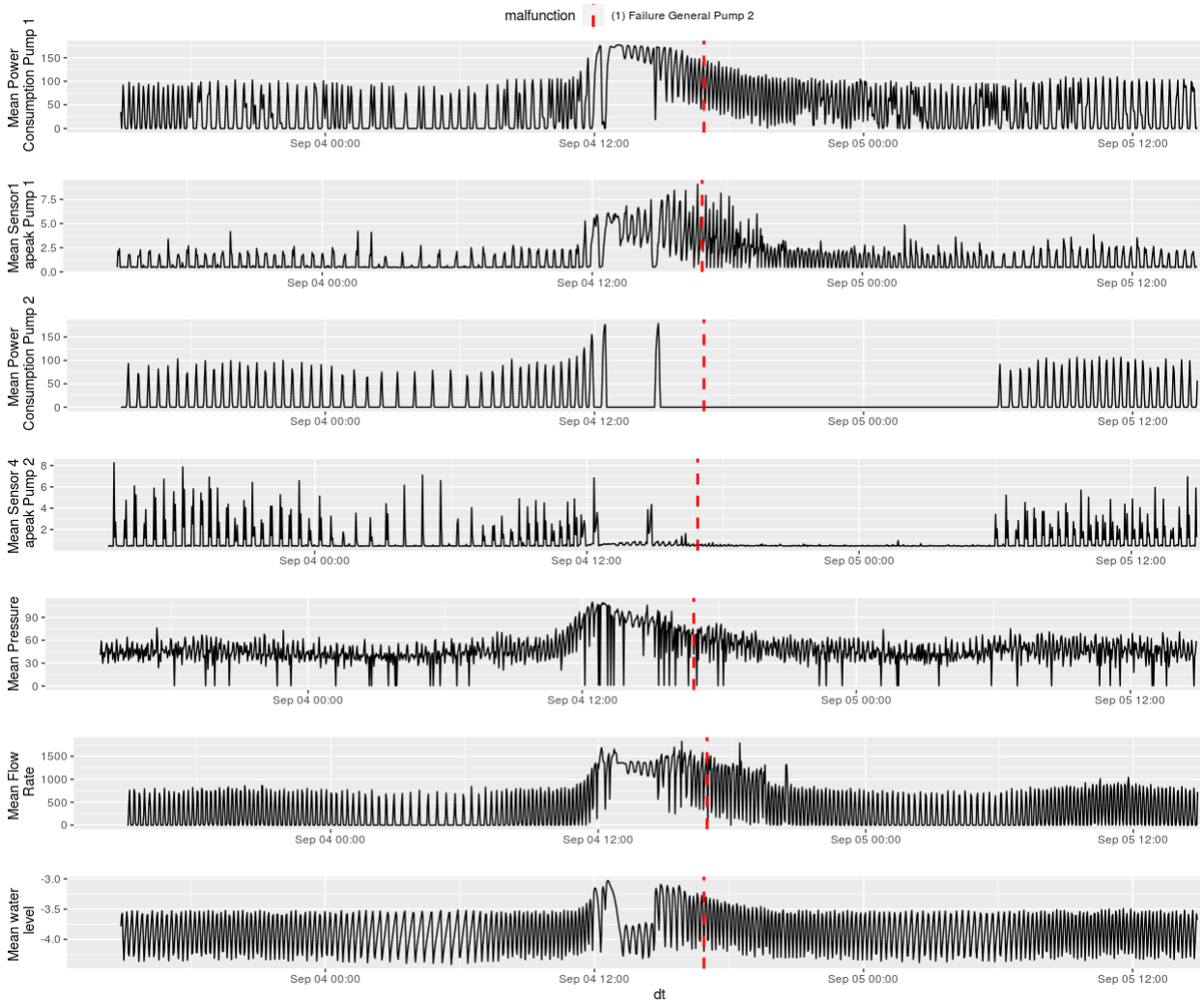



**Figure 3.2:** Failure General Pump 2 logged in September 2019. This shows an uptick in the flow rate, resulting in the pumps using more power and pump 2 suddenly shutting off around noon.

These examples show that a large number of the failures are not random, which makes predicting them impossible. So we will limit ourselves to the failures that occurred "randomly" on a single day, without previous malfunctions. Unfortunately, this only leaves 42 failures all of type Failure General Pump 2, and only 17 of these fall into the periods of which there is vibration data available. Figure 3.2 shows one of these failures, where there is an increase around noon in flow rate, pressure and water level. Both pumps also start using more power in order to pump away the excess amount of water when suddenly pump 2 shuts off during the peak. This seems to show a usable pattern that can be detected by machine learning algorithms.

## 3.3 Pre-Processing

### Data Aggregation

Table 3.1 shows that the vibration data is logged at a different frequency than the other attributes, respectively one minute vs two minutes. The first step is to decide a step-size or granularity ($\Delta t$) for this data set. Using a small granularity could cause noise to remain in the data which is frequently the case with, for example, sensor data from smart watches and smart phones [12]. Noise in the context of predictive maintenance could have a different meaning where it may already indicate something wrong with the machine. So, we will choose the smallest granularity possible. And because we want to be able to compare the predictive power of the vibrations with that of the power consumption of the pumps, we will use $\Delta t = 2$ minutes. For every step-size we will calculate the mean. In order to mitigate the run-time of the models, we will limit the step aggregation to the mean instead of also adding statistics such as the minimum and maximum.

Due to infrastructure issues, the vibration data collection encountered issues and as a result, there are various gaps in the vibration and pump temperature data, some spanning several months. This means that when we include the vibration and temperature data in the data set, we have a total of around 244,000 observations. Excluding the vibrations and temperature leaves us with 443,000 observations.

### Missing Values

After the data aggregation, we are left with some missing values for certain attributes. An overview is shown in Appendix A.1. Fortunately, there are not many missing values. The attribute "Pressure" contains the most, with around 1% missing. In order to retain as much valuable data as possible, we will impute the missing values using interpolation. This is easily done since all of our data is numerical and it is also the preferred method when handling temporal sequences since it results in much more natural values [12]. For this method we will take the average of the previous and next value of the same attribute. When given with the temporal sequence $x_i^1, x_2^i, ..., x_N^i$ for attribute $i$, with missing value $x_j^i$, it will be predicted as

$$x_j^i = \frac{x_{j-1}^i + x_{j+1}^i}{2} \tag{3.1}$$

### Feature Engineering

Now that the data has been aggregated and missing values have been interpolated, we will extract some useful attributes from our data set in order to improve the predictive performance of our model. Hoogendoorn and Funk [12] mention two main ways of doing this. One is creating attributes in the time domain and the other is creating them in the frequency domain. We will focus on the time domain. Because our data set is

a time series, applying a supervised learning model to predict a failure or no failure is difficult based on only the data at a specific time point. The graphs in Figure 3.2 are examples that show that anomalous behaviour doesn't necessarily translate to only peaks in the accelerations but also shorter wavelengths in the series leading up to a failure. In order to capture this historically observed behaviour, we will summarise these values using a window size $\lambda$.

Take $x_t^i$ as the attribute $i$ at time point $t$ and its corresponding window of observed values $[x_{t-\lambda}^i, ..., x_t^i]$. We will summarise the mean aggregated values with a window $\lambda = 3$ hours using the variance as follows:

$$x\_var_t^i = \frac{\sum_{n=t-\lambda}^{t}(x_n^i - \bar{x}^i)^2}{\lambda + 1} \tag{3.2}$$

Where

$$\bar{x}^i = \frac{\sum_{n=t-\lambda}^{t} x_n^i}{\lambda + 1}$$

Because we expect the observed values leading up to a failure to show constant faster accelerations, we expect that the variance of these values will be lower when compared to the "normal accelerations".

## 3.4 Final Data Set

The methods that we will apply, mentioned in Chapter 4, are computationally expensive. In an effort to mitigate the run time, we will limit the number of attributes and observations.

Figure 3.3 shows interesting patterns for the correlations of the mean values of the attributes for $\Delta t = 2$ minutes. We can see that the vibration sensors of pump 1 are highly correlated with each other but the sensors of pump 2 show a different pattern where sensor 4 shows less correlation with the sensors 5 and 6. The vibrations from pump 1 and pump 2 also show no correlation with each other, which is to be expected since they are two different machines, but each pump does show some correlation with its corresponding power consumption. The graph also shows some correlation between the measurements from the basement and the measurements from the pumps, which can be attributed to the fact that the water level and flow rate determine how hard the pumps will pump.

In regards to the attributes for the final data set, we will select the ones that show less correlation with others and only consider the mean values. For example, since the vibration sensors from pump 1 are highly correlated with each other, we will only take one of these values into consideration. For pump 2 we will take two values into consideration since sensor 4 does not show correlation with the other two. Also, because there are so few malfunctions, we will only use the data from the three days leading

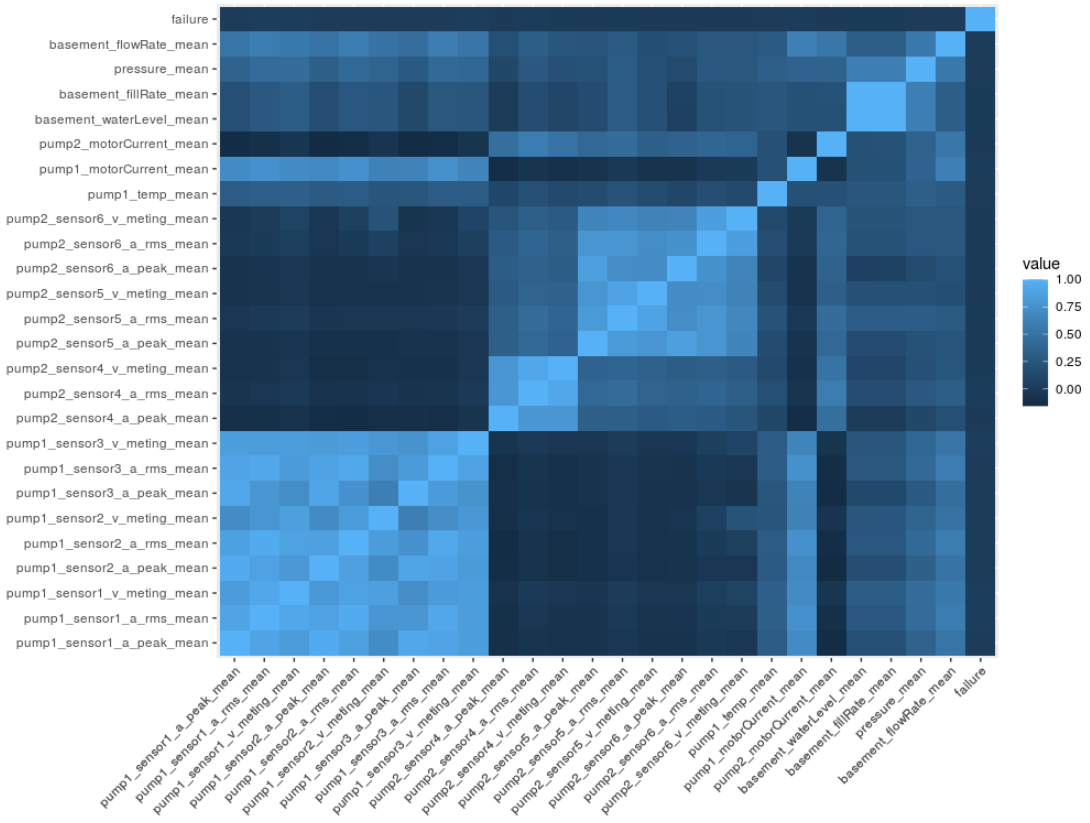



**Figure 3.3:** Correlation Matrix of the mean values of the attributes with $\Delta t = 2$, based on the Pearson correlation coefficient. The data from when pumps have shut off due to a malfunction, is excluded. The vibration sensors on pump 1 all show strong correlation with each other. Whereas one of the sensors on pump 2 shows less correlation with the other two sensors. There is also correlation between the vibrations and the power consumption.

up to a failure. We expect that this will give enough information on how the pumping station runs normally and when a malfunction is about to occur.

Since we also want to be able to compare the influence of the vibrations and power consumption on the model performance, we will create different data sets. An overview of the attributes in the different data sets is shown in Table 3.3. Data set 1 forms the base set and consists of the 2 minutes mean aggregated values of the vibrations, power consumption and various measurements from the basement. The remaining data sets form variations on this set. Data set 2 removes the vibrations and temperature leaving us a set to see the effects of the power consumption. Data set 3 is to see the effects of the vibrations and temperature and set 4 only has data from pump 2. Data set 5 also consists of the power consumption, but with more data since leaving out the vibrations gets rid of the gaps in the data. This last data set will help determine if the model performance when using the power consumption increases when using more data.

| Available attributes | Data set | Number of observations |
|---|---|---|
| timestamp<br>failure<br>pump 1 sensor 1 a-peak mean<br>pump 2 sensor 4 a-peak mean<br>pump 2 sensor 5 a-peak mean<br>pump 2 sensor 6 a-peak mean<br>pump 1 power consumption mean<br>pump 2 power consumption mean<br>pump 1 temperature mean<br>basement flow rate mean<br>basement pressure mean<br>basement water level mean<br>all attributes variance previous 3 hours | 1 | 15,000 |
| Data set 1 - vibrations - temperature | 2 | 15,000 |
| Data set 1 - power consumption | 3 | 15,000 |
| Data set 1 - pump 1 | 4 | 15,000 |
| Data set 2 complete power consumption | 5 | 54,500 |

**Table 3.3:** Final data sets with $\Delta t = 2$ minutes with data set 1 forming the base set. The remaining data sets will be based on variations of the base set. The data sets consist of the mean aggregated values for a selection of the vibration sensors. For each attribute we also calculate the variance of the previous 3 hours of every time stamp.

# Chapter 4

# Machine Learning Models

## 4.1 Random Forest

Tree-based classification models work by using a set of splitting rules to partition a feature space into smaller regions that have a similarity in response values [6]. A few of the many benefits of these decision trees, is that they produce easy to interpret rules that can also be visualized with tree diagrams. An example of one is shown in Figure 4.1. Kuhn and Johnson [15] explain that classification trees aim to partition the training data into smaller, more homogeneous groups. In this context, homogeneity means that the nodes of the split are more pure, meaning that each node contains a larger proportion of a class. One way to measure purity is with the Gini index. For a given node in a two-class problem, the Gini index is defined as:

$$p_1(1 - p_1) + p_2(1 - p_2) \tag{4.1}$$

with $p_1$ and $p_2$ being the class 1 and class 2 probabilities. Because $p_1 + p_2 = 1$, when either of the class probabilities is driven towards zero, the node will be pure in regards to one of the classes. The node will be least pure when $p_1 = p_2$.

A Random Forest for classification is an ensemble of decision trees where multiple classification trees are built on bootstrap samples of the train data. Each tree casts a vote for the classification and the proportion of these votes across the ensemble gives the predicted probabilities.

The data from pumping station Zuidbroek has an imbalance in the number of failures and non-failures that occur. And when it comes to modeling discrete classes, this imbalance can have a significant impact on how effective the model is. Kuhn and Johnson [15] discuss a number of methods to remedy severe class imbalance. These methods include, among others, model tuning to increase the sensitivity of the minority class, using alternative probability cutoffs, assigning unequal case weights and using different sampling methods. Because of time constraints, we will focus on the latter and apply different sampling methods to the random forest model.

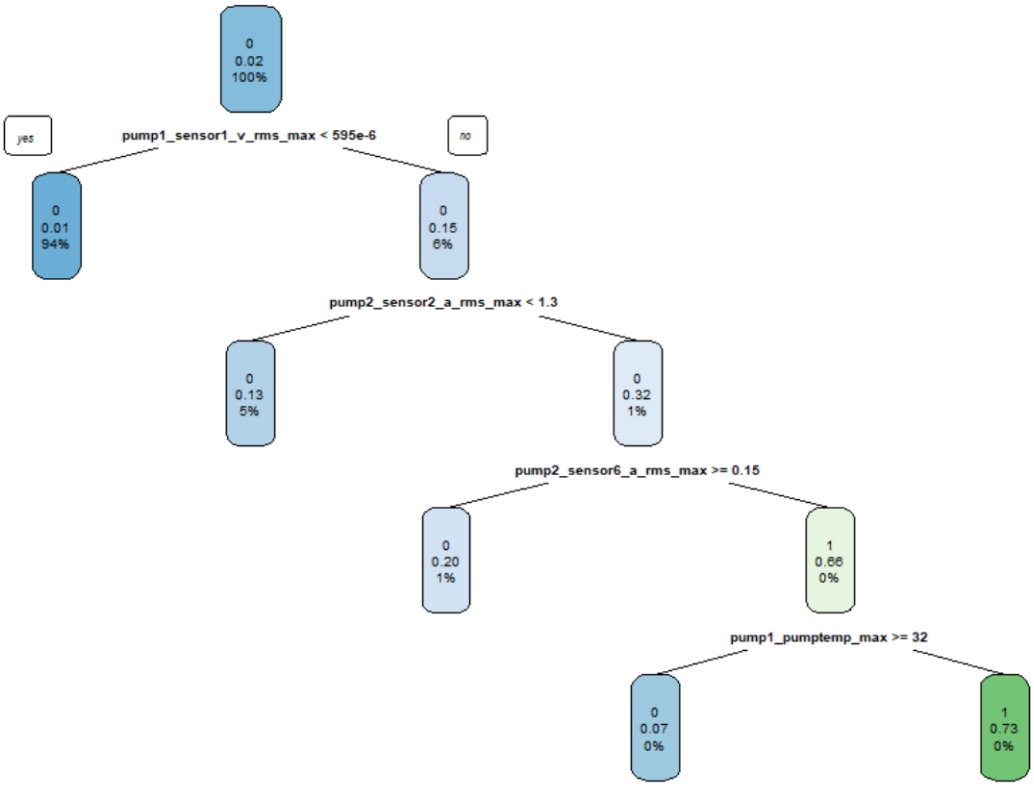



**Figure 4.1:** Example of a decision tree predicting whether or not a failure will occur (1 or 0) based on the maximum root mean square of the vibration speed of sensor 1, the maximum root mean square of the acceleration of sensor 2 and 6 and the maximum pump temperature of pump 1.

The data will be split according to a stratified split with 70% for the train set and 30% for the test set. For the training phase we will also apply k-fold cross validation where the train set is split into $k = 10$ non-overlapping chunks, and use $k - 1$ chunks to train on and one chunk to test upon. Further tuning of the model will be done by trying different values for the parameter $m_try$, which denotes the number of randomly selected predictors to choose from at each split.

### 4.1.1 Sampling Methods

A random forest model takes random samples of the training data to build decision trees on. With a large class imbalance, the minority class can be underrepresented in these samples which will most often result in a heavily over-fit model which will classify every instance as the majority class. A straightforward approach to avoid this, is to balance the class frequencies for the training phase so that the model does not have to deal with the imbalance [15]. While the train set will be sampled to be balanced, the test set shall remain unbalanced in order to compute honest estimates of future performance.

Kuhn and Johnson [15] mention a number of subsampling techniques which they apply with the `caret` package. They also warn for the pitfalls of sampling the training data before model fitting, namely overly optimistic estimates because there is a possibility that the holdout set that is generated during re-sampling does not reflect the class imbalance. And the performance of the model can become more uncertain due to the random subsampling process.

As an alternative, the subsampling can be included inside the usual re-sampling process. Fortunately, in recent versions of the `caret` package this can be achieved with relatively simple syntax. We will apply the following subsampling methods that are conducted inside of re-sampling:

- Under-sampling: Randomly sample from the majority class so that it's roughly the same size as the minority class. Take for example a training set where the majority class takes up 90% of the samples and the minority class the remaining 10%. Under-sampling will randomly sample from the majority class so that it also takes up 10% of the training set. This leaves 20% of the total training set to fit the model.
- Over-sampling: Randomly sample, with replacement, from the minority class until it's the same size as the majority class.
- Random Over-Sampling Examples (ROSE): Uses artificially generated balanced samples according to a smoothed bootstrap approach [17].
- Syntetic Minority Over-sampling Technique (SMOTE): Uses a combination of under-sampling the majority class and over-sampling the minority which can achieve a better performance (in ROC space). When over-sampling the minority, this method uses synthetically created minority class examples [9].

### 4.1.2 Performance Metrics

In order to compare the different models, we will use the performance metrics sensitivity, specificity, precision, F1 and AUC. For a description of these metrics, see Table 4.2. Because of the class imbalance, the accuracy is obviously not a reliable metric. Instead, we will focus on metrics that describe how well each of the classes are predicted, with an emphasis on how well the failures are predicted. Most of these metrics can be derived from the model's confusion matrix (see Table 4.1). This matrix gives an overview of the class predictions, namely:

- TN: True Negatives. These are the number of cases where the model predicts no failure and the true value is also no failure.
- FN: False Negatives. The number of cases predicted as no failure but are actually a failure.
- FP: False Positives. The number of cases predicted as a failure but are actually not a failure.
- TP: True Positives. The number of cases predicted as a failure and are actually a failure.

| | | True value | |
|---|---|---|---|
| | | No Failure | Failure |
| Predicted value | No Failure | TN | FN |
| | Failure | FP | TP |

**Table 4.1:** Confusion Matrix for a two-class classification problem

The Area Under Curve is the only metric that cannot be determined with the confusion matrix but is instead calculated with the Receiver Operating Characteristic (ROC) curve. The ROC curve is a tool to determine alternate thresholds for class probabilities when considering the sensitivity and specificity. When one of the Random Forest models results in a sensitivity of 0% and a specificity of 100%, these values are due to the default 10% probability threshold. A different threshold could capture more true positives and lead to an improved sensitivity.

The ROC curve is generated by using multiple thresholds when evaluating the class probabilities for the model [15]. For each threshold value, the sensitivity and specificity are plotted against each other. Figure 4.2 shows an example for one of the Random Forest models. With the 50% default threshold, the sensitivity is at its lowest at 0%. By decreasing the threshold, the sensitivity is improves to 80%. This does come with a cost of a decrease in specificity.

The area under the ROC curve can be used as a performance metric. Since a perfect model would have 100% specificity and sensitivity and therefore an area of one. While a bad model would result in a ROC curve that runs along the 45° diagonal, with an area of around 0.50.

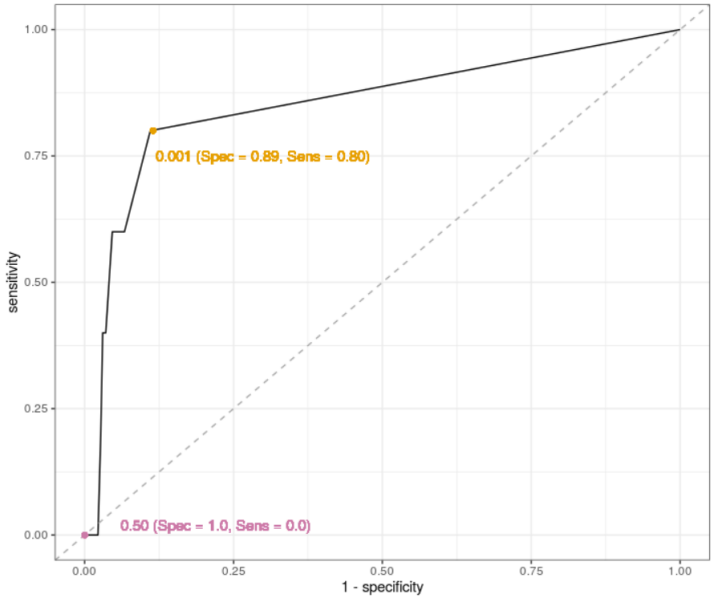



**Figure 4.2:** A receiver operator characteristic (ROC) curve for the Random Forest model. The orange dot indicates the values corresponding to a cutoff of 0.1 %, meaning probabilities greater than 0.001 are labeled as failures.

## 4.2 k-means Clustering

Whereas the Random Forest model from the previous sections is a supervised learning method, as in we supply the classification of the instances to the model, k-means clustering is an unsupervised learning method. This method is fairly intuitive, where it finds a predefined number of cluster, $k$, in the given data. Each cluster has a cluster center, which is chosen randomly at first, and is refined in a loop. Each point is assigned to a cluster based on the minimum distance to the cluster center. There are different distance metrics that can be used when it comes to numerical data, and the most well known is the Euclidean distance. This defines the distance between two point $x_i$ and $x_j$ as:

$$euclidean\_distance(x_i, x_j) = \sqrt{\sum_{k=1}^{p} (x_i^k - x_j^k)^2} \tag{4.2}$$

Once each data point is assigned to a cluster, the cluster centers are recalculated as the average of all the points in the cluster. This process keeps repeating until the cluster centers don't change (or change very little). The full algorithm is shown in Hoogendoorn and Funk [12].

There are a number of papers [21] [2] [4] that have successfully used k-means to identify machine conditions. These clusters, which can convey a machine's age or condition, are then often used as an extra attribute in the data that is fed into a supervised learning algorithm. But Uhlmann et al. [21] was able to identify the machine's behavioural groups as clusters. These were clusters that showed normal behaviour and faulty behaviour.

| Metric | Formula | Description |
|---|---|---|
| Sensitivity | $\frac{TP}{TP+FN}$ | The percentage of failures that were labeled correctly |
| Specificity | $\frac{TN}{TN+FP}$ | The percentage of non-failures that were labeled correctly |
| Precision | $\frac{TP}{TP+FP}$ | The percentage of predicted failures that were labeled correctly |
| F1 | $2\frac{Precision*Recall}{Precision+Recall}$ | Where recall is the same as sensitivity, this gives a harmonic mean of the precision and recall |
| Area Under Curve (AUC) | - | Area under the ROC curve |
| Average Time to Failure (ATF) | - | The average amount of time between a failure prediction and an actual failure |

**Table 4.2:** Performance metrics. The AUC is only applicable to the Random Forest and the Average Time to Failure only for k-means.

In our case we will see if the behaviour of the pumps can be captured in clusters. We will be able to see how the time series leading up to a failure is clustered and if any discernible pattern appears. Such a pattern can be used to determine a decision rule to evaluate whether the system is heading towards a failure. For example,

*"If the system stays in cluster 2, which represents an irregular phase, for at least x minutes, this could result in a failure."*

The duration of an "irregular" phase will be called the *warning signal* and based on such a decision rule, we will be able to calculate predictions of a time-series, which will also enable us to use the performance metrics mentioned in Table 4.2. The metric, Average Time to Failure, will also allow us to calculate the average time between a failure prediction and when the failure actually occurred.

In order to determine an optimum solution we will vary a number of parameters in this k-means and decision rule model. These are mentioned in Table 4.3, where the length of the warning signal is expected to have the highest impact on the model performance.

In regards to the data, there will be no regular train/test split since the three days leading up to a failure will function as a "test" set. These data will determine if the decision rule results in false positives during the moments before a failure. Data set 5 however, will be used to perform a train/test split, where the power consumption

| Parameter | Description | Values |
|---|---|---|
| centers | the number of clusters used in k-means | 2, 3 |
| iter.max | the maximum number of iterations allowed | 10, 50, 100 |
| nstart | the number of initial configurations for the randomly chosen centroids | 10, 50, 100, 1000 |
| warning signal | duration of an irregular phase | 2-90 minutes |

**Table 4.3:** Parameter values for tuning k-means and decision rule model.

model resulting from data set 2 will be applied to the failures in data set 5 that are not in data set 2.

### 4.2.1 Silhouette Score

The *Silhouette score* is a metric which measures the tightness of the clusters, relative to the distance to the closest cluster [12]. This is another metric that can be used to evaluate the number of clusters that we expect to find. Hoogendoorn and Funk [12] first define the average distance of a point to the other points in its cluster as:

$$a(x_i) = \frac{\sum_{\forall x_j \in C_l} distance(x_i, x_j)}{|C_l|} \text{ where } x_i \in C_l \tag{4.3}$$

Followed by the average distance to the points in the closest cluster:

$$b(x_i) = \min_{\forall C_m \neq C_l} \frac{\sum_{\forall x_j \in C_m} distance(x_i, x_j)}{|C_m|} \text{ where } x_i \in C_l \tag{4.4}$$

They then define the *silhouette score* as:

$$silhouette = \frac{\sum_{i=1}^{N} \frac{b(x_i) - a(x_i)}{max(a(x_i), b(x_i))}}{N} \tag{4.5}$$

This score compares the distances $a(x_i)$ and $b(x_i)$ and the resulting measure says how the clusters are relative to the distance to the clusters closest to them. This score ranges from -1 to 1, with -1 being the worst score and 1 the best. -1 would mean that the distance from points to the closest clusters is lower than the distance between points within the cluster. A score closer to 1 would mean low values for $a(x_i)$, which indicates a tight clustering, and high values for $b(x_i)$, which would indicate that the clusters are far apart.

# Chapter 5

# Results

This chapter discusses the evaluation of the results from the Random Forest and k-means clustering models. These models will be evaluated based on the performance metrics that were discussed in Chapter 4.

## 5.1 Random Forest

Training an original random forest model without re-sampling leads, as expected, to a very overfit model where the model classifies every instance of the train set correctly but when applied to the test set, classifies every instance as "no failure".

| | | True value | |
|---|---|---|---|
| | | No Failure | Failure |
| Predicted value | No Failure | 4654 | 5 |
| | Failure | 0 | 0 |

**Table 5.1:** Confusion Matrix of the test set for the original random forest model without sampling. This model is heavily overfit and predicts only no-failures.

When applying different sampling methods, we do see different results. The resulting performance metrics of these models are shown in Figure 5.1. All of these models show a high specificity meaning that a high percentage of the "no failures" are predicted correctly, which is usually the case in severely imbalanced data sets. But we can see that this is not the case for most models when looking at the sensitivity, precision and F1, which are all very low. This means that very few of the failures are predicted correctly and of the predicted failures, very few are actually failures (see the confusion matrix in Table 5.2). The smote- and under-sampling models however, seem to perform better in sensitivity. Interestingly, all models show a AUC of around 0.75 which means that while the current classifiers score badly in performance, there is a possibility to increase performance by using a different threshold. The performance metrics also do not differ much across the different data sets. The only difference seems to lie in the sensitivity where we can see that the under-sampling model performs the best with the data set that includes both the vibrations and the power consumption.

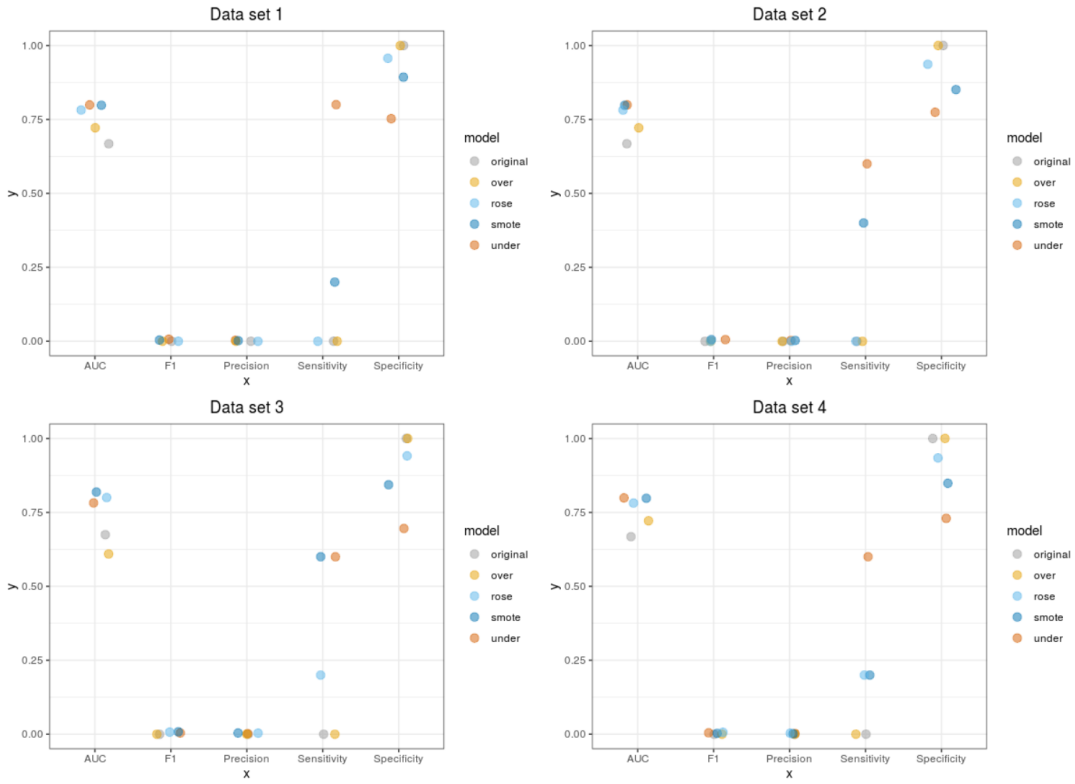



**Figure 5.1:** Performance metrics for different sampling methods when applying the models to the test set. The data has $\Delta t = 2$ minutes and the y value is the 2 hour horizon. Data set 1 includes both the vibrations and power consumption, data set 2 excludes the vibrations, data set 3 excludes the power consumption and data set 4 excludes the data from pump 1. The main difference between data sets lies in the achieved sensitivity of the smote- and under-sampling models.

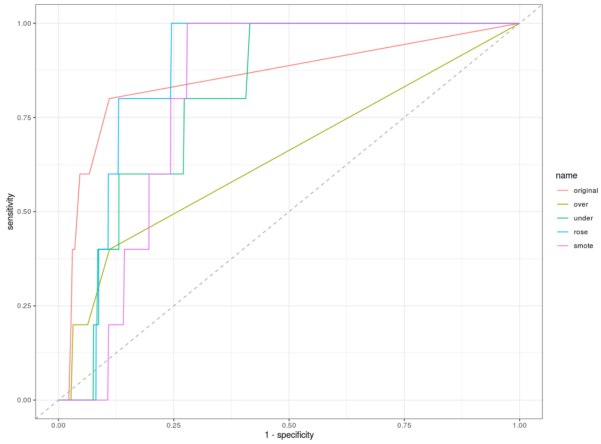



**Figure 5.2:** ROC curves for different sampling methods on data set 1. The data has $\Delta t = 2$ minutes and the y value is the 2 hour horizon. Data set 1 includes both the vibrations and power consumption.

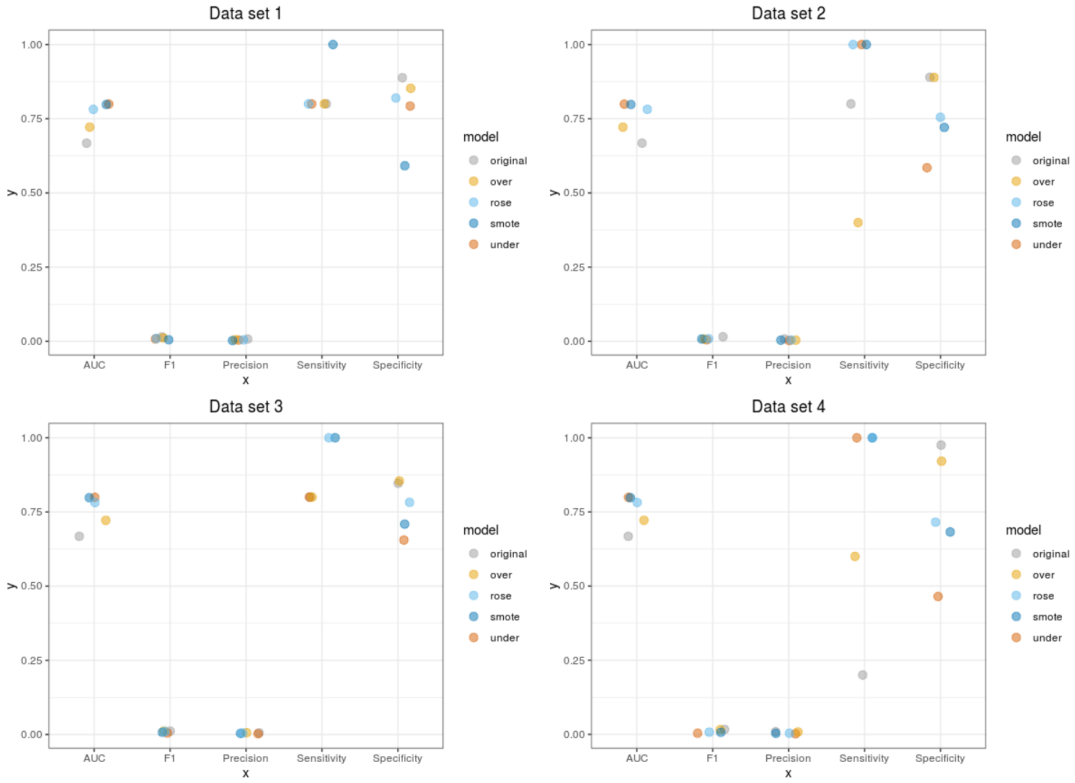



**Figure 5.3:** Performance metrics for different sampling methods when applying the models to the test set and using the optimal threshold from the corresponding ROC curve. The data has $\Delta t = 2$ minutes and the y value is the 2 hour horizon. Data set 1 includes both the vibrations and power consumption, data set 2 excludes the vibrations, data set 3 excludes the power consumption and data set 4 excludes the data from pump 1. .

When using the corresponding optimal threshold from the ROC curve for each model, we can see in Figure 5.3 that the sensitivity across all models increases. But as expected, this has no influence on the precision meaning that the classifiers predict a large number of false positives i.e. predicts failures that are not truly failures. The results also show the expected trade-off between specificity and sensitivity. The models with a higher sensitivity also have a lower specificity. The smote model seems to perform well across data sets, with a sensitivity of 100% and a specificity of around 75%. There also aren't any extreme differences between the results for the different data sets. The data set containing only the measurements from pump 1, does have a few models that perform slightly worse when compared to the other sets.

In Figure 5.4, the variable importance of the smote model shows that the variance of the previous 3 hours has the most impact. Especially for the water level, flow rate and power consumption of pump 2. This could be attributed to the fact that the operating intensity of the two pumps depends on the water level, flow rate and pressure. For example, an increase in flow rate would cause an increase in operating intensity for the pumps. While a higher water level would cause a decrease in operating intensity

| | | True value | |
|---|---|---|---|
| | | No Failure | Failure |
| Predicted value | No Failure | 3696 | 0 |
| | Failure | 952 | 5 |

**Table 5.2:** Confusion Matrix of the test set for the smote model when using data set 1. This model has a high sensitivity because it classified all the failures correctly but a low precision due to a large number of False Positives.

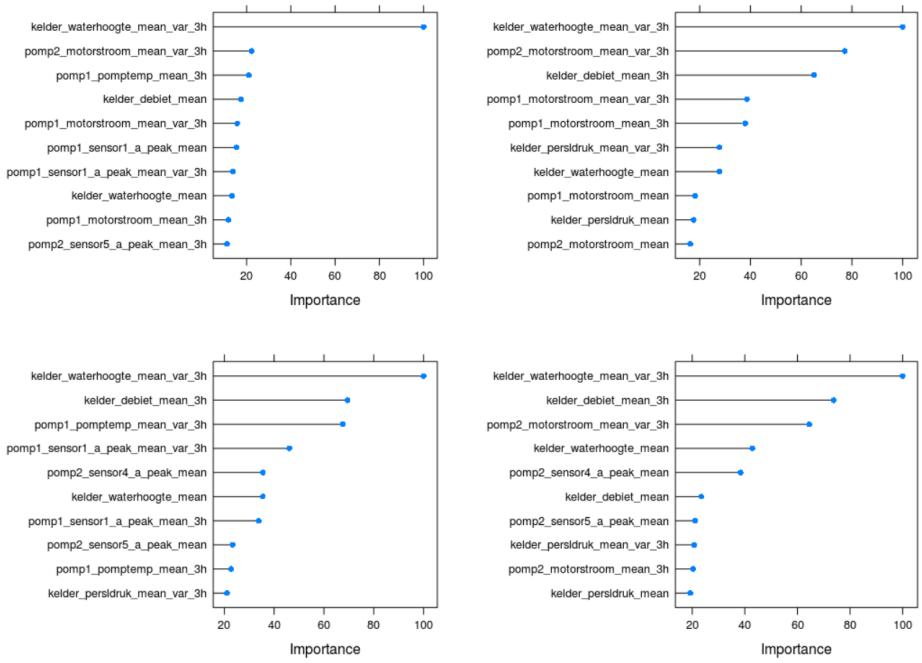



**Figure 5.4:** Variable importance for the top 10 variables of the smote model based on the Gini-index. The results are shown respectively from left-to-right and top-to-bottom for data set 1, 2, 3 and 4.

when the flow rate and pressure remain constant, since the higher water level will cause a higher pre-pressure, making it easier to pump the water away. The vibration and temperature measurements seem to have less impact, except when excluding the power consumption.

Finally, when trying to predict further into the future, one would expect a steady decline in performance. But Figure 5.5 shows how volatile the sensitivity and specificity behave when using data set 1 with both the vibrations and power consumption and data set 5 with the larger number of observations.

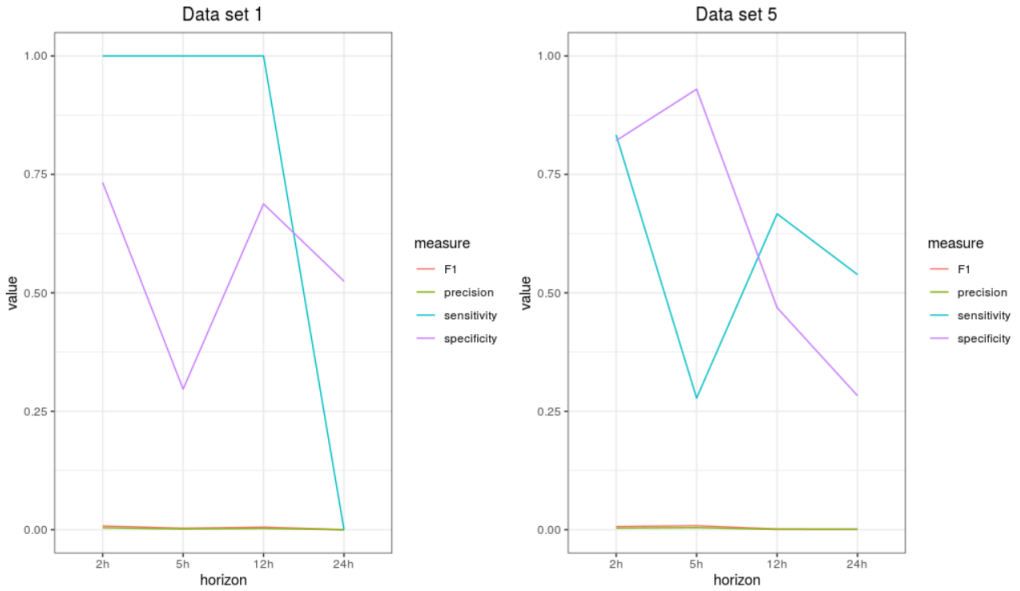



**Figure 5.5:** The smote model applied to data set 1 (with both the vibrations and power consumption) and data set 5 (the larger data set of the power consumption) for different prediction horizons.

## 5.2 k-means Clustering

As mentioned in chapter 2, k-means clustering has been successful in determining the types of faults and/or operating conditions of machines [5] [11]. Applying k-means clustering would ideally results in more than two clusters. One indicating a normal/standard condition, one indicating an irregular condition or period to a failure and one indicating when the pumps are in failure. Since we filtered out the data during the failures, finding two clusters would still be reasonable. When calculating the silhouette score for multiple numbers of clusters, Figure 5.6 shows that the highest score is reached with just two clusters. The highest score being just above average ($\sim$ 0.55), belongs to the data sets without the vibrations and temperature. This would indicate that the power consumption form tighter clusters when compared to the vibrations and temperature.

After applying k-means clustering for two clusters to the data that contains both the vibrations and power consumption (data set 1), we can look at how this clusters the time series leading up to the failure from September 2019 mentioned in Chapter 3.2. It would be preferable that there is a change in cluster some time before the failure occurs, which can be then used as a warning signal. The results in Figure 5.7 show how the data leading up to the failure are clustered. In this case, k-means is able to distinguish a different cluster, shown in red, around two hours before the first failure is registered. This is the moment where there is an increase in the mean water level en flow rate and pump 2 shuts off, resulting in pump 1 using more power to pick up the slack. The remainder of the series after the second failure was not used for clustering since we

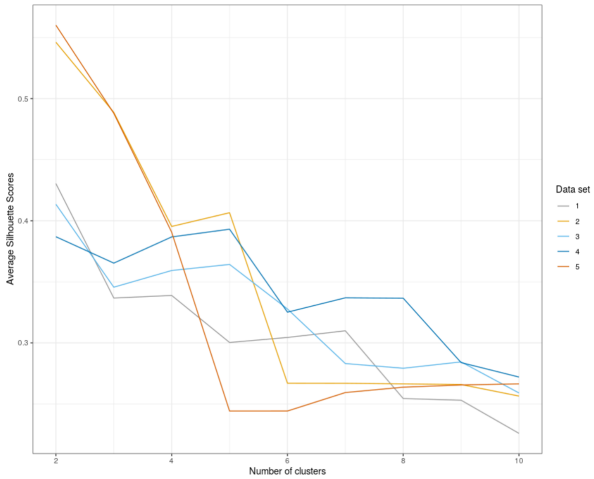



**Figure 5.6:** Silhouette score for the different data sets. Set 1: vibrations + power consumption, set 2: power consumption, set 3: vibrations, set 4: vibrations + power consumption pump 2 and set 5: larger set of power consumption.

only consider the data from three days prior to a failure.

Based on Figure 5.7, cluster 1 (green) could indicate a normal phase and cluster 2 (red) a irregular/warning phase. Using this we can look at how the time series are clustered for all 17 failures. Figure 5.8 shows horizontal bars for each failure which represent how often, in the three days before a failure, a certain cluster appeared. In Figure 5.8(a) we can see that during the three days before a failure, cluster 1 is prevalent, which could indicate that for most of the time the system is running in a normal phase. The last failure from the 26th of February 2020 being the exception in this case since 90% of the time the system ran in an irregular phase. Now lets look at what happens to the time series closer to a failure. During the three hours before a failure, shown in Figure 5.8(b), we see that cluster 2 occurs more often or that the system is running in an irregular phase. Note that not all of the bars add up to three days or three hours since some of the failures occurred on the same day.

By looking at the failure dates shown on the y-axis in Figure 5.8(b), we can see which failures occurred on the same day. For example, on the 4th of September 2019 there were two failures within a couple of hours of each other. The most failures on one day occurred on the 19th of November 2019. During the three hours before the first failure on the 19th of September 2019, the system ran an hour in a normal phase and two hours in an irregular phase. During the time leading up to the second failure of that day, the system stayed in an irregular phase. The same can be said of the other days with multiple failures. After the initial failure, the system seems to stay in an irregular phase. A look at the failure logs does show that these failures are logged as fixed before the next one occurs. So these recurring failure logs could be due to maintenance engineers who are, at that time, working on fixing the issue at hand.

Focusing on just the initial failure on days with multiple failures and the days with a

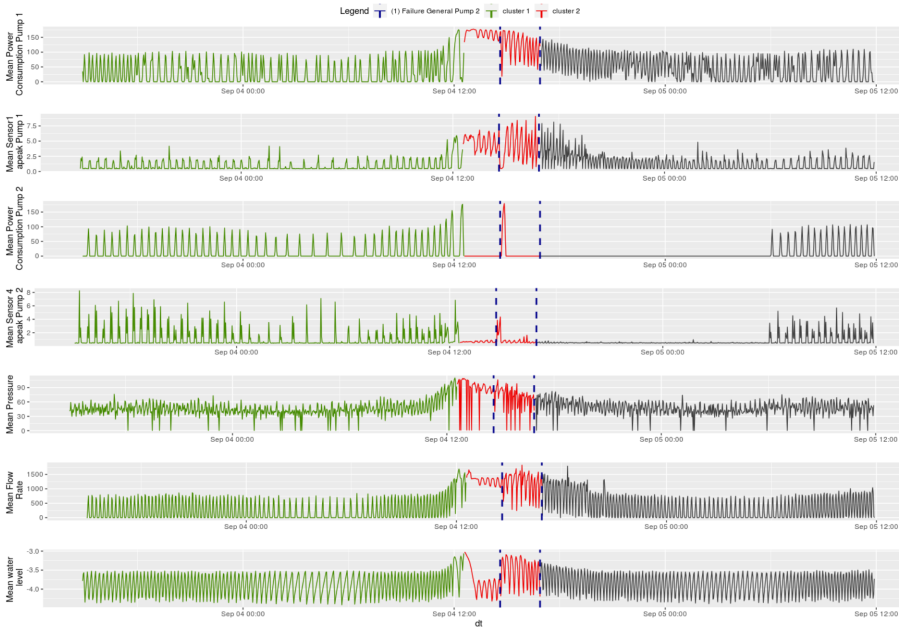



**Figure 5.7:** Two consecutive failures General Pump 2 from September 2019 after applying k-means clustering for two clusters on data set 1: vibrations + power consumption12. The model is able to distinguish a different cluster (red) around 2 hours before the first failure is registered.

single failure, we can see that there are failures where the system ran in an irregular phase for the entire three hours leading up to the failure (e.g. 13-11-2019, 19-11-2019, 17-02-2020 and 26-02-2020). Whereas others ran a shorter irregular phase. The three failures that occurred on 09-02-2020, 27-01-2020 and 22-02-2020 also show the shortest irregular phase of around half an hour.

Now, Figure 5.9 shows a similar picture but gives us the opportunity to see how often the system changes cluster/phase during the three days and three hours leading up to a failure. If we look at the days leading up to a failure in Figure 5.9(a), we can see that for the majority of the time the system runs in a normal phase and always ends in an irregular phase (red) which is then followed by a failure. We can also see that there are irregular periods during those days and periods where the system oscillates between phases. During this oscillation, the system switches between a normal and irregular phase for anywhere between 0 and 10 minutes. These irregular periods could be the result of malfunctions which, as mentioned previously, do not require intervention and tend to resolve on their own.

### 5.2.1 Decision Rule(s)

Based on the system during the three hours before a failure, shown in 5.9(b), we can determine a decision rule for when a failure is likely to occur according to how long the system has been in an irregular phase. Figure 5.9(b) shows a clear pattern where the system stays in an irregular phase for a certain period before the failure occurs. This


**(a)** 3 days before a failure

**(b)** 3 hours before a failure

**Figure 5.8:** The total amount of time that the system ran in a certain cluster/phase before each failure was registered along with the corresponding proportions, based on data set 1: vibrations + power consumption. It shows that for most failures the system runs in an irregular phase for the majority of time when closer to a failure. Note that some bars are shorter due to multiple failures on one day.

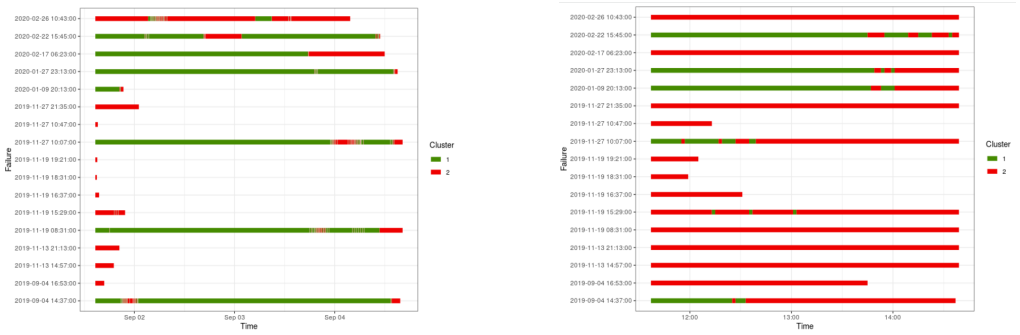
pattern changes when we use different data. Looking at similar plots for the other data sets shown in Appendix B, we can see that the data sets that incorporate the vibrations show a consistent irregular phase leading up to a failure whereas the data sets with just the power consumption shows an oscillation between the phases for a certain duration. When oscillating, the system quickly jumps back and forth between the two clusters. These two situations would require different decision rules. Based on the data sets that include vibrations we will decide that:

> *"If the system stays in an irregular phase (cluster 2) for at least x minutes, predict a failure."* (5.1)

For the data sets that include only power consumption this would be:

> *"If the system oscillates between regular and irregular phase for at least x minutes, predict a failure."* (5.2)

We will choose an initial value of 38 minutes for the warning signal based on Figure 5.9(b), as this value enables us to predict all failures. The first results of using different warning signals is shown in Figure 5.10 where both decision rules have been applied to the different data sets in order to calculate predictions during the three days leading up to a failure. These results show that decision rule 5.1 does work best for the data sets that include the vibrations and rule 5.2 works better for the power consumption. The sets that include the vibrations also perform better than the sets with just the power consumption. We can also see that a shorter warning signal of 20 minutes results in a much higher sensitivity where about 90% of failures have been predicted correctly.




**(a)** 3 days before a failure **(b)** 3 hours before a failure

**Figure 5.9:** The time series 3 days vs 3 hours before a failure labeled with the corresponding cluster, based on data set 1: vibrations + power consumption. This shows how often the system changes cluster during the time leading up to a failure. Note that some bars are shorter due to multiple failures on one day.

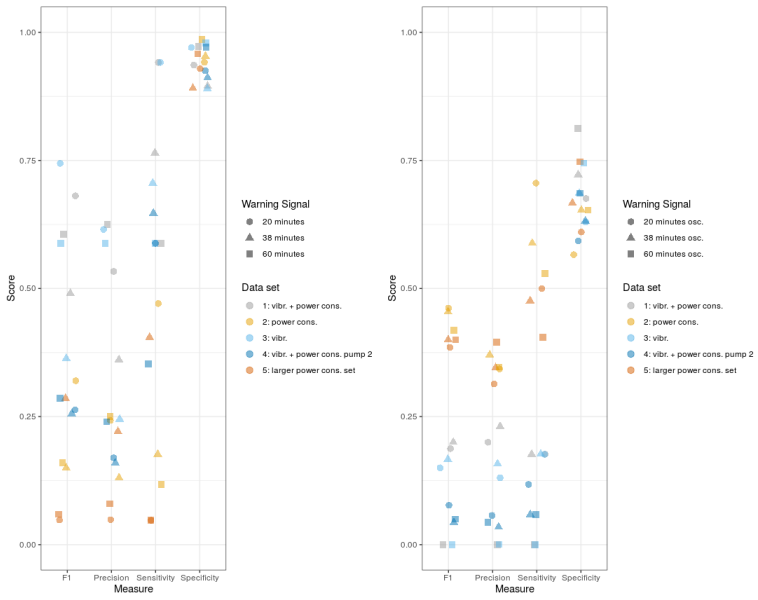
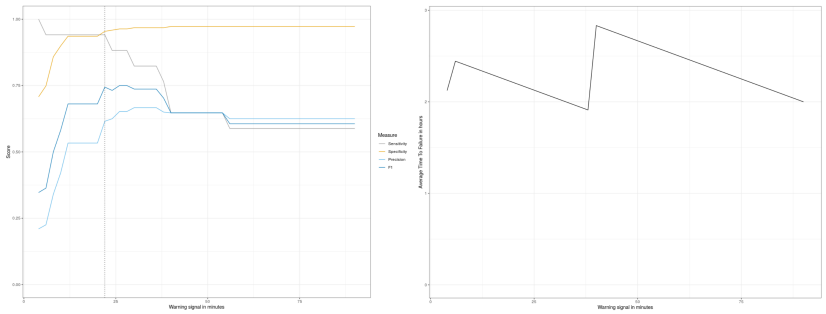



**Figure 5.10:** Performance measures based on prediction results using 2 clusters with left decision rule 5.1 and right decision rule 5.2. The Warning Signal represents the maximum (oscillating) duration of the irregular phase. A maximum duration rule works best for data that includes vibrations whereas a maximum oscillating rule works better for sets with the power consumption.

### 5.2.2 Parameter Tuning

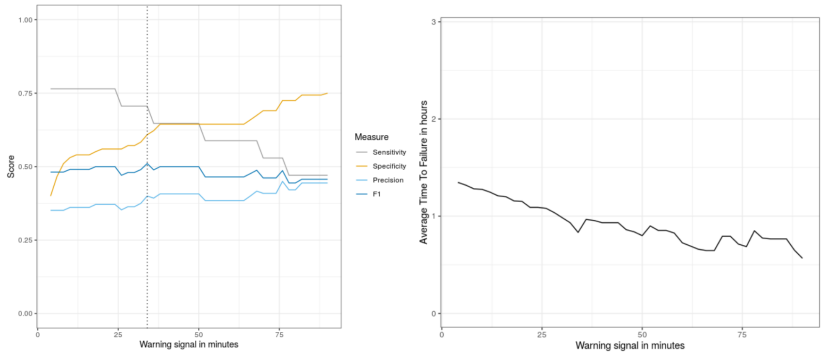
As expected, the parameters *iter.max* and *nstart* had zero effect on the performance of the decision rules and using three clusters for k-means also results in a pattern of system phases that make it difficult to determine a suitable decision rule (see Appendix B). As a result, we shall focus on the effect of the warning signal on the model performance.

When varying the length of the warning signal, we achieve the results shown in Figure 5.11. The dotted line shows an "optimum" based on four performance metrics. Note that the average time to failure for the vibrations appears jagged as a side effect from decision rule 5.1. Again the model based on the data set with the vibrations outperforms the model based on the set with the power consumption. When using a warning signal of 22 minutes, the vibration model shows a high specificity and sensitivity of respectively 0.95 and 0.94. This means that a 95% of non-failures and 94% of failures are predicted correctly. With the corresponding precision, 62% of the predicted failures are actual failures. The resulting average time to failure in this case is around 2 hours meaning that after a prediction is made, it will take on average 2 hours for the failure to occur. The optimum warning signal when using the power consumption model at 28 minutes, results in overall lower scores across across all metrics, where the specificity and sensitivity drop 30% and the precision and average time to failure is halved.



**(a)** Prediction scores data set 1: vibrations + power consumption

**(b)** Average Time to Failure data set 1: vibrations + power consumption




**(c)** Prediction scores data set 2: power consumption

**(d)** Average Time to Failure data set 2: power consumption

**Figure 5.11:** Performance metrics for varying lengths of the warning signal. The dotted line shows an "optimal" length for a warning signal.

| Score | Data set 1: vibrations + power consumption | Data set 2: power consumption | Data set 3: vibrations | Data set 4: pump 2 | Data set 5: power consumption |
|---|---|---|---|---|---|
| Sensitivity | 0.94 | 0.64 | 0.94 | 0.53 | 0.50 |
| Specificity | 0.95 | 0.58 | 0.97 | 0.96 | 0.62 |
| Precision | 0.62 | 0.33 | 0.64 | 0.28 | 0.32 |
| F1 | 0.74 | 0.44 | 0.76 | 0.37 | 0.39 |
| Average Time to Failure (in hours) | 2.2 | 1.02 | 2.2 | 4.4 | 1.5 |

**Table 5.3:** Results k-means + decision rule model.

The overall results in Table 5.3 also show that it is the vibrations which add the predictive power since data set 1 and 3 achieve similar results. Using only the data from pump 2 achieves the lowest results. We can also conclude that using almost four times the amount of power consumption data (data set 5) does not lead to a model with better results.

| | Sensitivity | Specificity | Precision | F1 | Average Time to Failure |
|---|---|---|---|---|---|
| Score | 0.55 | 0.62 | 0.27 | 0.37 | 1.02 |

**Table 5.4:** Power consumption decision rule model applied to data set 5 as test set. This model has an oscillating warning signal duration of 28 minutes. Note that this test set excludes the failures from data set 2.

# Chapter 6

# Conclusion

The main challenge of Predictive Maintenance is to determine an efficient maintenance strategy that maximizes operational hours while reducing the required maintenance en associated costs. This thesis investigated the use of applying Machine Learning techniques to see if such an efficient strategy is possible compared to the current maintenance schedule, where maintenance is performed twice a year. We also determined whether vibration data adds any predictive power to this strategy.

We developed two predictive models for sewage pumping station Zuidbroek; a supervised learning model (Random Forest) and a unsupervised learning model (k-means clustering). The data consists of various measures taken from the basement of the station and the power consumption and vibrations of the two pumps. This data is severely imbalanced, with the failures making up less than 1% of the data. Because of the class imbalance, we applied different sampling techniques to the Random Forest model. The Random Forest and k-means clustering model were evaluated on performance by using the sensitivity, specificity, precision, and F1.

Our research question was formulated as *"How well do certain Machine Learning models perform when predicting the failure of components of the sewage pump station Zuidbroek?"*. The Random Forest model achieves a sensitivity of 75% meaning that it was able to correctly classify 75% of failures. However, this model also has a precision of close to 0.00%. This means that this model produces a large number of false positives (i.e., classifies non-failures as failures). The large number of false positives could be due to disruptions occurring. According to domain experts these could show the same change in patterns as the failures we highlighted but will eventually correct themselves. These disruptions also show up in the k-means clustering results.

The k-means clustering model performs better when compared to the Random Forest. This is somewhat surprising since the highest Silhouette score was only 0.56 for 2 clusters, which would indicate that the data does not form very compact clusters. But with these two clusters we were able to distinguish two different system phases, namely a normal and irregular phase. Based on this behaviour we created two decision rules for the vibration and power consumption features, respectively. These rules predict whether a failure might occur when the system has stayed in an irregular phase for a

certain period of time. The best results were achieved when using the vibration features. When the system stayed in an irregular phase for at least 22 minutes, 94% of the failures were predicted correctly (sensitivity) and of the predicted failures, 62% were actually failures (precision). The vibration model also gives an average time to failure of 2 hours. Meaning that, on average, there is a 2 hour window to take action in order to avoid a possible failure.

To answer the sub-question *"Can the same predictions be achieved without the vibration data?"*, the Random Forest models show no difference in performance when using either the vibration or power consumption data or when using both. Based on these results, the same performance is achieved when using just the power consumption of the pumps. The variable importance of the best performing model also shows that the standard deviation of the past 3 hours for the different attributes, has the most predictive power. And of these attribute standard deviations, attributes such as water level and power consumption of pump 2 rank much higher than the vibration attributes. The k-means clustering also achieves a higher Silhouette score when using the data sets with just the power consumption. This indicates that the power consumption forms tighter clusters than the vibrations. But, this is contradictory to the fact that in this model the vibrations offer much better predictive results compared to the power consumption.

K-means clustering does show promising results in being able to distinguish two types of system phases. The small number of failures does take the model performance into question, however, the power consumption model yields similar results when using four times the number of failures.

It is important to note that these results are based on a limited number of actual failures, 17 in total. This does take into question how generalizable these conclusions are. A lesson learned is that in order to perform supervised learning in this case, would require a change in how the necessary data is collected. At the start of the data analyses we ran into a lot of issues with time zone differences due to data being collected from different systems. This was corrected to the best of our abilities. Logging failures would also require logging a cause since a large number of the logged failures were not actually due to a failure but due to sensor issues and maintenance being performed.

In conclusion, it is difficult to say how effective these models are in order to perform predictive maintenance on sewage pump station Zuidbroek and more data would be needed to validate this. But k-means clustering does give hopeful results that predictive maintenance is possible and that the vibration measures are more effective to do this than just the power consumption.

# Appendix A

# Data Attributes

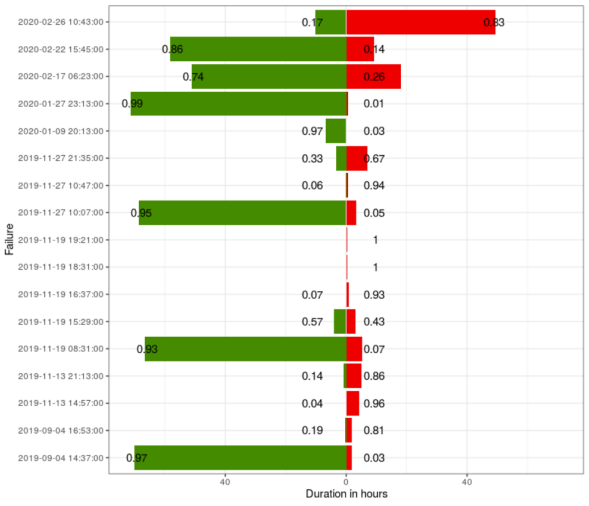
| Attribute | Percentage of missing values |
|---|---|
| Flow rate [1] | 0.00% |
| Water level [2] | 0.00% |
| Fill rate [3] | 0.48% |
| Pressure [4] | 1.10% |
| Power consumption pump 1 | 0.00% |
| Power consumption pump 2 | 0.00% |
| Pump 1: temperature | 0.12% |
| Pump 1: Sensor 1 | 0.00% |
| Pump 1: Sensor 2 | 0.00% |
| Pump 1: Sensor 3 | 0.00% |
| Pump 2: Sensor 4 | 0.00% |
| Pump 2: Sensor 5 | 0.04% |
| Pump 2: Sensor 6 | 0.04% |

**Table A.1:** Percentage of missing data per attribute with $\Delta t = 2$ min.

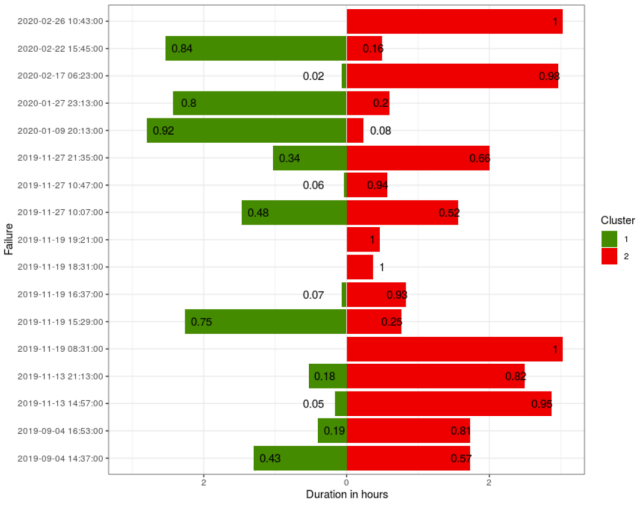
[1] In Dutch: Debiet
[2] In Dutch: Waterhoogte
[3] In Dutch: Vullingsgraad
[4] In Dutch: Persleidingdruk

# Appendix B

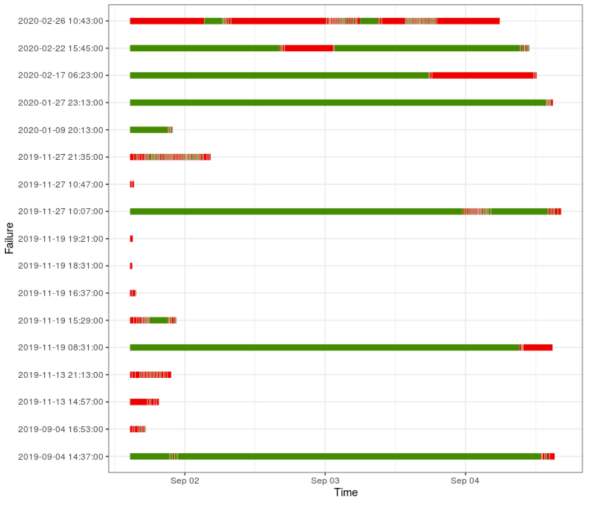
# K-means clustering




**(a)** Total amount of time in a phase 3 days before failure.




**(b)** Total amount of time in a phase 3 hours before failure.




**(c)** Phases of the time series 3 days before failure.


**(d)** Phases of the time series 3 hours before failure.

**Figure B.1:** Clusters or phases of each time series according to data set 2: power consumption.


**(a)** Total amount of time in a phase 3 days before failure.

**(b)** Total amount of time in a phase 3 hours before failure.

**(c)** Phases of the time series 3 days before failure.

**(d)** Phases of the time series 3 hours before failure.

**Figure B.2:** Clusters or phases of each time series according to data set 3: vibrations.

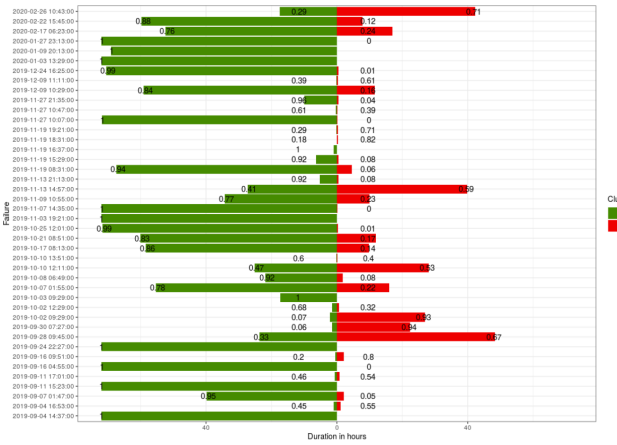

**(a)** Total amount of time in a phase 3 days before failure.

**(b)** Total amount of time in a phase 3 hours before failure.

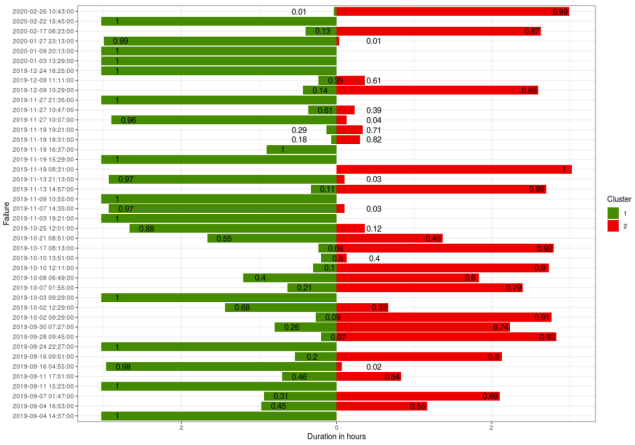
**(c)** Phases of the time series 3 days before failure.

**(d)** Phases of the time series 3 hours before failure.

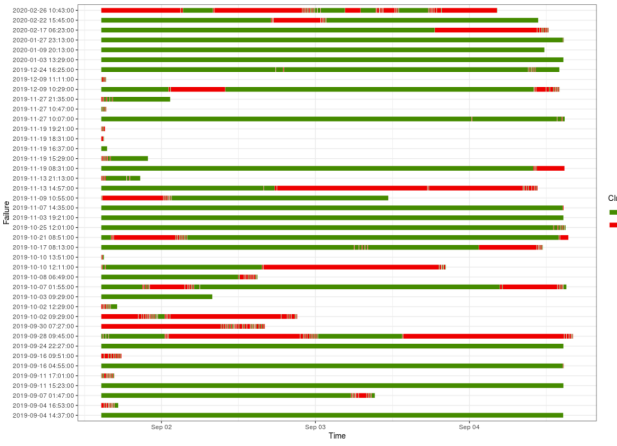
**Figure B.3:** Clusters or phases of each time series according to data set 4: pump 2.



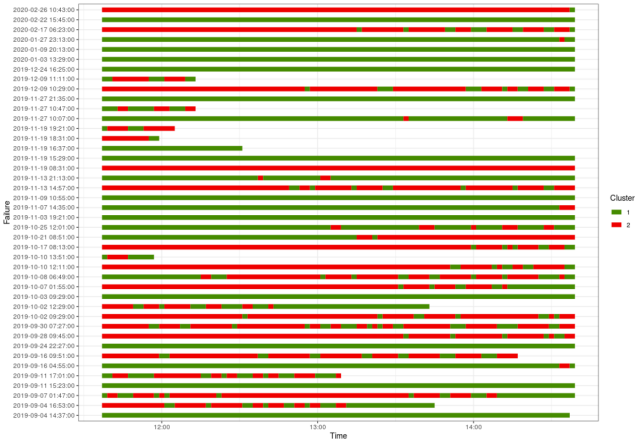

**(a)** Total amount of time in a phase 3 days before failure.




**(b)** Total amount of time in a phase 3 hours before failure.

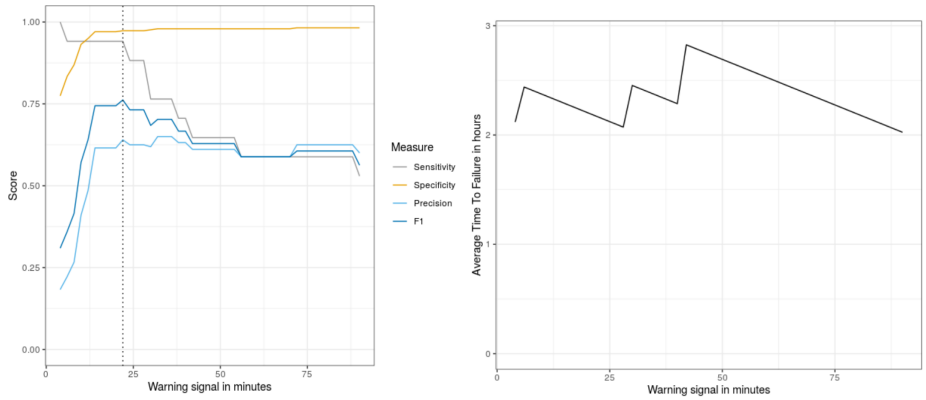



**(c)** Phases of the time series 3 days before failure.



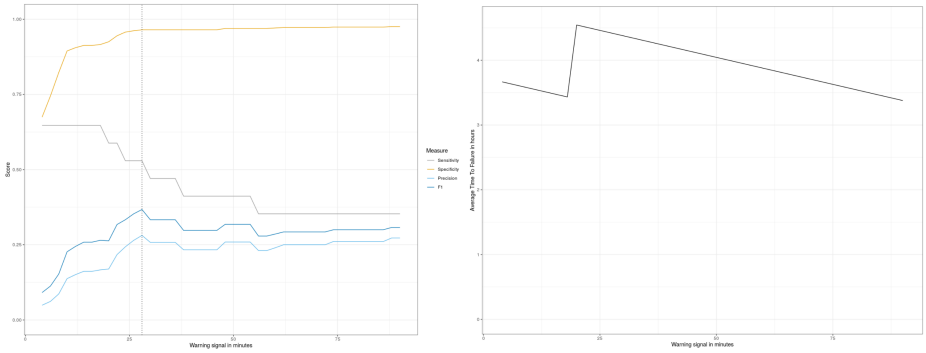

**(d)** Phases of the time series 3 hours before failure.

**Figure B.4:** Clusters or phases of each time series according to data set 5: expanded power consumption.




**(a)** Prediction scores data set 3: vibrations

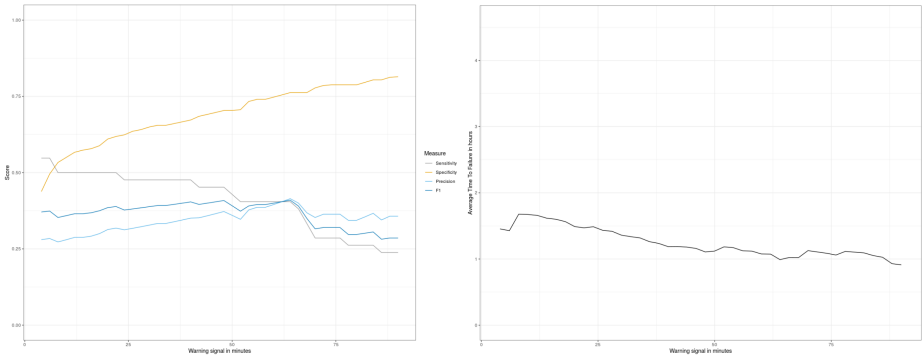
**(b)** Average Time to Failure data set 3: vibrations



**(c)** Prediction scores data set 4: vibrations + power consumption pump 2

**(d)** Average Time to Failure data set 4: vibrations+ power consumption pump 2

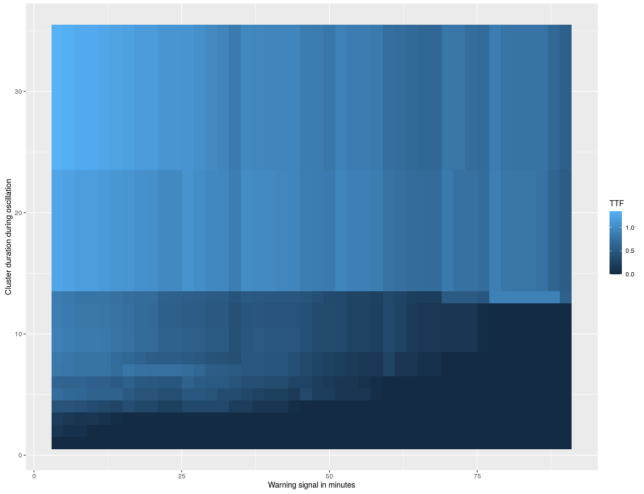
**Figure B.5:** Performance metrics for varying lengths of the warning signal. The dotted line shows the "optimal" length for a warning signal.



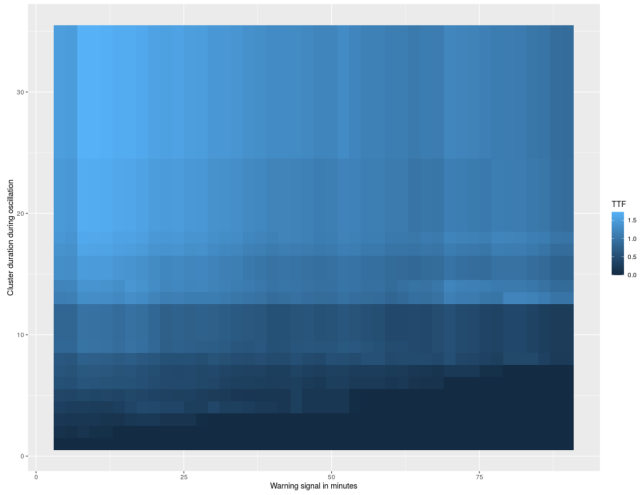

**(a)** Prediction scores data set 5: power consumption

**(b)** Average Time to Failure data set 5: power consumption

**Figure B.6:** Performance metrics for varying lengths of the warning signal when using data set 5 which contains a larger selection of power consumption data.




**(a)** Results data set 2: Power consumption




**(b)** Results data set 5: Power consumption (larger)

**Figure B.7:** The Average Time to Failure when varying the length of the warning signal and the cluster/phase duration when the system is oscillating.

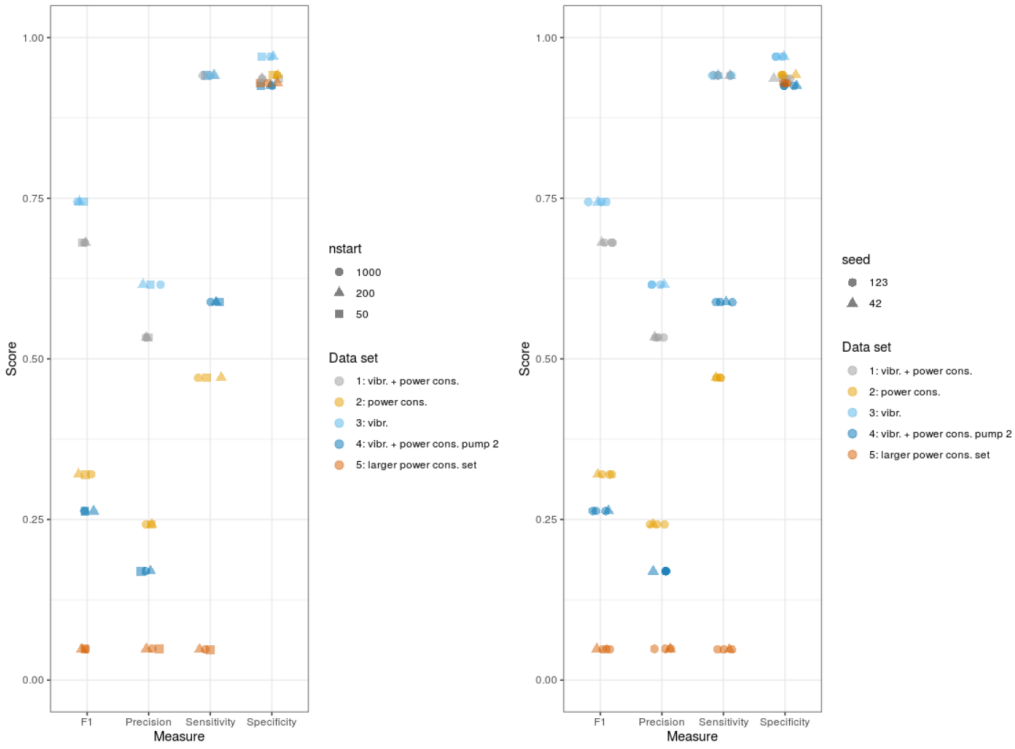



**Figure B.8:** Performance measures k-means clustering for varying values of the parameters *nstart* and *seed*.

# Bibliography


[1] R. Ahmad and S. Kamaruddin. "An overview of time-based and condition-based maintenance in industrial application". In: *Computers  Industrial Engineering* 63.1 (2012), pp. 135–149. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2012.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0360835212000484.

[2] I. Amihai, M. Chioua, R. Gitzel, A. M. Kotriwala, D. Pareschi, G. Sosale, and S. Subbiah. "Modeling Machine Health Using Gated Recurrent Units with Entity Embeddings and K-Means Clustering". In: *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. 2018, pp. 212–217. DOI: 10.1109/INDIN.2018.8472065.

[3] I. Amihai, R. Gitzel, A. M. Kotriwala, D. Pareschi, S. Subbiah, and G. Sosale. "An Industrial Case Study Using Vibration Data and Machine Learning to Predict Asset Health". In: *IEEE 20th Conference on Business Informatics* (2018), pp. 178–185. DOI: https://doi.org/10.1109/CBI.2018.00028.

[4] N. Amruthnath and T. Gupta. "A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance". In: *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*. 2018, pp. 355–361. DOI: 10.1109/IEA.2018.8387124.

[5] S. Biswal and G. Sabareesh. "Design and development of a wind turbine test rig for condition monitoring studies". In: *International Conference on Industrial Instrumentation and Control (ICIC)* (2015), pp. 891–896. DOI: https://doi.org/10.1109/IIC.2015.7150869.

[6] B. Boehmke and B. Greenwell. *Hands-On Machine Learning with R*. Chapman and Hall/CRC, 2020.

[7] S. Bruyn. "Predictive Maintenance for Sewer Systems using Machine Learning: A comparative study on the performance of three algorithms". graduation internship report. Faculty of Science , Vrije Universiteit, 2018. URL: https://beta.vu.nl/nl/Images/stageverslag-bruyn_tcm235-919760.pdf (visited on 06/19/2020).

[8] T. Carvalho, F. Soares, R. Vita, R. Francisco, J. Basto, and S. Alcalá. "A systematic literature review of machine learning methods applied to predictive maintenance". In: *Computers  Industrial Engineering* 137 (2019). DOI: https://doi.org/10.1016/j.cie.2019.106024.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357. DOI: https://doi.org/10.1613/jair.953.

[10] N. Davari, B. Veloso, G. d. A. Costa, P. M. Pereira, R. P. Ribeiro, and J. Gama. "A Survey on Data-Driven Predictive Maintenance for the Railway Industry". In: *Sensors* 21.17 (2021). ISSN: 1424-8220. DOI: 10.3390/s21175739. URL: https://www.mdpi.com/1424-8220/21/17/5739.

[11] G. K. Durbhaka and B. Selvaraj. "Predictive maintenance for wind turbine diagnostics using vibration signal analysis based on collaborative recommendation approach". In: *International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (2016). DOI: https://doi.org/10.1109/ICACCI.2016.7732316.

[12] M. Hoogendoorn and B. Funk. *Machine Learning for the Quantified Self. On the Art of Learning from Sensory Data*. Vol. 35. Springer Nature, 2018.

[13] A. K. Jardine, D. Lin, and D. Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance". In: *Mechanical Systems and Signal Processing* 20 (7 2006), pp. 1483–1510. DOI: https://doi.org/10.1016/j.ymssp.2005.09.012.

[14] M. Kuhn. *The caret Package*. 2019. URL: https://topepo.github.io/caret/index.html.

[15] M. Kuhn and K. Johnson. *Applied Predictive Modeling*. Springer, 2013.

[16] A. Kumar, R. Chinnam, and F. Tseng. "An HMM and polynomial regression based approach for remaining useful life and health state estimation of cutting tools". In: *Computers  Industrial Engineering* 128 (2019). DOI: https://doi.org/10.1016/j.cie.2018.05.017.

[17] N. Lunardon, G. Menardi, and N. Torelli. "ROSE: A Package for Binary Imbalanced Learning". In: *R Journal* 6 (1 2014), p. 79. DOI: https://doi.org/10.1613/jair.953.

[18] S. Schwendemann, Z. Amjad, and A. Sikora. "A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines". In: *Computers in Industry* 125 (2021), p. 103380. ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2020.103380. URL: https://www.sciencedirect.com/science/article/pii/S016636152030614X.

[19] O. Serradilla, E. Zugasti, J. Ramirez de Okariz, J. Rodriguez, and U. Zurutuza. "Adaptable and Explainable Predictive Maintenance: Semi-Supervised Deep Learning for Anomaly Detection and Diagnosis in Press Machine Data". In: *Applied Sciences* 11.16 (2021). ISSN: 2076-3417. DOI: 10.3390/app11167376. URL: https://www.mdpi.com/2076-3417/11/16/7376.

[20] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi. "Machine Learning for Predictive Maintenance: a Multiple Classifier Approach". In: *IEEE Transactions on Industrial Informatics* 11 (3 2015), pp. 812–820. DOI: https://doi.org/10.1109/TII.2014.2349359.

[21] E. Uhlmann, R. P. Pontes, C. Geisert, and E. Hohwieler. "Cluster identification of sensor data for predictive maintenance in a Selective Laser Melting machine tool". In: *Procedia Manufacturing* (24 2018), pp. 60–65. DOI: https://doi.org/10.1016/j.promfg.2018.06.009.

[22] Y. Wen, M. Fashiar Rahman, H. Xu, and T.-L. B. Tseng. "Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective". In: *Measurement* 187 (2022), p. 110276. ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2021.110276. URL: https://www.sciencedirect.com/science/article/pii/S0263224121011805.

[23] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li. "Predictive maintenance in the Industry 4.0: A systematic literature review". In: *Computers  Industrial Engineering* 150 (2020), p. 106889. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106889. URL: https://www.sciencedirect.com/science/article/pii/S0360835220305787.