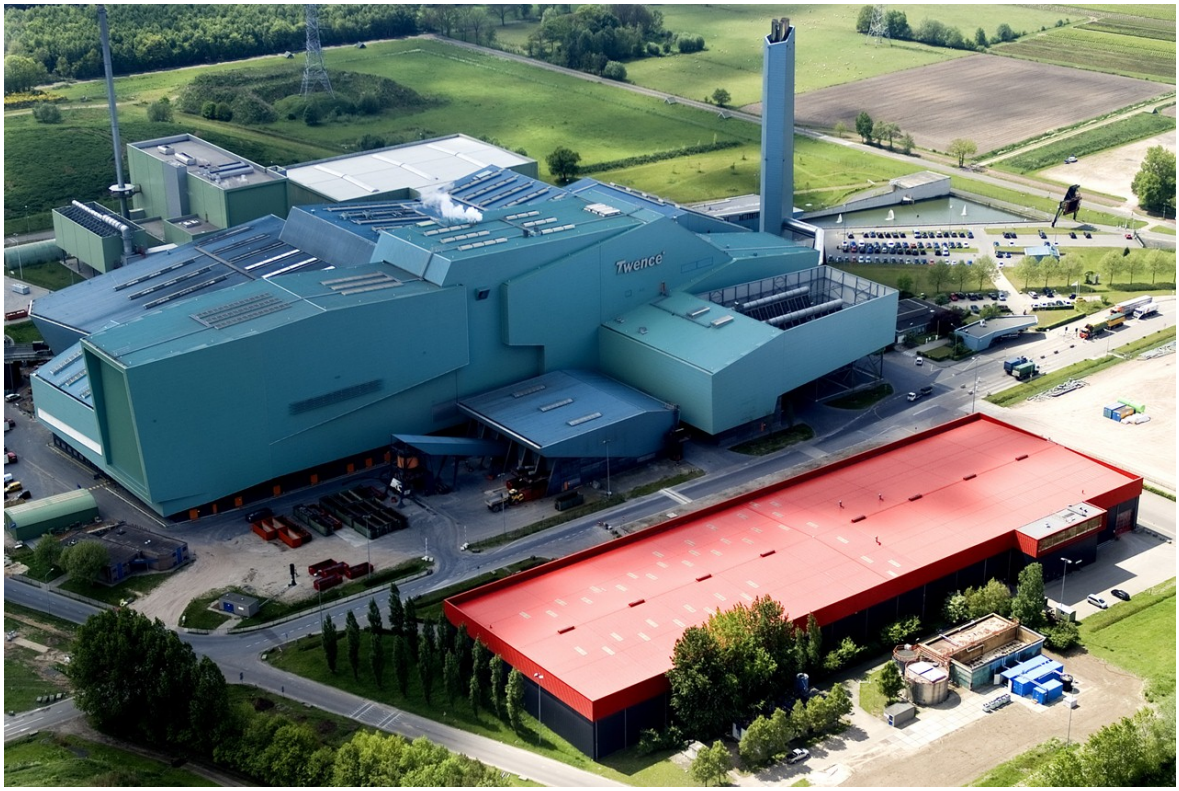
Analysing the Sodiumbicarbonate production of Twence

Maaike van de Ven, Jacco Wielaard

July 6, 2018



Contents

1 Introduction

Twence originally started as a landfill site in Hengelo, in the eastern part of the Netherlands. Over time they transformed into a recycling plant. Twence converts recyclable waste, biomass and non-reusable refuse-derived fuel into reusable component streams, compost, raw materials and energy. Due to the reliability of their supply of heat, steam and electricity they recently received the R1 status from the government [1]. This means they convert the processing heat created in their plants very efficiently and cleanly into energy carriers. Their goal is to recover as much energy and raw materials from waste as possible. [2]

One of their most recent projects is the capture of CO_2 from their Waste-To-Energy (WTE) plant. Their goal is reducing their CO_2 output, which is good for the environment, and making it available for useful application. There are various possibilities for the useful application of CO2, for instance as fertiliser in glasshouse horticulture, but also as a basic substance in the production of fuels. In this process the CO_2, that usually is emitted to the atmosphere, is removed from the flue gas with the use of a carbon dioxide scrubber. A scrubber is a device which absorbs CO_2. Besides being used to treat exhaust gases from industrial plants, scrubbers are also used to treat exhaled air in life support systems such as rebreathers. The obtained pure CO_2 stream is used to produce a sodium bicarbonate (SBC) slurry.

Instead of the conventional SBC flue gas scrubbing process, where dry particles are used, the slurry will be injected to remove acid components from the flue gas, before the gas is emitted to the atmosphere. Due to the implementation of this process the carbon footprint of the Twence installation is reduced. The new SBC plant should produce 8000 tons of sodium bicarbonate annually. For this 2000 ton CO_2 is captured per year from the flue gas. This is approximately 2-3% of the amount of CO_2 present in the flue gas and more than 90% of this amount will be used to produce the bicarbonate.

At Twence 1.6 tons of SBC is produced with 1 ton of soda, Na_2CO_3. This means fewer trucks have to drive and that is where most of the carbon reduction of this project is achieved. This means transportation of soda is more economic than transporting the SBC.

This rather ambitious process of making SBC has had multiple problems from the start, the most inconvenient one being the clogging of the cooling system. This cooling system is necessary to keep the process at a steady 45 °C. There have been many innovations regarding this project but it still clogs unexpectedly. If the system clogs the whole process is interrupted and the entire system needs to be cleaned, which is an expensive process. We were asked to look into this, and analyse when the system clogs.

Twence has collected a lot of data over the years. We will analyse the data to see if it is possible to predict when the cooling system will clog. We imagine the data will show specific features happening prior to a system failure. In that instance there are multiple ways to find that feature. First we need to identify the data from Twence. We will then look for a possible pattern indicating an oncoming clogging event using Fourier Transform methods. We will also try approach this problem of pattern finding with the use of machine learning, since

there may be patterns which are so subtle only a computer will be able to identify them.

For this we[1] will look into the exact process happening inside the reactors in Chapter 2. Then the process of the data extraction and interpretation will be looked at in Chapter 3. Then Fourier Transform methods will be shortly explained and applied in Chapter 4. Since that might prove insufficient the need for machine learning was found necessary. Since machine learning is a relatively new field in mathematics, a short explanation of this technique and the different types of machine learning is provided in Section 5.1. After this we will look into what can be done with this new technique in Section 5.2.

[1]Main contribution of Chapter 1, 2, 3 and 4 done by Jacco Wielaard. Main contribution of Chapter 5 and 6 done by Maaike van de Ven.

2 Process

To better understand the system we took part in a tour at Twence to learn about their state-of-the-art WTE plant. This helped us to understand and interpret the data provided to us more effectively. The process of making SBC will now be described in greater detail.

Since, as explained earlier, it is more economical to transport soda, Na_2CO_3, Twence tries to make as much SBC from soda to use in their flue gas scrubbing installation as possible. Sodium Bicarbonate reacts according to the following reaction mechanism,

$$Na_2CO_3 + CO_2 + H_2O \longrightarrow 2\,NaHCO_3. \tag{1}$$

Since the soda is dissolved in water in the first step of the process, the reaction mechanism becomes,

$$Na_2CO_3(aq) + CO_2 \longrightarrow 2\,NaHCO_3. \tag{2}$$

What Twence does to produce SBC is guiding an excess amount of CO_2 through the SBC reactors. A careful analysis has been done to determine the optimum reaction conditions of this process. This is at a temperature of 45.0 °C and a pH value of 8.5 [3].

Since heat is generated in this process the mixture needs to be cooled to keep the temperature at 45 °C. This is done in separate cooling installations, one for each SBC reactor. In this cooling installation the SBC mixture is pumped through seven consecutive pipes. In each of these pipes the SBC mixture is led trough small tubes, and water flows in the opposite direction on the outside of the small tubes to cool the mixture.

The reason why the cooling water flows in the opposite direction is that in this way more heat can be exchanged in the same pipe length, see Figure 1. The upper diagram shows the heat exchange if the cooling liquid and the mixture would both flow in the same direction (cocurrent). The lower diagram shows the heat exchange if the cooling liquid and the mixture flow in opposite directions (countercurrent). This shows that countercurrent flow is more effective at exchanging heat in the same space.

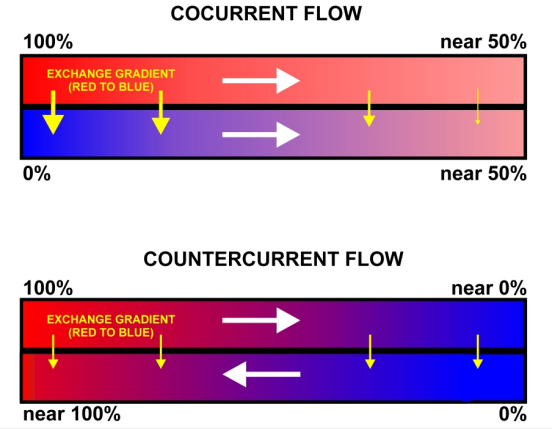


Figure 1: *A comparison between cocurrent flow (in the same direction) and countercurrent flow (in the opposite direction) [4].*

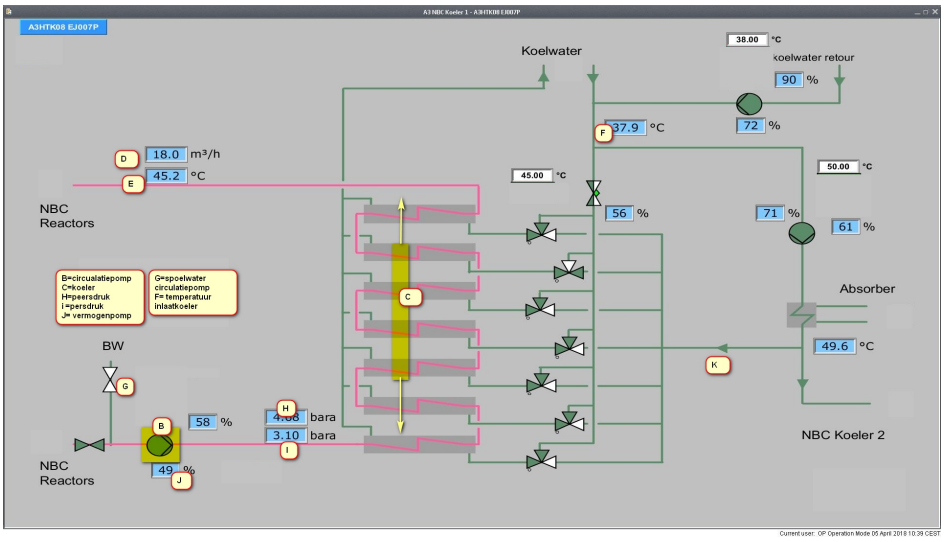


Figure 2: *Schematic view of one cooling installation. In the middle the cooling installation itself. Shown in pink is the flow of the mixture. On the left side is the input from the SBC (=NBC) reactor, and the output to the SBC reactor. On the right side is the input of the hot water flow to heat one of seven cooling tubes.*

During the cooling process, the SBC product may crystallise in the small tubes. The crystallisation silts up the small tubes, and with that prevents a continuous flow through the cooling installation. When the tubes are clogged, the system has to be shut down. The cooling installation then needs to be opened up and everything has to be cleaned thoroughly with water. The shutting down costs money and interrupts the process, which is what Twence wants to prevent.

Multiple solutions have been tried by Twence to arrive at a continuous process. One of the workarounds engineered was heating one of the seven cooling pipes a little bit, just enough to dissolve the built up crystals. All the while still cooling the other six pipes to keep the temperature constant. Heating, and thereby cleaning one of the pipes in this way means the process can run longer. It is just a workaround though, not the intended solution, since in the end the system still clogs, but more slowly.

Something else that has helped to extend the time between cleaning tasks is flushing the cooling installation. Pumping clean water through the cooling installation together with the SBC mixture for a short period of time, to flush out the built up crystallisation. This clears a lot of the crystallisation that has built up over time. This is also not a full solution since some crystallisation is still left behind, and it subsequently builds up quicker. Also the SBC mixture now consists of more water, which has a negative effect on the reaction process.

The process as it happens at Twence has been described, together with the crystallisation problem. In the next chapter we will look at the data of Twence.

3 Data

Twence collects a lot of data of all their systems. Since this data is tracked over long periods of time, analysing this information might give the solution to the problem of the unwanted crystallisation in the cooling installation. Handling all this data is a challenging task in itself, and consists of multiple steps. In this Section the process of extracting the data and taking a first look at it will be described.

The first step in this process is collecting the data from the system. This was done by Marc van Sonderen from Twence. He sent us a first sample of Excel files to work with. This first sample consisted of readings from different parts of the two SBC reactors and the cooling installation.

After this data is collected, it needs to be made available so it can be processed. MATLAB was chosen for this, and data was read using the easy to use XLSREAD function. It quickly became clear that the location of the data in the files, and the amount of datapoints was not entirely consistent across all the files. Some of the files had values of 10 minute averages, others of 1 minute averages. The start time and the end time also differed for some files. Since something went wrong with exporting the data of the first set and not the expected end times were used, we very quickly got a second set. This second set had some more consistency, but it differed once again at several key points from the first sample.

At that point we already had data spread over eighteen Excel files. These files differed so much, it was chosen to write a function to make it easier to automatically import the data provided by Twence into a usable format. The goal was to have a function that automatically imported all the needed information to MATLAB, given the file name. This function went through many iterations to eventually end up in its final form as specified in Appendix I. It determines the number of variables, which is then used to calculate the range of the values.

Since the date format specified in Excel cannot be read into Matlab without manual interference, a solution needed to be found. Since the start time, and the end time were readable, and with some tricks the interval time could be calculated, it was possible to make a time series. The size of this time interval then defines the maximum range for the values to be in, tremendously speeding up the importing function from almost 30 seconds to just 2.5 seconds. Since it is easier to interpret data when you have an idea what the data represents and entering the legend manually every time quickly became repetitive, these entries were also imported, and then combined, to also be outputted by the function to be displayed in a neat way.

The second step is actually doing something with this data. Since it is helpful to get a feeling for the data by plotting it, this was the first priority. One of the first datasets consisted of measurements from SBC reactor 1 from April 1st to April 7th, see Figure 3. Here you can see very clearly that sometimes the SBC production (pink line) dips down to zero. For example in the afternoon of the 4th, the system clogged up and had to be shut down to be cleaned.

Striking features are the spikes of the CO_2 supply. While the pH and temperature seem to be largely constant, the CO_2 supply spikes very consistently every hour. This does not seem to have any effect on the other values though. These spikes come from the supply of CO_2 gas of the flue gas installation.

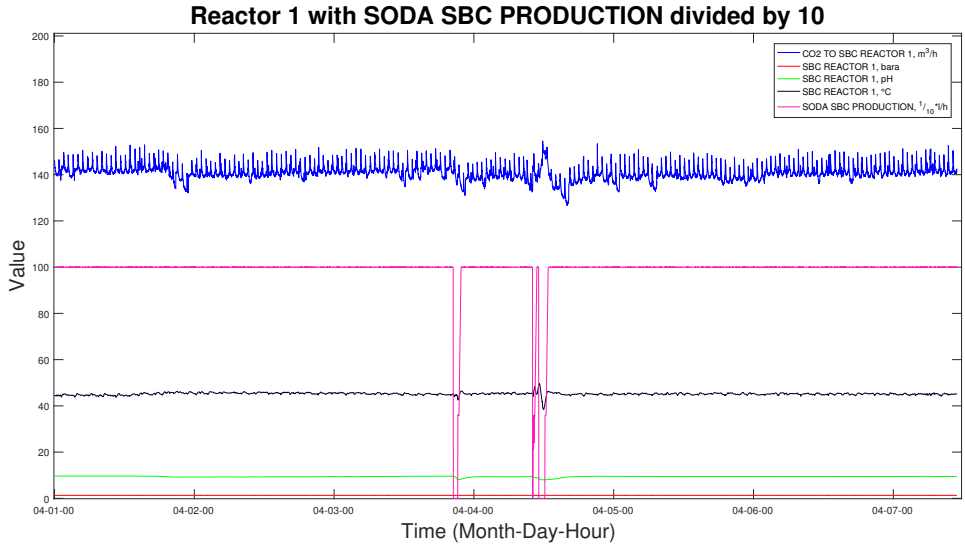


Figure 3: *The first dataset of Reactor 1*

A problem with the data that is not immediately apparent, is that sometimes values are missing. MATLAB reads over this when importing the data and writes NaN (Not a Number) in the matrix. For plotting purposes this is not a problem, there is just nothing plotted at that moment in time. When calculations are done with a NaN value an error is given by MATLAB. This happens for example when an average is calculated over the entire dataset with a NaN value somewhere in the set. For some of these an easy solution is already there, for the average this is the MATLAB function NANMEAN, which calculates the average without the NaN values. For other calculations this is not as easy, for example the Fourier Transform used in 4.1. This is something to keep in mind when working with MATLAB.

Using these techniques a big part of the data could be made available for processing in MATLAB. There were a couple of data files that consisted of data from flushing valves. The flushing valves given are the ones closest to the SBC reactors. In these files were just the names of the valves and times when they were turned on and off. Since this seemed like relevant information, time was spent to make this data available to MATLAB as well. This required a large amount of string manipulation to get the result in Figure 4. Every event, i.e., the pump is turned on, or turned off, is plotted against the number of the pump. Almost always, when a pump is turned on, it is turned off within ten seconds, a short flush. This means that in Figure 4 the circles for opening the valve and closing the valve are overlapping. Thus every circle corresponds to an entire flush.

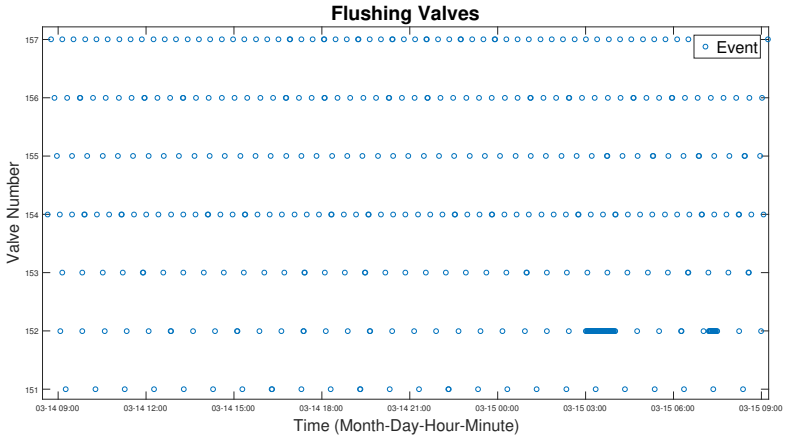


Figure 4: *The seven flushing valves closest to the SBC reactors. Every blue circle corresponds to a flush of the corresponding valve*

All of these series of dots seem to have an equal amount of space between them. Which means that there is an equal amount of time between flushes. The time on which this is not true is on the 15th of March. There valve number 152 is flushed repeatedly in a very short time period. This means the operators deemed it necessary to flush the system repeatedly with this particular valve. It indicates there might be something wrong with the system.

Since all of these times seem to be so consistent it could be useful to compare the valves between themselves. If there is a peculiarity in the data, there could be something wrong with the system. To have something to compare to, some statistics were extracted from the dataset.

```
151 is turned on 348 times, for an average of 00:01:07.187
152 is turned on 569 times, for an average of 00:00:20.511
153 is turned on 512 times, for an average of 00:00:09.574
154 is turned on 836 times, for an average of 00:00:09.788
155 is turned on 674 times, for an average of 00:00:09.583
156 is turned on 801 times, for an average of 00:00:12.771
157 is turned on 907 times, for an average of 00:00:09.572
```

It seems weird that valve number 151 is open for so long on average, so a better look at the data is necessary. For this, boxplots were deemed an easy way to visualise this data. This is because outliers are easily visible in a boxplot. In Figure 5 it is visible that valves 151, 152, and 154 have several outliers, where valve 151 has very high outliers. The other valve opening times are very consistently grouped around their medians. It would be useful to extract the time when these outliers happened. These times can then be compared with the SBC production at that time, to see if there is any correlation between these values.

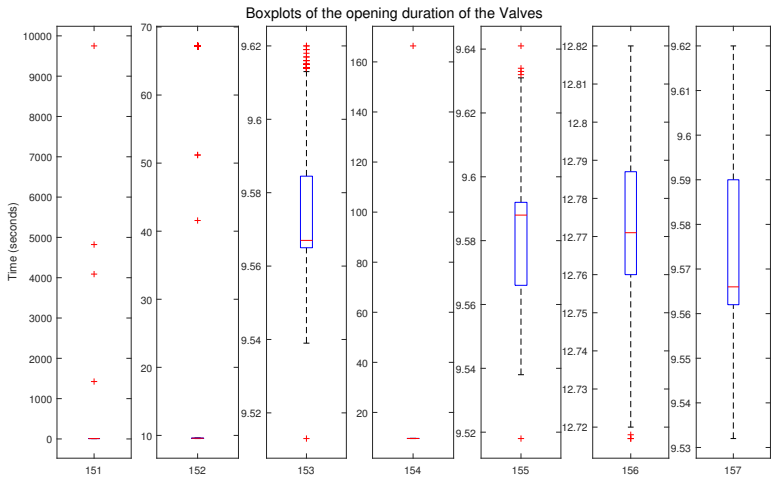


Figure 5: *Boxplots of the opening duration of different valves.*

The third step is to compare the data of where the system clogs to normal operation. For this it is necessary to have information on "normal" system data. After asking for this, Marc van Sonderen send us big datasets with hourly averages of January, February, March and April. This was again with different measurements than we already had, and thus again needed manual comparing to earlier datasets. This was where our *Import* function really started to pay off, since that made plotting the entire dataset a lot easier.

Something else that was deemed interesting, and was thus requested data from, was the valve in the cooling installation. On the second visit to Twence it was found that the operators were flushing when the pressure in the cooling installation became too high. So if we could decipher the information from this flushing valve, then this might have interesting information.

Sadly the values from this valve were garbled in the excel file, but this was the best we could get from Twence. After some manual work this file was made readable. The value for the earlier valves consisted of 0 (open) or 1 (closed). The values for these particular valves (two valves, one for each cooling installation), are 1 (open) or 0 (closed), which means rewriting the MATLAB program to also read this. When this was done it was possible to get the statistics in almost the same way as before. Number 57 is the valve of the cooling installation of SBC reactor 1, and number 58 is the valve of the cooling installation of SBC reactor 2.

```
57 is turned on 2336 times, for an average of 00:00:45.848
58 is turned on 1292 times, for an average of 00:01:07.057
```

Valve number 57 is turned open nearly twice as much as valve number 58. This has been confirmed by Twence, there are more problems with reactor 1 than with reactor 2.

Since all the data is now available in MATLAB, and most of the problems that have been encountered have been resolved, it is now possible to start with analysing the data to find the possible events that triggers the clogging of the cooling installation.

4 Signal Analysis

It is to be expected that something triggers the clogging, an event, or a certain series of events. There are multiple ways to investigate this. A possibility is Fourier analysis, decomposing the signal and making a prediction based on that. If the real value deviates too much from this prediction, there is a chance the system will clog soon. Another thing to check is correlation. If a signal has the same peaks and dips as the production signal, they are correlated. If signals are correlated, it is possible to say something about one signal using an other. A last thing to check, is the valve data, mostly the valves from the cooling installation themself, since those seem to be the most relevant in the process.

4.1 Fourier Analysis

Any periodic signal can be represented by a linear sum of sinusoidal waves, as was shown by Fourier in 1822 [5]. For example the signal "CO2 TO SBC REACTOR 1", the signal with the hourly spikes in Figure 3, could be approximated as periodic. Then it should be built up of such sinusoidal waves.

Using one of MATLAB's tools, FFT (Fast Fourier Transform) it is possible to transform a signal from the time domain to the frequency domain. Essentially decomposing a signal in all of its sinusoidal wave components. This is visualised in Figure 6.

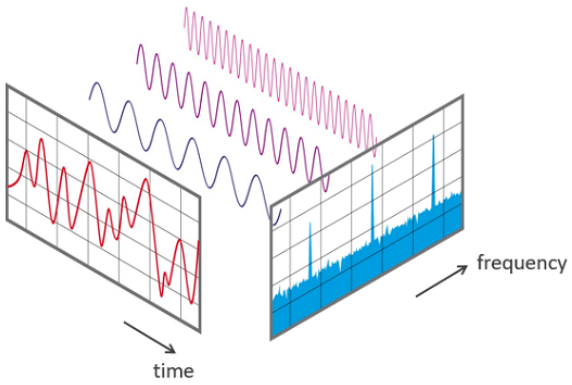


Figure 6: *View of a signal in the time and frequency domain [6].*

Shifting the signal "CO2 TO SBC REACTOR 1" to the frequency domain using FFT we get its spectrum as shown in Figure 7. Very clearly visible is the wave component with a frequency of 1.2 hours, and the multiples of this frequency.

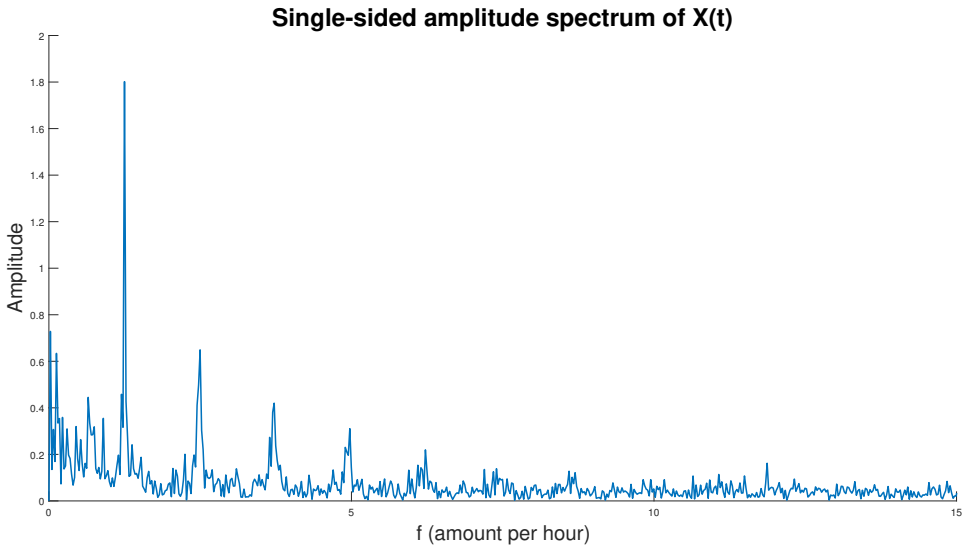


Figure 7: *Single-sided amplitude spectrum of the signal "CO_2 TO SBC REACTOR 1".*

Taking the location of the peaks from highest to lowest peak, we get a list of the most important frequencies, $freq_i$. Using these frequencies a least squares algorithm is used to get a best fit for each frequency. This minimisation problem is defined as,

$$\min_{x,\ y} \sum_{i=1}^{N} \sum_{k=1}^{K} \left[\text{data}_k - x \cdot \sin(2\pi \cdot t_k \cdot freq_i) - y \cdot \cos(2\pi \cdot t_k \cdot freq_i)\right]^2, \tag{3}$$

where data_k are the data points, and t_k are the time points. N is the amount of frequencies, K is the total amount of time points. This minimisation is then used to find the amplitudes, x and y, belonging to a certain frequency such that the combination of all the sines and cosines is as close to the real signal as possible.

Then the sines and cosines and their amplitudes are combined, this is visible in Figure 8. In blue is the original data, in orange is the prediction made using the FFT algorithm and the least squares method.

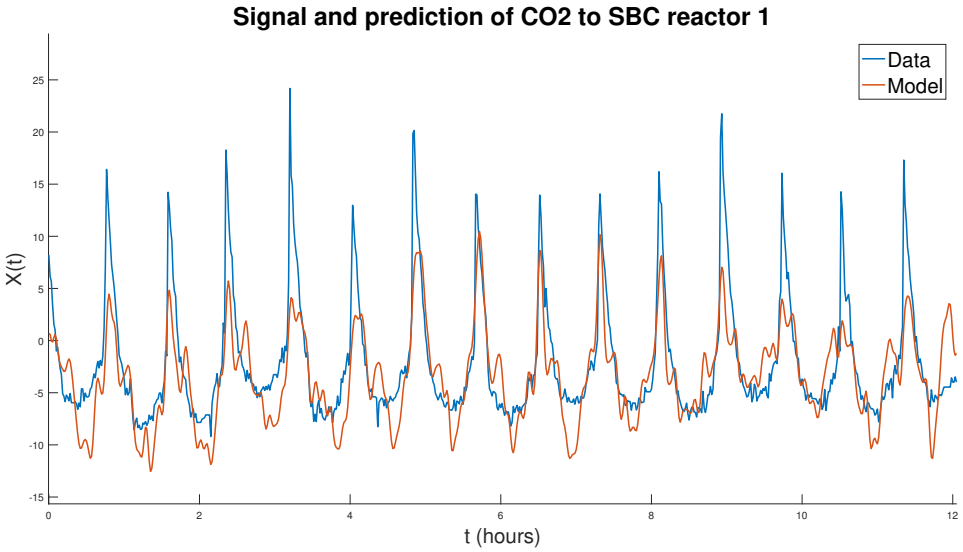


Figure 8: *Signal and prediction of "CO2 TO SBC REACTOR 1". The prediction was made using the 100 highest peaks of Figure 7.*

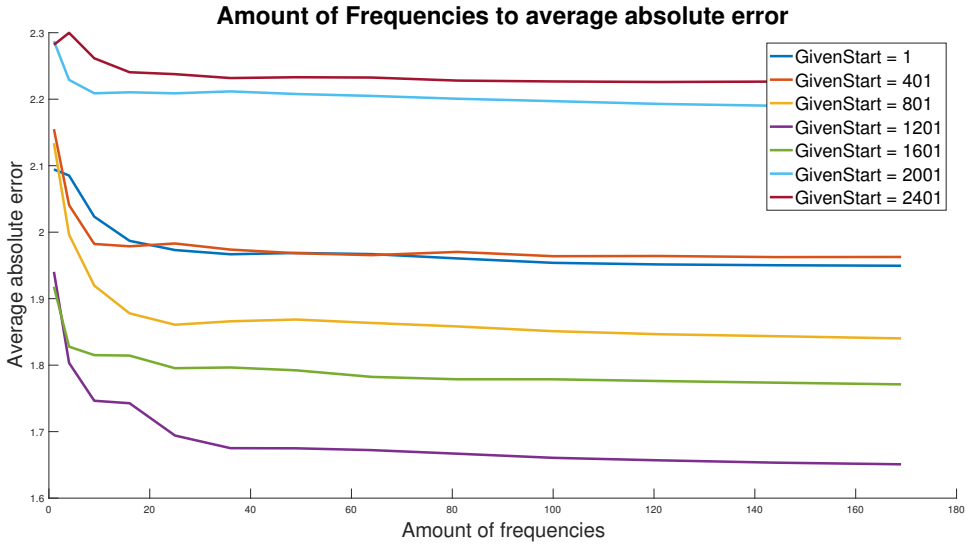


Figure 9: *The average absolute error depending on the amount of frequencies and the start time of the data when predicting 20 hours into the future. The absolute error is the difference between the data and the model and prediction.*

To determine how many frequencies are needed to get the best model, different amounts of frequencies and different start times were tested. This is visible in Figure 9. The error is calculated as the sum of the absolute difference between the data and the model. It seems

that for this signal, predicting from a start time of 1200 minutes, twenty hours, gives the best result. At about forty frequencies adding extra frequencies gives diminishing returns. It was thus chosen to model the prediction using forty frequencies.

To check the accuracy of the model, a prediction was made using data of hour 25 to hour 65 of the set, since this gave the smallest error according to our program. This prediction was then compared to the real data, see Figure 10. In green the data, in orange the model, and in blue the prediction. As could be seen in Figure 3, the production dips for the first time at the end of April 3rd and a second time in the afternoon of April 4th. This corresponds to hour 70 and hour 87 respectively. Since this is where the system is shut down, and the CO_2 has a dip, this is where our interest is.

At hour 70, the first dip happens, this is where the prediction based on our model goes wrong. Since the model has no knowledge of earlier dips, it is not to be expected that it can predict one. The same holds for the second dip at hour 85. This might thus indeed be useful to keep track of, since if the real CO_2 signal deviates to much from the expected value there is something wrong with the system. Upon closer look it seems to be the other way around, when the system is shut down the CO_2 deviates. It seems thus difficult to predict something on the basis of the CO_2 signal.

Further investigation with the large dataset determined that most of the smaller periodicity's disappeared when taking hourly averages. In this large dataset almost all of the signals are constant. These straight lines are not periodic, thus FFT is not a good way to find deviations on a large scale, although the CO_2 signal seemed interesting.

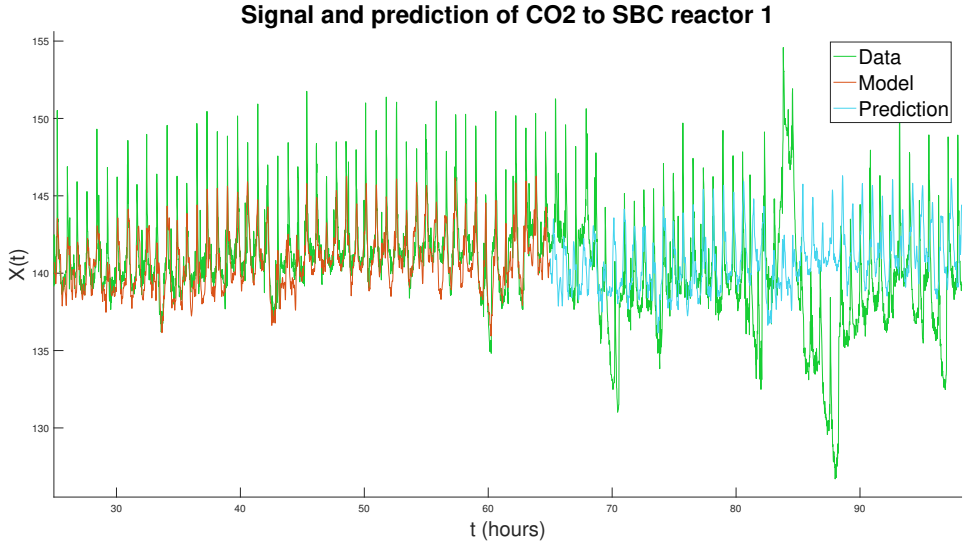


Figure 10: *The signal "CO_2 to SBC reactor 1 in green". The model based on data from hour 25 to hour 65 in orange and using forty frequencies. The prediction in blue.*

4.2 Correlation Coefficients

To predict what the SBC production is going to be, it is interesting to look at all the other signals and see if another signal has drops and peaks at (almost) the same time. If that holds, then this other signal, or these other signals might be the key to predict the SBC production.

For this Pearson's correlation coefficient is used [7]. This is for 2 signals, A and B, defined as

$$\rho(A, B) = \frac{\text{cov}(A, B)}{\sigma_A \sigma_B}, \tag{4}$$

where

$$\text{cov}(A, B) = E[(A - \mu_A)(B - \mu_B)], \tag{5}$$

is the covariance of A and B, μ_A and σ_A are the mean and standard deviation of A, respectively, and μ_B and σ_B are the mean and standard deviation of B.

The Pearson's correlation coefficient is always between -1 and 1. The closer the value is to 0 the less correlation there is between the two signals. For interpreting these values the convention in Table 1 is used [8].

Table 1: *Convention for interpreting the size of a correlation coefficient*

.90 to 1.00 (-.90 to -1.00)	Very high positive (negative) correlation
.70 to .90 (-.70 to -.90)	High positive (negative) correlation
.50 to .70 (-.50 to -.70)	Moderate positive (negative) correlation
.30 to .50 (-.30 to -.50)	Low positive (negative) correlation
.00 to .30 (.00 to -.30)	Negligible correlation

Using Equation 4 the correlation coefficients between the signals are calculated, see Figure 18 in Appendix II. The greener the value is, the higher the correlation coefficient. The values on the diagonal are of course equal to one, a signal is always directly correlated with itself. Some values have a high correlation, for example "SBC REACTOR 2, bara" and "SBC REACTOR 1, bara". Looking at the names of these signals, it seems to be expected that these values are correlated. On the other hand, "SBC REACTOR 2, ph" and "SBC REACTOR 1, ph" have a very low correlation while one could expect this value to be highly correlated as well. This can be explained by the fact that reactor 1 has more problems in the cooling installation and thus needs to be flushed more often. Flushing a cooling installation has a negative effect on the pH value of the reactor, but not on the other reactor.

The most important correlation coefficients in this case would seem to be the correlation coefficients linked to "SODA NBC PRODUCTION, l/h". The highest correlation coefficients for this signal are "MIXER SBC REACTOR 1, %" and "SBC REACTOR 1, ph", although these are still low. The use of correlation coefficients seems to lead nowhere, since none of the signals have a high correlation with the production signal. This means that predicting

the production using other signals seems impossible at this moment. In Chapter 5 machine learning will be applied to this problem to take a deeper look at this problem.

4.3 Valve data

Something else that seemed relevant to check, was the data of the cooling installation valves. In Figure 11 the events of the valve data are plotted together with the data of reactor 1. The data is divided by its maximum to get a better overview in the plot. As is visible in this figure, the time between flushes of the system decreases shortly before the production is shut down. The circles are then located closer together.

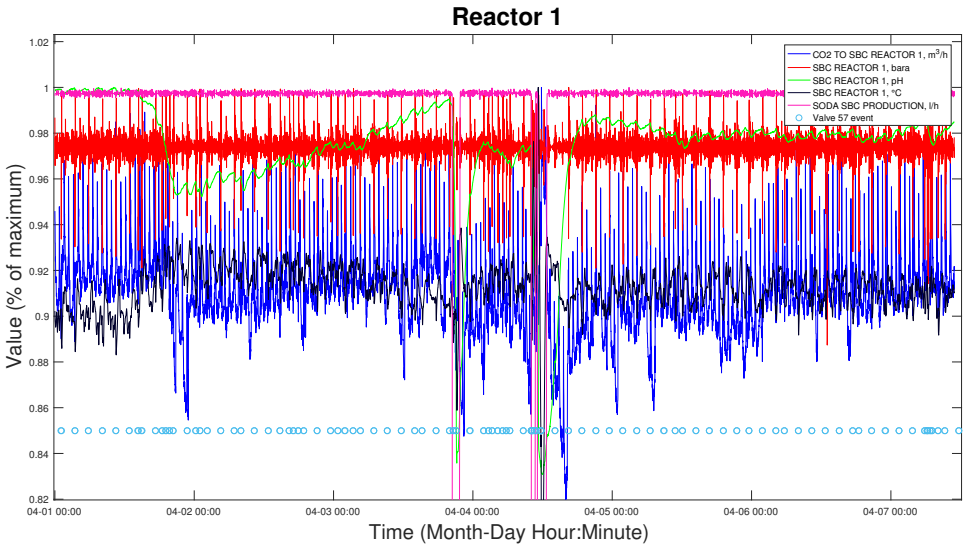


Figure 11: *The first dataset of reactor 1 where all signals are normalised. Also the events of valve 57, the valve of the cooling installation of reactor 1 is included.*

To further investigate this, the large dataset was plotted together with the valve data of the cooling installations. Valve number 57 is the valve in the cooling installation of reactor 1, number 58 is the valve in the cooling installation of reactor 2. This is visible in Figure 12. The result is best visible with valve number 57, close to a dip in the production, the valve is opened and closed more often. It seems that indeed, the system is flushed more often closer to a shut down. It is important to remember that correlation does not imply causation, but this warrants a closer look.

Upon taking this closer look it seems that this effect is caused by the data not aligning correctly, and the flushing in quick succesion is not the cause of the clogging, but the result.

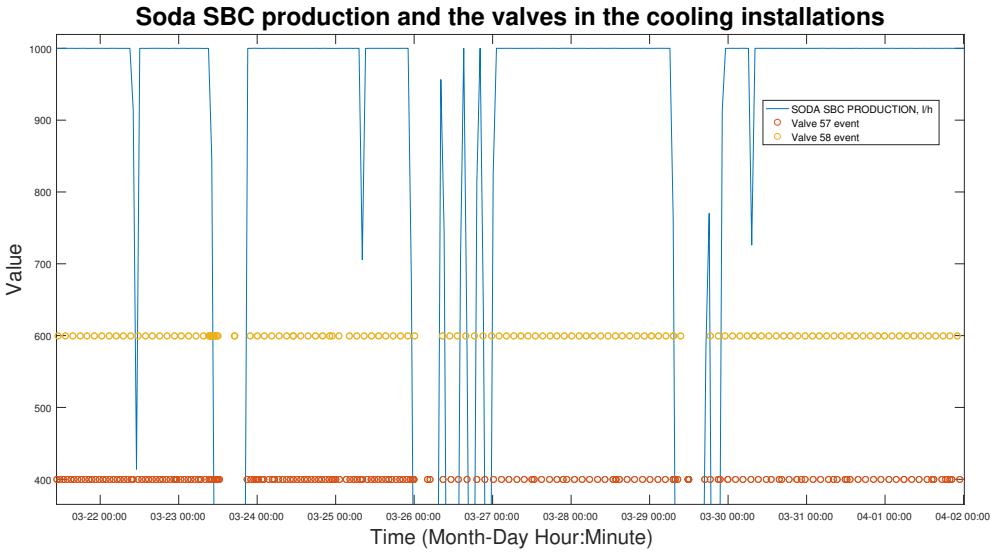


Figure 12: *Part of the large dataset of the SBC production and both of the valves in the cooling installation. Valve number 57 is of the cooling installation of reactor 1, valve number 58 is of the cooling installation of reactor 2.*

5 Machine Learning

To see if there are any relations in the data that could not be explained with signal analysis we look for relations in the data using machine learning. In Section 5.1 the general idea of machine learning and the different types of machine learning will be explained. In Section 5.2 the application for Twence will be elaborated.

5.1 What is Machine Learning

Machine learning is a term that is used quite often nowadays. After the development of the computer, algorithms like neural networks were made that used this newly developed computer to, for example, identify written numbers. However, in the nineties this research was not really popular anymore. Nonetheless, in 2006 Geoffrey Hinton changed the use of neural networks to a brute force approach [9]. To use a brute force approach a lot of data is needed. In the first years of neural networks there was not enough data available, but with the further development of the computer more data became available. With this data neural networks were useful again and got famous under the names deep learning or machine learning.

To get basic knowledge about machine learning, I watched a YouTube tutorial [10]. The YouTube tutorial explains how machine learning works with scikit-learn [11] in Python. The newly learned theory is practised using the iris dataset [12]. The iris dataset is created in 1936 and contains measurements of three species of Iris flowers. Measurements are made for fifty flowers of each species, so hundred and fifty flowers in total. Even though this dataset is quite old, it is often used to get to know machine learning or testing algorithms. To get an idea of what possibilities there are with machine learning and what should be used in which situation, the cheat-sheet from scikit-learn [13] and the documentation of MATLAB is used.

Basically there are three main types of machine learning; supervised learning, unsupervised learning and reinforcement learning. Supervised and unsupervised learning will be briefly explained. Reinforcement learning is harder to understand, implement and explain. Also, for reinforcement learning there is no standard implementation in MATLAB or PYTHON available. Therefore it was not applied and thus it will not be explained.
Before explaining supervised and unsupervised learning some terms related to machine learning will be defined. First of all the target, this is the measurement or type that you want to determine. Determining this will be done based on data, which is the information available except the target. For example with the Iris dataset the species of flower is the target and the measurements from the sepals and petals are the data. A group of data and a target that belong together is called an observation. When the target and the data are available an estimator can be chosen. The estimator is the type of machine learning model used. With this estimator the model can be trained based on (a part of) the available data. Training means that for each datapoint it will be determined to which target it belongs. This way a model is made that can be used to determine the target of not yet used data. The new data

can be data of which the target is not yet known or data of which the target is already known. In the case the target is already known, this data is often used to test or verify the model. Testing is done by comparing the prediction with the real target. Now that the terms target, data, observation, estimator, training and testing are explained supervised and unsupervised learning will be explained.

5.1.1 Supervised learning

Supervised learning uses current available data of which the targets are already known. The Iris dataset mentioned in Section 5.1 is an example of a dataset where supervised learning can be applied. The target can be a class, such as the species of the iris, but can also be a continuous number. If the target is in a class it is called classification learning and if the target is a continuous number it is called regression learning. The idea of supervised learning is that data and targets are used to train a model that is then used to predict the target based on other data.

k-nearest-neighbours classification

There are multiple estimators for supervised classification learning, one of them is the k-nearest-neighbours classification method. With this estimator it is easy to understand what happens and to get a general idea of supervised learning. The k-nearest-neighbours estimator uses the euclidean norm to determine how far a point is located from the known datapoints. Say $\vec{x_i}$ is the data at time i, y_i is the class at time i. $\vec{x}$ is the data for which the target has to be determined. Now determine $d_i = |\vec{x_i} - \vec{x}|$ for all i. Next select the k lowest d_i, these correspond with the k nearest neighbours. Next the classes corresponding with these k datapoints are determined and it is determined which class is most common. This can be done by just counting, or by adding a weight function depending on d_i.
Using what is explained above, the target of $\vec{x_i}$ is predicted. This method can be influenced by choosing k and choosing the weight function for determining the target. Especially k should be chosen carefully. If k is to big it can give the wrong idea of the situation, while if k is too small one outlier changes your model. If for all points in a certain range the class is determined as described above a map can be made as in Figure 13. For every new datapoint of which the class needs to be determined, the model will check what class this should be according to what it has learned before. Figure 13 is an example for a dataset with two variables and three classes. Of course more variables and classes are also possible, but more variables will make it harder to give a visualisation.

Next to the k-nearest-neighbour estimator explained above, there are multiply supervised learning estimators available. Next to the classification estimators there are also regression estimators, these predict a value instead of a class. The following estimators are available in Matlab:

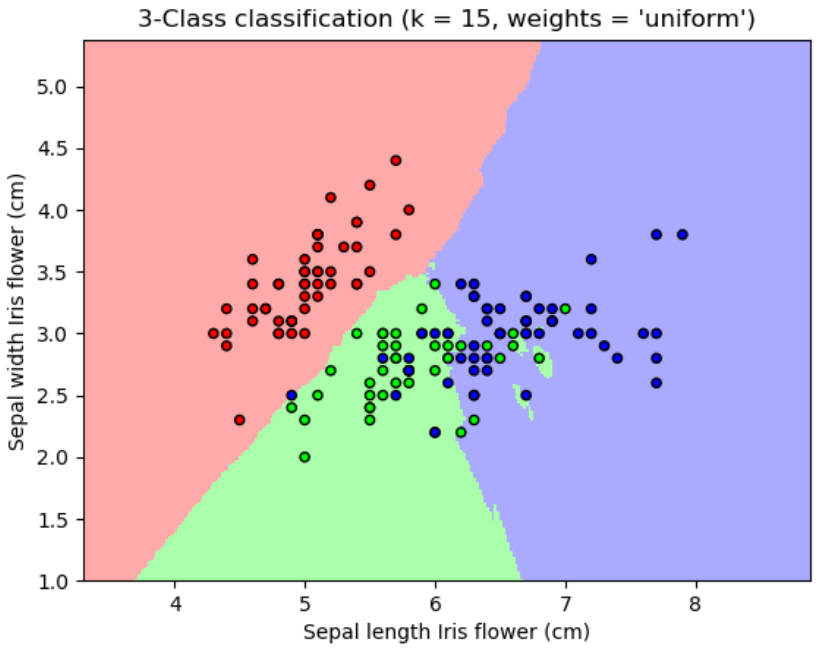


Figure 13: *Classification Iris dataset with k-nearest-neighbour method, k=15, uniform weights (the k nearest neighbours are counted equal independent of distance from point to determine class). The dots are the actual 150 measurements with a different colour for each class (if two measurements are close to each other they are one dot). The background colour is the class that will be predicted for a flower with those measurements [14]*

- Classification estimators
  - Support vector machines
  - Neural networks
  - Naïve Bayes classifier
  - Decision trees
  - Discriminant analysis
  - Nearest neighbours
- Regression estimators
  - Linear regression
  - Nonlinear regression
  - Generalised linear models
  - Decision trees
  - Neural networks

From the estimators mentioned above, the classification and regression decision tree will be explained, since these will be used later on.

Decision tree

Decision trees can be used for both categorisation supervised learning and regression supervised learning. The idea is that the data is split based on the available information. These splits are tracked by making a decision tree. For classification this will be done till all data following the same splits have the same target. So every leaf only contains one class if all observations from the training set are split using the tree. Every new datapoint finds a leaf in the tree according to the splits. The predicted class will be the class that belongs to that leaf of the tree. An example can be seen in Figure 14. In this figure there is decided based on the age and whether someone has a drivers license if this person is allowed to drive. If it is known whether someone has a drivers license and how old this person is, it can be determined if this person is allowed to drive by following the choices of the tree. This is just an easy example and can be extended when more data is available. The idea of using such a tree in machine learning is that choices are made in an optimal way to split the data into different classes using as few steps as possible. This way the model is trained and a tree is made. For the tree it is important that each observation will end up in exactly one node. If a certain observation has the option of ending up in no leaf you can get problems, since you want to be able to make a prediction for all observations. Next to that an observation should not end up in more than one leaf, since then no proper prediction can be made.
If the decision tree is used for regression, the data is again split and a tree representing these splits is made. However it is quite hard to end up with a leaf only having one class. Therefore the data will be split until every leaf contains one observation instead of one class. Next to that the approach is the same.

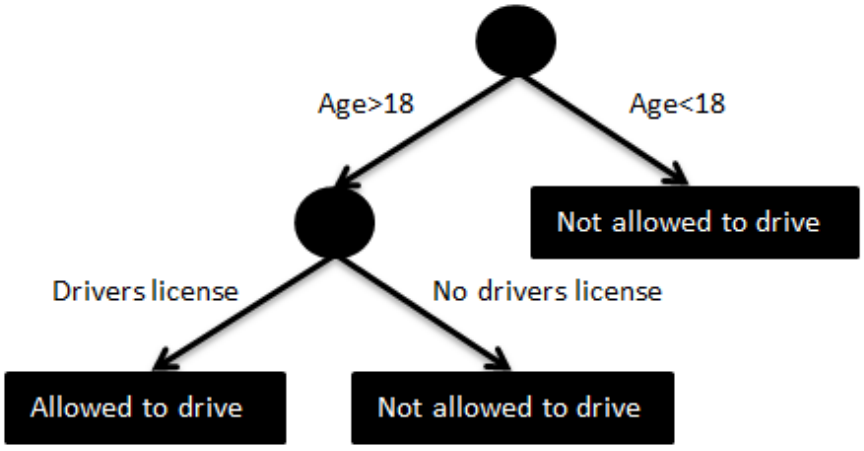


Figure 14: *Example of decision classification tree. Based on age and whether someone has a drivers license there can be determined if this person is allowed to drive or not.*

5.1.2 Unsupervised learning

The difference between supervised and unsupervised learning is that the targets that belong to the known data are unavailable. So the estimator tries to categorise the data itself. In most estimators the number of categories can be given as an input after which the model tries to group the data. Based on the way the training data is grouped, the testing data will be grouped. This method often does not work as good as supervised training. As can be expected we can make better predictions if we have more information.

5.1.3 What type to use?

Since there are two types of machine learning explained it is important to look at when to use which type. Whether to use supervised or unsupervised learning mainly depends on the data available. If you have targets with your data, you can use supervised learning and then it is always good to do this. If the targets belonging to the data are available, use supervised learning, otherwise use unsupervised learning. For both supervised and unsupervised learning there are implemented estimators in MATLAB and PYTHON. The library in PYTHON is further developed and has more estimators available.

5.2 Application for Twence

Before the approach or the results are explained we will explain how the data of Twence is used for machine learning. For the application at Twence a target is available, therefore supervised learning will be used. The target will be the soda input in l/h, since this is

directly dependent on the state of the clogging in the system. The data will be the other measurements from the system. The observations can be given as the average per minute, per ten minutes or per hour. What is used depends on what is available in the dataset given by Twence. Measurements from different time averages will not be used at the same time. It also differs per dataset which variables are given and therefore can be used as data. The target in this case is a continuous value, so it would make sense to use regression estimators. However, after visiting Twence it turned out that there are actually three states in which the system can be. It can work good, the soda input will then be thousand l/h or even more. There can be some problems, but the system is still working. The people in the control room will set the production and therefore the soda input lower if there are problems. This will be done to prevent or slow down further clogging and to keep the production going. The soda input will then be around eight hundred l/h. Or the system is not working and has to be cleaned, the soda input will then be zero l/h. The classes will become the following:

- 0-'broken' if soda input $< 300l/h$
- 1-'problems' if $300l/h \leq$ soda input $\leq 950l/h$
- 2-'good' if soda input $> 950l/h$

The values for the boundaries are based on the process, but are not known precisely. The classes are also used to check how well the regression works. If during testing the prediction is in the same class as the target it will be seen as a correct prediction. This way the percentage of correct predictions can be determined. Since the classes are just one way to determine the amount of correct predictions, it is also good to look at it in another way. Therefore the euclidean norm of the prediction and the original target will sometimes be used as well to determine the percentage. This norm then has to be below a certain value to be a correct prediction.

For the application at Twence MATLAB is used, since the supervised learning functions in MATLAB automatically ignore rows with non numbers and for PYTHON these rows have to be sorted out before applying supervised learning. As already explained in Section 3 the data contains wrong measurements that have a question mark in front. These are not numbers, so in Python it would be necessary to take these out.

In Subsection 5.2.1 there will be elaborated how supervised learning is applied for Twence. In Subsection 5.2.2 there will be looked for relations in the data.

5.2.1 Machine learning applied

With the interpretation of the data mentioned in Section 5.2 machine learning is applied with the estimators; k-nearest-neighbours classification, regression decision tree and classification decision tree. The k-nearest-neighbour method is explained before and is easy to understand, therefore it is good to also apply this method. The decisions trees are chosen since they are known for accuracy and fast training. After some experiments it turned out that it is the best

to use a regression tree. For the classification estimators the targets from both the training and the testing set are changed from the soda input in l/h to the classes as mentioned before. The estimator then predicts a class and it is checked if this is the correct class. For the regression the training targets are the soda input. The soda input is also predicted and then set to the belonging class. This class is compared to the one it actually should be. When training the data and making predictions we look at one time step a time. The target that is predicted, is the target belonging to the observation of one time step. This is different from what has been done in Chapter 4, since Chapter 4 looks at the whole dataset at once. The time dependency is not used with the supervised learning.

If the dataset is split randomly and 60% is used for the prediction and 40% for the testing, the regression tree predicts about 95% correct depending on the random division of the data. This looks like a good result. Also categorisation supervised learning is applied on the data of Twence. This is done by categorising the original targets. The categorisation tree and k-nearest-neighbours method is used. These both gave around 85% accuracy and therefore seem to work not as good as the regression tree. Therefore the regression tree will be used further on.

With the supervised learning as described above there is one problem though. The used data is time dependent. If a random part is taken for the training and a random part for the testing the results will be quite good, since the data that is close to the testing data is used as training data. Therefore from now on a part of the data will be used as training data and another part of the data will be used as testing data, instead of making a random division. When using the data divided in parts for machine learning, the results depend on which part was used as training data. If the training data contains enough examples of each category it works just as good as the random division. However if the training set mainly contains one category, the regression tree does not work that well anymore. Even though the results are good, there still is not guaranteed that the machine learning will give good results for other time spans that are further away from the training data.

As mentioned above it can be determined if a prediction is good or not in multiple ways. What is mainly done is using the categories described in Section 5.2. Another way to determine how well machine learning works, is counting the prediction as correct if it differs less than 100 l/h with the real value. Say y_i is the target at time i and $y_{i-predict}$ is the predicted target for time i. Then the prediction is correct if $|y_{i-predict} - y_i| < c$. Here c is a value that can be varied. However the c will most times be set to 100 l/h. In Figure 15 can be seen how good the prediction is for different values of c.

Furthermore a forecast is made in advance indicating how the system will be working. To do this the data from a certain time step is linked with the target of the next time step. So the data $\vec{x_i}$ will not be linked to the target y_i, but to the target y_{i+1}. This can be extended to multiple time steps. This gave the remarkable result that increasing the amount of time steps did not always mean a worse prediction. That this did not happen is weird and not yet

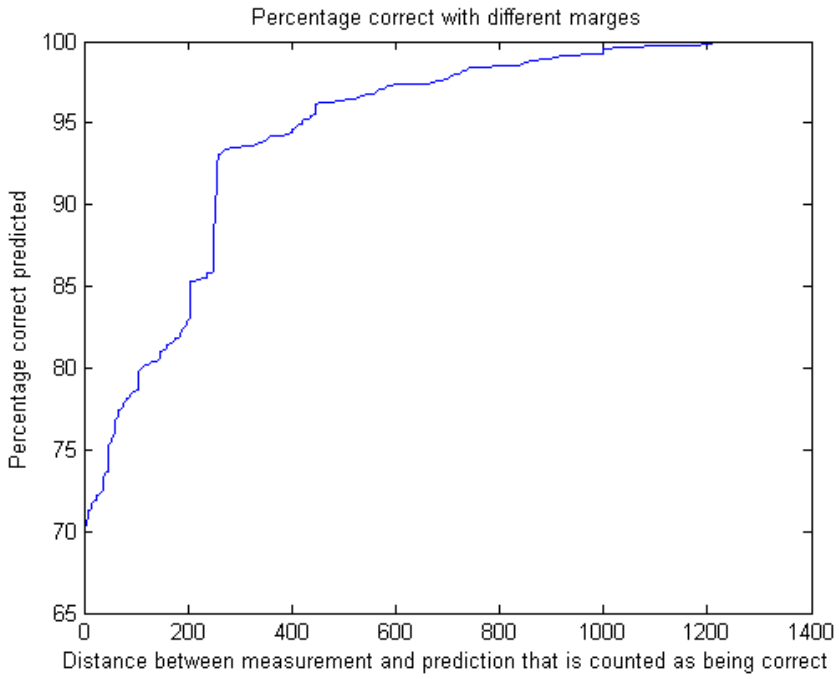


Figure 15: *Percentage predicted correctly with different marge for being correct for the five month data (first hundred days training, last twenty five days testing)*

explainable. This can be related to periodicity in the data or can be just coincidence. There still is one improvement, when predicting the future, the current target can also be used as data. It is not wrong if the current target is not taken along, but it is extra information that is otherwise not taken into account. So the data from time i will be $\vec{x_i}$ together with y_i and the corresponding target will be y_{i+1}.
In the next Subsection results obtained by the methods described above are discussed.

5.2.2 Relations in the data

Now that the basic implementation of machine learning is working, the next step is to figure out if there are relations in the data. This is done by looking at a certain combination of variables and the state of the system. To do this the data that is used for the regression tree will consist only out of two variables at a time, instead of all available. This result[2] can be seen in Table 2. In this table the percentage of correct prediction using two variables instead of all variables as data is shown. The diagonal, where the same variable is used twice, can be

[2]During the last day before the deadline a mistake in the implementation was found. Therefore the results discussed are not correct. Unfortunately there was no time to adjust the discussion. The correct version of Table 2 and 3 can be found in Appendix III. We also refer to Appendix III for a short discussion on the corrected percentages.

Table 2: *Regression tree method with one or two variables as data. Number is percentage predicted in the correct class. The colour represents relatively highest percentage (green) and relatively lowest percentage (red) and colours in between.*

CO2 input reactor 1 (m³/h)	SBC reactor 1 (°C)	SBC reactor 2 (°C)	Reactor 1 circulation pump (%)	Circulation SBC reactor 1 (m³/h)	Circulation SBC reactor 1 (bara)	Circulation SBC reactor 1 (°C)	Reactor 1 cooling water cold (°C)	Reactor 1 cooling water warm (°C)	CO2 input reactor 2 (m³/h)	Mixer SBC reactor 1 (%)	SBC reactor 1 (bara)	SBC reactor 1 (pH)	
92,9	92	94,3	92,8	94,1	93	90,9	94	93,4	94,4	94,3	83,8	95,2	CO2 input reactor 1 (m³/h)
	92,4	87	92,4	95,4	92,9	95,1	93,4	93,5	92	94	83,7	94,4	SBC reactor 1 (°C)
		91,3	93,3	91,9	91,3	91	92,7	93,9	93,7	92,1	84,3	96,1	SBC reactor 2 (°C)
			92,6	92,3	94,5	94,8	94,1	94,4	93,5	94,8	83	94,9	Reactor 1 circ ulation pump (%)
				94,4	94,1	95,2	96,1	95,7	93,8	94,2	84,4	95,4	Circulation SBC reactor 1 (m³/h)
					92,6	94,4	94	93,7	93,5	95,1	88,3	94,9	Circulation SBC reactor 1 (bara)
						90,9	94,3	88,3	91,2	93,6	83	95,8	Circulation SBC reactor 1 (°C)
							94	94,4	92,7	94,5	84,1	95,4	Reactor 1 cooling water cold (°C)
								94	94,2	94,9	84,4	95,5	Reactor 1 cooling water warm (°C)
									93,8	94,6	82,9	95,2	CO2 input reactor 2 (m³/h)
										94,4	91,8	95,8	Mixer SBC reactor 1 (%)
											84,8	93,6	SBC reactor 1 (bara)
												95,2	SBC reactor 1 (pH)

seen as if only that one variable is used. The table is coloured using a green colour for the relatively highest values and red for the relatively lowest values.

The pH-value of the NBC reactor gives really good results with quite some other variables, but also if only the pH-value is used already a good prediction is made. Apparently there is a relation between the pH-value and the soda input, so a plot has been made with the pH-value and our target, the soda input. As can be seen in Figure 16 the pH-value drops every time the soda input drops. This can be explained, since if the system does not work that well, the operators flush the cooler, or the entire system, with warm water. By warming the system the crystallised mixture will resolve and the system can be used longer. The pH-value of water is lower than the pH-value of the soda mixture, so the pH-value will drop. This is something that was not clear before, since the employee of Twence told us that pH-value does not really matter, but it can be seen in the data that there is a relation. However this relation probably will not help in preventing the system from clogging.

In Subsection 5.2.1 we already mentioned that there are multiple ways to determine the percentage that says something about how good the prediction is. In Table 2 the percentage is determined by checking whether the prediction and the real value of the target are in the same class. Now the question is whether the results stay the same if the percentage is determined using the norm of the prediction and the real value of the target. In Table 3 it can be seen that all percentages got lower. This makes sense, since the 'rules' for when a prediction is correct are stricter. In the last column it can be seen that the pH-value still gives the best predictions. However the pressure from reactor 1, that had relatively the lowest scores before, now ends up at a second place. So apparently most of the predictions from the pressure that

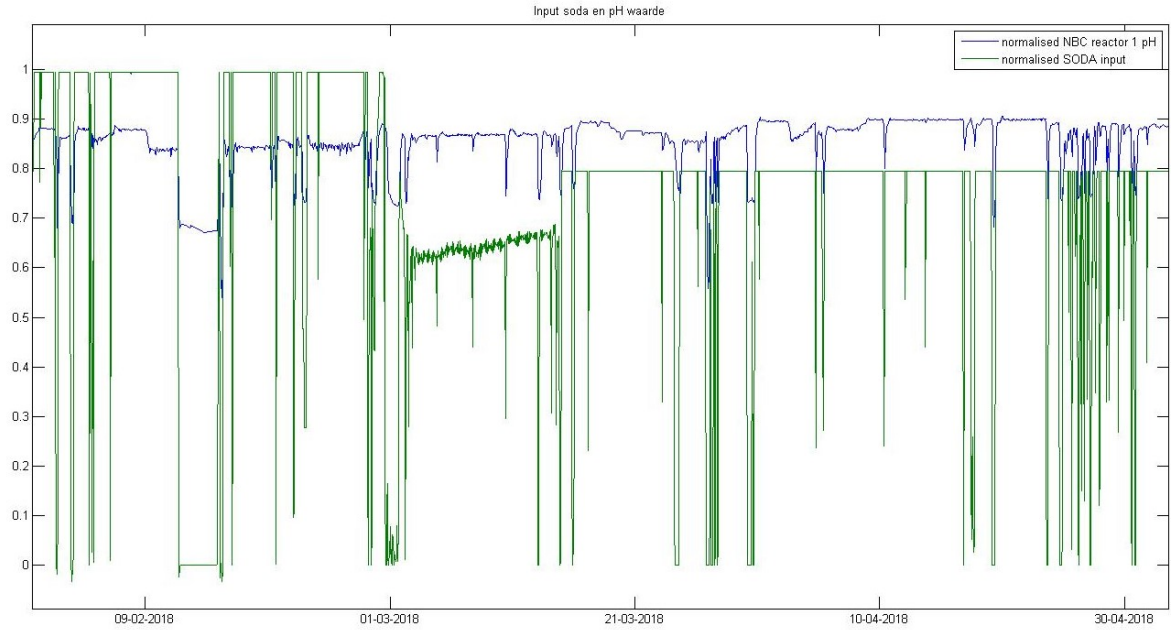


Figure 16: *Plot of the soda input (l/h) and pH-value normalised*

are in the right class are also pretty close. The other variables have more predictions in the right class, but the distance $|y_{i-predict} - y_i|$ is larger. To see which method to use, we should decide what is more important; being close to the target, or being in the same class. Most important is the division into classes, since the classes are accurate with the way the system is operated. If the prediction is also precise this is nice, but it is more important that the prediction is in the right class. The next step is to see if there are relations in the data that can tell something about why the systems clogs.

To get relations that can tell something about why the system clogs, forecasts are made as explained in Subsection 5.2.1. If a certain state of a variable will result in clogging a certain amount of time steps later, this can help us to prevent the clogging. When looking for one or two time steps in advance, in this case one or two hours, the results are quite the same as in Table 2 and 3. The average is a bit lower, but that makes sense. The only main difference is that the soda input is also used as data and that this variable gives good results. These good results can be explained by the fact that the soda input often stays the same for a long period. If the current input is taken and the next is predicted to be the same, quite often the prediction is correct.

The results that are achieved with machine learning unfortunately can't prevent the clogging of the system. The relations that are found can be logically explained and most likely don't give improvements for the system.

Table 3: *Regression tree method with one or two variables as data. The values are the percentages where* $|y_{predict} - y_{test}| < 100$. *The colour represents relatively highest percentage (green) and relatively lowest percentage (red) and colours in between.*

CO2 input reactor 1 (m³/h)	SBC reactor 1 (°C)	SBC reactor 2 (°C)	Reactor 1 circulation pump (%)	Circulation SBC reactor 1 (m³/h)	Circulation SBC reactor 1 (bara)	Circulation SBC reactor 1 (°C)	Reactor 1 cooling water cold (°C)	Reactor 1 cooling water warm (°C)	CO2 input reactor 2 (m³/h)	Mixer SBC reactor 1 (%)	SBC reactor 1 (bara)	SBC reactor 1 (pH)	
50,3	47,8	58,4	46,9	50,8	50,3	44,9	47,4	45,9	49,4	57,9	55	65,1	CO2 input reactor 1 (m³/h)
	43,8	54	50,4	47,1	50,7	46	42,7	44,7	55,2	53,4	56,6	72,9	SBC reactor 1 (°C)
		45,4	58,9	51,5	58,8	48	52,9	57,6	53,4	64,4	64,2	70,4	SBC reactor 2 (°C)
			48,8	51,1	52,3	41,5	49,6	42,9	50,6	50	58,6	72,2	Reactor 1 circulation pump (%)
				47,9	48,6	42,7	46,8	42,3	51,1	60,5	59	67,7	Circulation SBC reactor 1 (m³/h)
					41,9	52,4	47,1	47,4	46,6	53,5	57,3	70,5	Circulation SBC reactor 1 (bara)
						44,4	49	46,2	47	53,5	60,9	65,5	Circulation SBC reactor 1 (°C)
							47,4	38,7	50,1	55,1	54,9	67,5	Reactor 1 cooling water cold (°C)
								45,4	47,7	53,7	60,6	63,1	Reactor 1 cooling water warm (°C)
									48	52,1	57,8	63,9	CO2 input reactor 2 (m³/h)
										52,4	72,9	66,6	Mixer SBC reactor 1 (%)
											60,5	76,7	SBC reactor 1 (bara)
												66,4	SBC reactor 1 (pH)

6 Conclusion

While trying to understand the data from Twence a lot was learned about the process of making sodiumbicarbonate. The question asked was if we were able to analyse the data, and if we could say something meaningful about it. After understanding the process and the data, a classical approach to signal analysis was tried. First Fourier analysis was used. The CO_2 signal could be approximated very closely using this technique. This result might prove useful, but not for predicting the input of the system. The other signals were not periodic enough to use this technique.. Then correlation coefficients were applied to the data. This seemed like an interesting alley to explore, but proved less successful than originally thought. None of the signals had a high enough correlation with the input signal. A quick look at the valve data was taken, which also seemed interesting. It seemed that it was possible to predict something about the input signal, but there was not enough time to check the full extent of this result.

When applying machine learning for Twence we chose to look at supervised learning, since enough information was available to use supervised learning instead of unsupervised learning. Reinforcement learning is not applied since no implementation in MATLAB or PYTHON was available. Supervised learning again has two options; classification and regression. These are both applied on a dataset using the categorisation decision tree, k-nearest-neighbours and regression decision tree. The regression tree gave the best results. Therefore this estimator is used later on. The classification and regression where applied by splitting the data randomly and using 60 % of the data for training and 40 % for testing. Later on the data is not split randomly, but one part is used for training and another for testing. This is done to get good results, if the training data is time dependent with the testing data the results will get to good. This can also be seen if Appendix III is compared to Subsection 5.2.2.
After the implementation of supervised learning was working it was the goal to prevent the clogging. To prevent the clogging we looked into relations in the data. This is done by applying machine learning with one or two variables at a time are used instead of all variables. This gave the result that there are relations in the data, but that it is not possible to give useful information about preventing the system from clogging. Most relations that are found can be logically explained by the way the SBC is produced. Next to that also predicting when the system will clog is quite hard, so far no results are obtained that will work better then the experience of the operators. Based on what is described in Chapter 5 we expect that further research using supervised learning won't give a major break trough. What however can be done is looking into reinforcement learning. This can give better results, since the time dependency and the human control of the system can also be used in the model. With reinforcement learning an optimal strategy to operate the system can be determined. Therefore this can be useful in preventing the system from clogging.

7 References

References

[1] Rijkswaterstaat. Status avi's. https://lap3.nl/uitvoering-lap/status-avir1-d10)/. [accessed 2018-06-11].

[2] Twence. https://www.twence.nl/en/. [accessed 2018-06-24].

[3] Andy Roeloffzen Patrick Huttenhuis, G. F. Versteeg. Co2 capture and re-use at a waste incinerator, 2016.

[4] Cruithne9. Countercurrent exchange. https://commons.wikimedia.org/w/index.php?curid=57612048, 2017. [accessed 2018-06-14].

[5] Joseph Fourier. *Théorie analytique de la chaleur*. 1822.

[6] Phonical. View of a signal in the time and frequency domain. https://commons.wikimedia.org/w/index.php?curid=64473578, 2017. [accessed 2018-06-23].

[7] Karl Pearson. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.

[8] M. Mukaka. A guide to appropriate use of correlation coefficient in medical research. *Malawi Medical Journal*, 24(3):69–71, 2012.

[9] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[10] Data School. Machine learning in python with scikit-learn. https://https://www.youtube.com/watch?v=elojMnjn4kklist=PL5-da3qGB5ICeMbQuqbbCOQWcS6OYBr5A. [Online; accessed 24 April 2018].

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.

[13] Scikit-learn algorithm cheat-sheet. http://scikit-learn.org/stable/tutorial/machine_learning_map/index.htmlchoosing-the-right-estimator. [Online; accessed 26 April 2018].

[14] Scikit-learn nearest neighbors classification. http://scikit-learn.org/stable/modules/neighbors.htmlnearest-neighbors-classification. [Online; accessed 25 April 2018].

8 Appendix

I Matlab function Import

```
function [data,time,legendEntries,relData] = Import(filename,needAVG)
tic
%import the data from an excel file
%
% [data,time,legendEntries,relData] = Import(filename,needAVG)
% INPUT filename: The filename of the excel file
%       needAVG: If this is not specified, the part is skipped to make
%       the code run faster
%
% OUTPUT data: the data in a matrix
%        time: the time series
%        legendEntries: the legend as specified in the excel file
%        relData: the data divided by its mean

%%Determine the range where values are
vari = xlsread(filename,1,'F12:F17'); %This reads the amount of rows in
    use
variables = sum(round(vari)); %Round since sometimes the value is 0.999
dataPerDay = 1/xlsread(filename,1,'B7'); %Smart way to determine time
    steps
%If converted correctly the startTime and endTime are in these squares
startTime = x2mdate(xlsread(filename,1,'C6'));
endTime = x2mdate(xlsread(filename,1,'E6'));

%%start and endsquare and setting the ranges
startSquare = 15 + variables; %This is always where the data starts
endSquare = startSquare + round((endTime-startTime)*dataPerDay) - 1;
%The data always starts on the B column
R1 = {char(66),startSquare,char(66+variables),endSquare}; %char(66) = B
dataRange = sprintf('%s%d:%s%d',R1{:});
%The averages always start on E12 and depend on the amount of variables
R2 = {char(69),12,char(69),11+variables};
avgRange = sprintf('%s%d:%s%d',R2{:});
%The legend always starts on B12 and depend on the amount of variables
R3 = {char(66),12,char(67),11+variables};
legendRange = sprintf('%s%d:%s%d',R3{:});

%%reading the data, AVG and legendEntries
data = xlsread(filename,dataRange);
AVG = xlsread(filename,avgRange);
[~,legendEnt] = xlsread(filename,legendRange);
for i = 1:variables
    %Combine the legend matrix to include the name and unit
```

```
42      legendEntries(i) = cellstr(sprintf('%s, %s',legendEnt{i},legendEnt{i+
            variables}));
43      if exist('needAVG','var') %Check if this is needed, to speed up the
            code
44          relData(:,i) = data(:,i)/AVG(i);
45      end
46  end
47
48  %%making the time variable so it fits the data
49  time = linspace(startTime,endTime,length(data));
50
51  fprintf('Time in Import function was %f seconds \n', toc)
```

	A	B	C	D	E	F
1	SPPA-T3000					
2	Analog Interval Report					
3	Name:					
4	Comment:					
5	Created at:	07:11,6				
6	Time:	From	46:01,6	To	46:01,6	
7	Time Interval :	00:01:00				
8	Aggregate:	average values per time period				
9	Note:					
10						
11	Name	Designation	EngUnit	Time	Avg	QF
12	A3HTF23 CF002||XQ01	CO2 NAAR NBC REACTOR 1	m³/h		135,11182	1
13	A3HTF30 CQ001||XQ01	NBC REACTOR 1	pH		8,906181	1
14	A3HTF30 CT001||XQ01	NBC REACTOR 1	°C		44,782177	1
15	A3HTK61 CF001||XQ01	SODA NBC PRODUCTIE	l/h		742,4104	1
16						
17	Time	A3HTF23 CF002||XQ01	A3HTF30 CQ001||XQ01	A3HTF30 CT001||XQ01	A3HTK61 CF001||XQ01	
18						
19	25/03/2018 07:46:01.572 - 25/03/2018 07:47:01.572	142,59	9,203421	43,43998	1000,5596	
20	25/03/2018 07:47:01.572 - 25/03/2018 07:48:01.572	142,5527	9,2031975	43,43998	1000,5596	
21	25/03/2018 07:48:01.572 - 25/03/2018 07:49:01.572	142,30096	9,2031975	43,43998	999,66547	

Figure 17: *Screenshot of an excel file to be read by the Import function*

II Correlation coefficients

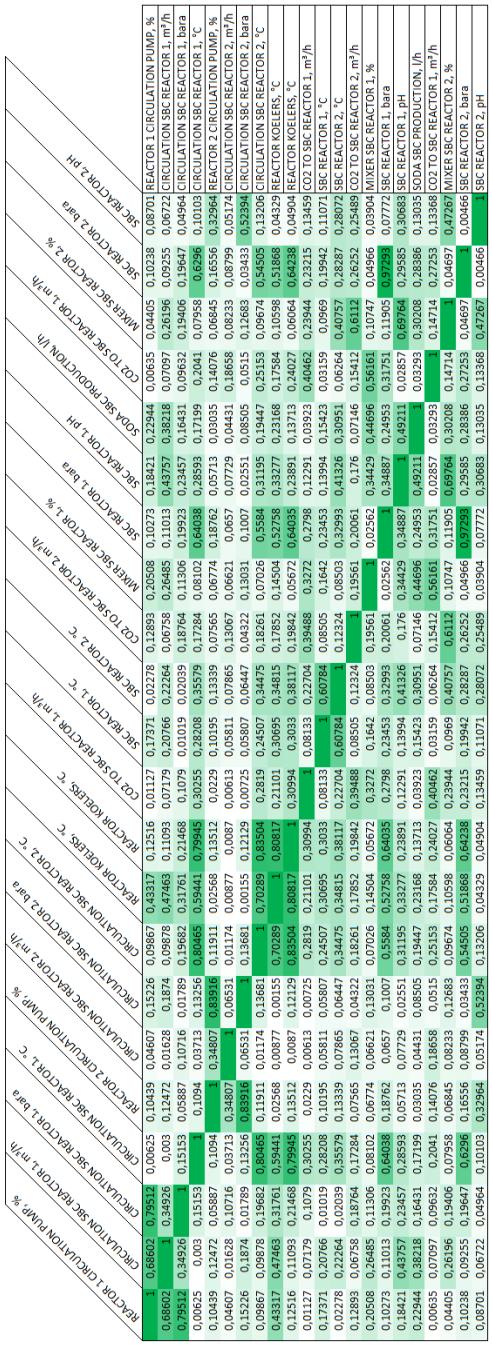


Figure 18: *Absolute value of the correlation coefficients between two signals. The colour represents a high correlation coefficient (green) and low correlation coefficient (white).*

III Results machine learning

With the implementation used in Subsection 5.2.2 there was a small mistake. Due to a typo the testing data and training data had an overlap. This gave results that are higher than is realistic. Since this mistake was only discovered a day before the deadline no full adjustment could be made. Therefore the correct versions of Tables 2 and 3 are given in this appendix. As can be seen in Table 4 the percentage predicted correct is dropped. The pH-value still gives good results, but also the pressure (bara) in reactor 1 gives good results. If the norm of the target and the prediction is used instead of the classes the results are really bad and only two combinations give results that really pop out. This can be seen in Table 5.

Table 4: *Regression tree method with one or two variables as data. The values are the percentages predicted in the correct class. For this table the first hundred days of the five month data are used for training and the last fifty for testing. The colour represents relatively highest percentage (green) and relatively lowest percentage (red) and colours in between.*

CO2 input reactor 1 (m³/h)	SBC reactor 1 (°C)	SBC reactor 2 (°C)	Reactor 1 circulation pump (%)	Circulation SBC reactor 1 (m³/h)	Circulation SBC reactor 1 (bara)	Circulation SBC reactor 1 (°C)	Reactor 1 cooling water cold (°C)	Reactor 1 cooling water warm (°C)	CO2 input reactor 2 (m³/h)	Mixer SBC reactor 1 (%)	SBC reactor 1 (bara)	SBC reactor 1 (pH)	
85,8	87	87,4	75,2	83	85,6	87,6	85,6	84,8	86,2	83,6	94,6	90,2	CO2 input reactor 1 (m³/h)
	88,6	86,2	83,8	94	81,6	91	83,8	86	84,8	88	95	92,4	SBC reactor 1 (°C)
		90,4	88,4	88,8	88,4	91	89,4	88,6	89,4	90,4	93	88,4	SBC reactor 2 (°C)
			81,6	81,4	78,8	89,2	78	83	82,6	88,2	86,8	91,4	Reactor 1 circulation pump (%)
				85,6	85,8	91,2	92	90,6	94	92,8	86,6	90	Circulation SBC reactor 1 (m³/h)
					90	91	89,6	90,8	91,2	87,6	92,4	89,4	Circulation SBC reactor 1 (bara)
						82	90	80	81,8	90,4	86,2	89	Circulation SBC reactor 1 (°C)
							88,8	87,8	86,4	84,8	96,6	90,4	Reactor 1 cooling water cold (°C)
								87,6	88,4	87,2	89,6	90,2	Reactor 1 cooling water warm (°C)
									83,8	91	90,8	91,6	CO2 input reactor 2 (m³/h)
										89	92	92,4	Mixer SBC reactor 1 (%)
											96	92,6	SBC reactor 1 (bara)
												93,8	SBC reactor 1 (pH)

Table 5: *Regression tree method with one or two variables as data. The values are the percentages for which* $|y_{predict} - y_{test}| < 100$. *For this table the first hundred days of the five month data are used for training and the last fifty for testing. The colour represents relatively highest percentage (green) and relatively lowest percentage (red) and colours in between.*

CO2 input reactor 1 (m³/h)	SBC reactor 1 (°C)	SBC reactor 2 (°C)	Reactor 1 circ ulation pump (%)	Circulation SBC reactor 1 (m³/h)	Circulation SBC reactor 1 (bara)	Circulation SBC reactor 1 (°C)	Reactor 1 cooling water cold (°C)	Reactor 1 cooling water warm (°C)	CO2 input reactor 2 (m³/h)	Mixer SBC reactor 1 (%)	SBC reactor 1 (bara)	SBC reactor 1 (pH)	
18,8	10,2	6	13	10,2	13,6	28,8	19,8	17,2	10,8	17,2	11,2	41,6	CO2 input reactor 1 (m³/h)
	9,6	1,8	7,6	5,4	11,2	10,2	1,8	4,8	21,6	22	5,8	6	SBC reactor 1 (°C)
		8,4	5,6	7,6	6,8	7,2	8,2	7,6	5,2	5,4	4,2	2	SBC reactor 2 (°C)
			17,6	12,2	18,2	16,8	12,2	14,2	4,8	9,4	26,4	9,8	Reactor 1 circ ulation pump (%)
				9	9,6	18,4	13,6	17,4	8,2	11	6,2	12,4	Circulation SBC reactor 1 (m³/h)
					7,4	12,6	10	8	11,2	8,8	17,4	3,6	Circulation SBC reactor 1 (bara)
						12,8	24	11,6	10,2	9,8	5	7	Circulation SBC reactor 1 (°C)
							25,4	20,2	19	20	5,8	14,6	Reactor 1 cooling water cold (°C)
								21,8	18,4	9,6	4,4	6,6	Reactor 1 cooling water warm (°C)
									10,6	8	3	53,4	CO2 input reactor 2 (m³/h)
										8,2	30,8	10,4	Mixer SBC reactor 1 (%)
											5,8	10,2	SBC reactor 1 (bara)
												8,4	SBC reactor 1 (pH)