Master Thesis

Prediction of wastewater treatment plants process performance parameters based on microbial communities using machine learning techniques

Emile Cornelissen
s2022893

November 2018

Submitted in partial fulfillment of the degree
MSc Industrial Engineering and Management

Supervisors:
prof. dr. B. Jayawardhana
prof. dr. G.J.W. Euverink

university of groningen
faculty of science and engineering

Master Thesis

Prediction of wastewater treatment plants process performance parameters based on microbial communities using machine learning techniques

Emile Cornelissen
s2022893

November 2018

Submitted in partial fulfillment of the degree
MSc Industrial Engineering and Management

Supervisors:
prof. dr. B. Jayawardhana
prof. dr. G.J.W. Euverink

university of groningen
faculty of science and engineering

Abstract

Wastewater treatment plants (WWTPs) use a wide variety of microorganisms to remove contaminants from the wastewater. This thesis researches the relationship between microbial communities and process performance. This relationship is crucial to improve process performance and provides insight in the diagnosis and prognosis of the process. The biological process of the WWTP is highly complex due to its nonlinear and dynamic behaviour and the diversity of the microbial community. Two machine learning techniques, artificial neural networks and support vector regression are used to model this complex system. Using data from next-generation sequencing, the microbial community composition was revealed. This data was used as input for the machine learning models to predict a selection of process performance parameters. Both models showed beyond satisfactory results in the training and test stages. By analyzing the sensitivity of each modeled process parameter to each microorganism, an indication of the influence of the microbial structure on process performance is established.

List of Abbreviations

ANN	Artificial Neural Network
ASP	Activated Sludge Process
AT	Aeration Tank
BOD	Biological Oxygen Demand
COD	Chemical Oxygen Demand
CV	Cross Validation
EC	Electroconductivity
KKT	Karush-Kuhn-Tucker conditions
MLP	Multilayer perceptron
MSE	Mean Squared Error
NGS	Next-generation Sequencing
NN	Neural Network
OFAT	One-Factor-At-a-Time
PCA	Principal Component Analysis
PCHIP	Piecewise Cubic Hermite Interpolating Polynomial
qPCR	quantitative Polymerase Chain Reaction
R^2	Coefficient of Determination
RBF	Radial Basis Function
ASM	Activated Sludge Model
SVI	Sludge Volume Index
SVM	Support Vector Machine
SVR	Support Vector Regression
TSS	Total Suspended Solids
WWTP	Wastewater Treatment Plant

Contents

Chapter 1

Introduction

1.1 Introduction

The impact of human activity on the environment is growing in importance as resources become scarcer and the climate changes. Wastewater from municipal and industrial sources is just one of the many results of human activity. Wastewater has a wide range of polluting effects when disposed directly into the environment (Friha et al., 2014). That is why wastewater is treated by a Wastewater Treatment Plants (WWTP). WWTPs use a sequence of physical, chemical and biochemical processes to remove pollutants from the wastewater, before the water flows back into the environment.

Most of the WWTPs use a biological process called the Activated Sludge Process (ASP) to remove contaminants as organic materials, nitrogen and phosphorous from the influent (i.e. the incoming stream of wastewater). There are thousands of different microbial species present in the process (Ofiteru et al., 2010). These microorganisms can be seen as the 'engine' of this ASP, as they form groups ('flocs') and respire and grow using the contaminants in the wastewaster.

The ASP is a dynamic and complex process that is difficult to control. One of the reasons for this is that the influent wastewater flow and composition varies over time and follows dynamic patterns. Due to this complexity, operating the WWTP to maintain a good quality outflow ('effluent') from the WWTP is a difficult task. Subsequently, modeling and control of the process has gained interest over the past decades. Commonly, mathematical models are used, constructed from the physical, chemical and biological principles.

A recent development is the use of next generation sequencing of microbial communities. With this technique, the taxonomic fingerprint of the entire microbial population present in an environment can be determined. This technique opens up a wide array of research to be performed in various research areas, including WWTPs.

1.2 Research question

According to Werner et al. (2011), modeling a WWTP based on solely mathematical models is not enough. An understanding of microbial communities and how they influence the process performance is crucial in order to improve the process performance. This is also the conclusion of the papers by Muszyński et al. (2015); Liu et al. (2016); Bassin et al. (2017).

The abundance of different microorganisms as well as the time-varying and highly nonlinear characteristics of the WWTP process makes the relation with process performance a difficult one to investigate. Machine learning techniques, such as Artificial Neural Networks (ANN), have a relatively high accuracy for dealing with complicated systems, while just relying on the inputs and outputs of the system. That is why the aim of this research is to use machine learning to cope with the large numbers of variables and dynamics of the system.

Subsequently, this thesis focuses on the following question:

Can machine learning be used to model the relationship between microbial communities and parameters of the process?

In order to answer this research question, datasets with microbial communities and the process parameters characteristics from one WWTP will be used. The output of this research is a model that simulates the relation between the bacterial communities and process parameters and is able to predict these characteristics based on the microbial communities dataset. The quality and performance of the model is measured using statistical accuracy measurements. This way, it is tested whether the model is a realistic representation of the process at the WWTP. Finally, by performing a sensitivity analysis of the model, an indication of the relationship between the microbial communities and parameters of the process can be quantified.

1.3 Thesis outline

This thesis consists of six chapters as follows:

Chapter one (Introduction): chapter one contains a general introduction to the subject, the research question and an outline of this thesis.

Chapter two (Background): chapter two includes general literature search of the WWTP and its process, next-generation sequencing and gives an overview of the business and technological context.

Chapter three (Machine learning): chapter three covers the three machine learning methods used in this research: Principal Component Analysis for dimen-

sionality reduction and the two machine learning techniques Support Vector Regression and Artificial Neural Networks.

Chapter four (Implementation): chapter four covers the implementation of methods in a case study at NorthWater. Steps of the implementation include: data pre-processing, selection of parameters, building the machine learning models and training and testing of the developed models.

Chapter five (Results): chapter five discusses the obtained results.

Chapter six (Conclusion): chapter six presents the main conclusion and recommendations made during this study.

Chapter 2

Background

2.1 Background information

2.1.1 Wastewater Treatment Plant

A Wastewater Treatment Plant (WWTP) purifies polluted water so that it is returned to the environment properly. There are two types of WWTPs: a municipal WWTP treats wastewater from domestic sources and an industrial WWTPs treats wastewater from industrial sources. This thesis focuses on industrial WWTPs.

In Figure 2.1, a simplified schematic overview of a WWTP is shown, based on NorthWater (2014). Only the main components of the plant are included. Influent water consists of wastewater from different industries. Thus, the content of the water differs substantially. Firstly, the water passes through bar screens, where the larger parts are filtered out and disposed. Secondly, the water enters the equalization basin. In this basin, the water from different sources is mixed. This minimizes the variability of the influent characteristics of the wastewater, such as pH, temperature, electrical conductivity, flow rate and organic compound concentrations. This is an important process step, since high variation in wastewater composition may result in a severe degradation in overall process performance (Goel et al., 2005).

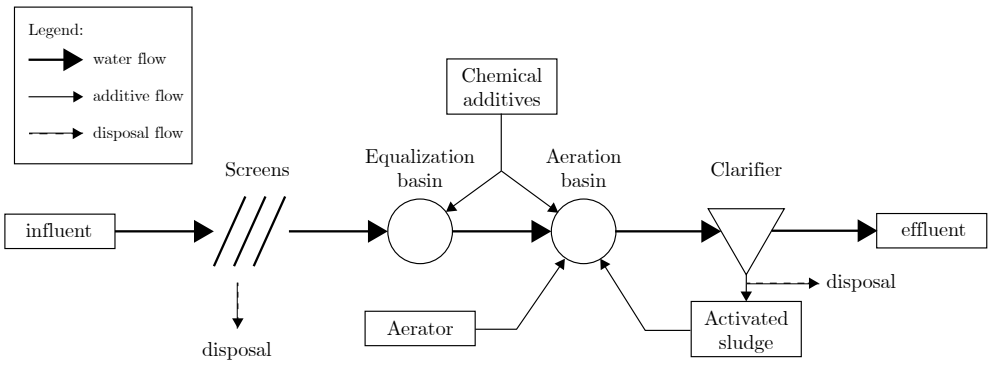

Figure 2.1: Simplified schematic overview of a WWTP at NorthWater

Thirdly, the filtered and equalized water flows into aeration tank. A large vari-

ety of microorganisms, called the activated sludge, resides in this tank. An aerator provides the mix of water and microorganisms with enough oxygen. The combination of oxygen and microorganisms creates an environment where the organic compounds in the wastewater is degraded by the microorganisms (Muralikrishna and Manickam, 2017). We will discuss more on this activated sludge in the next section.

Fourthly, the wastewater passes through a clarifier. The microorganisms will sink to the bottom of the clarifier and the clean water from the top of the clarifier passes further. The remaining sludge from the bottom is partly pumped back in the system and partly disposed (Eckenfelder and Grau, 1998).

2.1.2 Activated Sludge Process

The activated sludge is a complex ecosystem of competing microorganisms. This biomass, consisting mostly out of bacteria, is responsible for the removal of pollution in the wastewater, such as organic matter, nitrogen and phosphorus. The microorganisms feed on organic matter to stay alive and, at the same time, create new individuals (Grady Jr et al., 1998; Seviour and Nielsen, 2010). This process, called the Activated Sludge Process (ASP) is the most widely used process in WWTP's worldwide, since it is a reliable, flexible method and results in a high quality effluent (Eckenfelder and Grau, 1998; Scholz, 2006).

The aerator provides oxygen to the microorganisms and simultaneously mixes the blend of wastewater and biomass (Grady Jr et al., 1998). As the microorganisms grow, they stick together and form particles ('flocking'). These particles will sink to the bottom of the clarifier, leaving a relatively clean effluent (Muralikrishna and Manickam, 2017).

Three major challenges regarding modeling this process are found in literature. Firstly, the diversity of the biomass is immense. Thousands of species of microorganisms are found in the activated sludge, all having their own function in the process (Shchegolkova et al., 2016). Secondly, the ASP exhibits time-varying and highly nonlinear characteristics. There are numerous factors influencing the process of which not all are known (Hong et al., 2003). Thirdly, the quality and quantity of the influent of wastewater varies widely. This requires a flexible process that requires experienced personnel controlling and operating the process 24 hours a day (Grady Jr et al., 1998).

2.1.3 Process parameters

In a WWTP, there are certain key explanatory variables which are used to assess the plant performance, in this thesis referred to as *process parameters*.

Two important process parameters are the amount of nitrogen and phosphorous in the water. Presence of these elements is associated with eutrophication in the water, which should be prevented. Therefore, the phosphorous and nitrogen

compounds should be eliminated from wastewaters (Yamashita and Yamamoto-Ikemoto, 2014; Bassin et al., 2017).

The organisms that carry out the removal of organic matter, nitrogen and phosphorous are very sensitive to operating conditions, such as the pH, electrocondictivity and dissolved oxygen (Martín de la Vega et al., 2018). Heterotrophic bacteria use organic matter as carbon source in their metabolism, resulting in oxygen consumption. Dissolved oxygen in the water is thus crucial in this process. A depletion of oxygen in the water would result in the death of these organisms. Therefore, oxygen demand measures are an important process parameter in WWTPs. Chemical (COD) and Biochemical Oxygen Demand (BOD) are two examples of these measures (Bassin et al., 2017).

2.1.4 Metagenomics

Metagenomics is the analysis of the genomes of microorganisms by extracting and cloning the DNA-sequences of a sample of microorganisms. Analyzing the genomic data provides insight of the microbial population of an environment (Handelsman, 2005).

In order to analyze the bacterial communities of an activated sludge, Next-Generation Sequencing (NGS) is applied to a sample of the sludge. NGS analyzes millions of different sequences sampled from the aeration basin at the same time. These sequences are compared with DNA and protein databases. Based on this comparison, an overview of which organisms occur in the sample is created (Mardis, 2008; Xu, 2014). Conventional methods, such as qPCR detection, are only able to detect one or a few microorganisms at once. With NGS, one is able to detect the entire microbial population and assess the entire sample. This technique enables researchers to open up new biological research areas. (Caporaso et al., 2012; Soon et al., 2014)

The result of this analysis at the WWTP is an extensive list of microorganisms that exist in the sample. In addition, the amount of sequences corresponding to a certain microorganism relative to the total amount of sequences is given.

2.2 Business context

The business context of this research takes place at the Wastewater Treatment Plant. A WWTP is a costly process and gaining insight in its process could result in a higher efficiency at the plant. The performance of a WWTP currently depends mainly on the decisions made by a process engineer, based on his experience at the plant (Hong et al., 2003).

The activated sludge process is of major importance, since the management and operation of the activated sludge is accountable for at least 50% of the construction

and operating costs of a WWTP (Campos et al., 2009; Foladori et al., 2010; Guo et al., 2013). Thus, increasing the efficiency of the ASP and decreasing its costs will have a large impact on the total costs of the plant.

The WWTP of NorthWater in Oosterhorn will function as case study for this research. This WWTP is located near the industrial port area of Delfzijl in Groningen, the Netherlands. The plant purifies the water from industrial companies in the area, mainly operating in the chemical sector. What distinguishes this WWTP from other plants is that the salinity of the wastewater is relatively high. This increases the complexity of the management and control of the plant (NorthWater, 2014).

2.3 Engineering context

The engineering context of this research is two-fold. On the one hand, the microbiological processes are to be considered. On the other hand, data science techniques will play a major role in this research.

As stated in Section 2.1.2, the ASP is a highly complex, multi-variable and nonlinear system. The traditional approach to simulate, control and evaluate the process of a WWTP is by using deterministic models. Henze et al. (2000) has devoted effort in creating a model that is based on the fundamental biokinetics. This Activated Sludge Model (ASM), is used by operators of WWTPs worldwide. However, modeling and controlling remains challenging in a real WWTP (Côté et al., 1995; Moral et al., 2008; Bagheri et al., 2015).

Another way to deal with the complexity of the WWTP is by applying a data-driven approach, in which only the inputs and outputs of the system are considered. The major advantage of data-driven models over deterministic models is that little knowledge about the intrinsic characteristics of the system is required (Côté et al., 1995; Lou and Zhao, 2012; Hamed et al., 2004).

Machine learning techniques have been used extensively by researchers regarding predicting the performance of WWTP's (Hong et al., 2003; Hamed et al., 2004; Mjalli et al., 2007; Moral et al., 2008; Han and Qiao, 2012; Wei, 2013; Ay and Kisi, 2014; Guo et al., 2015; Fernandez de Canete et al., 2016). In a literature review by Corominas et al. (2018), peer-reviewed papers that used data-driven techniques to improve the operation of WWTP's are analyzed. The most cited techniques are Artificial Neural Networks (ANN), Principal Component Analysis (PCA), Fuzzy logic and partial least square regression.

Corominas et al. (2018) also noted that due to the large amounts of data existing in the area of WWTP's, the development of black-box models such as ANN's and Support Vector Regression (SVR) was stimulated. These techniques are used for process optimization. Both these techniques were used in a study by Guo et al. (2015), where these techniques were used to effectively predict effluent character-

istics. Bagheri et al. (2015) used two different ANN models to predict the Sludge Volume Index (SVI) in a WWTP. In addition, they used a genetic algorithm to optimize the weights within the model.

The majority of the papers in this field used influent characteristics and environmental conditions as inputs for their machine learning models. However, Seshan et al. (2014) and Liu et al. (2016) were both able to predict effluent quality using the microbial communities as input parameters with the aid of a Support Vector Regression model. Liu et al. (2016) concluded that the understanding of bacterial communities and their influence on the performance is crucial in improving the performance of a WWTP. This paper by Liu et al. (2016) is used as foundation and guideline for this research.

Chapter 3

Machine learning

Machine learning (ML) is a research area within Artificial Intelligence (AI) regarding the design and development of algorithms where computers learn and adapt from data and develop knowledge (Mitchell, 1997). ML includes a variety of techniques that allows computers to improve automatically. The primary advantage of these techniques is that they are able to represent complex, non-linear systems, without having to know and model the deterministic mathematics of the system (Côté et al., 1995; Hamed et al., 2004; Lou and Zhao, 2012).

Machine learning can roughly be divided into three categories. Figure 3.1 shows these three categories and where the techniques in this thesis are categorized.

1. **Supervised learning.** This type of machine learning requires both input values as target values. The model learns from the target values. The trained model is used to make prediction on new input values. Examples: Classification and regression.

2. **Unsupervised learning.** This type of machine learning only uses input values to learn. Examples: Clustering, dimensionality reduction and noise reduction.

3. **Reinforcement learning.** This is a system of learning that trains based on rewards and punishments. It doesn't have fixed targets, but learns by interacting with its environment. When the algorithm acts correctly, it gets rewarded and vice versa.

For this research, three different machine learning models are used. The first, Principal Component Analysis, is used to perform dimensionality reduction. The other two, Support Vector Regression and Artificial Neural Network, are used as regression methods.

3.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most used methods of reducing the dimensionality of a dataset (Jolliffe, 2011). The information in a given data set

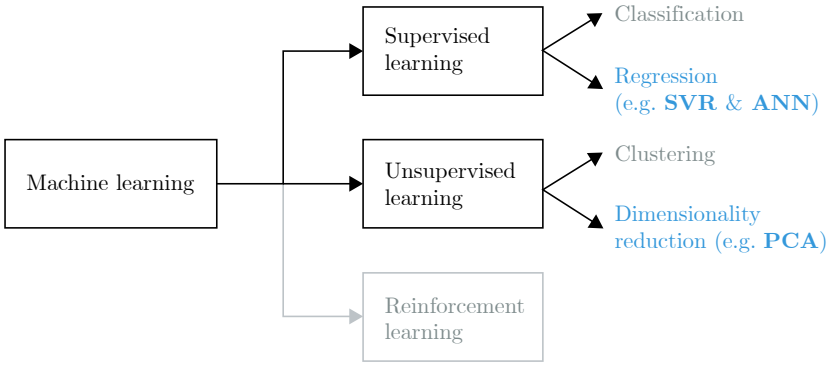

Figure 3.1: Different types of machine learning. Marked in blue are the approaches used in this research. Marked in grey are the approaches not elaborated in this research.

corresponds to the total variation it contains. With PCA, one can extract important variables in the form of components from a large set of variables in a dataset. There are two main advantages of a lower dimensionality (fewer variables) in regression techniques. First, the chance of overfitting on the model reduces. If a model has p input vectors and n samples where $p \gg n$, the model has a high chance of having a high performance on the training data, but a low performance on the test data. Reducing p to a smaller set of q input vectors will improve generalization of the model and reduce the chance of overfitting (Hastie et al., 2009). Second, the speed of running machine learning models increases significantly. With a smaller size of input vectors q, machine learning models require less storage and time to train and run (Burges, 2009).

PCA searches for linear combinations with the largest variances, and divides them into Principal Components (PC) where the largest variance is captured by the highest component in order to extract the most important information. The goal of PCA is to reduce the number of features whilst maintaining a high level of retained variance. Following is an explanation of the technique, based on the books by Hastie et al. (2009); Abdi and Williams (2010); Jolliffe (2011)

Let a dataset consist out of feature vectors $x_1, x_2, \ldots, x_p$, so with a total of p variables. The mean of these features is denoted as $\bar{x}$. The first step is to calculate the covariance matrix Σ, defined by

$$\Sigma = \frac{1}{p} \sum_{i=1}^{p} (x_i - \bar{x})(x_i - \bar{x})^T \tag{3.1}$$

.

The first step of PCA is to solve the problem

$$\Sigma v_i = \lambda_i v_i \tag{3.2}$$

where λ_i $(i = 1, 2, ..., n)$ are the eigenvalues and v_i $(i = 1, 2, ..., n)$ are the corresponding eigenvectors.

The eigenvalues λ_i are then sorted from highest to lowest and the top q eigenvalues are selected ($q < n$). The parameter q is the number of principal components and can either be chosen beforehand or be calculated based on the total variance to be retained by the principal components.

Let

$$\phi = \begin{Bmatrix} v_1 \\ v_2 \\ \dots \\ v_q \end{Bmatrix} \qquad \text{and} \qquad \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_q \end{bmatrix} \tag{3.3}$$

The retained variance is calculated by dividing the sum of the selected eigenvalues by the sum of all the eigenvalues. If the total variance to be retained (r) is chosen as parameter to calculate the number of PCs, then the following equations holds.

$$\frac{\sum_{i=1}^{q}(\lambda_i)}{\sum_{i=1}^{n}(\lambda_i)} \geq r \tag{3.4}$$

The new matrix x_{PC} is calculated as follows:

$$x_{PC} = \phi^T x \tag{3.5}$$

This results in a new matrix with q features and a retained variance of at least r. This is the final result of the PCA.

3.2 Support Vector Regression

A Support Vector Machine (SVM) is a supervised learning models. It was originally designed for classification problems (Cortes and Vapnik, 1995), but later extended for regression as well (Drucker et al., 1996). The method used for regression is called Support Vector Regression (SVR).

In this section, the general concept of SVR is explained and kernel functions will be explained. This explanation is mainly based on books by Scholkopf and Smola (2002) and Haykin et al. (2008).

In SVR, the nonlinear function between a vector of input variables x_i ($i = 1, 2, \dots, n$) and output variable y is estimated. The input x_i is mapped to a higher dimensional feature space $\phi(x_i)$. This way, the nonlinear relationship between x_i and y is converted to a linear regression problem between $\phi(x_i)$ and y. This linear regression relationship is described by

$$f(x_i) = w^T \phi(x_i) + b \tag{3.6}$$

where w is the weight vector and b is the bias constant.

The goal of the SVR is to optimize the w, b and the parameters of the function $\phi(x_i)$ (kernel-related parameters). These kernel-related parameters are described in Section 3.2.1.

To quantify the performance of a regression, a loss function is introduced. In normal regression problems, this loss function can be in the form of a quadratic loss function, as shown in Equation 3.7. However, in SVR, a ϵ-insensitive loss function is used. This function is equal to zero when the loss is within a range of ϵ (Equation 3.8).

$$L(y, f(x_i)) = (y - f(x_i))^2 \tag{3.7}$$

$$L_\epsilon(y, f(x_i)) = \begin{cases} 0 & \text{if } |y - f(x_i)| \leq \epsilon \\ |y - f(x_i)| - \epsilon & \text{if } |y - f(x_i)| > \epsilon \end{cases} \tag{3.8}$$

Ideally, by optimizing the weight vector w and bias b, a function $f(x_i)$ where the loss is within a range of ϵ of the actual output y. In that case, the convex optimization problem is feasible (Scholkopf and Smola, 2002). However, this is not always the case and errors greater than ϵ exist. When the problem with its constraints is infeasible, two slack variables are introduced; ξ_i and $\hat{\xi}$. Figure 3.2 shows an ϵ-insensitive tube with slack variables and the corresponding ϵ-insensitive loss function.

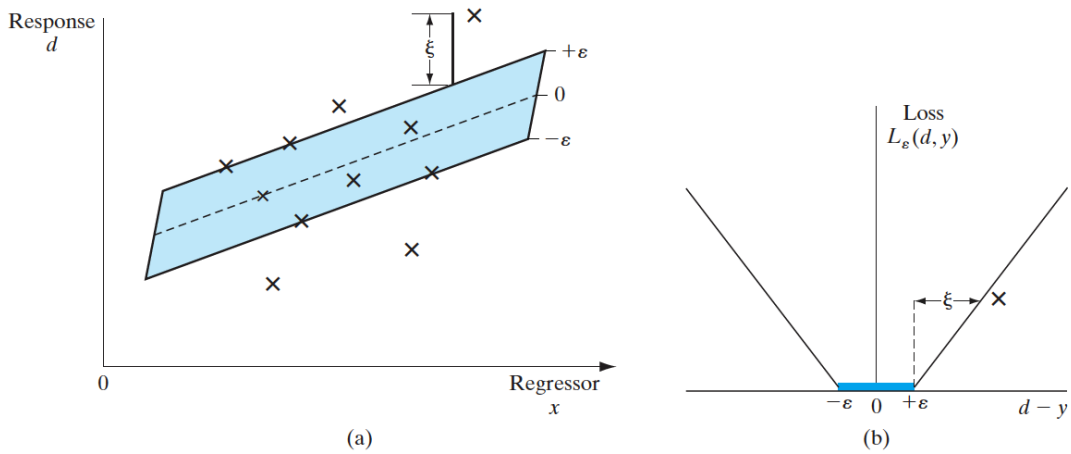

Figure 3.2: (a) Graph of a linear regression with an ϵ-insensitive tube, fitted to data points, noted with $\times$'s. (b) The corresponding ϵ-insensitive loss function. From Haykin et al. (2008).

The minimization problem for optimizing $f(x_i)$ and its constraints are formulated as follows.

$$\text{minimize} \quad \left[\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \hat{\xi}_i)\right] \tag{3.9}$$

$$\text{subject to} \quad \begin{cases} y - f(X) \leq \epsilon + \xi_i \\ f(X) - y \leq \epsilon + \hat{\xi}_i \\ \xi_i, \hat{\xi}_i \geq 0 \end{cases} \tag{3.10}$$

where n is the number of training samples.

The minimization problem consists of two parts; the first term $\frac{1}{2}\|w\|^2$ represents the generalization term of the function and the second term represent the training error. The parameter $C > 0$ controls to what extent training errors are allowed. Equation 3.9 is called the primal objective and its variables are called primal variables.

To solve the minimization problem, we introduce Lagrange multipliers α and α^* and the objective is reformulated into

$$\text{maximize} \quad \left[-\epsilon\sum_{i=1}^{n}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{n}(\alpha_i^* - \alpha_i) - \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(x_i, x_j)\right] \tag{3.11}$$

$$\text{subject to} \quad \begin{cases} 0 \leq \alpha_i^* \leq C \\ 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \end{cases} \tag{3.12}$$

where n is the number of training samples and $k(x_i, x_j)$ is the chosen kernel method. This Equation is called the dual objective. It's constraints are based on the Karush-Kuhn-Tucker (KKT) conditions.

Once the α and α^* are found that maximize this dual objective, the regression function $f(x)$ becomes

$$f(x) = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)k(x_i, x) + b \tag{3.13}$$

3.2.1 Kernels

The performance of a SVR depends highly on the choice of kernel method. A kernel method is an algorithm that maps vectors into a high dimensional vector space by computing the dot product of the vectors. The main advantage of this method is that it allows the use of linear regression techniques for non-linear regression

problems at low computational costs.

A variety of kernel methods exists, four of the most used are shown below. For a linear regression model, the kernel function is just a simple sum of the cross products. The other three kernel functions are non-linear.

$$\textbf{Linear:} \quad k(x_i, x) = x_i \cdot x \tag{3.14}$$

$$\textbf{Polynomial:} \quad k(x_i, x) = (x_i \cdot x)^d \tag{3.15}$$

$$\textbf{Radial Basis Function:} \quad k(x_i, x) = \exp\left(-\gamma ||x_i - x||^2\right) \tag{3.16}$$

$$\textbf{Sigmoid:} \quad k(x_i, x) = \tanh(\gamma x_i \cdot x + r) \tag{3.17}$$

with d, γ and r as kernel parameters.

The choice for a kernel as well as the adjustable kernel parameters have an important influence on the performance of the kernel. Attention should be paid to optimize these parameters (Eitrich and Lang, 2006). Usually, this is done by doing a grid search, where all possible combinations of parameters are investigated (Hsu et al., 2010).

3.3 Artificial Neural Networks

An Artificial Neural Network (ANN) is an information processing system that resembles an actual neural network as present in a human brain. It is capable of approximating processes that relate the input and output of any system. It does not need to know the explicit internal relations of the system itself, nor the physical meaning of the system. Therefore, ANN's are considered as 'black-box' models. This section is mainly based on the books by Haykin et al. (2008), Han et al. (2011) and Nielsen (2015).

The structure of an ANN mimics biological neural networks, based on the assumptions that:

1. Information processing occurs at many simple elements called neurons.
2. Signals are passed between neurons over connection links. Each neuron of a layer is connected to all neurons of the next layer.
3. Each connection link has an associated weight, which, in a typical neural network, multiplies the signal transmitted.
4. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.
5. The output signal of that neuron is then transmitted to all neurons of the next layer.

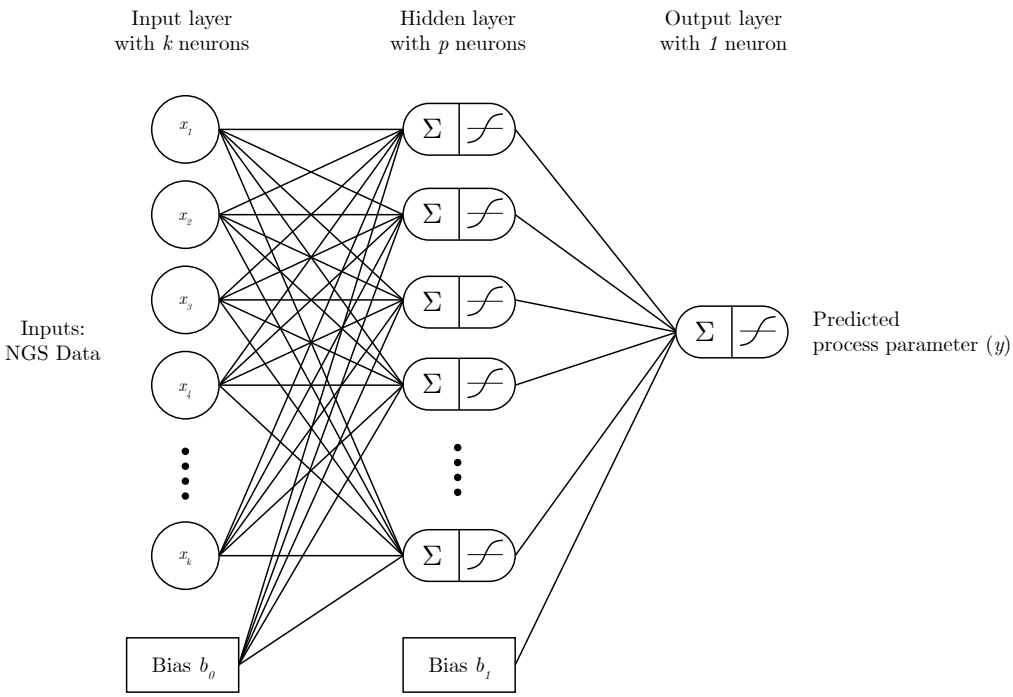

Figure 3.3: Architecture of a neural network with one hidden layer with p neurons.

The most commonly used ANN structure, a multilayer perceptron (MLP) consists of three types of layers, as shown in Figure 3.3. When ANN or NN is mentioned in this thesis, specifically an MLP is meant.

- Input layer. This layer consists of neurons that receive the values of the input parameters. From these neurons, the values of the input parameters are fed into the network.
- Hidden layer. This layer (or multiple layers) connects the the input and output layers.
- Output layer. This layer consists of one or multiple neurons that determine the output of the neural network.

Considering an ANN with k neurons in the input layer, 1 hidden layer with p neurons and a single output as in Figure 3, there will be two types of connecting weights. First of all, a $k \times p$ matrix w connecting the input to the hidden layer and a $p \times 1$ vector v connecting the hidden layer to the output layer. All k neurons and the output will have a bias. The input data used will be the $n \times k$ matrix X with n observations and k variables and the output variable used is the $n \times 1$ dependent variable y. Thus, each of the k input neurons as well as the single output vector has a vector with n observations.

In an ANN, we forward propagate the input of the model through the different layers into an output. Then, using a loss function, the output is compared with the target value. In order to minimize the loss, backward propagation is used by

finding the derivative of the error for all weights and biases and adjusting these accordingly. This process is shown in Figure 3.4.

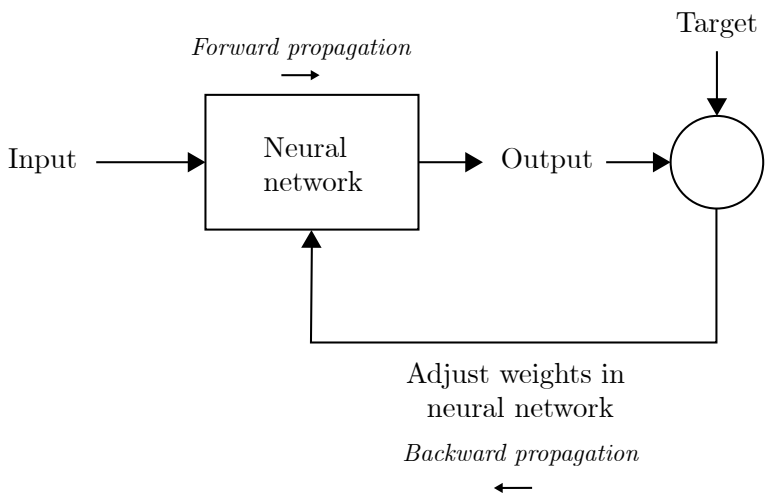

Figure 3.4: Simplified process diagram of backward and forward propagation in a neural network

3.3.1 Forward propagation

Zooming in on one neuron in the network, Figure 3.5 schematically describes what happens at each node in the hidden and output layers. The value of each node a_i from previous layer i is multiplied with its corresponding weight w_{ij}. Then, the results of these multiplications are summed up and a bias is added. Finally, the output o_j is determined by a chosen activation function $g(\cdot)$. Various commonly used activation functions are shown in Figure 3.6. The process at a neuron in the hidden layer is described by

$$o_j = g\left(\sum_{i=1}^{k} w_{ij}a_i + b_j\right) \tag{3.18}$$

At first, the model will propagate forward using random initial weights and biases. With these random values, the output of the model might differ a lot from the actual values the model is trained on. The ANN learns from these errors and updates the weights by using backward propagation.

3.3.2 Backward propagation

Backward propagation (also known as backpropagation) is applied in order to improve the current model's fit and consequently get the predicted values $\hat{y}$ closer to the target values y. In other words, the cost should be minimized. The cost is a function of the error e, which is equal to $\hat{y} - y$. The goal of backpropagation is to find the partial derivatives $\frac{\delta C(e)}{\delta w}$ and $\frac{\delta C(e)}{\delta b}$ of the cost function $C(e)$ to all weights vectors and biases in the network. In case of an ANN with one hidden layer, this

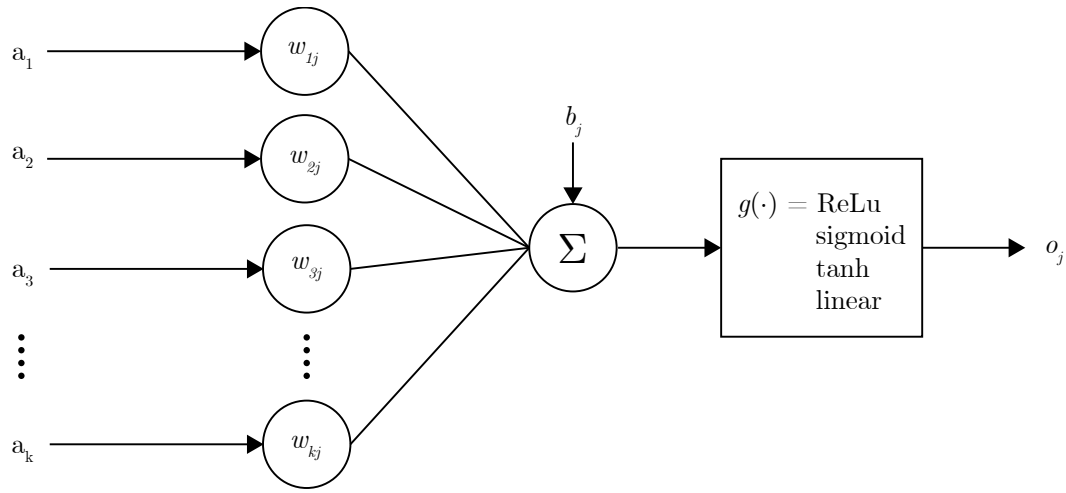

Figure 3.5: Schematic overview the process of one neuron. It shows how the output of a neuron o_j is calculated from the inputs a_i $(i = 1, 2, \ldots, k)$ of k incoming nodes.

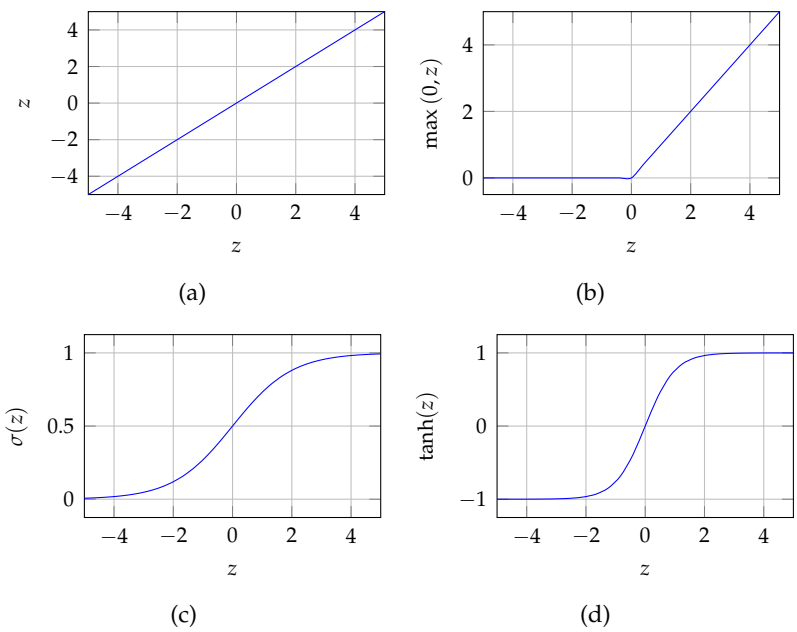

Figure 3.6: Common used activation functions include (a) the linear function $g(z) = z$, (b) the rectified linear unit (ReLu) $g(z) = \max[0, z]$, (c) logistic sigmoid $g(z) = \sigma(z) = \frac{1}{1+e^{-z}}$ and (d) the hyperbolic tangent $g(z) = tanh(z)$.

means that four partial derivatives of the error have to be calculated: for the two weight matrices and the two biases. These partial derivatives denote how a change in weight and bias affects the cost of the network. Since the loss is to be minimized, the weights and biases are updated according to

$$w^{+} = w + \alpha \frac{\delta C(e)}{\delta w} \tag{3.19}$$

where α is the learning rate, determining how much the weight matrix is updated each iteration and w^{+} denotes the updated weight vector.

The same updates are applied to all weight matrices and biases. This process will iterate until a stopping criterion is reached. Stopping criteria for a network are for example the number of iterations or that the cost is smaller than a certain threshold.

Elaborate derivation of the formulas for the partial derivatives is left out in this thesis, but can be found in Nielsen (2015).

Chapter 4

Implementation

In this chapter, the three machine learning techniques described in Chapter 3 are implemented in a case study at NorthWater using Python programming language. Figure 4.1 shows the road map of the steps taken in this chapter.

4.1 NGS Data pre-processing

In this study, the NGS data from NorthWater is used, in particular data from the WWTP in Oosterhorn. This dataset consists out of 32 samples. The first sample was taken in week 42 of 2014 and the last sample was taken in week 6 of 2017. This dataset is created by BioClear and was made available for this thesis.

4.1.1 Choice of taxonomical rank

The dataset consists of multiple sheets of data, each sheet representing one hierarchical rank of taxonomy. The available taxonomical ranks in the dataset are (in hierarchical order): Class, Order, Family and Genus.

For this research, the genus rank is chosen to develop and build the models. This rank shows the highest level of detail of the four available ranks. However, the dimensionality of this rank is also the highest. 1236 different genera in total have been measured in all 32 samples.

The genus dataset is setup as shown in Table 4.1 (redundant columns have been omitted), where the first three rows out of a total 11881 rows are shown. The Result-column shows the relative occurrence of the genus from column 'Parameter' at a particular sample. This negative value is the base 10 logarithm of the sequence ratio. The sequence ratio is the number of sequences related to that genus divided by the total amount of sequences of the sample. Only genera with at least three sequences are added to the dataset. Genera with less than three sequences found in the sample are omitted from the dataset for that particular sample. This threshold is called the detection limit and differs per sample, since the number of sequences per sample varies.

Table 4.1: NGS Genera Dataset

Year Sample	Weeknumber	Parameter	Result
2014	42	Acetobacterium	-2,225221459
2014	42	Acholeplasma	-3,857359972
2014	42	Acidaminobacter	-3,918057419
...	...	...	...

4.1.2 Microbial community analysis

The total number of sequences detected in the samples ranged between 1 million to 10 million. From these sequences, 1236 unique genera were identified across all samples. Per sample, an average of 368 genera are detected, ranging between 139 and 610.

Sedimenticola is the most dominant genus, represented by nearly 9% of all sequences across all samples, followed by *Methanolobus* ($>$ 6%), *Methylophaga* and *Methylomicrobium* (both $>$ 3%) and *Cryptocaryon* ($>$ 2.5%).

4.1.3 Creating matrix

The dataset as shown in Table 4.1 is imported in Python and stored into a 'dataframe', using the pandas library (McKinney, 2011). Then, the dataframe is transformed into a matrix with columns describing the type of genera and rows (or index) describing the sample date by combining the year and weeknumber. The cells are filled with the corresponding values from the column 'Result'. This is done by using the pivot function from pandas.

In the resulting matrix, all values listed in the original dataset are listed in the new matrix. However, for the majority of genera, in one or more sample, the number of strings in the sample was below the detection limit and had no presence in the dataset for that sample. This results in an empty cell for that particular sample and genus. These cells are filled with a value well below the detection limit of that sample. At last, the values are raised to the power of 10.

The resulting matrix is shown in Figure 4.2, where the first five rows are shown. The entire matrix consists out of 1236 columns, equal to the number of unique genera in all samples and 32 rows, equal to the number of samples.

4.1.4 Interpolation

In the NGS dataset, samples are taken on average every four weeks, resulting in 32 total samples. Unfortunately, these samples are not spread evenly over the total period, but differ in interval time. Since the process data has a higher, weekly frequency, the NGS dataset is re-sampled into a weekly frequency as well. This is

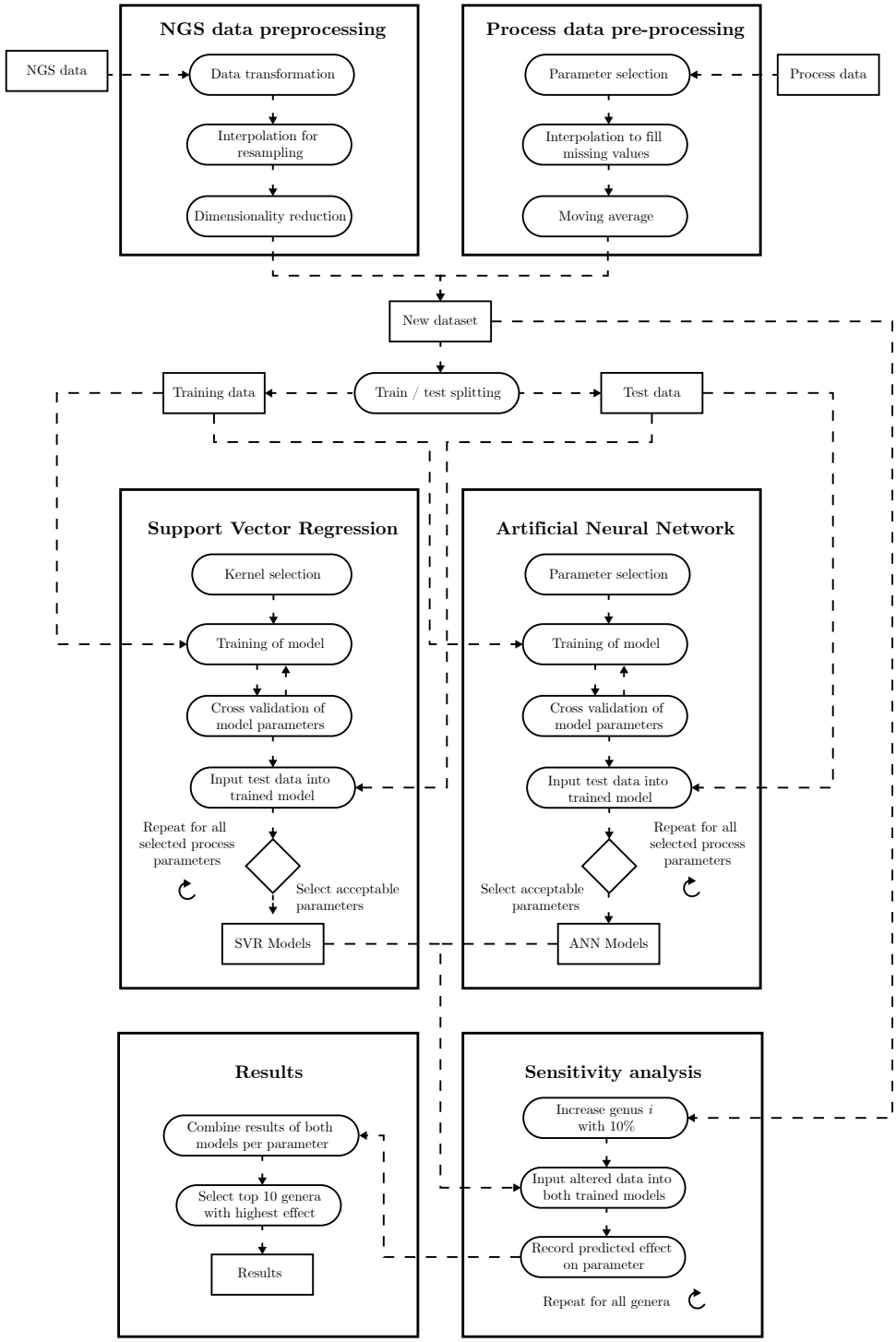

Figure 4.1: Global road map of the methodology

	Sedimenticola	Methanolobus	Methylophaga	Methylomicrobium	Cryptocaryon	Vannella	Arcobacter	Desulfobacterium	Desulfocapsa	Dechloromarinus	...
0	0.010343	0.145335	0.000085	0.162669	0.000235	0.029490	0.012408	0.003490	0.011756	0.000719	...
1	0.012061	0.125933	0.000037	0.141913	0.007414	0.019848	0.009865	0.007878	0.012375	0.001319	...
2	0.013298	0.108500	0.000012	0.123269	0.013032	0.011240	0.007594	0.011852	0.012818	0.001899	...
3	0.014046	0.094329	0.000003	0.108142	0.016695	0.004945	0.005927	0.015195	0.013084	0.002456	...
4	0.014297	0.084713	0.000002	0.097936	0.018003	0.002241	0.005194	0.017693	0.013173	0.002986	...

Figure 4.2: Matrix resulting from the pre-processing using the pandas library in Python.

done using interpolation. A variety of interpolation techniques are available in the pandas module, such as 'linear', 'cubic', 'spline' and 'pchip'. The main two criteria for an interpolation techniques are that it approximates the missing data points in a natural way and that it cannot be lower than zero. Values lower than zero are not acceptable since the occurrence of genera cannot be negative. One of the interpolation techniques that satisfies these two criteria is Piecewise Cubic Hermite Interpolating Polynomial (PCHIP). Therefore, the PCHIP interpolation technique is chosen and applied to the dataset. The result of the PCHIP interpolation and resampling on two genera, *Sedimenticola* and *Vannella*, is shown in Figure 4.3.

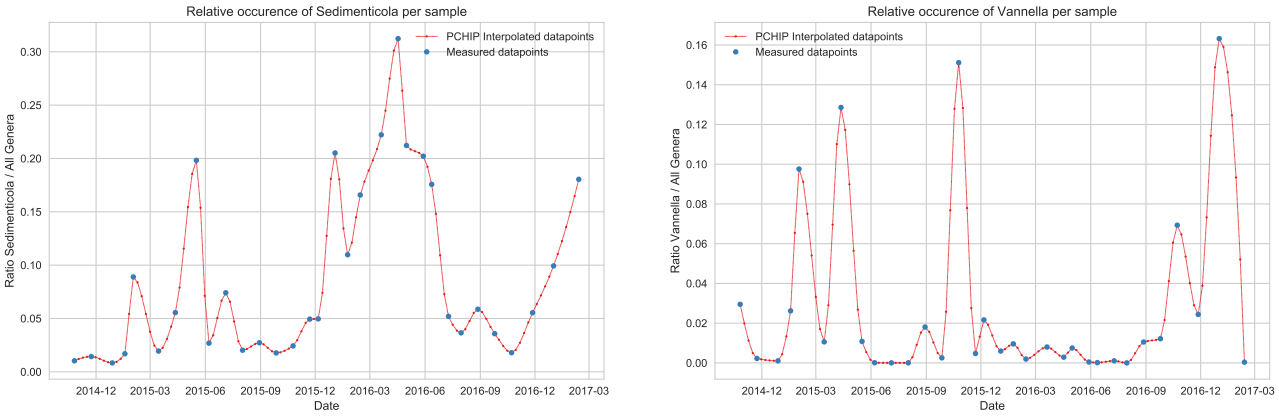

Figure 4.3: Plot of the relative occurrence of Sedimenticola (left) and Vannella (right). Raw data (blue) was interpolated using PCHIP interpolation (red).

4.1.5 Dimensionality reduction

1236 columns or features in a dataset is extremely high for any machine learning model to deal with. Therefore, measures have to be taken in order to reduce the number of features in the dataset, also called dimensionality reduction.

Feature selection

Feature selection is a very simple measure of dimensionality reduction. A predetermined number of features are selected from the total set of features. This is done by sorting the features based on the sum of the values from all sample for that genus.

The main disadvantage of this technique is that (perhaps important) data from omitted genera are left out of the model and will not be taken into account. A technique that doesn't have this disadvantage is PCA.

Principal Component Analysis

Principal Component Analysis (PCA) is a technique described in Section 3.1 that does not omit any features, but instead aims at reducing the number of features into a new set of components. PCA requires the input data to be normalized, so that each input is scaled to the same range.

PCA is carried out using the sci-kit learn library in Python (Pedregosa et al., 2011). In Figure 4.4, the individual and cumulative retained variance per component are shown for the selected dataset.

The retained variance per component reduces quickly after the first components and with 28 components, the cumulative retained variance exceeds 0.99 (Table 4.2). Thus, with that number of components, 99% of the variance from the original dataset is retained. Subsequently, a number of 28 principal components is chosen to proceed with in this research. Concluding, with PCA, the dimensions of the original dataset have been reduced by 97.7% (Equation 4.1), whilst retaining 99% of the information in the dataset.

$$\frac{1236 - 28}{1236} \times 100\% = 97.7\% \tag{4.1}$$

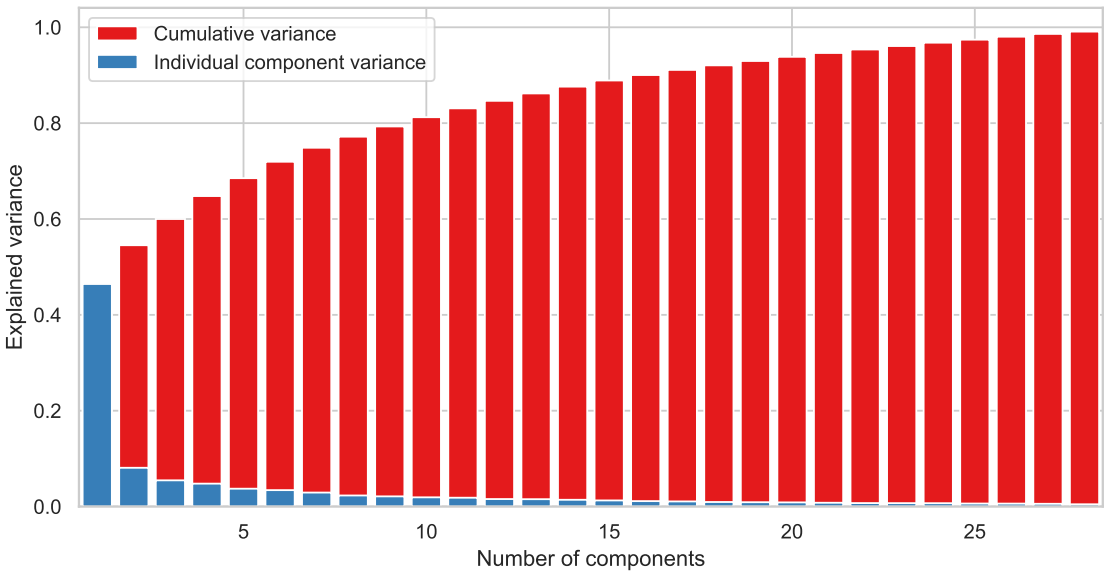

Figure 4.4: Bar plot of the retained variance versus number of principal components.

Table 4.2: Cumulative variance versus number of principal components.

Number of PC	**1**	**2**	**4**	**8**	**10**	**15**	**20**	**25**	**28**
Cum. variance	46.44%	54.53%	64.79%	77.18%	81.24%	88.90%	93.84%	97.42%	99.10%

4.2 Process data preprocessing

In this study, the process data from NorthWater is used, from the same WWTP in Oosterhorn. This dataset consists out of 108 samples, where the first sample was taken in week 40 of 2014 and the last sample was taken in week 10 of 2017.

4.2.1 Parameter selection & grouping

Table 4.3: Selected parameters, grouped.

Influent parameters	**Effluent parameters**	**Aeration Tank parameters**
Influent COD	Effluent COD	Dryweight AT
Influent flow	Effluent pH	Sludge load COD
Influent pH	Effluent EC	Vol aeration
Influent EC	Effluent N	
Influent N	Effluent PO4	
Influent PO4	Effluent SO4	

In Table 4.3, the parameters that are used for this research are shown. These parameters are selected based on the relevance as determined by the WWTP research group. In addition, the importance of these process parameters is described in Section 2.1.3.

The process parameters are grouped based on their sample point in the plant. Figure 4.5 shows three different sample points in the WWTP. Influent parameters are measured at Point 1, Aeration Tank parameters are measured at Point 2 and Effluent Parameters are measured at Point 3.

4.2.2 Interpolation

In the available dataset of the process parameters, there are some missing values for certain weeks. To fill in these values, the same PCHIP technique as in Section 4.1.4 is used. In contrary, the data points are not resampled as done with the NGS dataset, since the frequency of the data is already weekly. The result of the interpolation is shown in Figure 4.6 for the Influent Nitrogen and Effluent BOD parameters.

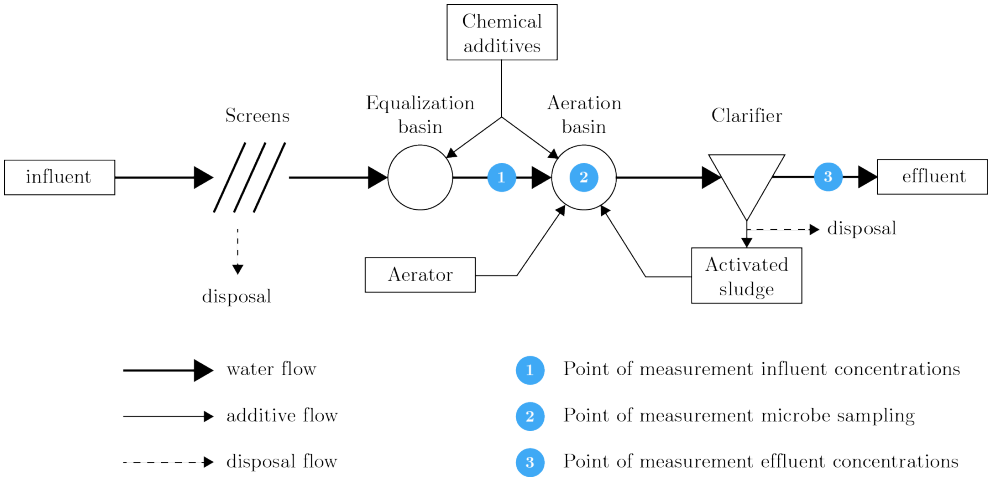

Figure 4.5: Simplified schematic overview of a WWTP at NorthWater

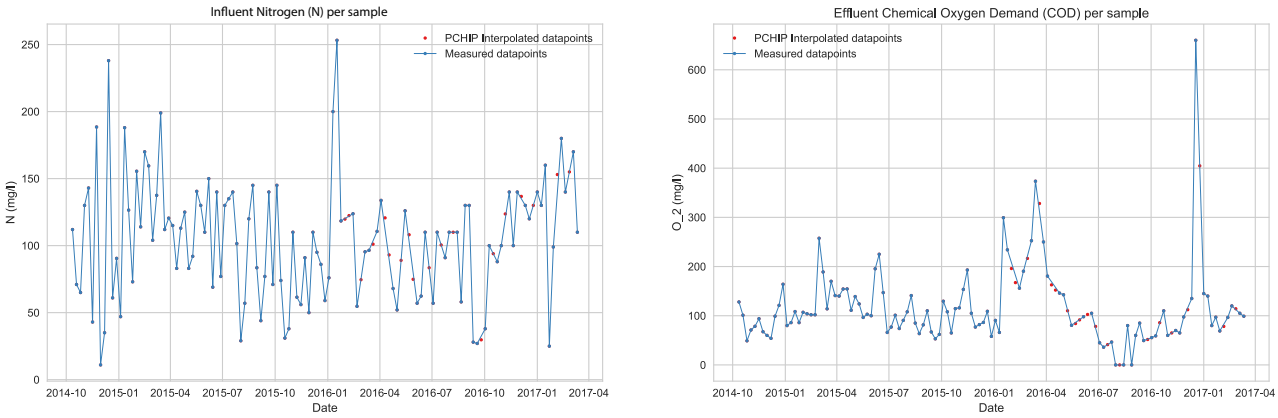

Figure 4.6: Plot of the Influent Nitrogen (left) and Effluent COD (right). Raw data (blue) was interpolated using PCHIP interpolation (red).

4.2.3 Low-pass filter

As seen in the plots of Figure 4.6, the volatility of the process parameters is significantly higher than that of the NGS data in Figure 4.3. A low-pass filter can help overcome this difference by removing the higher frequencies in the data, leaving out a smoother graph.

There is variety of methods how the process data can be smoothed. One of the easiest and most used method is a moving average filter (MA). Each value in the series is recalculated by taking the weighted average of its neighbors. The number of neighbors depends on the chosen size and shape of the window. A new value for x_p, $\hat{x}_p$ is calculated with a window size of N is calculated by

$$\hat{x}_p = \frac{1}{N} \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_{p-i} x_{p-i} \tag{4.2}$$

where $w_p - i$ is the corresponding weight of the value x_{p-i} and N is the window size.

The simplest form of an MA is a rectangular window, where the the weights assigned to all values within the window is equal. The formula for this window function is given by

$$w_{rect}(n) = 1 \qquad 0 \leq n \leq N-1 \tag{4.3}$$

where N is the window size. The disadvantage of this method is that every neighbor of x_p has an equal influence on $\hat{x}_p$. However, in reality, this isn't a realistic representation, as the influence of a sample will decay over time. Therefore, a different window type is chosen for this research: a Hann window, given by

$$w_{Hann}(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \qquad 0 \leq n \leq N-1 \tag{4.4}$$

where N is the window size. Both window types are shown in Figure 4.7. A window size of 7 was chosen, based on trial and error. This results in a smoother graph as shown in Figure 4.10(a).

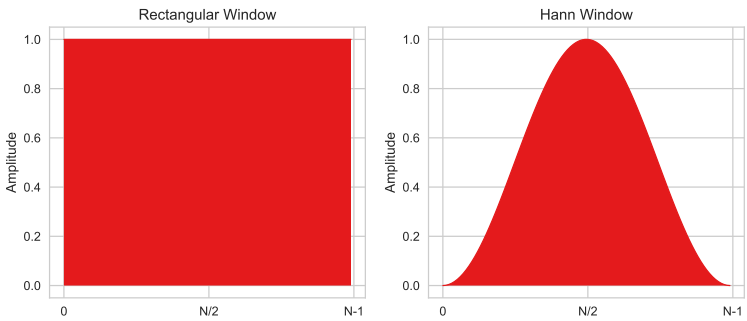

Figure 4.7: Plot of window functions of a rectangular window (left) and a Hann Window (right).

4.2.4 Lag

As stated in Section 4.2.1, the selected parameters are measured at different points of the WWTP. Subsequently, the timing of measurement varies. This difference in timing results in a lag of the system. In addition, as a moving average is taken, only values of the water parameters prior to the sample time should be taken into account.

Let t_s be the time the NGS sample is taken from the water. Only values of influent process parameters for $t \leq t_s$ are influencing the microbial communities and thus should be included in the model. On the contrary, for effluent parameters, only values for $t \geq t_s$ should be included. Therefore, parameters from both groups are shifted with a certain number of weeks; a shift of i for influent parameters and e for effluent parameters. The values for i and e are determined by comparing the

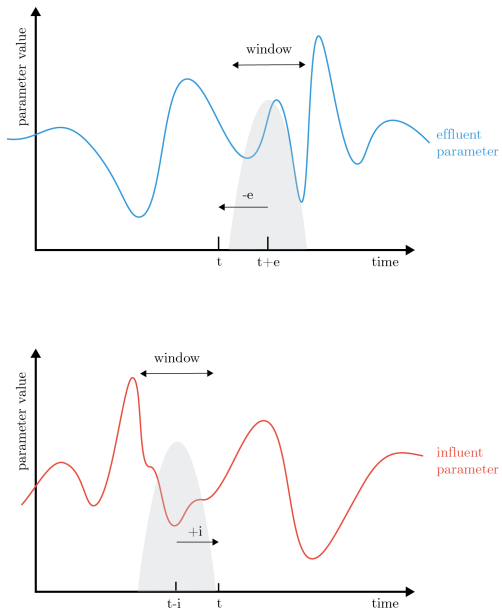

Figure 4.8: Schematic overview of the window function and shift for effluent (top) and influent (bottom) process parameters.

performance of the SVR model for different values of i and e. The performance criteria will be explained in Section 4.4. Figure 4.8 shows schematically how a window is placed for effluent and influent parameters and how the lag (or shift) affects the placement of this window.

Figure 4.9 shows the average R^2 score of the test set for $0 \leq i \leq 10$ and $-10 \leq e \leq 0$ for the influent and effluent parameters, respectively. The values for e and i with the highest score are chosen: $i = 3$ and $e = -3$. Subsequently, the data of the effluent parameter group is shifted three weeks back in time and the data of the influent parameter group is shifted 3 weeks forward in time. The data of the aeration tank is not shifted, since the point and time of measurement matches with the NGS data.

Figure 4.10 shows the effect of an applied moving average filter (left plot) and lag (right plot). As seen from the plots, the high frequencies are filtered out, leaving a smooth graph.

4.2.5 Normalization

According to Haykin et al. (2008), it is important that the target values of a neural network are within the range of the activation function. Therefore, the data is normalized to values in the range $[0, 1]$, where the maximum value of each process parameter is changed to 1 and the minimum value is changed to 0.

Figure 4.9: Plot of the average R^2 scores for influent (left) effluent (right) parameters versus shift in weeks. The optimal shift is marked with a red cross at +3 for influent and -3 for effluent parameters.

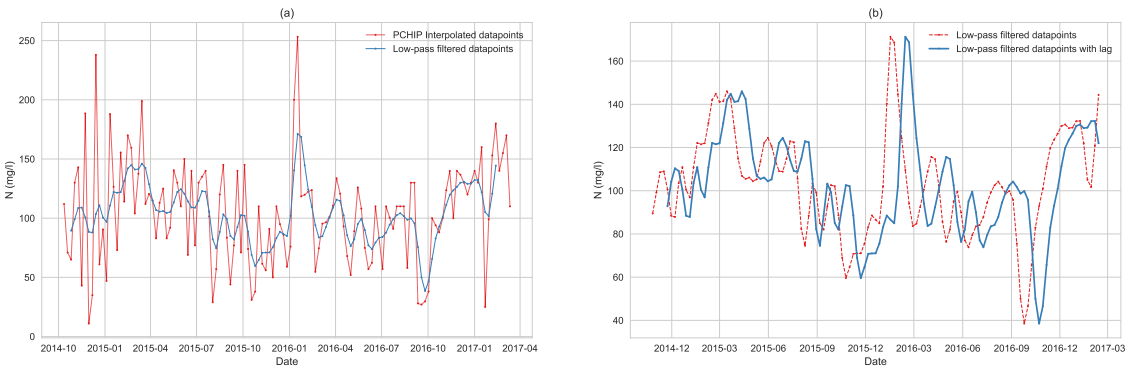

Figure 4.10: (a) Plot of the Influent Nitrogen (in red) and the result of a moving average filter (in blue). (b) Plot of the result of a shift forward in time for Influent Nitrogen

4.3 Splitting the data

It is well known method in machine learning to divide the collected data into three subsets: a training, a validation and a test set. The training set is used to fit the model to the data, the validation set is used to tune the parameters of the model, whereas the test set will be used after the model is created to evaluate the performance of the model. Subsequently, the model will not 'see' the test set until the evaluation.

In this work, 20% of the dataset is used for testing the developed models. The remaining 80% of the dataset is then further divided into a training set and validation set. When there isn't an abundance of data, setting aside data for validation purposes can come at a high cost for the performance of the model. k-Fold Cross Validation can overcome this problem by creating k subsets of the entire dataset (Hastie et al., 2009). The first $k-1$ subsets are chosen as training set and the other subset is used as validation set. This process is repeated k times, such that each subset is used as validation set once. Figure 4.11 shows this schematically with

$k = 5$.

The advantage of this technique is that more data is used for training the model, while the model is still validated properly. An additional advantage of this technique is that it prevents overfitting of the model. Choosing a value for k is a trade-off between computing time and accuracy. Commonly used values are $k = 5$ and $k = 10$. (Fushiki, 2011)

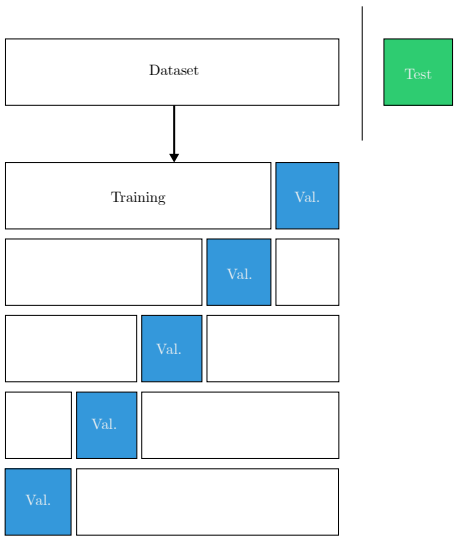

Figure 4.11: K-Fold Cross Validation with k = 5

4.4 Performance criteria

The performance of the models will be assessed on the test dataset using both the mean square error (MSE) and coefficient of determination (R^2). These are commonly used criteria for regression machine learning methods (Seshan et al., 2014; Ay and Kisi, 2014; Guo et al., 2015; Liu et al., 2016; Isa Abba Gozen Elkiran, 2017). Both criteria will be calculated for the test set as well as the training set. These two criteria are calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{4.5}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{4.6}$$

where $\hat{y}_i$ is the predicted output, y_i is the measured output and $\bar{y}_i$ is the mean measured output for sample i. n is the number of samples.

Tbe coefficient R^2 is used to indicate what level of variability is explained by the model. It is well known in trend analysis and has a maximum value of 1. The closer the value is to 1, the better the relationship between two variables (Glantz and Slinker, 1990). On the contrary, the lower the MSE, the higher the accuracy of the regression.

There is no specific rule for the limits of both criteria. Acceptable values for MSE and R^2 depend on the field of research and model. That is why the research by Liu et al. (2016) is used as framework of reference. A $R^2 > 0.8$ and a MSE $<$ 0.008 for the test set was considered as very good by this paper. Subsequently, for this research, a model will be considered as 'good' when it has a $R^2 > 0.75$ and a MSE < 0.015 for both the training as the test set.

4.5 SVR model development

A Support Vector Regression model is implemented using an inbuilt module from the sci-kit learn library in Python, called 'SVM'. This module can be used for both classification and regression. For this research, the regression module 'SVM.SVR()' is used. For all process parameters, a new model is developed, with different model parameters and numbers of support vectors.

The function from sci-kit learn requires six inputs to fit the model to the data:

- training data ($x_{training}$)
- observed training values ($y_{training}$)
- type of kernel
- kernel-related parameter(s)
- C, the regularization cost parameter
- ϵ, determining the ϵ-insensitive loss function

4.5.1 Kernel choice

For this research, the radial basis function (RBF) is chosen for two reasons. First, it has a good general performance, since it is able to cope with nonlinearity. Second, it has only one kernel parameter (γ) that needs to be adjusted (Hsu et al., 2010).

4.5.2 Parameter optimization

Grid search methodology with 5-fold cross validation on the training set is applied to retrieve the optimal values for the model parameters C, ϵ and γ for each of the process parameters. These three parameters are also known as hyperparameters. The search was done by using exponential sequences for C, ϵ and γ in the ranges:

Table 4.4: Optimized SVR parameters.

Parameter	C	γ	ϵ	Support Vectors
Dryweight AT	128.0	8.0	0.031623	58
Effluent COD	32768.0	1.0	0.003162	91
Effluent EC	32768.0	2.0	0.031623	56
Effluent N	4096.0	2.0	0.001	94
Effluent pH	512.0	4.0	0.003162	77
Effluent PO4	2048.0	8.0	0.01	78
Effluent SO4	16384.0	2.0	0.003162	86
Influent BOD	128.0	8.0	0.003162	94
Influent COD	128.0	8.0	0.003162	89
Influent EC	512.0	8.0	0.01	70
Influent Flow	16384.0	4.0	0.001	93
Influent N	512.0	8.0	0.01	72
Influent PO4	32.0	8.0	0.001	95
Sludge load COD	16384.0	0.125	0.01	76
Vol. Aeration	128.0	2.0	0.001	87

$$
\begin{aligned}
C &= [2^1, 2^3, 2^5, \ldots, 2^{15}] \\
\gamma &= [2^{-15}, 2^{-13}, 2^{-11}, \ldots, 2^3] \\
\epsilon &= [2^{-11}, 2^{-9}, 2^{-7}, \ldots, 2^{-1}]
\end{aligned}
$$

These three ranges result in a total of 420 different combinations. In combination with the 5-fold cross validation, this resulted in 2100 runs of the model per process parameter. The optimized SVR hyperparameters per process parameter are shown in Table 4.4.

4.5.3 Test set validation

After fitting the model with the optimized model parameters, the test set is used as input for the model. Since the model hasn't 'seen' this dataset yet, it is a validation of the model. The test output is compared with the measured output using the performance criteria MSE and R^2.

Figure 4.12 shows the plots of all process parameters where measured output is compared with predicted output of both the test set as the training set. Ideally, the predicted values are equal to the actual measured values. In that case, the data points are placed on the line $y = x$. The results show that the large majority of points on the plots are very close to this ideal line.

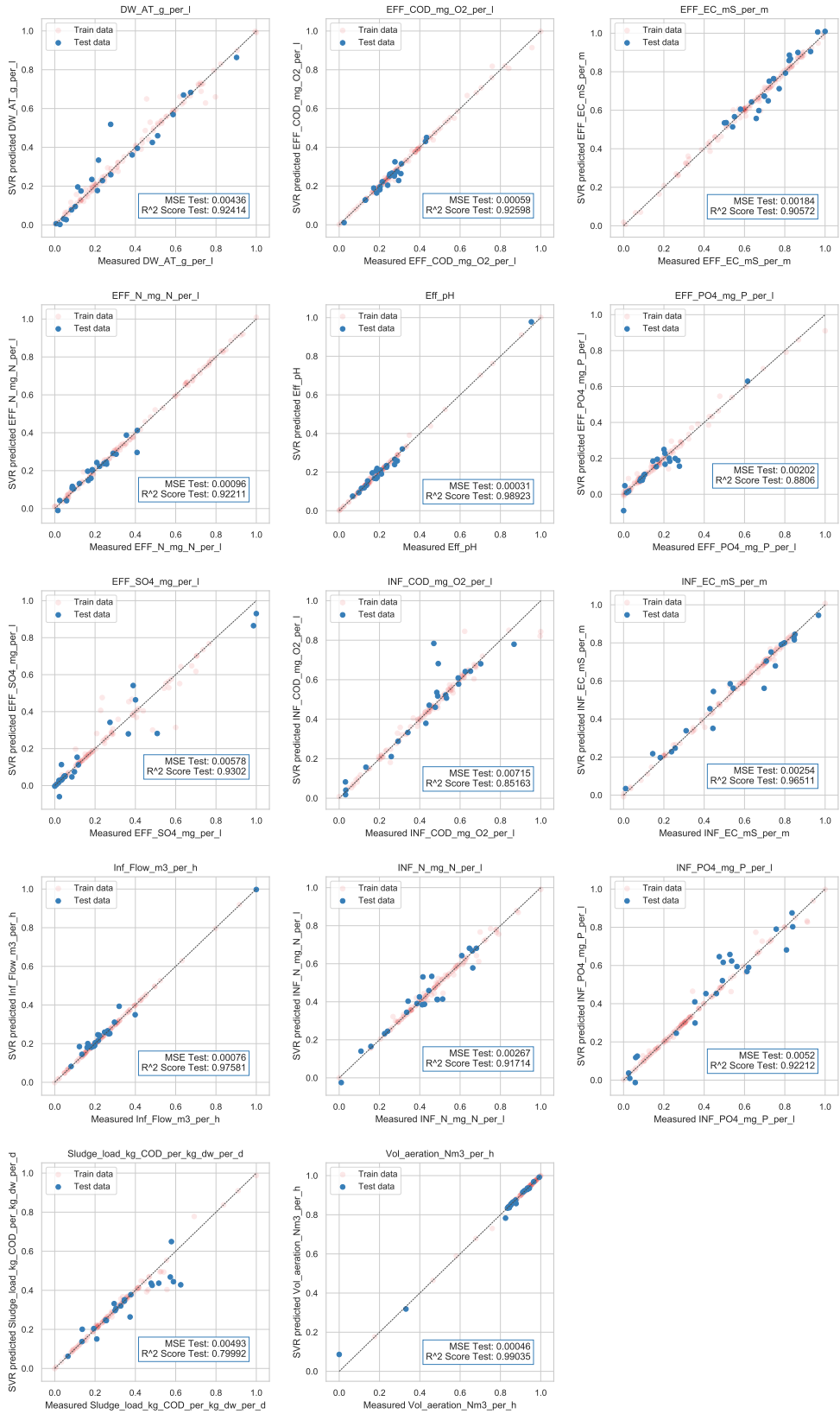

Figure 4.12: SVR model results showing predicted concentrations versus corresponding measured (normalized) concentrations for all process parameters. A distinction is made between the test set (in blue) and training set (red). The dotted line shows the ideal $y = x$ line that would perfectly fit the dataset.

4.6 Neural network model development

The Neural Network is modeled using the tensorflow library, created by the Google Brain team (Abadi et al., 2016). This open source library is designed for machine learning. Its extensive documentation and possibilities for finetuning the the model are the main reasons this library is chosen.

The model is set to run for 4000 iterations. The solver used in this neural network is 'AdaGrad' a gradient descent optimization method. Mean squared error is used as loss function.

Since modeling of an NN is computationally heavy, the number of parameters that can be optimized is limited. Two parameters are chosen to be fine-tuned by cross-validation:

1. Hidden layer composition. This parameter defines the number of hidden layers and the number of neurons per layer.
2. Activation function. The type of activation function determines the output of each neuron, including that of the final output neuron.

Due to the high required computing power for a model to run, a set of six combinations of hidden layers and three activation functions are investigated, resulting in a total of eighteen options. The notation $(5,)$ indicates only one hidden layer of five neurons, whereas $(5,5)$ means two hidden layers of equal size of five neurons. The following ranges for the model parameters are used.

$$\begin{aligned} \text{Hidden layers} &= [(5,),(15,),(30,),(5,5),(15,15),(30,30)] \\ \text{Activation function} &= [\text{tanh, sigmoid, ReLu}] \end{aligned}$$

The best performing activation function is the sigmoid function, based on both the R^2-score and the MSE for the test set. This makes sense, since the target values of the output (i.e. the process parameters) are normalized to a range of [0,1]. The sigmoid activation function will also return values between 0 and 1, as seen in Figure 3.6. The sigmoid activation function is depicted in Equation 4.7.

A neural network with two hidden layers outperforms a single hidden layer network on all process parameters. The network with two hidden layers of 15 neurons is chosen.

$$g(z) = \sigma(z) = \frac{1}{1+e^{-z}} \tag{4.7}$$

4.6.1 Test set validation

For this model, the same procedure as in Section 4.5.3 is followed. The test output is compared with the measured output using the performance criteria MSE and R^2.

Figure 4.13 shows the plots of all process parameters where measured output is compared with predicted output of both the test set as the training set. As said in Section 4.5.3, the ideal situation is where all points are on the line $y = x$. For the ANN model, all points are very close to this line.

4.7 Sensitivity analysis

Both the SVR and NN are black-box models. Thus, the intrinsic relations between the inputs of microbial communities and the predicted outputs of process parameters are not known. A sensitivity analysis (SA) was performed on all selected trained models, to show the relative importance of each genus to the prediction of the process parameters. SA is performed on both the SVR as the NN, so that both sensitivities are compared, resulting in a higher robustness of the sensitivity.

One-Factor-At-a-Time (OFAT) technique is used for this SA. OFAT is a local sensitivity analysis technique, only measuring the influence of a genus on a process parameter for a certain change of that genus. Thus, only a *local* area of the influence of each genus is explored. Algorithm 1 shows a pseudo code of this technique.

for *all selected process parameters* j **do**
  load model;
  load all input variables x;
  y_j = model.predict(x);
  for *all genera* i **do**
    $x^* = x$;
    $x_i^* = x_i * 1.1$;
    $x * [x_i] = x[x_i^*]$;
    y_j^* = model.predict(x);
    $\Delta y_j = y_j^* - y_j$;
    $S_{y_j \leftarrow x_i} = \frac{\frac{\Delta y_j}{y_j}}{\frac{\Delta x_i}{x_i}}$;
  end
end

Algorithm 1: One Factor At A Time pseudocode

The variable $S_{y_j \leftarrow x_i}$ from Algorithm 1 is the sensitivity of a change in output y_j to a change in the input x_i. j is a range of all selected process parameters and i is a range of all genera in the NGS dataset.

The result of this technique is a matrix of size $[i, j]$ with the effect of an increase of 10% of each genus i on the prediction of each selected process parameter j. The top 10 highest absolute values for $S_{y_j \leftarrow x_i}$ per parameter j are selected and shown in Chapter 5.

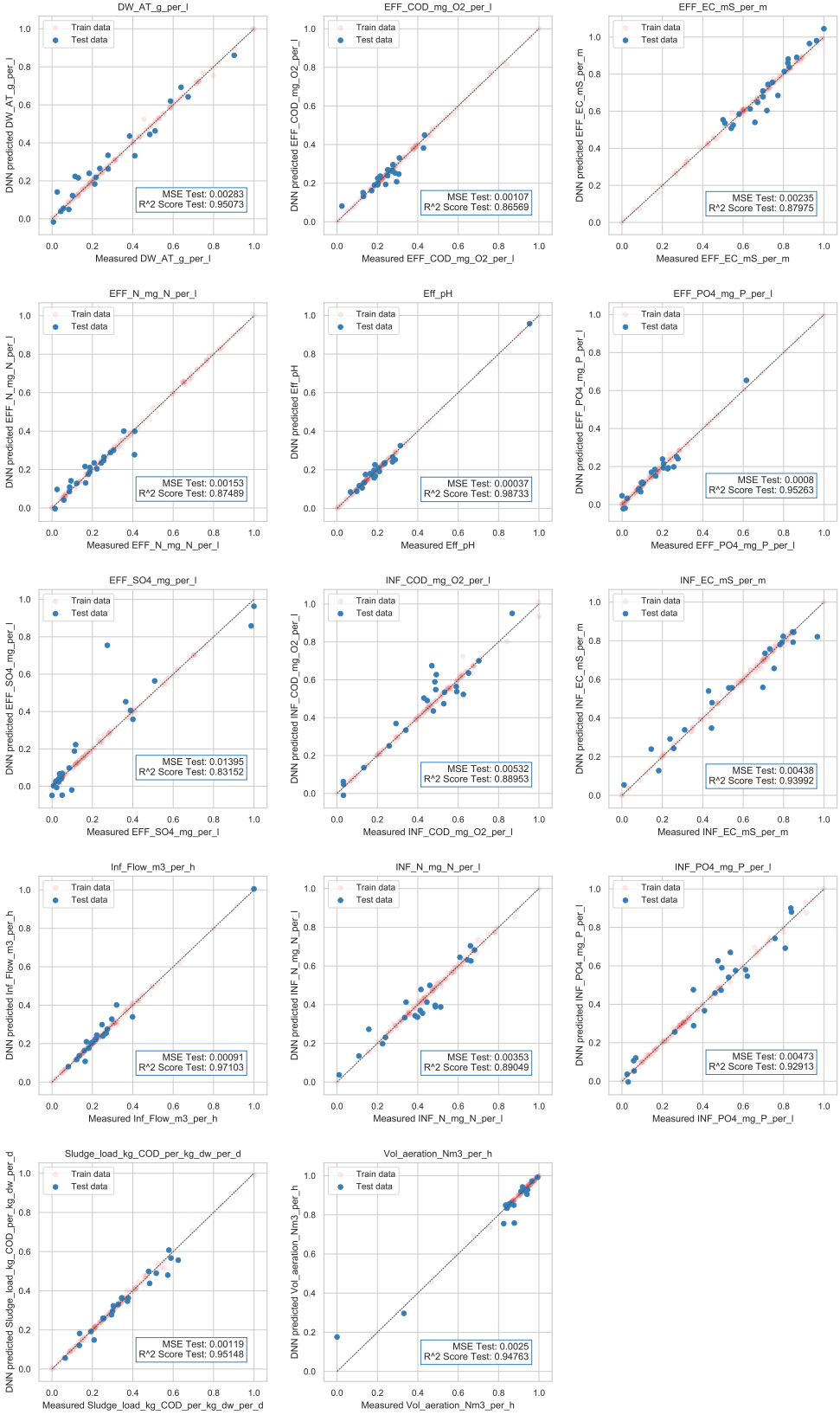

Figure 4.13: NN model results showing predicted concentrations versus corresponding measured (normalized) concentrations for all process parameters. A distinction is made between the test set (in blue) and training set (red). The dotted line shows the ideal $x = y$ line that would perfectly fit the dataset.

Chapter 5

Results

5.1 Model performance

Table 5.1 shows the performance of both models in the test stage. For nine of the process parameters, the SVR model outperforms the neural network. Five of the neural network models had a higher accuracy. In general, all models scored very good according to the performance criteria, since all models had an R^2 score higher than 0.8 and a MSE < 0.015. Therefore, all process parameters are selected for the sensitivity analysis.

Table 5.1: Mean square error (MSE) and coefficient of determination (R^2) for the test set of all process parameters. Best scoring values for each process parameter are made bold.

	SVR		**DNN**	
Parameter	**MSE**	$\boldsymbol{R^2}$	**MSE**	$\boldsymbol{R^2}$
Dryweight AT	0.00436	0.92414	**0.00283**	**0.95073**
Effluent COD	**0.00059**	**0.92598**	0.00107	0.86569
Effluent EC	**0.00184**	**0.90572**	0.00235	0.87975
Effluent N	**0.00096**	**0.92211**	0.00153	0.87489
Effluent pH	**0.00031**	**0.98923**	0.00037	0.98733
Effluent PO4	0.00202	0.8806	**0.0008**	**0.95263**
Effluent SO4	**0.00578**	**0.9302**	0.01395	0.83152
Influent COD	0.00715	0.85163	**0.00532**	**0.88953**
Influent EC	**0.00254**	**0.96511**	0.00438	0.93992
Influent Flow	**0.00076**	**0.97581**	0.00091	0.97103
Influent N	**0.00267**	**0.91714**	0.00353	0.89049
Influent PO4	0.0052	0.92212	**0.00473**	**0.92913**
Sludge load COD	0.00493	0.79992	**0.00119**	**0.95148**
Vol. Aeration	**0.00046**	**0.99035**	0.0025	0.94763

5.2 Results of sensitivity analysis

Figure 5.1 shows the results of the sensitivity analysis. For each process parameter, the sensitivity ranking of the top 10 genera are shown.

Both models showed very similar results for all parameters. The correlation between the results of the SVR model and the NN model ranged from 0.58 to 0.92 for all process parameters (Table 5.2). Since both models indicate similar sensitivities, the plausibility of this analysis increases.

Table 5.2: Correlation coefficient (Pearson) between OFAT results of DNN and SVR models per process parameter.

Parameter	Correlation
Dryweight AT	0.78
Effluent COD	0.87
Effluent EC	0.90
Effluent N	0.76
Effluent pH	0.93
Effluent PO4	0.87
Effluent SO4	0.58
Influent COD	0.93
Influent EC	0.74
Influent flow	0.60
Influent N	0.69
Influent PO4	0.88
Sludge load COD	0.89
Vol aeration	0.90

It is difficult to further assess the plausibility of this sensitivity analysis, due to two reasons. First, the functions of many genera are still unknown. Second, a strong sensitivity doesn't automatically mean that there is a causal relation between two factors. Nonetheless, for the following process parameters, a possible explanation was found for the strong sensitivity.

Sulfate

It is expected that genera from the group of sulfate-reducing bacteria (SRB) appear in the top 10 genera for the process parameter effluent sulfate. Higher concentrations of these bacteria would theoretically result in a lower effluent sulfate as SRBs reduce sulfate to sulfide. Thus, a negative relation is expected. From the top 10, three genera are known as sulfate-reducing, to wit *Desulfocapsa*, *Desulfovibrio* and *Desulfomicrobium*, all with significant negative sensitivities. Furthermore, these three genera are the only ones from the major groups of SRBs (Gerardi, 2006) that were present in the majority of the samples in the NGS data.

Nitrogen

Dechloromonas, a genus frequently found in WWTPs, is known as a nitrate reducing bacterium (NRB) (Liu et al., 2005). The same applies to the genus *Thalassospira* (Kodama et al., 2008; Kamarisima et al., 2018). This corresponds with their negative sensitivity to the predicted effluent nitrogen.

Electrical conductivity

This parameter is a general indicator of water quality, especially a function of the amount of dissolved salt (i.e. NaCl concentration). For the influent EC, *Boseongicola* and *Thalassospira* showed the highest sensitivity with the predicted process parameters (all positive). Park et al. (2014) found that *Boseongicola* grow in the presence of dissolved salt (0.5-5.0% NaCl). Thus, a higher EC in the influent could possibly be a reason that the occurrence of this genus increases. Similar research results were found for species of *Thalassospira* (Plotnikova et al., 2011; Zhao et al., 2010).

Phosphate

Polyphosphate-accumulating organisms (PAOs) are able to store phosphorous. *Candidatus Accumulibacter* and *Tetrasphaera* are identified as the most important PAOs (Carvalho et al., 2007; Crocetti et al., 2000). Unfortunately, the latter was not represented enough in the dataset. *Candidatus Accumulibacter* was only present in 7 out of the 32 samples and showed a sensitivity of -1.56% to effluent phosphate. Two other PAOs present in the dataset, *Pseudomonas* and *Vibrio* (Gerardi, 2006), had negligible negative sensitivity values with the effluent phosphate parameter.

5.3 Comparing results with existing work

Liu et al. (2016) used SVR to research the influence of microorganisms on five effluent process parameters: BOD, Suspended Solids (SS), Total Nitrogen (TN), NH_4^+-N and Total Phosphorous (TP). Unfortunately, the results of SVR models for TP and NH_4^+-N were not satisfactory and a sensitivity analysis was only applied to the remaining three process parameters. From these three parameters, only TN corresponds with a process parameter used in this thesis.

It is difficult to compare the articles' results with the results of this research. Most of the genera names that are listed in the top 10 sensitivity parameters for TN are either *other* or *uncultured*. When we look at a higher hierarchical level (family), we find that two genera from the articles' top 10 belong to the family *Rhodocyclaceae*. *Dechloromonas* belongs to that family and ranks number three in this thesis' top 10 of the effluent nitrogen. The other families of the articles top 10 do not correspond with this thesis' top 10.

Seshan et al. (2014) also used SVR to research process parameters, namely COD, nitrogen and 3-CA. Unfortunately, they did not include a list of genera that have a high sensitivity to those process parameters.

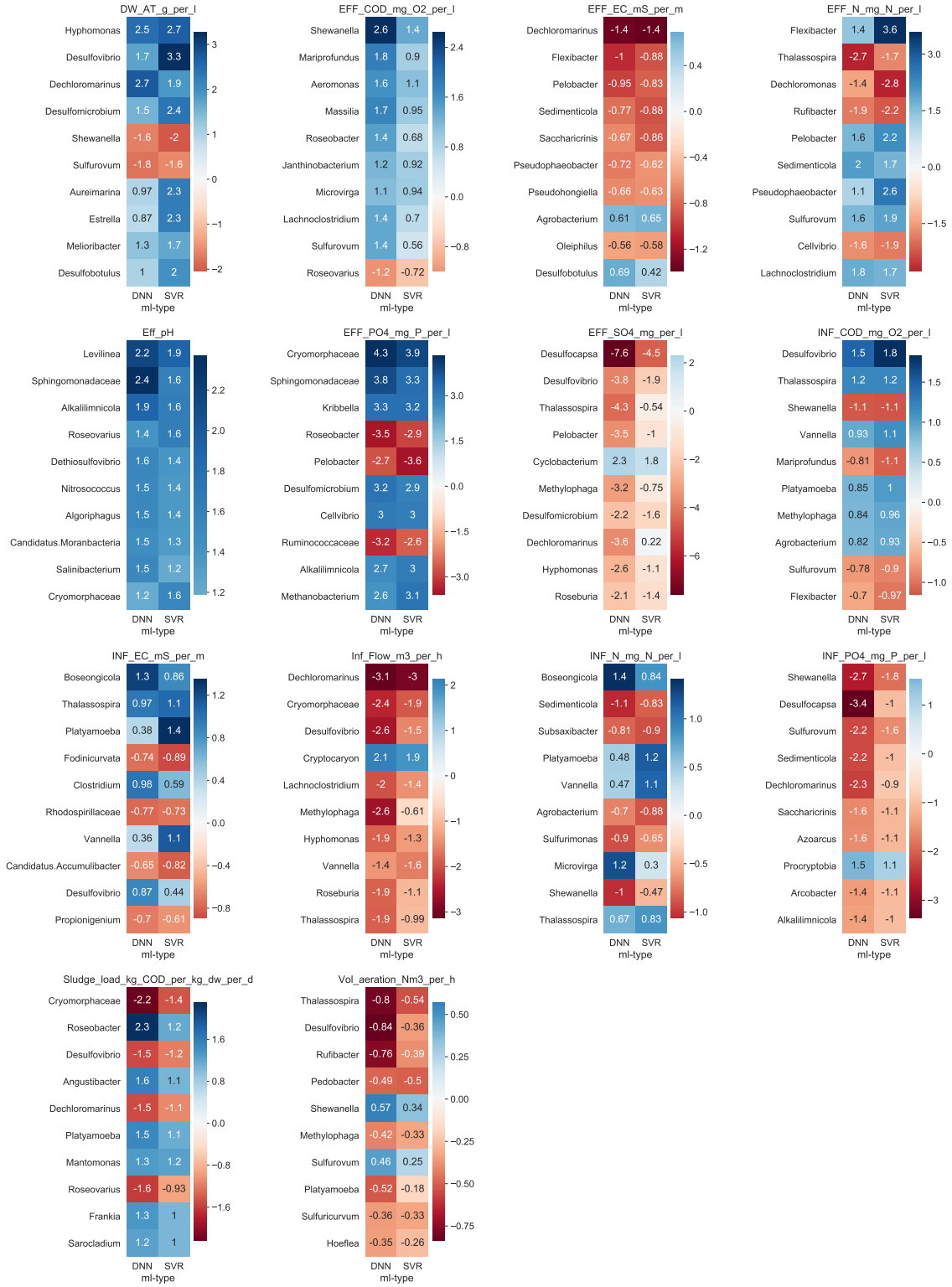

Figure 5.1: Comparison of OFAT results for selected process parameters. Each graph shows a heatmap of the top 10 genera with the strongest sensitivity. The left side shows the results of the Neural Network and the right side shows the results of the Support Vector Regression.

Chapter 6

Conclusion

6.1 Conclusion

The research question: *'Can machine learning be used to model the relationship between microbial communities and parameters of the process?'* is answered by applying two regression techniques to data from NorthWater WWTP. A dataset with microorganisms, derived from next-generation sequencing, was used as input for the model. In order to reduce the dimensionality of this input, PCA was used to reduce the number of input variables by 97.7%. The input data was fed into a support vector regression model as well as a neural network. After model parameter optimization and training of the models, both models showed beyond satisfactory results in the training and test stages.

By analyzing the (local) sensitivity of each modeled process parameter to each input (each genera), an indication of the influence of the microbial structure on process performance was found. Some of these sensitivity scores can be explained by literature, but most of them still remain unknown.

Concluding, machine learning models could be a reliable method for modeling the relationship between microbial communities and parameters of the process. However, more research is needed to assess the plausibility of the results.

6.2 Further research

Following is a list of points on how this research can be continued.

- Increase the frequency of NGS sampling.
- Apply a global sensitivity analysis.
- Use datasets from different WWTPs as input for the models as comparison.
- Build a tool that incorporates microbial communities in predicting effluent concentration. This is also the suggestion of Seshan et al. (2014).

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., and Brain, G. (2016). TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*.

Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459.

Ay, M. and Kisi, O. (2014). Modelling of chemical oxygen demand by using ANNs, ANFIS and k-means clustering techniques. *Journal of Hydrology*, 511:279–289.

Bagheri, M., Mirbagheri, S. A., Bagheri, Z., and Kamarkhani, A. M. (2015). Modeling and optimization of activated sludge bulking for a real wastewater treatment plant using hybrid artificial neural networks-genetic algorithm approach. *Process Safety and Environmental Protection*, 95:12–25.

Bassin, J. P., Rachid, C. T., Vilela, C., Cao, S. M., Peixoto, R. S., and Dezotti, M. (2017). Revealing the bacterial profile of an anoxic-aerobic moving-bed biofilm reactor system treating a chemical industry wastewater. *International Biodeterioration & Biodegradation*, 120:152–160.

Burges, C. J. C. (2009). Dimension Reduction: A Guided Tour. *Foundations and Trends® in Machine Learning*, 2(4):275–364.

Campos, J., Otero, L., Franco, A., Mosquera-Corral, A., and Roca, E. (2009). Ozonation strategies to reduce sludge production of a seafood industry WWTP. *Bioresource Technology*, 100(3):1069–1073.

Caporaso, J. G., Lauber, C. L., Walters, W. A., Berg-Lyons, D., Huntley, J., Fierer, N., Owens, S. M., Betley, J., Fraser, L., Bauer, M., Gormley, N., Gilbert, J. A., Smith, G., and Knight, R. (2012). Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. *ISME Journal*, 6(8):1621–1624.

Carvalho, G., Lemos, P. C., Oehmen, A., and Reis, M. A. (2007). Denitrifying phosphorus removal: Linking the process performance with the microbial community structure. *Water Research*, 41(19):4383–4396.

Corominas, L., Garrido-Baserba, M., Villez, K., Olsson, G., Cortés, U., and Poch, M. (2018). Transforming data into knowledge for improved wastewater treatment operation: A critical review of techniques.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Côté, M., Grandjean, B. P., Lessard, P., and Thibault, J. (1995). Dynamic modelling of the activated sludge process: Improving prediction using neural networks. *Water Research*, 29(4):995–1004.

Crocetti, G. R., Hugenholtz, P., Bond, P. L., Schuler, A., Keller, J., Jenkins, D., and Blackall, L. L. (2000). Identification of polyphosphate-accumulating organisms and design of 16S rRNA-directed probes for their detection and quantitation. *Applied and environmental microbiology*, 66(3):1175–82.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., and Vapnik, V. (1996). Support vector regression machines.

Eckenfelder, W. and Grau, P. (1998). *Activated sludge: Process Design and Control, Second Edition*. Technomic, Lancaster.

Eitrich, T. and Lang, B. (2006). Efficient optimization of support vector machine learning parameters for unbalanced datasets. *Journal of Computational and Applied Mathematics*, 196(2):425–436.

Fernandez de Canete, J., Del Saz-Orozco, P., Baratti, R., Mulas, M., Ruano, A., and Garcia-Cerezo, A. (2016). Soft-sensing estimation of plant effluent concentrations in a biological wastewater treatment plant using an optimal neural network. *Expert Systems with Applications*, 63:8–19.

Foladori, P., Andreottola, G., and Ziglio, G. (2010). *Sludge reduction technologies in wastewater treatment plants*. IWA Publishing.

Friha, I., Karray, F., Feki, F., Jlaiel, L., and Sayadi, S. (2014). Treatment of cosmetic industry wastewater by submerged membrane bioreactor with consideration of microbial community dynamics. *International Biodeterioration & Biodegradation*, 88:125–133.

Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing*, 21(2):137–146.

Gerardi, M. H. (2006). *Wastewater bacteria*, volume 5. John Wiley & Sons.

Glantz, S. A. and Slinker, B. K. (1990). *Primer of applied regression and analysis of variance*. McGraw-Hill, Health Professions Division, New York.

Goel, R. K., Flora, J. R. V., and Chen, J. P. (2005). Flow Equalization and Neutralization. In *Physicochemical Treatment Processes*, pages 21–45. Humana Press, Totowa, NJ.

Grady Jr, C. L., Daigger, G. T., and Lim, H. C. (1998). *Biological Wastewater Treatment, Second Edition, Revised and Expanded*. Marcel Dekker, INC, New York, 2nd edition.

Guo, H., Jeong, K., Lim, J., Jo, J., Kim, Y. M., Park, J. p., Kim, J. H., and Cho, K. H. (2015). Prediction of effluent concentration in a wastewater treatment plant using machine learning models. *Journal of Environmental Sciences (China)*, 32:90–101.

Guo, W. Q., Yang, S. S., Xiang, W. S., Wang, X. J., and Ren, N. Q. (2013). Minimization of excess sludge production by in-situ activated sludge treatment processes - A comprehensive review.

Hamed, M. M., Khalafallah, M. G., and Hassanien, E. A. (2004). Prediction of wastewater treatment plant performance using artificial neural networks. *Environmental Modelling & Software*, 19(10):919–928.

Han, H.-G. and Qiao, J.-F. (2012). Prediction of activated sludge bulking based on a self-organizing RBF neural network. *Journal of Process Control*, 22(6):1103–1112.

Han, J., Kamber, M., and Pei, J. (2011). *Data mining : concepts and techniques*. Elsevier Science.

Handelsman, J. (2005). Metagenomics: Application of Genomics to Uncultured Microorganisms. *Microbiology and Molecular Biology Reviews*, 69(1):195–195.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY.

Haykin, S., Mcdowell, L., Manager, M., and Galligan, T. (2008). *Neural Networks and Learning Machines Third Edition*.

Henze, M., Gujer, W., Mino, T., and van Loosedrecht, M. (2000). Activated Sludge Models ASM1, ASM2, ASM2d and ASM3. *Water Intelligence Online*, 5:130.

Hong, Y.-S. T., Rosen, M. R., and Bhamidimarri, R. (2003). Analysis of a municipal wastewater treatment plant using a neural network-based pattern analysis. *Water Research*, 37(7):1608–1618.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2010). A Practical Guide to Support Vector Classification. Technical report.

Isa Abba Gozen Elkiran, S. (2017). Effluent prediction of chemical oxygen demand from the astewater treatment plant using artificial neural network application. *Procedia Computer Science*, 120:156–163.

Jolliffe, I. (2011). Principal Component Analysis. In *International Encyclopedia of Statistical Science*, pages 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kamarisima, Hidaka, K., Miyanaga, K., and Tanji, Y. (2018). The presence of nitrate- and sulfate-reducing bacteria contributes to ineffectiveness souring control by nitrate injection. *International Biodeterioration and Biodegradation*, 129(December 2017):81–88.

Kodama, Y., Stiknowati, L. I., Ueki, A., Ueki, K., and Watanabe, K. (2008). Thalassospira tepidiphila sp. nov., a polycyclic aromatic hydrocarbon-degrading bacterium isolated from seawater. *INTERNATIONAL JOURNAL OF SYSTEMATIC AND EVOLUTIONARY MICROBIOLOGY*, 58(3):711–715.

Liu, T., Liu, S., Zheng, M., Chen, Q., and Ni, J. (2016). Performance assessment of full-scale wastewater treatment plants based on seasonal variability of microbial communities via high-throughput sequencing. *PLoS ONE*, 11(4).

Liu, Y., Zhang, T., and Fang, H. H. (2005). Microbial community analysis and performance of a phosphate-removing activated sludge. *Bioresource Technology*, 96(11):1205–1214.

Lou, I. and Zhao, Y. (2012). Sludge Bulking Prediction Using Principle Component Regression and Artificial Neural Network. *Mathematical Problems in Engineering*, 2012:1–17.

Mardis, E. R. (2008). The impact of next-generation sequencing technology on genetics. *Trends in genetics : TIG*, 24(3):133–41.

Martín de la Vega, P., Jaramillo-Morán, M., Martín de la Vega, P. T., and Jaramillo-Morán, M. A. (2018). Obtaining Key Parameters and Working Conditions of Wastewater Biological Nutrient Removal by Means of Artificial Intelligence Tools. *Water*, 10(6):685.

McKinney, W. (2011). pandas: a Foundational Python Library for Data Analysis and Statistics. *PyHPC*, pages 1–9.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math.

Mjalli, F. S., Al-Asheh, S., and Alfadala, H. E. (2007). Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance. *Journal of Environmental Management*, 83(3):329–338.

Moral, H., Aksoy, A., and Gokcay, C. F. (2008). Modeling of the activated sludge process by using artificial neural networks with automated architecture screening. *Computers & Chemical Engineering*, 32(10):2471–2478.

Muralikrishna, I. V. and Manickam, V. (2017). Wastewater Treatment Technologies. In *Environmental Management*, pages 249–293. Elsevier.

Muszyński, A., Tabernacka, A., and Miłobędzka, A. (2015). Long-term dynamics of the microbial community in a full-scale wastewater treatment plant. *International Biodeterioration & Biodegradation*, 100:44–51.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

NorthWater (2014). Sustainable processing of wastewater: Saline Wastewater Treatment Plant (SWWTP).

Ofiteru, I. D., Lunn, M., Curtis, T. P., Wells, G. F., Criddle, C. S., Francis, C. A., and Sloan, W. T. (2010). Combined niche and neutral effects in a microbial wastewater treatment community. *Proceedings of the National Academy of Sciences*, 107(35):15345–15350.

Park, S., Park, J.-M., Lee, K.-C., Bae, K. S., and Yoon, J.-H. (2014). Boseongicola aestuarii gen. nov., sp. nov., isolated from a tidal flat sediment. *INTERNATIONAL JOURNAL OF SYSTEMATIC AND EVOLUTIONARY MICROBIOLOGY*, 64(Pt 8):2618–2624.

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

Plotnikova, E. G., Anan'ina, L. N., Krausova, V. I., Ariskina, E. V., Prisyazhnaya, N. V., Lebedev, A. T., Demakov, V. A., and Evtushenko, L. I. (2011). Thalassospira permensis sp. nov., a new terrestrial halotolerant bacterium isolated from a naphthalene-utilizing microbial consortium. *Microbiology*, 80(5):703–712.

Scholkopf, B. and Smola, A. J. (2002). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT Press.

Scholz, M. (2006). *Wetland systems to control urban runoff*. Elsevier.

Seshan, H., Goyal, M. K., Falk, M. W., and Wuertz, S. (2014). Support vector regression model of wastewater bioreactor performance using microbial community diversity indices: Effect of stress and bioaugmentation. *Water Research*, 53:282–296.

Seviour, R. and Nielsen, P. H. (2010). *Microbial Ecology of Activated Sludge*. IWA Publishing Company, London.

Shchegolkova, N. M., Krasnov, G. S., Belova, A. A., Dmitriev, A. A., Kharitonov, S. L., Klimina, K. M., Melnikova, N. V., and Kudryavtseva, A. V. (2016). Microbial Community Structure of Activated Sludge in Treatment Plants with Different Wastewater Compositions. *Frontiers in microbiology*, 7:90.

Soon, W. W., Hariharan, M., and Snyder, M. P. (2014). High-throughput sequencing for biology and medicine. *Molecular Systems Biology*, 9(1):640–640.

Wei, X. (2013). Modeling and optimization of wastewater treatment process with a data-driven approach. *Industrial and Systems Engineering Research Conference*.

Werner, J. J., Knights, D., Garcia, M. L., Scalfone, N. B., Smith, S., Yarasheski, K., Cummings, T. A., Beers, A. R., Knight, R., and Angenent, L. T. (2011). Bacterial community structures are unique and resilient in full-scale bioenergy systems. *Proceedings of the National Academy of Sciences of the United States of America*, 108(10):4158–63.

Xu, J. (2014). *Next-generation Sequencing*. Caister Academic Press, Norfolk, UK, 1st edition.

Yamashita, T. and Yamamoto-Ikemoto, R. (2014). Nitrogen and phosphorus removal from wastewater treatment plant effluent via bacterial sulfate reduction in an anoxic bioreactor packed with wood and iron. *International Journal of Environmental Research and Public Health*, 11(9):9835–9853.

Zhao, B., Wang, H., Li, R., and Mao, X. (2010). Thalassospira xianhensis sp. nov., a polycyclic aromatic hydrocarbon-degrading marine bacterium. *International Journal of Systematic and Evolutionary Microbiology*, 60(5):1125–1129.