



UNIVERSITETET I AGDER

Towards Data-Driven Modelling of Saulekilen Wastewater Treatment Plant based on Artificial Intelligence

Magnus Gabriel Ersdal

SUPERVISORS

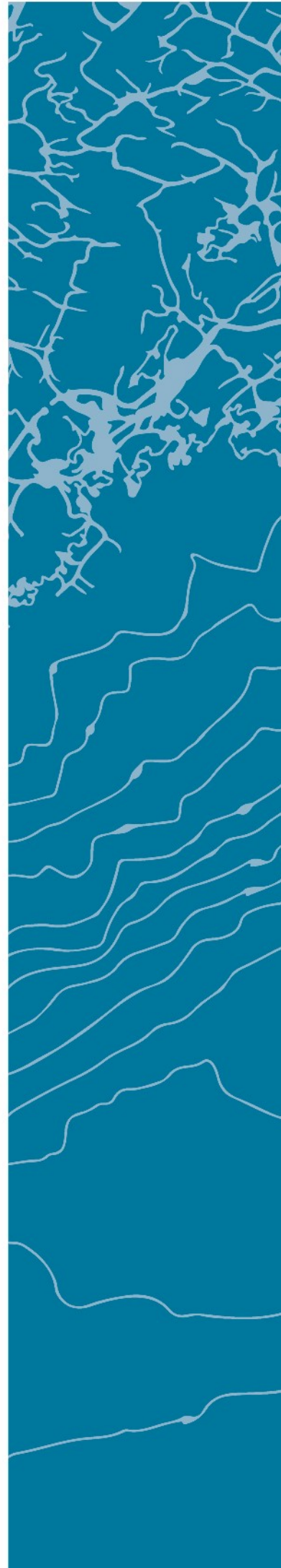
Morten Ottestad

Andreas Klausen

University of Agder, 2019

Faculty of Technology and Science

Department of Engineering



Preface

This report presents the process and results of a master thesis given by the University of Agder (UiA) within the mechatronics study. The thesis was part of the MAS-500 course which constitutes 30 out of 120 study points necessary to receive a master's degree. The main purpose of this thesis was for the students to show their knowledge within the field of mechatronics field based upon what was lectured during the study.

The work done in this thesis involves the subjects of artificial neural networks, wastewater treatment, and data analysis. This has been an exciting five months of researching topics that were well known, and completely new to me. The thesis has been written with the hope that the results can be useful for further development at Saulekilen Renseanlegg (Wastewater treatment plant).

First, I want to thank my supervisors from University of Agder, Assistant Professor Morten Ottestad and Dr. Andreas Klausen. The help they gave me was crucial to the completion of the thesis and their supervision guided me through the steps of the process. Secondly, I would like to thank the staff of Saulekilen Renseanlegg for teaching me how the plant functioned and by answering all my questions about the processes and data. Lastly, I would like to thank my parents for giving useful feedback about the writing process.

Magnus Gabriel Ersdal

Grimstad, May 24, 2019

Abstract

The processes at a wastewater treatment plant (WWTP) are complex systems that clean the wastewater before it is released into the environment. Total phosphorous (TP), biological oxygen demand (BOD) and chemical oxygen demand (COD) are critical measurements of the water quality. Measuring BOD and COD are too slow to effectively control the wastewater treatment processes (5 days and a couple of hours respectively) as the wastewater treatment process at Saulekilen WWTP uses about 30-40 minutes from influent to effluent. Measurements of effluent TP are also performed and are quicker but controlling by the use of effluent TP is too slow. Hence it is a goal for this thesis to propose models that can predict the values of BOD, COD and TP quicker by the use of multiple linear regression (MLR) and artificial neural networks (ANN). Results of this thesis indicate that measurements of effluent TP can be predicted with reasonable accuracy. Further, the influent BOD and COD measurements can be predicted with good accuracy. These promising predictions may be important in further work for an improved wastewater treatment.

A sensitivity analysis has been performed to determine which input parameters being the most effective in modelling the output. Input parameters in this thesis have consisted of TP, water discharge, the local temperature and the date for predictions of TP, COD and BOD. Measurements of influent BOD and COD has also been used as inputs for prediction of effluent BOD and COD, while total solids (TS), total percent solids (TS(%)), reject water, pH, the process additive PIX and turbidity has been used as inputs for predictions of effluent TP. The results of the models have been considered with respects to correlation, mean squared error (MSE) and mean absolute percentage error (MAPE).

Contents

Contents	iii
List of Figures	vii
List of Tables	x
1 Introduction	1
1.1 Thesis overview	1
1.2 Task description and motivation	2
1.3 Wastewater treatment	2
1.4 Artificial neural network	4
1.5 State-of-the-art of WWTP modelling	4
2 Wastewater treatment plants	8
2.1 Treatment cycle	8
2.1.1 Measurements	11
2.2 Saulekilen treatment plant	12
2.2.1 Measurement methods	13
2.2.2 Available data	14
3 Data analysis and modelling theory	17
3.1 Data regression	17
3.2 Data analysis	18
3.3 History of artificial neural networks	20
3.4 The artificial neural network	21
3.5 Backpropagation	24
3.6 Improving the way ANN learns	27
3.6.1 Underfitting and overfitting	28
3.6.2 Hidden layer design	28
3.6.3 Activation functions	29

3.6.4	Regularisation	32
3.6.5	Dropout	32
3.6.6	Artificially expanding the training data	33
3.6.7	Normalising inputs and activations	34
3.6.8	Hyper-parameter tuning	35
4	Experimental methodology	37
4.1	Data correlation and analysis	37
4.2	Regression models	40
4.3	Artificial neural network setup	41
5	Data analysis results	44
5.1	Data trends	44
5.1.1	Bi-weekly data set trends	46
5.1.2	Daily data set trends	48
5.2	Data correlations	50
5.3	Comparison of bi-weekly and daily overlapping measurements	54
6	ANN and MLR model results	56
6.1	BOD and COD influent models	56
6.2	MLR effluent models	58
6.2.1	Bi-weekly data set models	59
6.2.2	Daily data set models	61
6.3	MLR effluent models sensitivity	62
6.3.1	Bi-weekly data set	62
6.3.2	Daily data set	66
6.4	ANN model improvements	68
6.5	ANN effluent models	70
6.5.1	Bi-weekly data set models	71

6.5.2	Daily data set models	73
6.6	ANN effluent models sensitivity	74
6.6.1	Bi-weekly data set	74
6.6.2	Daily data set	78
7	Discussions	81
7.1	Data analysis	81
7.1.1	Trends of influent and effluent parameters	81
7.1.2	Time periods	83
7.1.3	Data correlations	84
7.1.4	Available data parameters and differences between bi-weekly and daily data set	85
7.2	MLR and ANN models in general	86
7.3	Bi-weekly data set influent models	87
7.4	Bi-weekly data set models	88
7.4.1	Sensitivity analysis	90
7.5	Daily data set models	91
7.5.1	Sensitivity analysis	93
7.6	ANN model improvements	94
8	Conclusions	96
A	Code to produce model results	102
A.1	MLR code from Matlab	103
A.2	ANN code from python	117
B	Agendas and minutes of meetings	126
B.1	Agenda #1	127
B.2	Minutes of meeting #1	128

CONTENTS

B.3 Agenda #2	129
B.4 Minutes of meeting #2	130
B.5 Agenda #3	131
B.6 Minutes of meeting #3	132
B.7 Agenda #4	133
B.8 Minutes of meeting #4	134
B.9 Agenda #5	135
B.10 Minutes of meeting #5	136
B.11 Agenda #6	137
B.12 Minutes of meeting #6	138
B.13 Agenda #7	139
B.14 Minutes of meeting #7	140
B.15 Agenda #8	141
B.16 Minutes of meeting #8	142
B.17 Agenda #9	143
B.18 Minutes of meeting #9	144
B.19 Agenda #10	145
B.20 Minutes of meeting #10	146

List of Figures

2.1	Four main stages of wastewater treatment.	9
2.2	Overview over all main process in a WWTP [20].	10
2.3	Overview over positioning of measurement points at Saulekilen. . . .	14
3.1	A single neuron.	22
3.2	Classification of layers in a artificial neural network [11].	23
3.3	Gradient descent, given inputs x	26
3.4	Response of different activation functions.	31
3.5	Illustration of dropout in an arbitrary artificial neural network.	33
5.1	Bi-weekly data set parameters plots, 04.04.06 - 13.12.18.	46
5.1	Bi-weekly data set parameters plots, 04.04.06 - 13.12.18 (Cont.). . . .	46
5.1	Bi-weekly data set parameters plots, 04.04.06 - 13.12.18 (Cont.). . . .	47
5.1	Bi-weekly data set parameters plots, 04.04.06 - 13.12.18 (Cont.). . . .	47
5.2	Daily data set parameters plots, 02.01.17 - 12.03.19.	48
5.2	Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).	48
5.2	Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).	49
5.2	Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).	49
5.2	Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).	50
5.2	Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).	50
5.3	Comparison of overlapping data parameters for bi-weekly and daily data set.	54
5.3	Comparison of overlapping data parameters for bi-weekly and daily data set (Cont.).	55
5.4	Histogram of deviations between overlapping parameters.	55
6.1	MLR models for generating BOD_{in} and COD_{in} , using time period 2, 04.04.06 - 04.08.15.	57

6.2	MLR models for generating BOD_{in} and COD_{in} , using time period 2-6, 04.04.06 - 13.12.18.	57
6.3	MLR models for generating BOD_{in} and COD_{in} , using time periods 3-6, 14.08.15 - 13.12.18.	58
6.4	MLR models of the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.	59
6.4	MLR models of the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).	60
6.5	MLR models for daily data set using time periods 3-6, 02.01.17 - 13.03.19.	61
6.6	MLR models for daily data set using time period 6, 07.04.18 - 12.03.19.	61
6.7	TP_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.	63
6.7	TP_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).	63
6.8	BOD_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.	64
6.8	BOD_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).	64
6.9	COD_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.	65
6.9	COD_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).	65
6.10	TP_{out} sensitivity of dropping input parameters for the daily data set using time periods 3-6, 02.01.17 - 12.03.19.	66
6.10	TP_{out} sensitivity of dropping input parameters for the daily data set using time periods 3-6, 02.01.17 - 12.03.19 (Cont.).	67
6.11	TP_{out} sensitivity of dropping input parameters for the daily data set using time period 6, 07.04.18 - 12.03.19.	67

6.11	TP_{out} sensitivity of dropping input parameters for the daily data set using time period 6, 07.04.18 - 12.03.19 (Cont.).	68
6.12	ANN models of the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.	71
6.12	ANN models of the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).	72
6.13	ANN models for daily data set using time periods 3-6, 02.01.17 - 13.03.19.	73
6.14	ANN models for daily data set using time period 6, 07.04.18 - 12.03.19.	73
6.15	TP_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15.	75
6.15	TP_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15 (Cont.).	75
6.16	BOD_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15.	76
6.16	BOD_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15 (Cont.).	76
6.17	BOD_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15.	77
6.17	COD_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15 (Cont.).	77
6.18	TP_{out} sensitivity of dropping input parameters for the daily data set using time periods 3-6, 02.01.17 - 12.03.19.	78
6.18	TP_{out} sensitivity of dropping input parameters for the daily data set using time periods 3-6, 02.01.17 - 12.03.19 (Cont.).	79
6.19	TP_{out} sensitivity of dropping input parameters for the daily data set using time period 6, 07.04.18 - 12.03.19.	79
6.19	TP_{out} sensitivity of dropping input parameters for the daily data set using time period 6, 07.04.18 - 12.03.19 (Cont.).	80

List of Tables

1.1	Graphical overview of thesis	1
2.1	Statistical analysis of bi-weekly data.	15
2.2	Statistical analysis of daily data.	16
5.1	Correlation between parameters in bi-weekly data set for time period 2, 04.04.06 - 13.08.15.	51
5.2	Correlation between parameters in bi-weekly data set for time periods 2-6, 04.04.06 - 13.12.18.	52
5.3	Correlation between parameters in bi-weekly data set using time peri- ods 3-6, 14.08.15 - 13.12.18.	52
5.4	Correlation between parameters in daily data set for time period 3-6, 02.01.17 - 12.03.19.	53
5.5	Correlation between parameters in daily data set using time period 6, 07.04.18 - 12.03.19.	53
6.1	Remaining MLR models of bi-weekly data set.	60
6.2	Abbreviations used in ANN model explanation	69
6.3	Final ANN models for bi-weekly data set	69
6.4	Final ANN models for daily data set	70
6.5	Remaining ANN models of bi-weekly data set.	72
B.1	Minutes of meeting #1	128
B.2	Minutes of meeting #2	130
B.3	Minutes of meeting #3	132
B.4	Minutes of meeting #4	134
B.5	Minutes of meeting #6	136
B.6	Minutes of meeting #6	138
B.7	Minutes of meeting #7	140
B.8	Minutes of meeting #8	142

LIST OF TABLES

B.9 Minutes of meeting #9	144
B.10 Minutes of meeting #10	146

Nomenclature

∇C	Gradient descent
a	Neuron activation
b	Neuron bias
Q_{rw}	Reject water discharge
Q_w	Water discharge
$Temp.$	Temperature
$Turb.$	Turbidity
w	Neuron connection weight
x	Input parameter
y	Output parameter
z	Inactivated neuron output
AI	Artificial intelligence
ANN	Artificial neural network
BOD	Biological oxygen demand
COD	Chemical oxygen demand
ELU	Exponential linear unit
MAPE	Mean absolute percentage error
MSE	Mean squared error
r	correlation coefficient
ReLU	Rectified linear unit
tanh	Hyperbolic tangent

LIST OF TABLES

TP	Total phosphorus amount
TS	Total solids
WWTP	Wastewater treatment plant

Chapter 1

Introduction

1.1 Thesis overview

This thesis explores the possibilities of introducing an artificial neural network (ANN) to model some of the processes of Saulekilen wastewater treatment plant (WWTP). This introduction chapter will explain what the target of the thesis is and in short detail also the main purposes of a WWTP and an ANN before these are more deeply elaborated upon in further chapters. Table 1.1 presents a graphical overview of the thesis.

Table 1.1: Graphical overview of thesis

Chapter 1: Introduction	
Chapter 2: Wastewater treatment plants	Chapter 3: Data analysis and modelling theory
Overview over the processes of a WWTP, its main measurement methods and how this is all applied at Saulekilen WWTP.	Theoretical background for model generation based on multiple linear regression (MLR) and ANN.
Chapter 4: Experimental methodology	
Methods used to analyse data and setting up experiments for MLR and ANN models.	
Chapter 5: Data analysis results	Chapter 6: ANN and MLR model results
Data trends, correlations and time period splits.	MLR and ANN models with respective sensitivity analyses
Chapter 7: Discussions	
Chapter 8: Conclusions	

1.2 Task description and motivation

The municipality of Arendal is the owner and operator of the Saulekilen wastewater treatment plant, which receives wastewater equivalent to 70,000 people from its own municipality and parts of Grimstad municipality. The cleaning process takes place in four steps; mechanical coarse cleaning, mechanical screening, biological cleaning and chemical cleaning. Extensive renovations and extensions have been carried out at the facility since its establishment in 1978. The plant now meets all state requirements but the operational costs for the plant have escalated. Although the treatment efficiency of the plant is significantly better today, it still drops below the accepted level and sometimes it can also become unnecessarily high. The increased costs are related to increased consumption of energy and chemicals, which in the purification process are the major input factors. The municipality sits on varying amounts of historical measurement data from the treatment plant which can be used to optimise the use of resources. Arendal municipality in collaboration with UiA seeks to investigate the possibilities of generating a model of the wastewater treatment plant based on artificial intelligence (AI). An artificial neural network can be trained by historical data to represent the wastewater treatment plant, and help an operator control the plant. Modelling the processes of a WWTP with the use of an ANN has had some success in existing literature. Such a model will enable the optimisation of the chemical use in the plant which could help generate a more stable treatment process and reduce the operating costs.

1.3 Wastewater treatment

Wastewater treatment is an important field for the preservation of the natural environments in urban and industrial areas. Activities in these areas utilise a huge amount of natural water and discharges chemicals that need, to the extent possible, to be ex-

tracted before the water is released to nature again [1][2]. A wastewater treatment plant brings discharge water through a multiple of processes to extract these chemicals and essentially clean the water [2][3]. The effluent¹ water from the plant needs to satisfy certain governmental criteria to be considered safe for the environment [4]. To be able to uphold these standards a reliable plant model needs to be created in order to predict which performance parameters the plant should follow. Measurements of key parameters of the influent² can be time consuming, and the correlation between the different chemicals seldom exist [1]. In Norway it is required to report Biological Oxygen Demand (BOD), Chemical Oxygen Demand (COD), Nitrogen and the total Phosphorous amount (TP) for both influent and effluent [4]. Additionally, suspended solids³ and dissolved solids⁴ are measurements that gives a good indication of how clean the water is [2]. The traditional models currently used does not always give satisfying results compared to the amount of time and resources used to obtain them [5]. This is especially a problem concerning measurements of BOD which requires 5 days to obtain [6][7]. Additional complexity to model the WWTP comes from the dynamic behaviour of wastewater. This can be due to the following: weekend effects, where industrial wastewater can be lower; seasonal weather effects in the sense of different rainfall and temperature; holiday periods where wastewater can both decrease or increase based on the local norm; and heavy rain falls, dry seasons, and other abnormal weather effects that does not necessarily follow the seasonal norm. Modelling of the plant can therefore require at least one year of data to make sure most types of influents are covered [8].

¹Water flowing out of the plant

²Water flowing in to the plant

³Solids still in suspension in water, measured as the amount of solids not able to pass through a filter

⁴Solids dissolved in water, measured as the amount of solids able to pass through a filter

1.4 Artificial neural network

Artificial neural networks (ANN) is a computer technique that is commonly used for feature extraction, pattern recognition and classification [9]. E.g. this can be to detect certain features in an image [10], recognise a pattern in a database [11, Chapter 4] or classify written numbers [11, Chapter 1]. The neurons are meant to artificially simulate the human brain and to harness its powerful ability of learning, which computers previously has been incapable of [12][13][14]. The implementation of artificial neural networks to the processes of wastewater treatment introduces new opportunities for both economic and environmental optimisation. An ANN's ability to learn the correlation between input and output can be an incredibly useful method of optimising the amount of additives during the treatment process. Additionally, it can be useful to estimate values quickly that would otherwise be time consuming by using traditional methods.

1.5 State-of-the-art of WWTP modelling

The implementation of artificial neural networks to the processes of wastewater treatment is not entirely new, but has acquired more consideration in the recent decade [9]. By studying the existing literature in the field, it is evident that ANN models are capable of predicting both effluent and influent values of important wastewater quality parameters.

Rene et al., 2008, proposed an ANN model of a WWTP, training with only 103 data points on a model with a single hidden layer of less than 10 neurons with over 5000 epochs. They concluded that an ANN model was an accurate and effective way of predicting unknown concentrations of effluent BOD and COD values. During training of the ANN model, they found that they could increase their network accuracy by

varying the training count and number of neurons in the hidden layer, while varying the learning rates did not have any significant impact. They also found that removing certain input variables could increase the accuracy and that using a single-output ANN model was superior to a multiple-output model. The best performing models included total organic carbon, phenol concentration, total suspended solids, total dissolved solids, Kjeldahl's nitrogen and ammoniacal nitrogen [1].

In a paper from 2008, Dogan et al. attempted to use an ANN to estimate the influent values of BOD in a biological WWTP. Good results were recorded with training on 244 data points stretching over a single year using only small single hidden layer. In this paper the results of the ANN model were compared to a multiple linear regression (MLR) model which in this case proved that a MLR model can give good results but that the ANN was superior. They used a sensitivity analysis which compared the effect on the BOD prediction by removing input parameters one by one. This analysis showed that all the available data inputs (total phosphorus, total nitrogen, suspended solids, water discharge and COD) provided some explanatory power for the ANN model but that water discharge decreased the MLR performance. COD had the biggest contribution to the results in both models [5]. The results of this paper is interesting as it shows that accurate predictions of influent BOD values can be obtained without waiting five days for a laboratory test.

Ozkan et al., 2009, developed an ANN model based on the Levenberg-Marquet learning algorithm which estimated effluent BOD values based on input values of total nitrogen, total phosphorus, suspended solids, COD and total dissolved solids. With 910 data points used for training stretching over a four-year period, the model managed to predict BOD with high accuracy. The conclusion was that effluent BOD values could be predicted much faster with an ANN model than by laboratory tests [15].

Nasr et al., 2010, applied a deep neural network to predict effluent COD, BOD and total suspended solids amounts based only on influent values of the same parameters. One year worth of measurements were used, training with the Levenber-Marquet learning algorithm which resulted in high correlation values for every output [16].

A different method was proposed by Zhang et al., 2012, with the use of a NW multi-layer feedforward small world ANN which was used because of its ability to offset one neurons weakness by learning from other neurons strong points. The network used a normal feedforward layer connection while also connecting a few sets of neurons which were not already connected. This made some neurons connect to neurons of two different layers. The network used inputs of influent COD, pH, BOD and suspended solids and predicted effluent BOD values with incredible accuracy, which compared to be superior to a normal ANN model which was also produced [17].

Similarly to Ozkan et al., a study of using the Levenberg-Marquet learning algorithm with an ANN was performed by Abba et al., 2015, to predict effluent COD values. Good results were achieved by using influent values of COD, BOD, ph, conductivity, total nitrogen, total phosphates, total suspended solids and suspended solids. A sensitivity analysis showed that using all input parameters gave a superior model and that an ANN model had a better performance than an equivalent MLR model. Data used in this paper was 312 data points based on a three year period [9].

Hamada et al., 2018, compared the performance of a normal ANN model with a MLR model and a radial basis function (RBF) neural network on the prediction of effluent BOD, COD and total suspended solids. The models were trained with 65 data points containing input data of influent temperature, pH, BOD, COD, TSS. The conclusions of the paper was that an ANN model had a better explanatory power over both the RBF and MLR models, while a sensitivity analysis showed that influent pH had almost no contribution to the result [18].

Some repeating factors within the existing literature is: the 70%-30% split of training and test data respectively; the low amount of available data, often being in the range of 100-300 observations; the use of small single hidden layer ANN architectures, often using only around 3-8 neurons; and the use of Matlab as a model generator, which has the Levenber-Marquet learning algorithm easily available.

Chapter 2

Wastewater treatment plants

This chapter will give an overview over the main processes of a WWTP. This includes explaining a typical treatment cycle and what, why and how pollutants are measured. The last section explains how this is performed at Saulekilen WWTP. Some information about the data used in this thesis will also be given.

2.1 Treatment cycle

Wastewater is normally divided into two major groups: sewage water, which is wastewater from urban areas, and industrial wastewater, which is wastewater from industrial activities. In addition, this is mixed with unabsorbed rain water which can often be a key factor in the fluctuations of influent wastewater for the WWTP [2][19]. At the treatment plants, the wastewater normally undergoes five major processes: preliminary treatment, primary treatment, secondary treatment, disinfection and sludge treatment. To balance all the processes correctly, water or particles from some processes can be diverted back to previous processes which helps with stabilisation. The full process can take about seven hours to complete before the cleaner water is released to the environment, but some of the sludge processing can take weeks to handle. The information about the treatment cycle in this section is mainly gathered from [3], but also [2].

In Figure 2.1 a simple overview over the four stages often used for wastewater treatment is shown. This process begins with sewage and wastewater being brought to the WWTP through the sewage system and ends by discharging the treated water into the environment. An explanation of the steps is shown next to the figure. A more in depth overview is shown in Figure 2.2 where the process of managing overflow and sludge is also shown.

2.1. TREATMENT CYCLE

The preliminary treatment is the first step and is a mechanical process which filters out larger objects that could damage the sewage pumps and other equipment. The objects are stopped by a filter (e.g. metal bars) and is removed and transported to landfills.

The primary treatment is a settling process which slows down the water flow to make heavier solids settle to the bottom of the tank and let lighter materials float to the top of the water. The solids, or sludge, and materials are then removed by mechanical scrapers and surface-skimmers respectively, and the remaining water is pumped to the next treatment process.

The secondary treatment takes the wastewater through aeration tanks that injects air so that microorganisms can consume most of the remaining organic materials. The heavier particles that are then formed, settles to the bottom of another settling tank. At this point, 85-95% of the pollutants from the untreated wastewater has been removed.

In the disinfection process, special made chemical mixes are added to the water which kill harmful organisms. The water is now considered clean enough to be released into the environment and measurements of the water cleanliness is performed to make sure the WWTP is upholding the required level of quality.

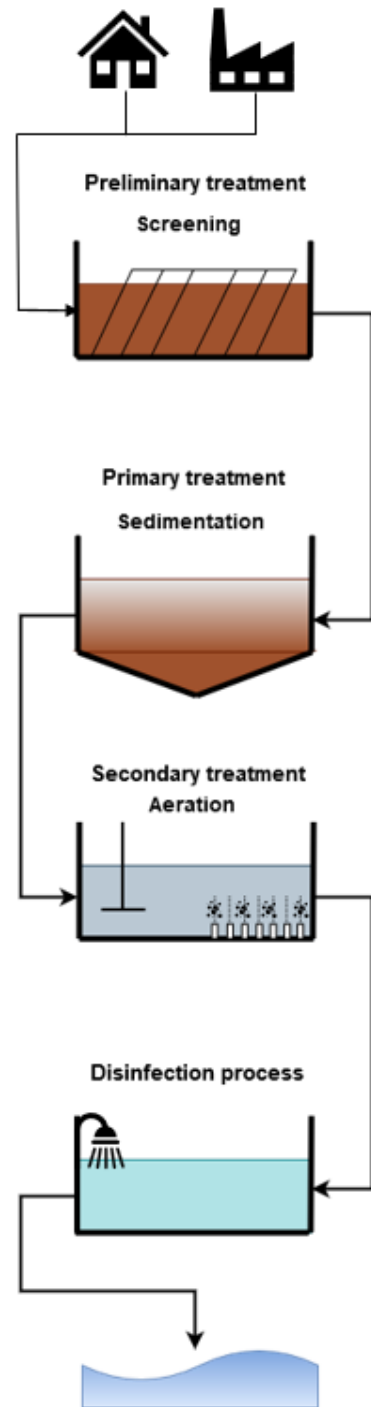


Figure 2.1: Four main stages of wastewater treatment.

2.1. TREATMENT CYCLE

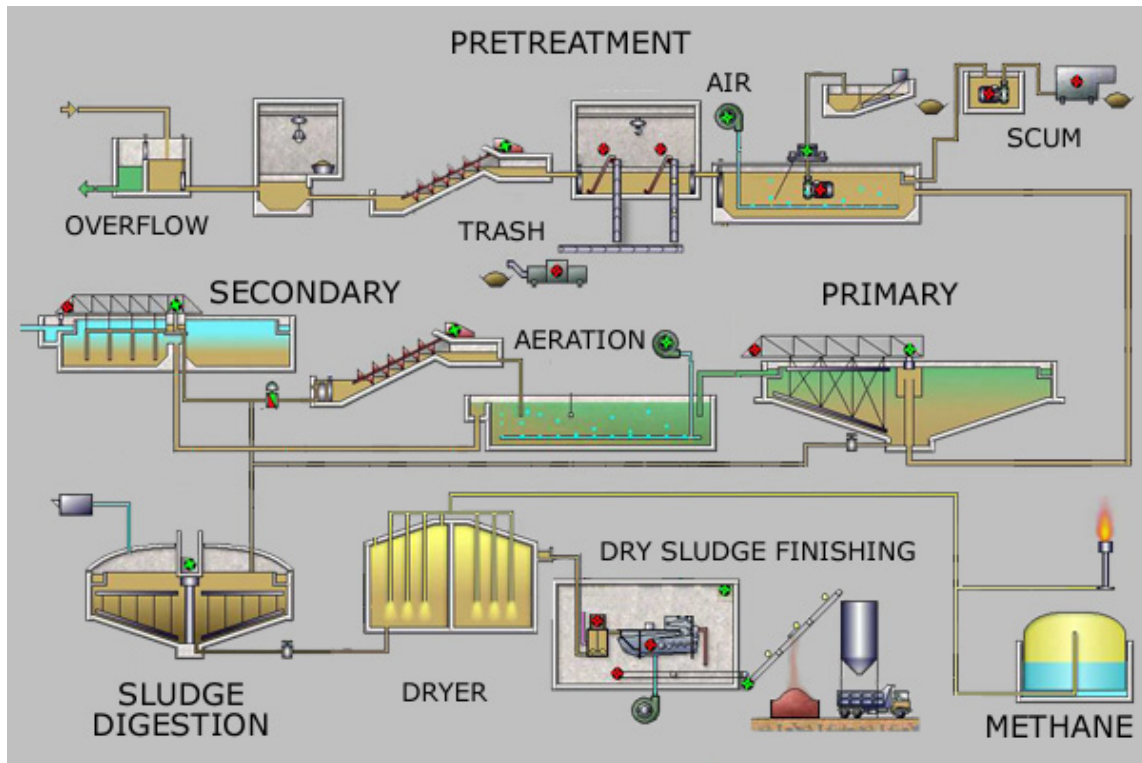


Figure 2.2: Overview over all main process in a WWTP [20].

The last step is sludge treatment, which is only applied to the sludge that is removed from the earlier steps. This is a much longer treatment and is mostly to create a more manageable waste product. Here the sludge is first put in settling tanks where it thickens after settling from the water. After thickening, the sludge is placed in an anaerobic environment at around 35 degrees Celsius for a couple of weeks to stimulate growth of bacteria. This process produces gas, water and a more concentrated sludge. At last, the sludge is dewatered by using centrifuges that forces water out of the sludge and leaves a more solid substance also known as biosolids which can be used as fertilisers for agriculture or be buried underground in sanitary landfills. In cases where the sludge could contain toxic industrial chemicals, it is not used on agriculture meant for human consumption.

2.1.1 Measurements

To maintain control over the treatment cycle and especially to make sure that unwanted pollutants are not unknowingly released into the environment, some key measurements has to be taken in the plant.

Two of the major methods of measuring the water quality of wastewater are biological oxygen demand (BOD) [5][15] and chemical oxygen demand (COD) [1][9]. They are both a measure of the oxygen required for microorganisms to carry out biological decomposition of dissolved solids or organic matter¹ [9]. A body of water with a high BOD or COD amount provides less dissolved oxygen for higher animals such as fish [21].

For the BOD test, a sample of the wastewater is sealed in a small bottle for 5 days at 20 degrees Celsius together with dissolved oxygen, nutrients and a buffer. The resulting oxygen demand is referred to as the BOD₅ [6][7]. On the contrary to the BOD test, the COD test uses a strong chemical oxidising agent to chemically oxidise the organic material in a wastewater sample, and only uses a couple of hours to complete [6][22]. The test indicates the amount of non-biodegradable organic material in wastewater. The ratio between COD and BOD can indicate the amount of biodegradable material in the water. If the ratio is lower than 2, the water is readily biodegradable, but if the ratio is more than 2, it can be suspected portions of the organic material in the sample is not biodegradable by ordinary microorganisms [22].

Additionally, each WWTP can have different standards they need to relate to from their municipality. In Norway it is required to report total nitrogen (TN), total phosphorous (TP), BOD, COD and suspended solids (SS) together with total water discharge (Q_w) [23]. Nitrogen and phosphorous can lead to overfertilisation of fresh

¹Carbon-based compounds found in water

waters, fjords and coastal waters [4].

2.2 Saulekilen treatment plant

The Saulekilen treatment plant is a wastewater treatment plant located south west of Arendal in southern Norway. Around 90-95% of the influent water is received from Arendal while the remaining water comes from the neighbouring town Grimstad. The wastewater is mostly from urban districts but also includes some industrial wastewater which can at times fluctuate the water quality. The amount of sewer is rather constant, so the main varying factor is therefore the amount of rain water. While the plant is dimensioned for 70,000 people, the rain water can sometimes cause certain parts of the plant to either flood or to require an unrecommended high velocity. This can cause a decrease in quality of the treatment process.

The Saulekilen plant is a mechanical, biological, and chemical plant. Instead of using sedimentation, the Saulekilen plant uses flotation for sludge removal. The wastewater is first taken through the preliminary treatment which removes larger and smaller objects before it goes straight in to a sand trap which removes sand and grease by using compressed air. Then it enters six filters which removes solids with a mechanical belt, transporting solids out of the water. The next step is the biological step where microorganisms remove biological materials that has been dissolved in the water. The wastewater is then transported to a floc reactor where a solution called PIX is added to create a metal reaction between the negatively loaded ions of the wastewater and the positively loaded ions of the additive. The result is that the phosphorus in the water separates and sludge is created. Polymer is then added to help coagulate² the wastewater and form larger masses. In the last step before the water is considered “clean”, water and air is used to force the sludge to the surface

²The process of a liquid changing to a solid or semi-solid state.

where it can be scraped off for further treatment. The water in the sludge is removed by entering a cylindrical apparatus which spins the water and relies on centrifugal forces to separate the water from the solids. The water normally takes about 30-40 minutes from it enters the preliminary treatment step until it reaches a floc reactor. The amount of polymer added to the flocculation step is depended on the amount of influent water, i.e. more water means more polymers. The amount of PIX that is added per litre of water decreases when there is a higher water influent. This is due to there being a very stable amount of sewage entering the plant at all times and excess water is often regarded as being rain water which does not need to be excessively treated. A measurement of the influent pH is also used to determine the correct amount of PIX to add. The reject water is taken from the centrifuges and sludge thickening steps of the sludge treatment and added to the influent water.

2.2.1 Measurement methods

At Saulekilen, measurements are performed in a variety of methods. Certain parameters are measured continuously and are used to display the state of the plant for the operators at any time. These are the many water flow measurements, water volumes, pressures and pH. A daily sensor check is also performed where the most important parameters are noted, which in addition to the continuous measurements also include TP, total solids (TS) and additives (PIX). Turbidity is also included on the daily check and is measured by lowering a disk into the effluent water and noting the depth which it can no longer be seen. This can lead to some inaccuracies in the measurements as different operators can measure it differently. TP is measured with a sensor in an automatic sampler of the wastewater. 24 times a year, BOD and COD tests get sent to a laboratory to control the performance of the plant. In addition, a few pollutant tests like quicksilver, lead and zinc are control checked a couple of times each year. It should be noted that the phosphorous sensor had not been calibrated

before this thesis started and could potentially be incorrect. In Figure 2.3 a simple overview over the measurement points at Saulekilen is shown. This highlights where the the influent and effluent measurements are in regard to PIX_{add} .

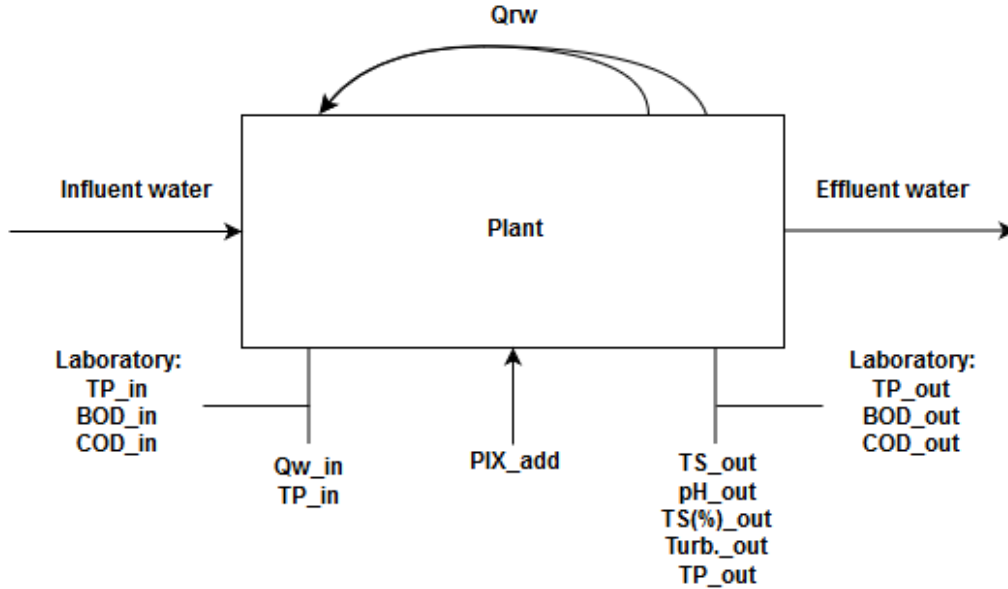


Figure 2.3: Overview over positioning of measurement points at Saulekilen.

2.2.2 Available data

Two different data sets (bi-weekly³ data set and daily data set) were given by Saulekilen. The first included around 22 observations per year from 2001 to 2018 with COD, BOD and TP measurements for both influent and effluent as well as Q_w for influent. The second included daily measurements from 2017 to 2019 with a wider range of measurements. For influent measurements there were: Q_w and TP_{in} as well as the additive PIX and reject water Q_{rw} . For effluent measurements there were: TS_{out} , pH_{out} , $TS(\%)_{out}$, TP_{out} , and turbidity ($Turb.$).

An advantage of these data sets was their inclusion of all seasons over multiple years.

³Every two weeks

2.2. SAULEKILEN TREATMENT PLANT

The difference in weather for these periods should reflect the different water qualities that the plant would have to treat. Considering the daily data set, a longer span of years would give a better guarantee for the inclusion of certain extreme weather periods, like the one that was experienced in the Norwegian summer of 2018 [24], the average weather of the area should be well represented over a two year period. In addition to the data given from Saulekilen, temperature readings from the local area was retrieved from Yr.no for every hour from 04.04.2006 to 01.05.2019 and was averaged for each day and then paired with the Saulekilen data.

A few important events for Saulekilen should be noted that could have had some effect on the measurements. From October 2016 to the middle of March 2017 the bio-reactor was out of order and its area was being rebuilt. From September 2017 the plant started precipitation of iron. In addition, a few malfunctions had been registered.

A statistical analysis of the data that was used are shown in Tables 2.1 and 2.2.

Table 2.1: Statistical analysis of bi-weekly data.

Parameters	Unit	Mean	Max	Min	Standard deviation
TP_{in}	mg/l	3.782	9.8	0.34	1.591
TP_{out}	mg/l	0.321	1.3	0.044	0.2055
BOD_{in}	mg/l	154.2	500	16	72.16
BOD_{out}	mg/l	49.02	180	4.2	35.26
COD_{in}	mg/l	418.4	1100	105	163.7
COD_{out}	mg/l	120.3	310	16	61.61
Qw_{in}	m^3/day	16871	59896	2386	7305
$Temp.$	$^{\circ}C$	8.64	22.7	-9.083	6.755

2.2. SAULEKILEN TREATMENT PLANT

Table 2.2: Statistical analysis of daily data.

Parameters	Unit	Mean	Max	Min	Standard deviation
Qw_{in}	m^3/day	15402	40603	3721	6500
TP_{in}	mg/l	2.587	6.7	0.25	0.9123
Qrw	m^3/day	133	401	20.33	72.5
TS_{out}	m^3/day	74.98	183	8.833	32.21
pH_{out}	pH	7.373	8.04	6.36	0.1819
$TS(\%)_{out}$	%	4.299	9.13	2.59	0.7252
PIX_{add}	l/d	3080	5786	786.5	1097
$Turb_{out}$	meters	0.6776	1.7	0	0.2345
TP_{out}	mg/l	0.19	0.9	0	0.1337
$Temp.$	$^{\circ}C$	8.314	22.4	-9.2417	6.557

Chapter 3

Data analysis and modelling theory

A foundation for every claim that is made during the report should be based upon scientific theories and mathematical equations. In this chapter the main theories used will be explained to introduce the reader to the science behind the report. The first section will describe the basis of data analysis which is the foundation of the data modelling methods used in this thesis. The second section will describe how data can be approximated with the help of data regression. An introduction to the history of artificial neural networks is then given in the third section which will simultaneously describe an ANN's foundations. The following two sections will further elaborate on the functionality of an ANN and how this computer technique is able to learn patterns in a similar way to humans. Finally, the last section will give an overview over the different techniques that can be used when improving an ANN.

3.1 Data regression

There are different methods to make a regression model fit to certain data. Multiple linear regression (MLR) attempts to model the relationship between the explanatory (independent, input) variables, $\{x_1, x_2, x_3, \dots, x_p\}$, and response (dependent, output) variable, y . The linear Equation (3.1) gives the best fit based on all the explanatory variables, and is in essence an expanded version of the least-squares method that only uses a single explanatory variable. For $i=1, 2, \dots, n$ observations,

$$y_i = \beta_0 + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i2} + \dots + \beta_p \cdot x_{ip} + \epsilon_i \quad (3.1)$$

where y_i is the dependent variable, x_{ip} are the explanatory variables, β_0 is the y-intercept (constant term), β_p are the slope coefficients for each explanatory variable, and ϵ_i is the models error term. The chosen values for the constant terms, $\beta_0, \beta_1, \dots, \beta_p$,

are chosen based on the least squares criterion given by

$$\min\left(\sum_i (y_i - \hat{y}_i)^2\right), \quad (3.2)$$

where \hat{y}_i is the estimated value of y_i , given x_i for the given observation i [25][26][27].

MLR can be scaled down to just a single explanatory variable as well, which gives a normal linear regression model by only using the first part of Equation (3.1): $y_i = \beta_0 + \beta_1 \cdot x_{i1} + \epsilon_i$ [25]. This model can then be expanded to create a non-linear model, while still only using a single explanatory variable. The formula remains fairly similar, but instead returns a polynomial of the q th degree that best fits the data as shown in Equation (3.3) [28]:

$$y_i = \beta_0 + \beta_1 \cdot x_i + \beta_2 \cdot x_i^2 + \dots + \beta_p \cdot x_i^q + \epsilon_i \quad (3.3)$$

3.2 Data analysis

Data analysis is a method of describing data with the use of statistics. Several experiments or surveys are performed to collect variables of interest which are then further analysed. The usual statistical measures are a data's mean, median, mode, percentiles, range, variance, and standard deviation. These measures can be used to give significant information about a data set, or they can be used to compare multiple different data sets. Another method of comparing data sets is the use of regression and correlation analyses. Data regression is often used as an initial hypothesis for the relationship between a measured and an estimated data set. To determine the goodness of fit of the model, a common method used is the coefficient of determination. For $i=1, 2, \dots, n$ observations,

$$SST = \sum_i (y_i - \bar{y}_i)^2, \quad (3.4)$$

where SST is the measure of total variation, y_i is the measured value being estimated, \bar{y}_i is the mean of all y_i ,

$$SSE = \sum_i (y_i - \hat{y}_i)^2, \quad (3.5)$$

where SSE is the measure of unexplained variation, \hat{y}_i is the estimated value by the model,

$$SSR + SSE = SST, \quad (3.6)$$

where SSR is the part explained by the estimated regression equation, and

$$r^2 = \frac{SSR}{SST}, \quad (3.7)$$

where r^2 is the coefficient of determination. A value of $r^2 = 1$ means there is a perfect linear relationship between the data sets, while a value of $r^2 = 0$ means there is absolutely no such relationship.

“For typical data found in the social sciences, values of r^2 as low as 0.25 are often considered useful. For data in the physical sciences, r^2 values of 0.60 or greater are frequently found.” [25]

Similar to determination is correlation which is sometimes calculated as the square root of the determination coefficient. The correlation coefficient is denoted r , and has the same sign as the first slope coefficient, β_1 , of the linear regression equation (Equation (3.1)). This means it has a range of $[-1, 1]$, where a value of 1 means the data sets are perfectly positively related, a value of -1 means they are perfectly negatively related, and a value of 0 means there is no linear relationship. It is important to note that correlation does not necessarily mean causation [25]. Another way of measuring the correlation coefficient, and the most commonly used one, is the Pearson’s linear correlation coefficient which is defined as:

$$r_{XY} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}, \quad (3.8)$$

where r_{XY} is the Pearson's correlation coefficient given data sets X and Y , containing values X_i and Y_i for $i=1, 2, \dots, n$ observations, and data set means \bar{X} and \bar{Y} respectively [29].

3.3 History of artificial neural networks

The idea of artificial neural networks can be traced all the way back to 1943 where neurophysiologist Warren McCulloch and mathematician Walter Pitts described binary threshold neurons. They proposed these neurons as a simple model of how the neurons in the brain might work [12][13][14]. These neurons worked by receiving an input value and multiplying it by a weight before comparing it to a threshold value. It then ultimately outputted either a 1 or a 0, depending on whether or not it was above the threshold value. This system was proven to be capable of replicating the basic OR/AND/NOT functions. While this was a big step forward in the science of AI, the lack of a learning mechanism meant that it was not satisfactory in its current state [13]. As neurons were thought of as a representation of the brain, scientific advances and new ideas within the field of cognitive psychology had the potential to be beneficial also for the development of AI. Donald Hebb (1949) explored the methods of which the brain learned, and pointed out that neural pathways are strengthened each time they are used [13][14].

This idea led to the introduction of perceptrons, popularised by Frank Rosenblatt (1958), which included this crucial learning mechanism [12][13][30]. Rosenblatt discussed the crucial importance of an AI's ability to learn instead of just remembering. A system that had been given a huge data set containing e.g. pictures of different poses of a person could only recognise another person if a too large number of factors were met. This could include angle, posture, position, lighting, colour, background, other objects and sizes. For the system to be able to recognise people effectively, the

data set would have to be ridiculously large and the searching method, incredibly quick [30].

“In general, identities of this sort must be learned, or acquired from experience, and if the system is to be economical, the number of functional units in the storage system, or memory, should be much less than the number of forms or memories to be retained.” [30, p. 1]

The learning mechanism of the perceptron will train the neuron to produce the correct output by altering the weights and biases. This process would be repeated for all data sets that were chosen for training until the perceptron did not make any more mistakes [13][30]. While such a system seemed promising, a publication by Marvin Minsky and Seymour Papert (1969) proved, among other things, that the perceptrons were unable to learn the simple Boolean function XOR because it is not linearly separable. The idea that artificial neural networks were fundamentally flawed therefore falsely spread and lead to a decrease in development in the field [12][13][14].

During the 1980’s, the introduction of both multilayered networks, non-binary activation functions and backpropagation meant that artificial neural networks were beginning to look much more like how they are used today. These improvements made it possible to approximate more advanced functions which makes it more applicable to the real world [13].

3.4 The artificial neural network

Most of the theories described in this thesis about artificial neural networks are based on the “neural network and deep learning” e-book by Michael Nielsen [11]. In an artificial neural network, the neurons have the job of calculating an output based on a set number of inputs. The network works together to create an output that is

more complicated than what a single neuron could create. The reason it is called a network is because the neurons are connected in a web-like fashion with so-called weights. These weights, w , multiply with the input neurons value, x , before they are added with a bias, b , which then becomes the output of the neuron in question, z . A demonstration of this is shown in Figure 3.1 and should be considered together with Equation (3.9). This equation is called an activation function and can take both linear and non-linear forms [11].

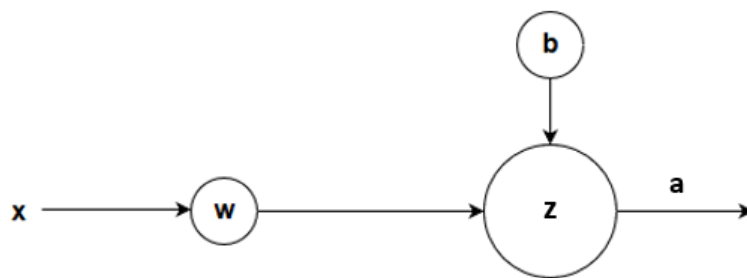


Figure 3.1: A single neuron.

$$z = x \cdot w + b \tag{3.9}$$

Equation (3.9) is actually only part of the full equation which is normally used in the network. Most networks utilise multiple inputs and multiple neurons in what is called layers. The first layer, as can be seen in Figure 3.2, is the input layer, and contains the input values. The middle layers are 'hidden' layers which are named thus as they are not inputs or outputs of the network. The output layer is where the result of the network can be gathered. Using a non-linear activation function for these neurons, it is possible to derive an equation that fully represents the output of a neuron in the hidden and output layer as seen in Equation (3.10). An activation function called the sigmoid function, as shown in Equation (3.11), is showcased here, and is one of many

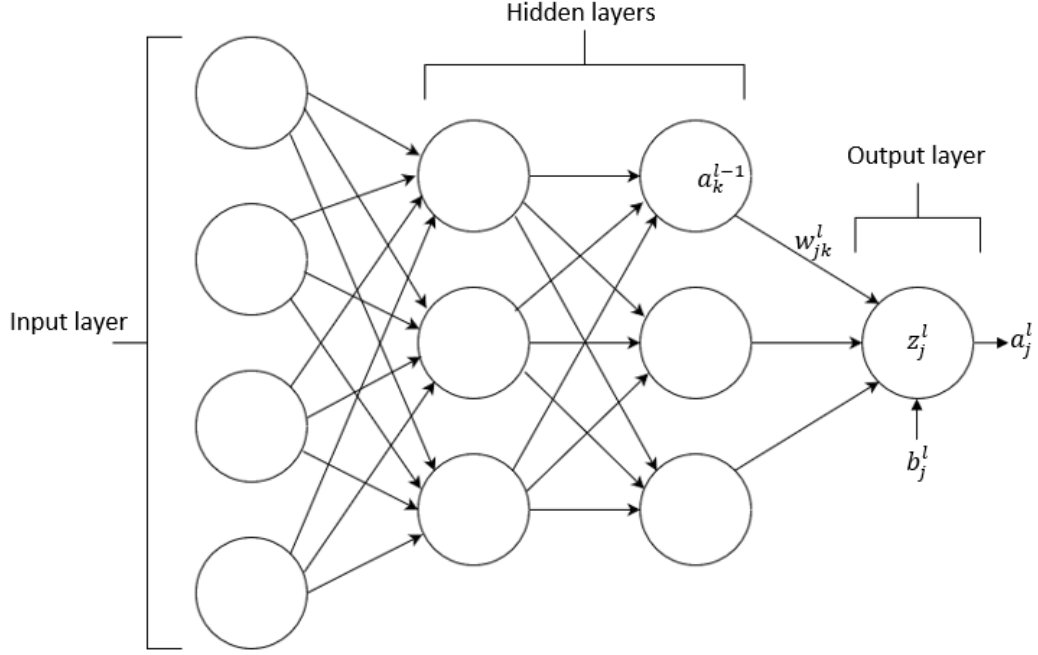


Figure 3.2: Classification of layers in a artificial neural network [11].

non-linear activation functions [11].

$$a_j^l = \frac{1}{1 - \exp\{-\sum_k (w_{jk}^l \cdot a_k^{l-1} + b_j^l)\}} \quad (3.10)$$

$$a_j^l = f(z_j^l) = \frac{1}{1 - \exp\{-z_j^l\}} \quad (3.11)$$

Beware that the variables of Equation (3.10) differ slightly from Equation (3.9) where the activation of the previous layer, a_k^{l-1} , is used instead of the data input x to the input layer, and the activation of the current layer, a_j^l , is used instead of the linearly activated function z . As an example, w_{jk}^l should be read as; the weight for the connection from the k^{th} neuron in the $(l-1)^{th}$ layer to the j^{th} neuron in the l^{th} layer [11].

3.5 Backpropagation

Backpropagation is a method of finding suitable weights and biases based on the error of the current network, or in other words, a training method for the network. The network improves by making small changes to its weights and biases for every training sequence, or epoch. These small changes are the reason the classical perceptron does not work well to model, as even a tiny change could completely flip the output from 0 to 1 or the opposite. The use of non-binary activation functions therefore becomes important. Instead of the complete switch, small changes Δw_j in the weights and Δb in the bias will produce a small change Δa in the output from the neuron. This can be represented by

$$\Delta a \approx \sum_j \frac{\delta a}{\delta w_j} \Delta w_j + \frac{\delta a}{\delta b} \Delta b, \quad (3.12)$$

where w_j is the sum over all the weights and $\delta a/\delta w_j$ and $\delta a/\delta b$ denote partial derivatives of the output with respect to w_j and b , respectively. To determine how well the weights and biases chosen are performing, a loss function is used, as shown in Equation (3.13).

$$C_{MSE}(a) = \frac{1}{n} \sum_{i=1}^n (y_i - a_i)^2 \quad (3.13)$$

Where a_i represents the activation vector of the neurons for the $i = 1, 2, 3, \dots, n$ observation. This activation is then compared to y_i which are the actual values being approximated and the desired output. This loss function is known as the mean squared error function. The sum over all observations i , indicate the networks performance and should with training approach 0, $C(a) \approx 0$. Another loss function is the mean absolute percentage error (MAPE), as shown in Equation (3.14), which instead of squaring the difference between y_i and a_i , uses the absolute value of them. The main difference between the two equations is that MSE give more weight to larger

errors, while MAPE treats all errors linearly.

$$C_{MAPE}(a) = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - a_i}{y_i} \right| \quad (3.14)$$

The loss function gives information about how much the activation has to increase or decrease to reach the desired value. As the activation is a function of the weight, bias and previous layer's activation, all these needs to change in the desired direction. Again, the previous layer's activation is a function of the previous layer's weights, biases and the activation before that layer again. This all backpropagates to the input layer, and to make sure the final output changes correctly, all previous weights and biases needs to change.

One way to approach 0, or at least the lowest possible value, is by the use of gradient descent. The following describes the basics of gradient descent optimisation. The idea here is that for every combination of weights and biases, the value of the loss function equals a value which can, at least for a one- or two-dimensional system, be represented by a graph. The lowest point in the graph represents the optimal weights and biases. To get there is therefore the goal. As all weights and biases are, generally, randomly chosen before the first epoch of the network, the fastest way "downhill" is also the fastest way to find the optimal weights and biases. In Figure 3.3, this process is visualised by the use of a ball which represents the initial value of the loss function. The ball would through each epoch roll closer and closer to the lowest point. This graph is notably fairly simple and only uses one input value, while a larger network would use too many to visually represent [11].

One problem with gradient descent is that the loss function is calculated as an average for individual training inputs. In other words, the gradient needs to be computed separately for each training input before averaging them, which takes time. A solution to this is to use multiple training inputs together before calculating the gradient. It turns out that the gradient of a randomly selected number of training inputs is

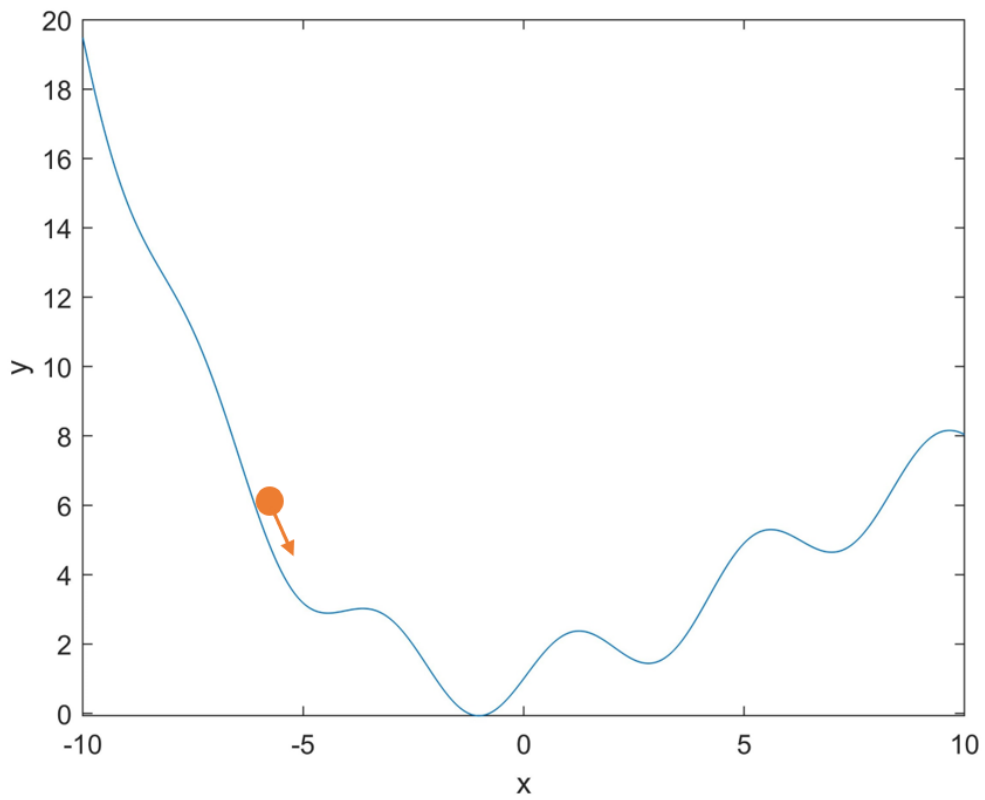


Figure 3.3: Gradient descent, given inputs x .

approximately equal to the gradient of individual training inputs. This can be shown as in Equation (3.15),

$$\frac{\sum_{j=1}^m \nabla C_{x_j}}{m} \approx \frac{\sum_x \nabla C_x}{n} = \nabla C, \quad (3.15)$$

where ∇C is the notation for gradient descent and m is the number of random training inputs, X_1, X_2, \dots, X_m are chosen for what is called a mini-batch of training. As long as the mini-batch of m training inputs is large enough, this approximation is accurate. When training with a mini-batch, the weights and biases in the network are changed, before a new mini-batch is chosen and used for training until all training inputs has been exhausted. At this point one epoch has been completed and the process can start again. Several algorithms use this theory for optimising an artificial neural network [11]. Notable examples are the Stochastic gradient descent algorithm (SGD) and the Adam optimiser. The Adam optimiser tends to perform well in the initial portion of training, but does not generalise as well as SGD. SGD does however have a disadvantage of scaling the gradient uniformly in all directions, which the Adam optimiser attempts to correct by diagonally scaling the gradient via estimates of the function's curvature [31].

3.6 Improving the way ANN learns

There are a great range of methods that can be used to improve the way the ANN learns. These includes methods for e.g. normalising the input data, generating larger data sets, and choosing the best hyperparameters [11, Chapter 3]. The point is to make the network be able to "fit" the data better, but when improving the network, it is important to be wary of underfitting and overfitting.

3.6.1 Underfitting and overfitting

When a network is unable to model the complexity of a model, i.e. by trying to explain an exponential function with a linear model, it is said that the network is underfitted. The network wrongly assumes that the model is simple and can be explained with only a few parameters. This can happen when too few neurons are used [5]. Overfitting on the other hand is when the network is no longer generalising to the data. The network could be trying to make a model which is either too complex or only learns how to remember the training data instead of generalising to new data. This can be detected by the increase of the loss function, the stabilising of the accuracy or when the training accuracy reaches close to 100% while the test accuracy stays low. This does not necessarily mean that when these instances happen, that overfitting is happening, but it can be safe to assume so. In general, one of the best ways of reducing overfitting is to increase the size of the training data [11, Chapter 3]. Some of the reasons why overfitting occurs will be explained later in this section.

3.6.2 Hidden layer design

The process of designing the hidden layers does not follow any ordinary rules, but researchers have instead developed many design heuristics¹ which can be used. The size and the number of the layers are what to be chosen. While it could be tempting to simply use a large deep artificial neural network², this could have a few drawbacks. One is the time needed for computation, but the other is the occurrence of overfitting.

The power of the ANN means that if there is a possible model that describes the data well, the network should be able to find it given enough complexity in the hidden

¹“A method of learning or solving problems that allows people to discover things themselves and learn from their own experiences.” [32]

²A deep artificial neural network is a network which uses more than one hidden layer [11]

layers. This is well described by the universal approximation theorem, as described by Cybenko, G. (1989):

“(…) Any continuous function can be uniformly approximated by a continuous artificial neural network having only one internal, hidden layer and with an arbitrary continuous sigmoidal nonlinearity.” [33, p. 312]

Even with just a single layer, any continuous function, regardless of complexity, may be approximated using a neural net with a non-linear activation function. For practical terms, this means that, as long as the input and output data is sufficient and contains some sort of relationship, a neural net should be powerful enough to approximate it [11, Chapter 4]

3.6.3 Activation functions

As there are many activation functions to choose from, picking the correct one could prove difficult as there are no solid rules on how to choose and how to modify the best activation function. Once again, these choices are based on design heuristics which mostly prove that certain functions are only marginally better than others in some tasks. However, a few functions are more used than others as they generally generate good results. Slightly different from the sigmoid function, which is more traditionally used, is the hyperbolic tangent (tanh) function, which is fairly similar in shape and also includes negative values, where the sigmoid only has positive. A problem that is sometimes present with these functions is the saturation they experience when the output gets closer to their maximum or minimum ($[0,1]$ and $[-1,1]$ for sigmoid and tanh respectively). This phenomenon is called the vanishing gradient problem and effectively hinders the weight from changing its value as the derivative of the gradient is approximately zero. Another function, called the Rectified Linear Unit (ReLU) does not experience this saturation, and will not lead to learning slowdown in the

same way as it is linear for positive values and zero for negative values [11, Chapter 3]. ReLU produces another problem however, as it is entirely non-negative it has a mean activation larger than zero. This can lead to a phenomenon called “bias shift” or a “dead” activation function. In short this means that a neuron learns a large negative bias which makes the activation function unlikely to ever activate. This can result in a network only producing zero-values for its estimations. [34] introduces the Exponential Linear Unit (ELU) which is similar to ReLU but uses an exponential function multiplied by a constant when below zero. This shifts the activation mean towards zero which speeds up learning, ensures a noise-robust deactivation state, and learning does not result in zero-values for estimations. Additionally, the linear function, which is directly linear to z can be mentioned as it is the most basic activation function. The activation functions mentioned in this section are shown in Equations (3.16) through (3.20) and Figure 3.4.

Sigmoid:

$$f(z) = \frac{1}{1 + \exp\{-z\}} \quad (3.16)$$

ReLU:

$$f(z) = \max(z, 0) \quad (3.17)$$

Tanh:

$$f(z) = \tanh(z) \quad (3.18)$$

Linear:

$$f(z) = z \quad (3.19)$$

ELU:

$$f(z) = \begin{cases} z & \text{for } z > 0 \\ c \cdot (\exp\{z\} - 1) & \text{for } z \leq 0 \end{cases} \quad (3.20)$$

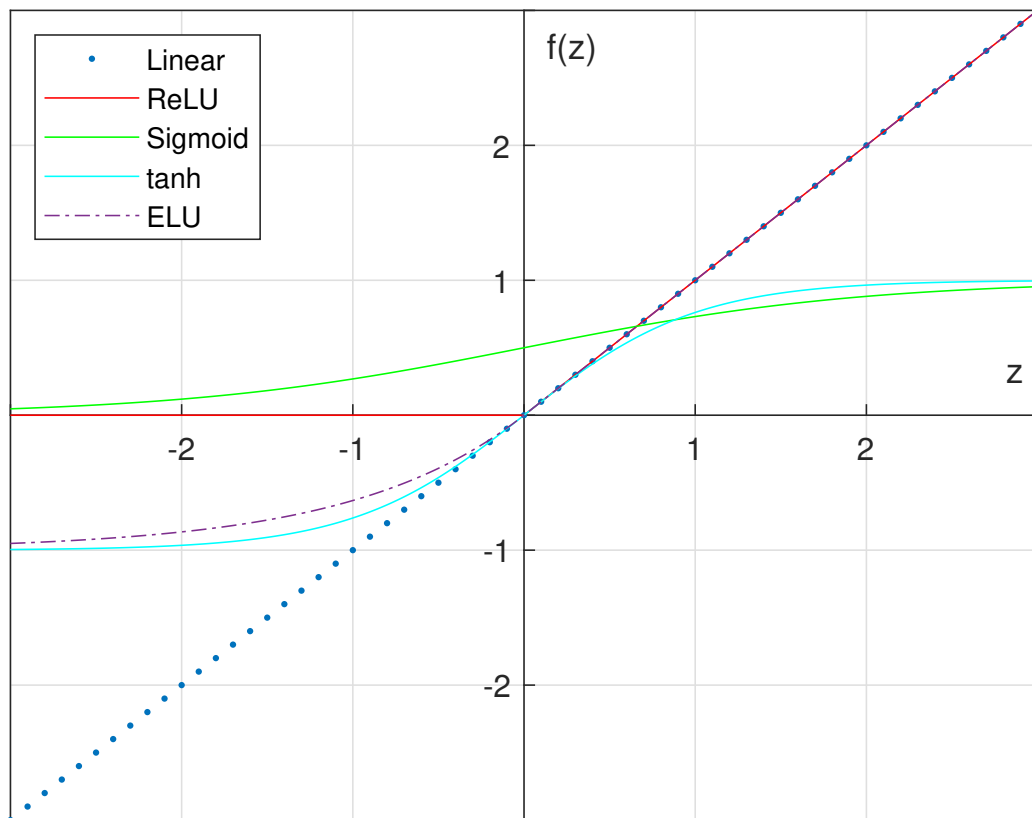


Figure 3.4: Response of different activation functions.

3.6.4 Regularisation

A method of reducing overfitting is what is known as weight decay or L2 regularisation. This method adds an extra term to the cost function. Equation (3.21) shows the regularisation term together with an arbitrary loss function, C_0 .

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2 \quad (3.21)$$

where λ is the regularisation parameter, n is the size of the training set, and $\sum_w w^2$ is the sum of all weights in the network. The effect of this method is to punish the network for choosing larger weights. This can also help the network from getting stuck in local minima of the cost function at times. For more complex systems it could however be beneficial to use an unregularised network, as the large weights helps modelling the complexities. Similar to L2 regularisation is L1 regularisation, where the difference is that it uses the absolute value of the weight instead of the square. This method results in weights being reduced by a constant amount instead of a proportional one and means that larger weights gets reduced less than smaller weights. In effect, this leads to the network concentrating on a smaller number of high importance connections and reduces other weights towards zero.

3.6.5 Dropout

Dropout is a method that modifies the network structure continuously during training. Given an ordinary network, as shown in Figure 3.5a, the input values are forward-propagating through the connections, and then backpropagated. With dropout, a specified amount of random neurons are temporarily deleted (as shown in Figure 3.5b) before the usual forward-propagating and backpropogating takes place in the remaining connections. This happens once for each mini-batch and for every time a new set of random neurons are deleted. With this method, it is like multiple networks

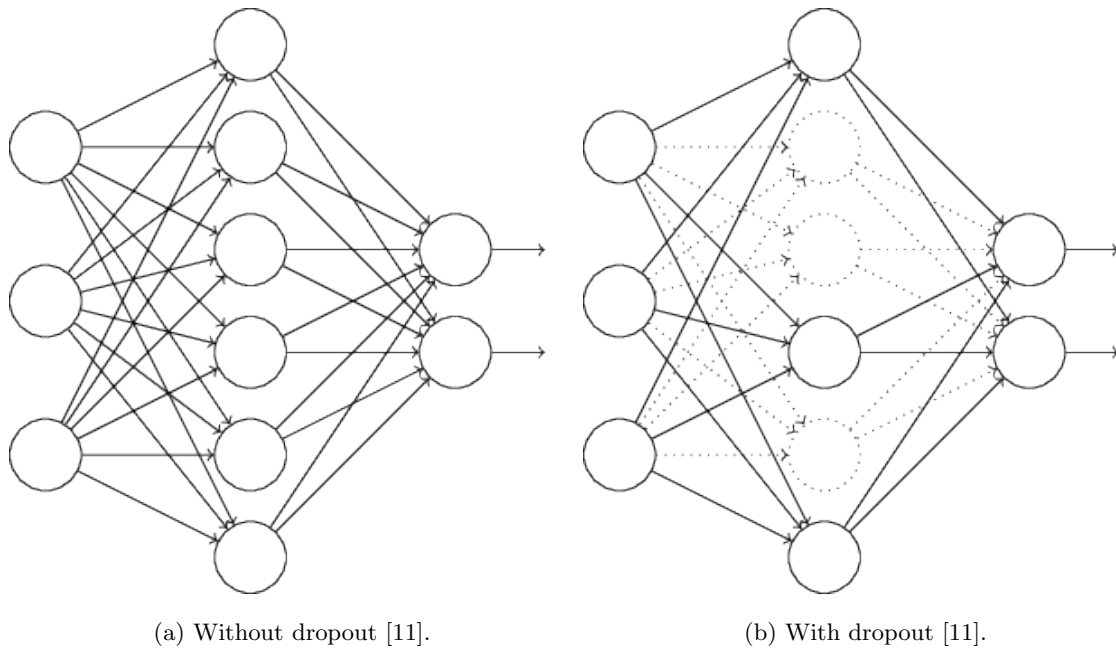


Figure 3.5: Illustration of dropout in an arbitrary artificial neural network.

are being trained and the results are being averaged between them. This usually gives a better result and helps reducing overfitting. In addition, the effect of dropout can be described as by A. Krizhevsky et.al. (2012):

“This technique reduces complex co-adaptations of neurons, since a neuron cannot rely on the presence of particular other neurons. It is, therefore, forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.” [35, p. 6]

3.6.6 Artificially expanding the training data

As previously mentioned, a network usually creates better results if it is given a larger data set for input. While it can be difficult to obtain additional data, it is possible to artificially expand the existing data set. E.g. for number recognition, where numbers

written by humans are to be identify by the ANN, this could be done by expanding, reducing, rotating or skewing the number slightly. While the input values would be different, it would for the human eye still be easily recognisable as the same number. It should therefore still be possible for the ANN to identify the number correctly. Adding or reducing noise could also help create a larger data set. The important factor is that the new data needs to still represent reality and not create relationships or correlations where there are none. An ANN which is given pure noise would, given enough complexity, still be able to find some sort of pattern and correlations, which is unwanted in most cases.

3.6.7 Normalising inputs and activations

When solving practical tasks, it is not unusual for input parameters to vary in sizes and ranges. One parameter could e.g. have a range of [1000, 50000] while another has [0.2, 2.5]. Having such a difference in ranges between input parameters can slow down learning. By normalising the input parameters to have similar ranges, this problem can be avoided [36]. In Equation (3.22) a normalisation method is shown:

$$\hat{x}_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (3.22)$$

where x_i is the unnormalised value, x is the parameter vector and \hat{x}_i is the updated, normalised value. This results in a range of [0, 1] which is easier to use.

Normalising can also be included as a part of the model architecture and is usually called batch-normalisation. This method was developed by Sergey Ioffe and Christian Szegedy [37] and is effective as it speeds up training of deep neural networks and allows for higher learning rates. The batch-normalisation works by fixing the means and variances of layer activations for every individual mini-batch. Equations (3.23) through (3.25) shows the algorithm used to accomplish the batch-normalisation. First

calculating the mean:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m a_i \quad (3.23)$$

where μ_B is the mini-batch mean, a_i is the activation of $i = 1, 2, \dots, m$ observations in the mini-batch. Then calculating the variance:

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (a_i - \mu_B)^2 \quad (3.24)$$

where σ_B^2 is the mini-batch variance, and then finally calculating the normalised value:

$$\hat{a}_i = \frac{a_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (3.25)$$

where \hat{a}_i is the updated and normalised activation value and ϵ is a constant added for numerical stability.

3.6.8 Hyper-parameter tuning

When improving the network, choosing the optimal hyper-parameters can be beneficial. These are parameters that are set before training is started which determines the structure of the network and the variables which determine how the network is trained. Knowing which hyper-parameter to change to get a better result is difficult and usually requires testing. In previous sections, hyper-parameters of dropout, L1 and L2 regularisation and hidden layer size has been discussed. While changing these might help, it is important to also consider the following parameters: Learning rate, epochs and mini-batch size.

The learning rate is a measure of how large of a step the gradient descent is taking. Too large and it might skip right past the optimal value of the loss function, too small and it might take forever to get there. Finding a good learning rate needs some testing to understand how changing the learning rate affects the loss function. Even with an optimal learning rate, the loss could begin to stagnate without it actually having

reached the optimal loss value. When the loss quickly decreases before it stagnates, it could be beneficial to combine it with a learning rate decay. The decay means that the learning rate gets smaller for every epoch, resulting in a learning rate that is better suited for the current epoch. When getting closer and closer to the optimal value of the loss function, smaller and smaller steps are required to not overstep it, which is why a learning rate decay can give slightly better results.

The epoch, as explained earlier, is the amount of times the network runs through all observations in the training data set. Choosing the number of epochs should be based on when the network stops learning anything of value. If both the accuracy on the test data and the loss has stagnated, the remaining epochs are usually unnecessary. It is however important to experiment with more epochs as a network can sometimes stagnate for a while before it suddenly starts improving again.

A mini-batch is the randomly chosen training inputs that are used during gradient descent when trying to decrease the loss. An option is to use a mini-batch size of just 1, which would give some noticeable errors for each mini-batch. However, due to how often the gradient updates, the general direction of the gradient is maintained. As the network has to loop over each mini-batch until all training inputs has been exhausted, having such a small mini-batch can be time consuming. A similar but more directionally reliable method is to use a small mini-batch size of between 2 or 32 as proposed by Dominic Masters and Carlo Luschi [38]. This allows for faster computation while avoiding memory problems and still being able to update the gradient frequently.

Chapter 4

Experimental methodology

This chapter will explain the methods used in this thesis and was based on the theories from the theory chapter (Chapter 3). These methods were mainly completed with the help of three programs: Microsoft Excel, MATLAB and Python. Excel was used to have good visual control over the data. MATLAB was used to import and export data between programs and create multiple linear regression models. Python was used to create the artificial neural networks with the Keras package. As the Levenberg-Marquet algorithm which has been frequently used in recent studies of this topic is not as easily available in the Keras package as it is in Matlab, it has been chosen to use the well performing optimisers available in Keras instead.

4.1 Data correlation and analysis

The data that was available for the WWTP had already been divided into months for the daily data set and years for the bi-weekly data set. This meant that it was not easy to look at long term trends and to export the data to different software. To more easily analysis the data, the data sets were structured into two tables. However, both data sets had some incomplete and false series. The false series were series where some of the values were obviously misentered as the values were incoherent with the rest of the data set. I.e. values that were larger than the possible limits or values that were lower than the previous in an ever-increasing series. The incomplete series were deleted, while the false series were fixed by entering the values that were assumed to be correct. During the time spans of the data sets there had been made changes to the plant in different ways which could potentially alter how each parameter correlated to each other. For instances where the plant had been partly out of service or malfunctioning, the data series were removed as they did not represent the normal behaviour of the

plant.

As the plant has been running for many years, there was a possibility that the data did not behave consistently in correlation to each other. Physical alterations of the plant and process changes could mean that one model would not be able to describe the plant well both before and after the changes. With larger differences between the different time periods, it would be necessary to make models which described the more recent time period in addition to a model which described the plant for the complete data set. The following dates had comments either about process changes, installations or malfunctions:

- 05.09.14 - 07.10.14: Only one pool was in operation.
- 06.08.15: The plant was out of service.
- 05.03.16: Bio reactor start.
- 14.06.16: Salsnes filter installed.
- 24.09.16: Centrifuges installed.
- 05.10.16 - 14.03.17: Bio reactor out of service.
- 08.12.16: Machine malfunction.
- 28.08.17: Precipitation¹ of iron begins.
- 13.09.17-14.09.17: Flotation out of service.
- 26.09.17, 28.09.17 - 29.09.17: Salsnes filter out of service.
- 01.03.18: Failure of polymer pump for flotation.
- 17.07.18: Ran for a few hours without course treatment.

¹“Chemical precipitation in water and wastewater treatment is the change in form of materials dissolved in water into solid particles.” [39]

- 01.08.18: Large parts of the day without a blower in the grit remover.
- 30.08.18: Line 1 closed due to defect in Nikuni pumps.
- 17.10.18: Troubles with polymers due to full scale test.
- 08.11.18 - 09.11.18: Displacement pump stopped for 2.5h for maintenance work.
- 07.12.18: Change of PLS which affected the polymer additives.
- 16.01.19: Downscaling of line 1. New point for adding iron.

Note that these dates may be inaccurate compared to when the events actually happened at the plant and that they might have been of a slightly different duration, start or stop date due to how the events were logged.

The data was analysed to find any existing correlations using the Pearson's correlation coefficient. Parameters with high correlation could be crucial in describing each other either directly or in combination with other parameters. If there proved to be noticeable differences during the time span of the data either due to the mentioned plant deviations or other reasons, it could also be necessary to compare correlations from only specific time periods. Even though it would be beneficial for training to use as much data as possible, using all of it could be a misstep due to how much the plant had changed over the years. Using only the more recent data could therefore be a possibility.

It could also be seen in Tables 2.1 and 2.2 that the overlapping values of TP_{in} , TP_{out} , and Qw_{in} gave deviating values. As these were supposed to be identical for the same time period (note that the referenced tables are of different time periods) a comparison of the values were performed. This was performed by extracting values that were measured on the same day and comparing them.

4.2 Regression models

Before making the more complicated ANN models, regression models were made for each data set. The results of these models could be used as a comparison to the ANN models, and as the MLR models were faster to train and experiment with, they were used to get a good idea about the optimal inputs and data ranges to use. As the bi-weekly data set used four inputs and three outputs, three models were needed, one for each output. Because of the single output of the daily data set, one model was sufficient. In addition, there was made models for the daily data set which contained either all input parameters, and only influent parameters. As the time between the measurement of TP_{in} and the flocculation step was only about 30-40 minutes, it was theorised that measurements could be collected over this time period and combined for a single set of model inputs which would control the next 30-40 minutes of wastewater. I.e. the measurements of Qw_{in} and TP_{in} would be measured for a single sample of water which would then use some time to reach the next measurement points. Once the sample had reached all measurement points the data could be used by the model to predict the influent water and then control the amount of PIX added to the water. This model would then repeat continuously. While such a model was not made in this thesis due to lack of time continuous measurements, testing the quality of both influent and effluent input parameters would give valuable information about its possibilities.

To determine if one or more of the parameters reduced the quality of the model, a sensitivity analysis was performed by using different combinations of input parameters and considering the different effects the parameters had on the result. This would also give information about which parameters were the most important to include in a data-driven model of the plant and which could potentially have a negative effect on the result. The models were made in MATLAB and were based on the methods

described in Chapter 3.1 and used one constant multiplier for each parameter used. In addition, any values below zero were changed to zero as negative numbers would not be possible. The data were split into 70% training data and 30% test data, both sampled in a random order. To avoid obtaining models that were based on statistical outliers, the tests were re-run 1001 times. The median result, in regard to correlation between model output and actual output, was chosen as the MLR model. This was also the model that was used for the sensitivity analysis. As the median of a series finds the average of the two middle values if the series is of even numbers, an odd number was used.

The quality of the models were considered with respect to correlation, r , mean squared error (MSE) and mean absolute percentage error (MAPE). The correlation coefficient indicates how much the estimated values increase or decrease in correlation with the actual values. Both MSE and MAPE describes how far each value is from the expected value, and as MSE punishes larger errors more, it suggests more about the larger errors which are unwanted. MSE and r were considered the major quality parameters as it was important that the models had as few large deviations as possible and responded proportionally with the reference values. When models were compared to each other, MSE and MAPE values were only considered when comparing models with the same setup. This was to avoid problems with a superior model having a higher MSE or MAPE values than an inferior model due to using data with higher values in general.

4.3 Artificial neural network setup

The ANN models were made with the same approach as with the MLR models regarding inputs, outputs and amount of data used. Data was also randomly split into 70% training data and 30% test data. While the MLR models were obtained by choosing a median result based on multiple training attempts, the ANN models were

not trained in this way. This was because an ANN model takes considerably longer to train compared to an MLR model. Every test was also run on the same random state for easier parameter tuning. Although it would be possible to use only a single model with three outputs for the bi-weekly data set, this was not done. It was instead decided that a more tailored model would be more beneficial as the optimal model for the different output parameters would not necessarily be the same. This has also shown to be a valid concern in existing literature [1]. Due to the different architecture of the ANN models compared to the MLR models, a new sensitivity analysis was also performed.

The setups that proved the best for the MLR models were used as base models for the ANN before new setups were examined. This was in regard to time periods and input parameters. The quality of the ANN models were also considered based on the same parameters as the MLR models: correlation (r), MSE and MAPE. In addition, any negative values were changed to zero as negative numbers would not be possible.

As the ANN models had more room for improvement than the MLR models, they were tested by adding and removing the different functions described in Chapter 3 as well as changing the different hyper-parameters. Initially, the base model for the bi-weekly data set composed of 3 input neurons, 10 neurons in a single hidden layer, and 1 output layer. This is shortened as a 3-10-1 network. The daily data set had a base model structure of 7-10-1 due to more available input parameters. As new parameters like $Temp.$, $date$, Q_{rw} , and $Q_{w_{in}}$ were added, the base of the models were changed to 8-10-1 and 12-10-1 respectively, using all input parameters. An 8-10-1 layer setup was also used for the daily data set which removed the input parameters that had their measurement point after PIX_{add} was added. In other words, this model only used influent input parameters. All of the models used an MSE loss function, stochastic gradient descent optimiser function, sigmoid activation function, 100 epochs, a batch

size of 10 and a 0-1 normalised input with all the default options to begin with. These models were then optimised separately with the use of the following methods and functions: noise, batch normalisation, regularisers and dropout. In addition, the hyper-parameters of the base model were optimised, and different optimisers, activations, and loss functions were tried.

Chapter 5

Data analysis results

This chapter shows the results in regards to the data analysis. First the data trends will be shown and the data will be split into 6 different time periods which is meant to better represent the plant at its different stages through 2002 - 2019. The correlation between each parameter is then calculated based on these time periods. At last, a comparison between overlapping values of the bi-weekly and daily data set is shown regarding TP_{in} , TP_{out} , and Qw_{in} .

5.1 Data trends

In Figures 5.1a through 5.1g the different parameters of the bi-weekly data set is shown. The first observation is at 04.04.06 and the last is at 13.12.18, with about 2 years and 6 month to 3 years and 4 month between every 50th observation (about 3 years on average). These figures exclude data prior to 04.04.06 as measurements of Qw_{in} and $Temp.$ were not present. In Figures 5.2a through 5.2k the different parameters of the daily data set is shown. The first observation is at 02.01.17 and the last is at 12.03.19, with about 5 months between every 100th observation. There were a few instances where noticeable changes had happened: a drop in values of BOD_{out} and COD_{out} occurred in August 2015 in the bi-weekly data set (observation 156-226); a steady decrease of TS_{out} happened between September 2017 and April 2018 before it instantaneously increased five-fold, which resembled values present previously in the daily data set (observation 178-310); PIX_{add} more than doubled in value after July of 2017 in the daily data set (observation 130-543). In regards to the rebuilding and process changes of the plant, there was not noticed any obvious direct change in the behaviour of the future data except for when precipitation of iron begun, which lined up well with the steady decrease of TS_{out} . However these results may be unrelated.

Note that even though the treatment efficiency $Treat_{eff}$ has been shown in Figure 5.2j, it has not been used in any of the models and has only been included to showcase its trends. Malfunctions and other process stops had at times some potential direct effect on the data, and these were removed from the data sets. The data were split into six different time periods with similar behaviour as these changes could mean that there would be a reduced correlation between the different periods. These time periods are shown in each of the data trends. In Figures 5.1d, 5.1f, 5.2g and 5.2d the time period splits can be seen matching up with the mentioned changes. The daily data set starts in the middle of time period 3 as the first data of the set is at 02.01.17. The bi-weekly data set stops in the middle of time period 6 as the last data of the set is at 13.12.18.

- Period 1: 20.Feb 2002 - 03.Apr 2006
- Period 2: 04.Apr 2006 - 13.Aug 2015
- Period 3: 14.Aug 2015 - 10.Jul 2017
- Period 4: 11.Jul 2017 - 14.Sep 2017
- Period 5: 15.Sep 2017 - 06.Apr 2018
- Period 6: 07.Apr 2018 - 12.Mar 2019

5.1.1 Bi-weekly data set trends

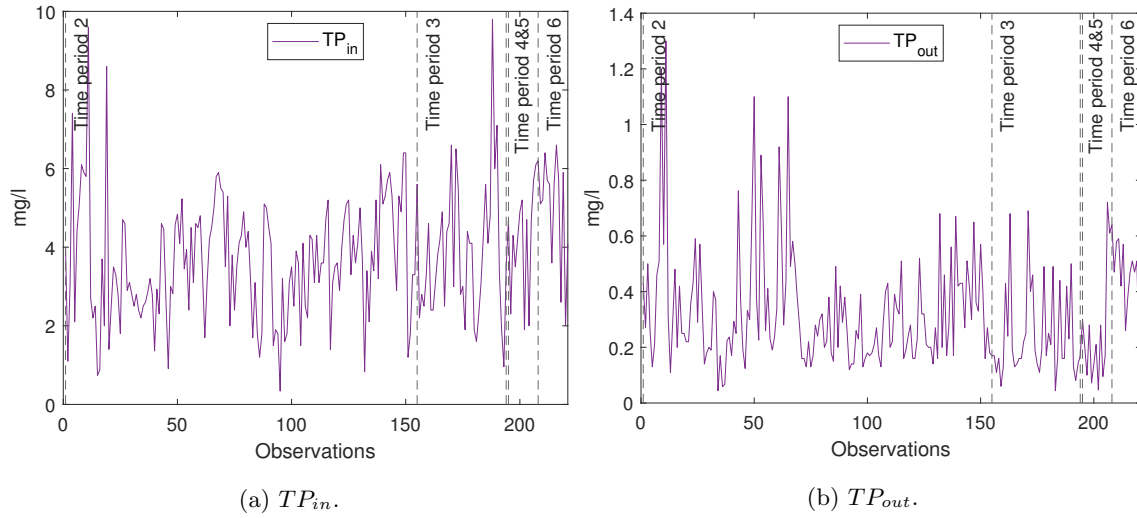


Figure 5.1: Bi-weekly data set parameters plots, 04.04.06 - 13.12.18.

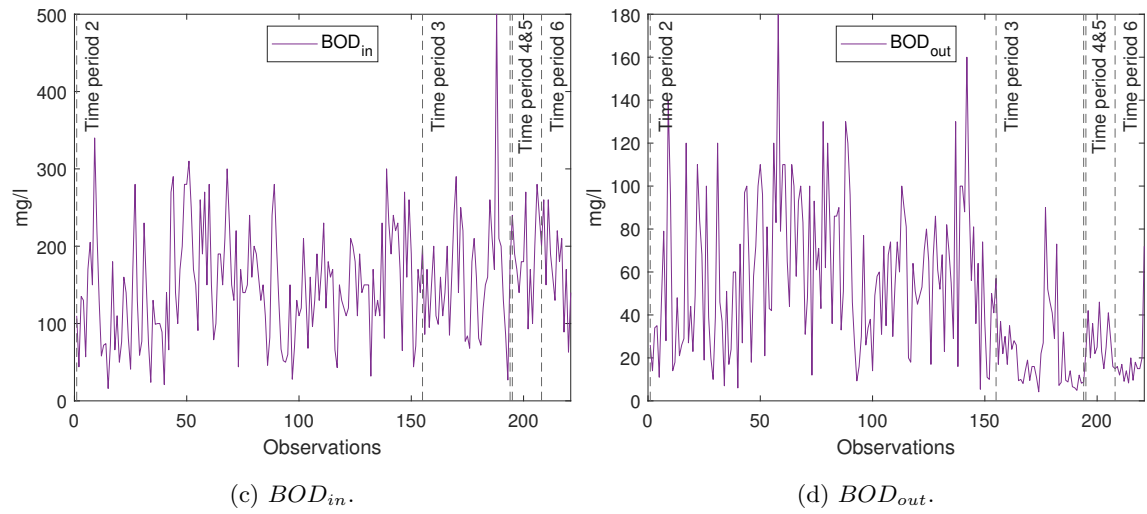


Figure 5.1: Bi-weekly data set parameters plots, 04.04.06 - 13.12.18 (Cont.).

5.1. DATA TRENDS

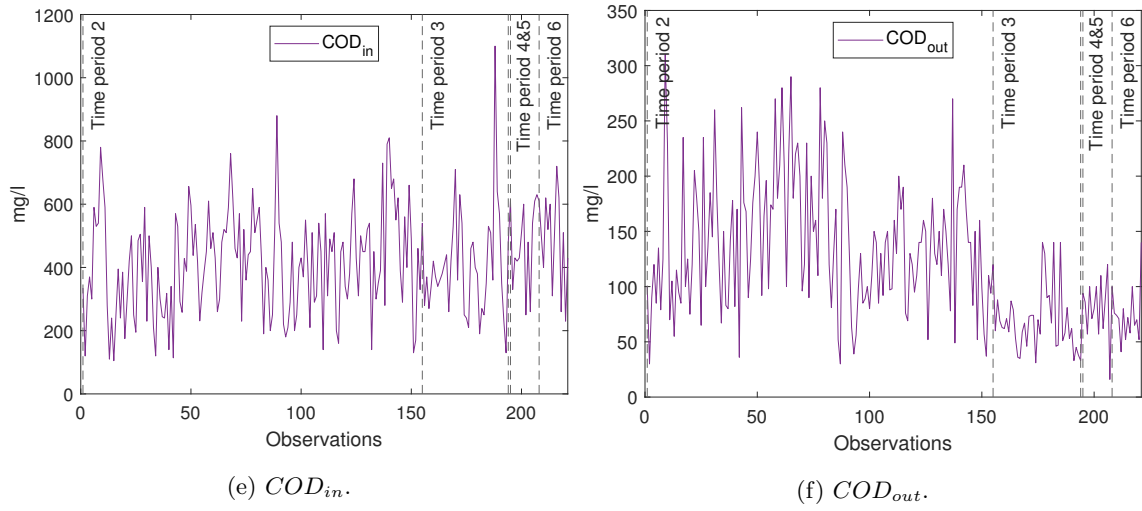


Figure 5.1: Bi-weekly data set parameters plots, 04.04.06 - 13.12.18 (Cont.).

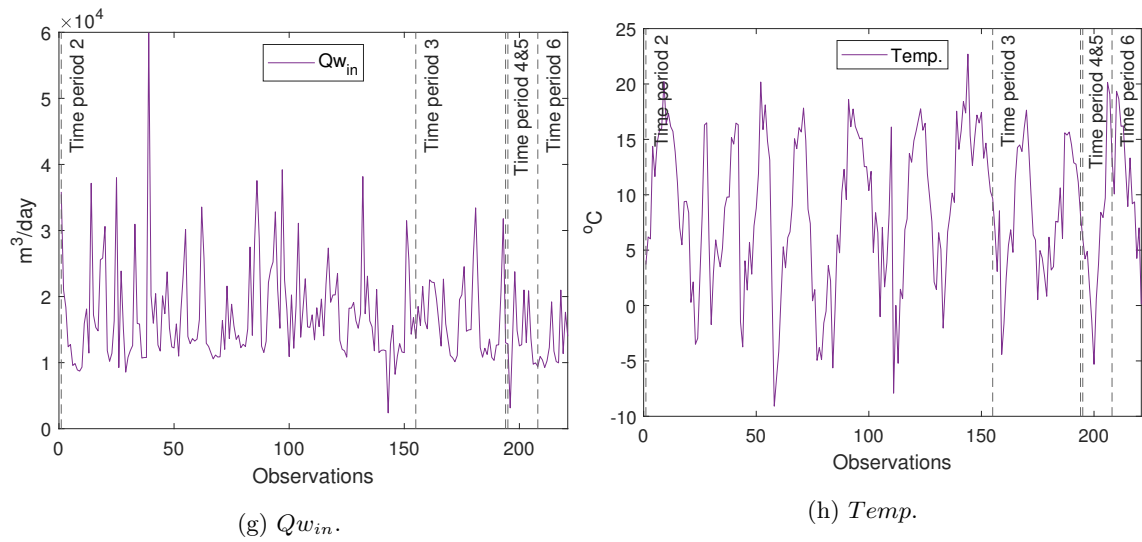


Figure 5.1: Bi-weekly data set parameters plots, 04.04.06 - 13.12.18 (Cont.).

5.1.2 Daily data set trends

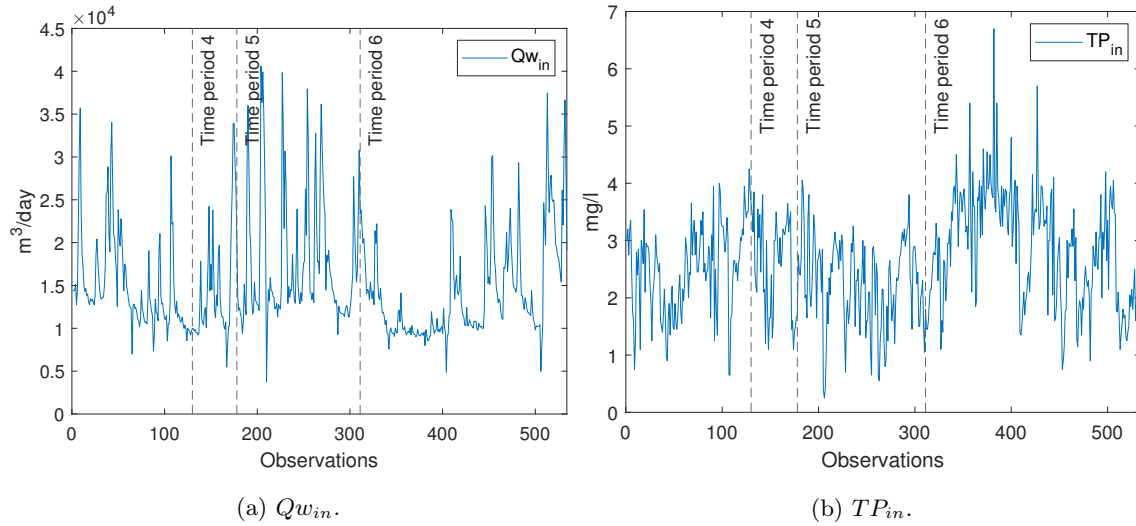


Figure 5.2: Daily data set parameters plots, 02.01.17 - 12.03.19.

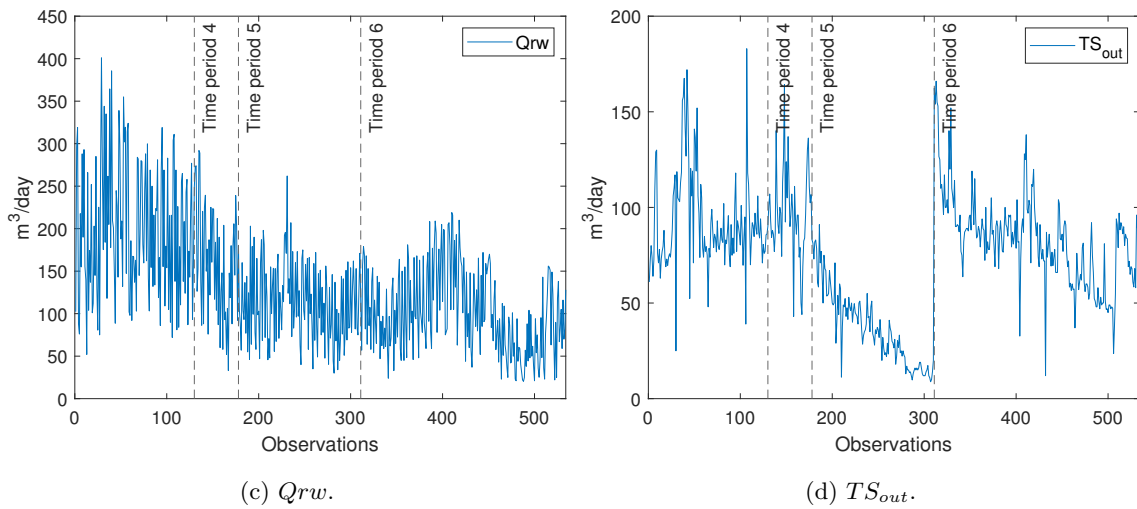


Figure 5.2: Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).

5.1. DATA TRENDS

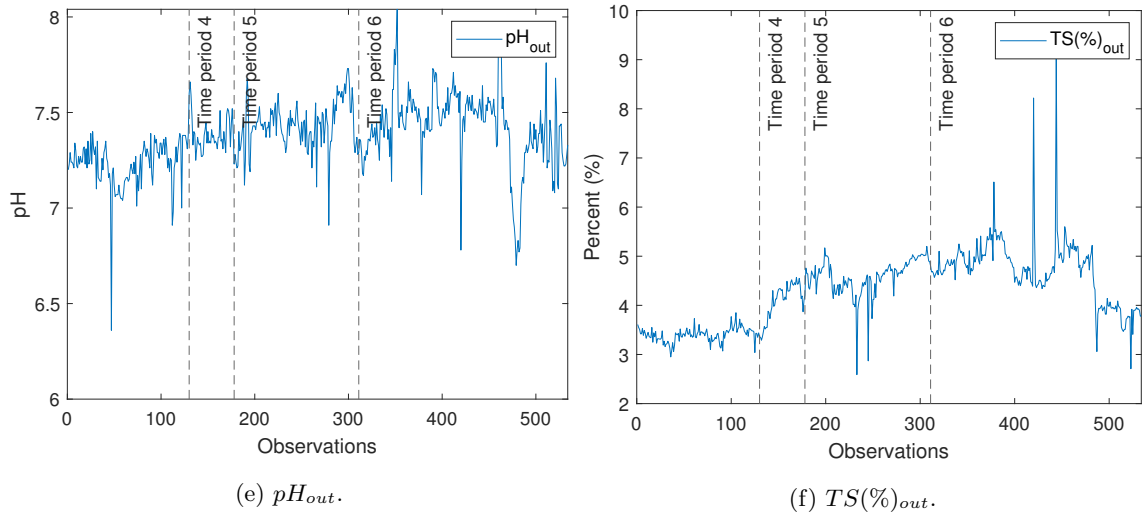


Figure 5.2: Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).

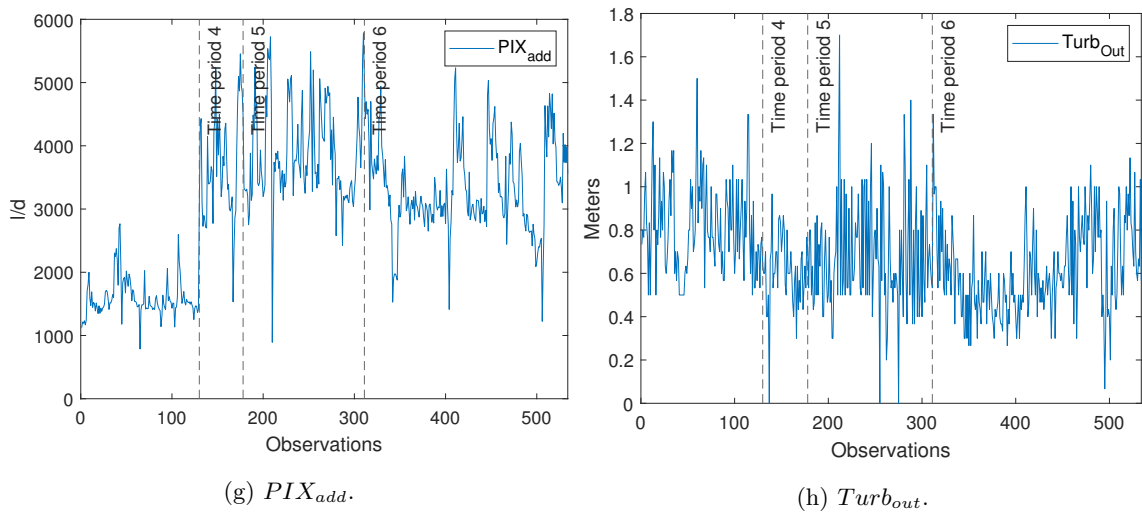


Figure 5.2: Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).

5.2. DATA CORRELATIONS

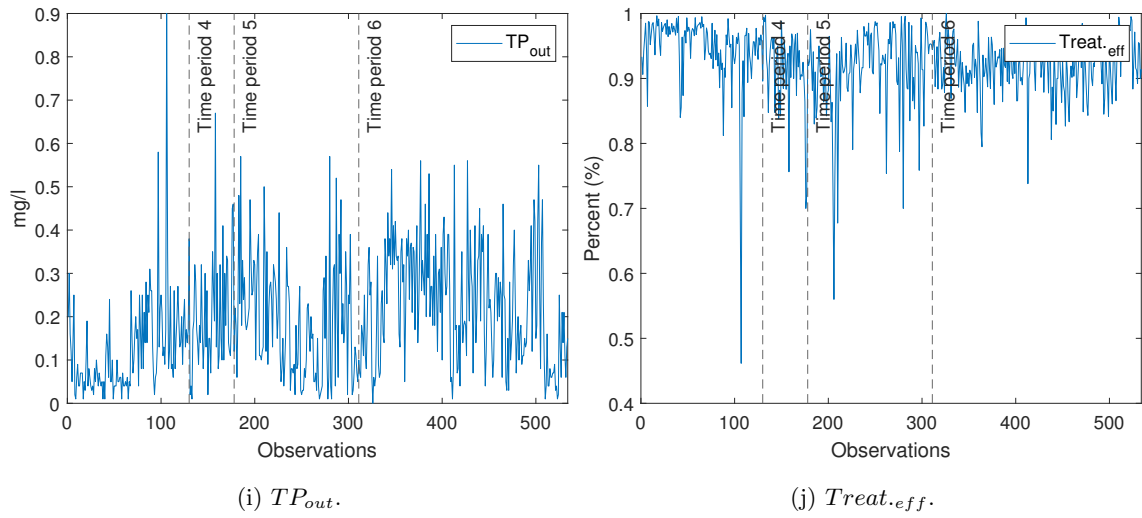


Figure 5.2: Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).

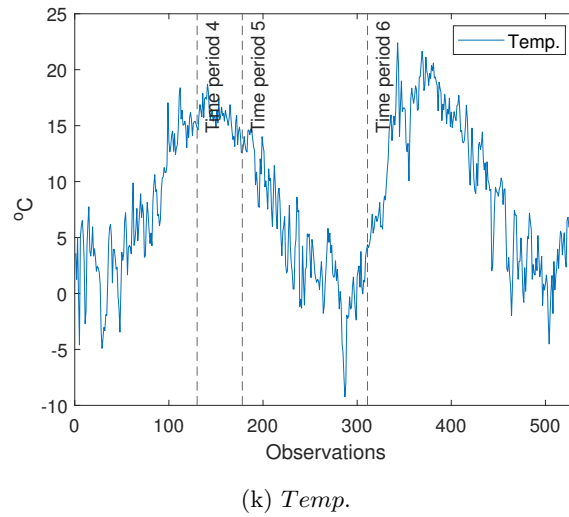


Figure 5.2: Daily data set parameters plots, 02.01.17 - 12.03.19 (Cont.).

5.2 Data correlations

The Pearson's correlation coefficient was used to find correlations between the different parameters. Correlation between output and input data did not give values

higher than 0.5 for neither the bi-weekly or daily data set using time periods 2-6 and 3-6 respectively. Correlations were however in general much higher when using time period 3-6 for the bi-weekly data set and time period 6 for the daily data set. When using the bi-weekly data set and time period 2, BOD_{out} and COD_{out} has much higher correlations with influent values than in later time periods. In Tables 5.2 and 5.4 the correlation between each parameter is shown for all data of their respective data sets, and in Tables 5.3 and 5.5 the correlation between each parameter is shown for only the later periods for the bi-weekly and daily data sets respectively. Table 5.1 shows the correlation of the bi-weekly data set using time period 2. Notably there was a high correlation between TP_{in} , COD_{in} and BOD_{in} for the bi-weekly data set, as well as between BOD_{out} and COD_{out} . There was a large negative correlation between Qw_{in} and the influent parameters TP_{in} , BOD_{in} and COD_{in} . For the daily data set, there was only a significant correlation between Qw_{in} and TP_{in} using all time periods. When using only time period 6, there were however much higher correlations in regards to both PIX_{add} and TP_{out} with other parameters.

Table 5.1: Correlation between parameters in bi-weekly data set for time period 2, 04.04.06 - 13.08.15.

Parameter	TP_{in}	TP_{out}	BOD_{in}	BOD_{out}	COD_{in}	COD_{out}	Qw_{in}	$Temp.$
TP_{in}	1.00	0.40	0.68	0.41	0.75	0.42	-0.52	0.20
TP_{out}	-	1.00	0.38	0.34	0.35	0.44	-0.11	0.13
BOD_{in}	-	-	1.00	0.76	0.84	0.69	-0.55	0.02
BOD_{out}	-	-	-	1.00	0.66	0.86	-0.38	-0.25
COD_{in}	-	-	-	-	1.00	0.67	-0.58	0.04
COD_{out}	-	-	-	-	-	1.00	-0.38	-0.24
Qw_{in}	-	-	-	-	-	-	1.00	0.00
$Temp.$	-	-	-	-	-	-	-	1.00

5.2. DATA CORRELATIONS

Table 5.2: Correlation between parameters in bi-weekly data set for time periods 2-6, 04.04.06 - 13.12.18.

Parameter	TP_{in}	TP_{out}	BOD_{in}	BOD_{out}	COD_{in}	COD_{out}	Qw_{in}	$Temp.$
TP_{in}	1.00	0.40	0.75	0.17	0.80	0.22	-0.57	0.25
TP_{out}	-	1.00	0.36	0.29	0.37	0.40	-0.14	0.20
BOD_{in}	-	-	1.00	0.45	0.86	0.43	-0.58	0.09
BOD_{out}	-	-	-	1.00	0.42	0.89	-0.21	-0.23
COD_{in}	-	-	-	-	1.00	0.46	-0.59	0.12
COD_{out}	-	-	-	-	-	1.00	-0.23	-0.20
Qw_{in}	-	-	-	-	-	-	1.00	-0.06
$Temp.$	-	-	-	-	-	-	-	1.00

Table 5.3: Correlation between parameters in bi-weekly data set using time periods 3-6, 14.08.15 - 13.12.18.

Parameter	TP_{in}	TP_{out}	BOD_{in}	BOD_{out}	COD_{in}	COD_{out}	Qw_{in}	$Temp.$
TP_{in}	1.00	0.44	0.87	-0.14	0.93	0.16	-0.74	0.35
TP_{out}	-	1.00	0.34	0.03	0.43	0.34	-0.29	0.39
BOD_{in}	-	-	1.00	0.01	0.91	0.26	-0.72	0.25
BOD_{out}	-	-	-	1.00	-0.02	0.79	0.08	-0.45
COD_{in}	-	-	-	-	1.00	0.24	-0.70	0.29
COD_{out}	-	-	-	-	-	1.00	-0.14	-0.28
Qw_{in}	-	-	-	-	-	-	1.00	-0.23
$Temp.$	-	-	-	-	-	-	-	1.00

5.2. DATA CORRELATIONS

Table 5.4: Correlation between parameters in daily data set for time period 3-6, 02.01.17 - 12.03.19.

Parameter	Qw_{in}	TP_{in}	Qrw	TS_{out}	pH_{out}	$TS(\%)_{out}$	PIX_{add}	$Turb_{out}$	TP_{out}	$Temp.$
Qw_{in}	1.00	-0.72	0.08	0.11	-0.15	-0.08	0.53	0.11	-0.39	-0.28
TP_{in}	-	1.00	-0.14	-0.06	0.22	0.17	-0.36	-0.22	0.47	0.37
Qrw	-	-	1.00	0.39	-0.16	-0.39	-0.24	-0.10	-0.08	0.07
TS_{out}	-	-	-	1.00	-0.17	-0.21	-0.11	0.04	-0.13	0.42
pH_{out}	-	-	-	-	1.00	0.38	0.32	-0.27	0.26	0.2
$TS(\%)_{out}$	-	-	-	-	-	1.00	0.5	-0.33	0.33	0.2
PIX_{add}	-	-	-	-	-	-	1.00	-0.14	-0.05	0.01
$Turb_{out}$	-	-	-	-	-	-	-	1.00	-0.45	-0.29
TP_{out}	-	-	-	-	-	-	-	-	1.00	0.3
$Temp.$	-	-	-	-	-	-	-	-	-	1.00

Table 5.5: Correlation between parameters in daily data set using time period 6, 07.04.18 - 12.03.19.

Parameter	Qw_{in}	TP_{in}	Qrw	TS_{out}	pH_{out}	$TS(\%)_{out}$	PIX_{add}	$Turb_{out}$	TP_{out}	$Temp.$
Qw_{in}	1.00	-0.78	-0.00	0.31	-0.32	-0.29	0.77	0.49	-0.59	-0.47
TP_{in}	-	1.00	-0.00	-0.23	0.29	0.25	-0.66	-0.48	0.58	0.49
Qrw	-	-	1.00	0.47	0.23	0.19	0.29	-0.34	0.21	0.36
TS_{out}	-	-	-	1.00	0.10	0.20	0.63	0.10	-0.22	0.32
pH_{out}	-	-	-	-	1.00	0.10	-0.08	-0.32	0.21	0.38
$TS(\%)_{out}$	-	-	-	-	-	1.00	-0.04	-0.21	0.26	0.42
PIX_{add}	-	-	-	-	-	-	1.00	0.42	-0.54	-0.18
$Turb_{out}$	-	-	-	-	-	-	-	1.00	-0.58	-0.46
TP_{out}	-	-	-	-	-	-	-	-	1.00	0.31
$Temp.$	-	-	-	-	-	-	-	-	-	1.00

5.3 Comparison of bi-weekly and daily overlapping measurements

The three parameters TP_{in} , TP_{out} , and Qw_{in} which were present in both data sets are compared in Figures 5.3a through 5.3c. Only measurements that were taken on the same day has been compared. The bi-weekly data set generally gives higher values for TP_{in} and TP_{out} , but values are fairly similar for Qw_{in} . A histogram of the deviations are shown in Figures 5.4a and 5.4b together with their respective normal distribution.

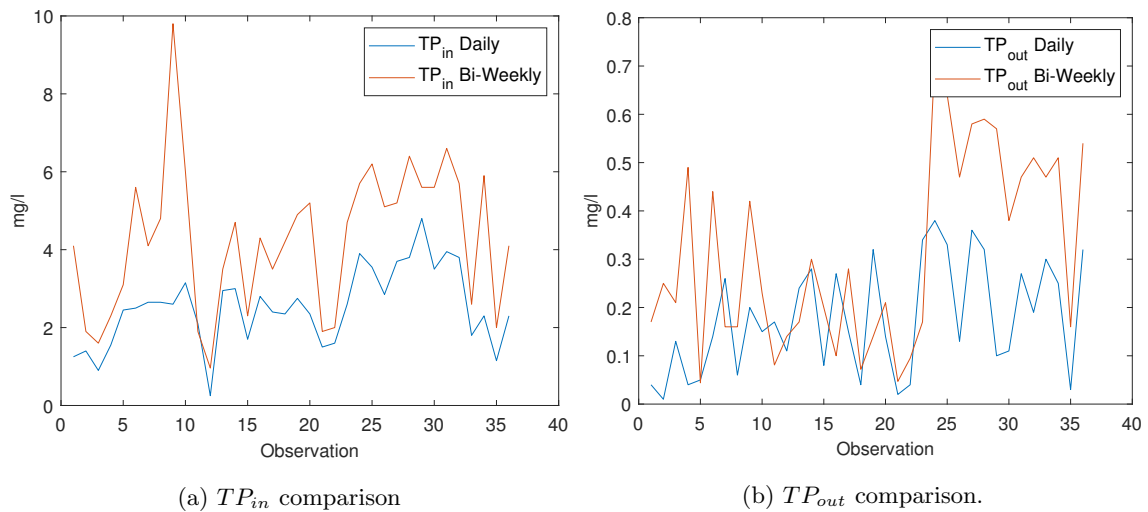
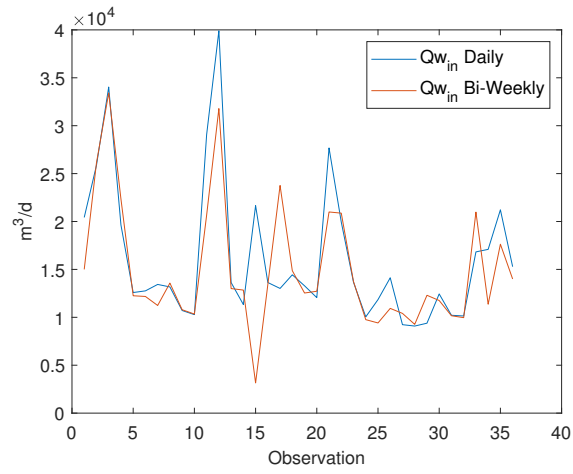


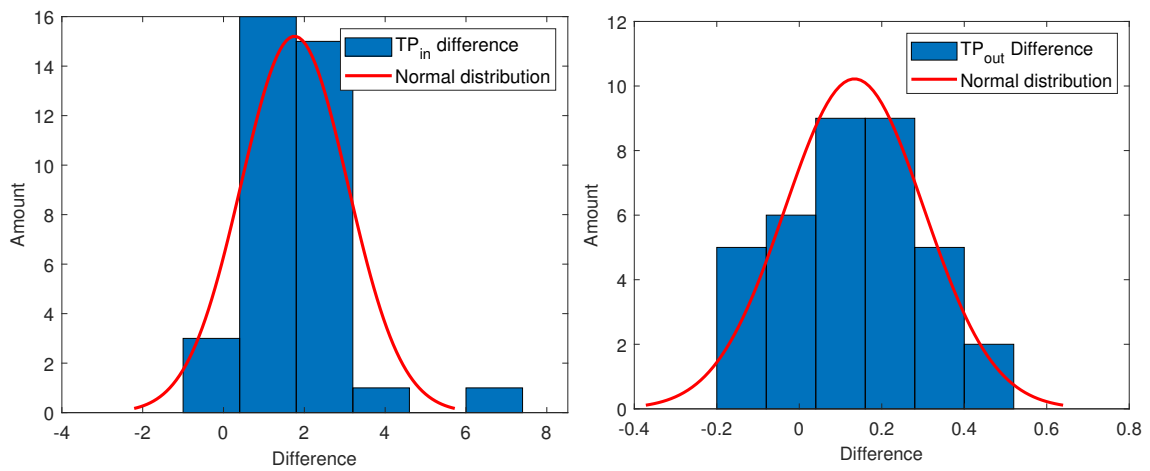
Figure 5.3: Comparison of overlapping data parameters for bi-weekly and daily data set.

5.3. COMPARISON OF BI-WEEKLY AND DAILY OVERLAPPING MEASUREMENTS



(c) Qw_{in} comparison.

Figure 5.3: Comparison of overlapping data parameters for bi-weekly and daily data set (Cont.).



(a) TP_{in} deviations.

(b) TP_{out} deviations.

Figure 5.4: Histogram of deviations between overlapping parameters.

Chapter 6

ANN and MLR model results

This chapter show the results from the ANN and MLR models. At first a high correlation between TP_{in} , COD_{in} and BOD_{in} is used to generate MLR models which could be used to guess upcoming COD_{in} and BOD_{in} values before they are measured by a laboratory. Secondly the MLR models for the bi-weekly data set is shown using time period 2 but also the results of time periods 2-6 and 3-6 are shown. This is also performed for the daily data set in the same way with time periods 3-6 and then 6. The results show how well the data can be estimated by a simple model. An output sensitivity analysis is also given to determine which parameters give the best and worst effect on the result. The same procedure is repeated for ANN models. All models has been generated with 70% training data and 30% test data which was randomly sampled. Data will be referred to as either training data, test data or all data, where all data includes both training and test data. Plots of the models are shown using all data, and correlations and MSE values are shown for both all data and for test data. Sensitivity analyses are compared in regards to all data to keep some consistency in the results. Note that the trends in this chapter has been randomly sorted before training, but has been sorted by date after testing was finished.

6.1 BOD and COD influent models

The high correlation between TP_{in} , BOD_{in} and COD_{in} was used to generate a linear approximation of their relationships shown in Figures 6.1a and 6.2b using all time periods. The day, month, year, TP_{in} , Qw , and $Temp$. were used as input for the MLR models. The correlation coefficient (r) and the mean squared error (MSE) are also shown for both test data and all data. Figures 6.3a and 6.3b uses only time periods 3-6 but show higher correlations. The models were also made by using only TP_{in} as

6.1. BOD AND COD INFLUENT MODELS

an input. These models gave a slightly higher correlation but much worse MSE for time period 3-6, slightly lower correlations and worse MSE for time period 2-6 and time period 2. These models are not shown.

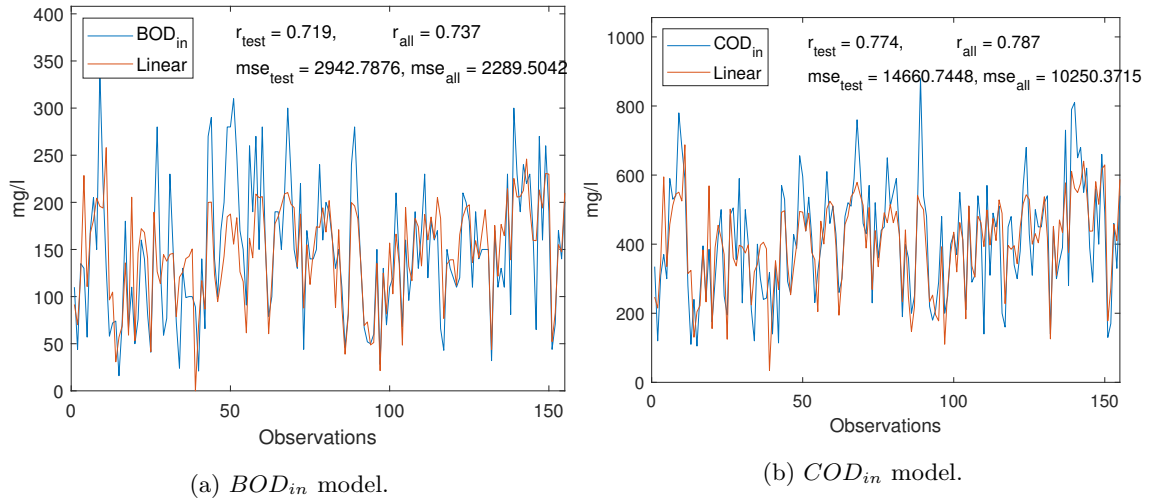


Figure 6.1: MLR models for generating BOD_{in} and COD_{in} , using time period 2, 04.04.06 - 04.08.15.

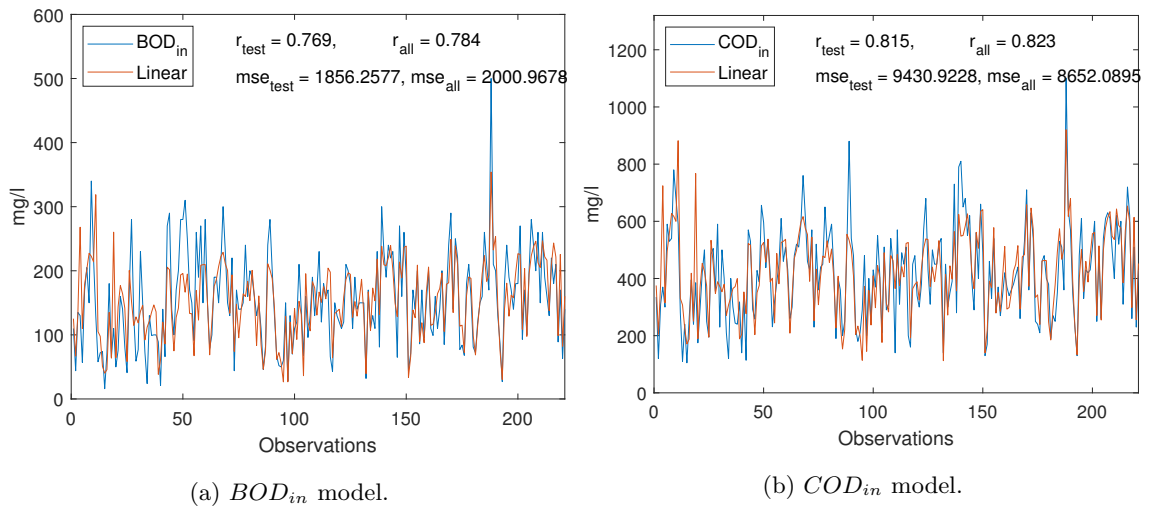


Figure 6.2: MLR models for generating BOD_{in} and COD_{in} , using time period 2-6, 04.04.06 - 13.12.18.

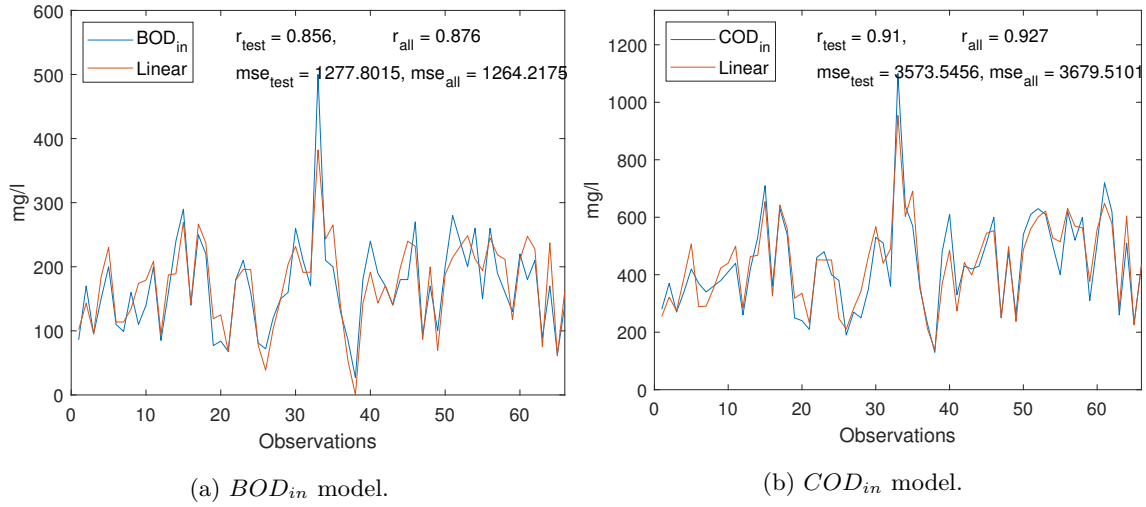


Figure 6.3: MLR models for generating BOD_{in} and COD_{in} , using time periods 3-6, 14.08.15 - 13.12.18.

6.2 MLR effluent models

Figures 6.4a, 6.4b, and 6.4c show the results of the MLR models for the bi-weekly data set using time period 2. Results for time periods 2-6 and 3-6 are shown in Table 6.1. This split was performed to show how correlations are generally high for time period 2 but are low for 3-6. Correlations are also much higher for BOD_{out} and COD_{out} than for TP_{out} . For these models, the month, year, TP_{in} , BOD_{in} , COD_{in} , Qw and $Temp.$ were used as inputs.

For the daily data set Figures 6.5a and 6.5b show the results of MLR models using data from time period 3-6. These models uses only influent parameters (month, year, Qw_{in} , TP_{in} , Qrw and PIX_{add}) and all parameters (month, year, Qw_{in} , TP_{in} , Qrw , TS_{out} , pH_{out} , $TS(\%)_{out}$, PIX_{add} and $Turb_{out}$) respectively. This is also the case for Figures 6.6a and 6.6b, but these models uses only data from time period 6. The results are better when using all inputs compared to using only influent inputs

(with correlation increasing from 0.58 to 0.61 and 0.68 to 0.72 respectively), but improves more significantly when using only data from time period 6 (with correlation increasing from 0.58 to 0.68 and 0.61 to 0.72 respectively). In the plots the correlation coefficient (r) and the mean squared error (MSE) are shown.

6.2.1 Bi-weekly data set models

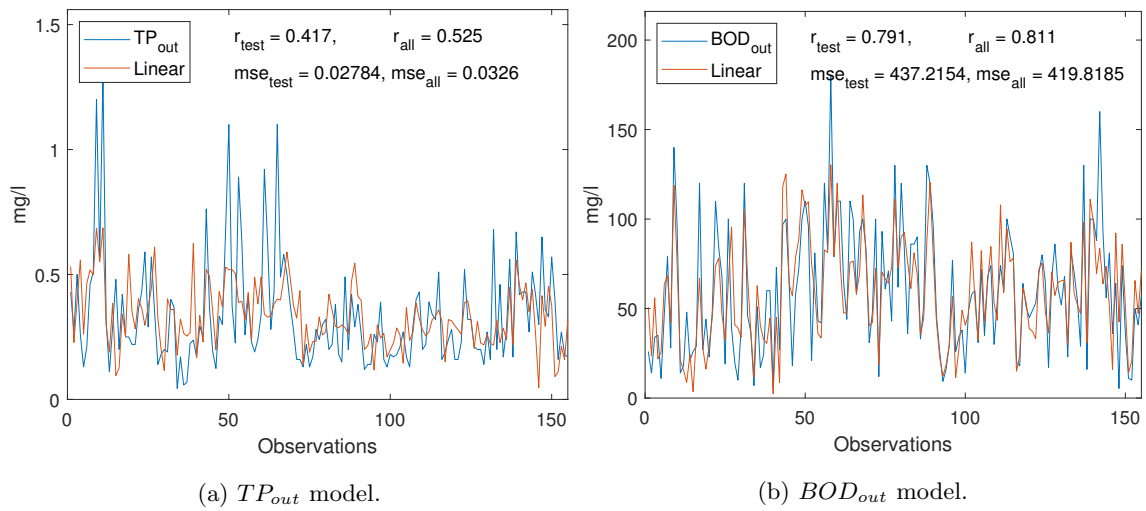
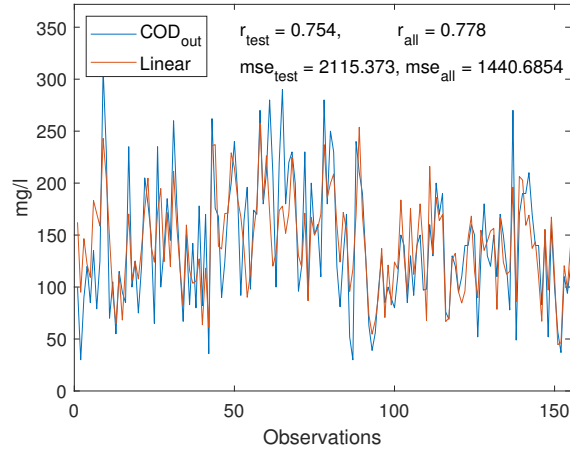


Figure 6.4: MLR models of the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.

6.2. MLR EFFLUENT MODELS



(c) COD_{out} model.

Figure 6.4: MLR models of the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).

Table 6.1: Remaining MLR models of bi-weekly data set.

Output	Time period	Dates	r_{test}	MSE_{test}	r_{all}	MSE_{all}
TP_{out}	2-6	04.04.06 - 13.12.18	0.415	0.04113	0.488	0.03206
BOD_{out}	2-6	04.04.06 - 13.12.18	0.683	674.2	0.712	610.8
COD_{out}	2-6	04.04.06 - 13.12.18	0.732	1781	0.754	1634
TP_{out}	3-6	14.08.15 - 13.12.18	0.425	0.02626	0.497	0.02708
BOD_{out}	3-6	14.08.15 - 13.12.18	0.485	147.9	0.589	185.3
COD_{out}	3-6	14.08.15 - 13.12.18	0.321	1147	0.46	617.0

6.2.2 Daily data set models

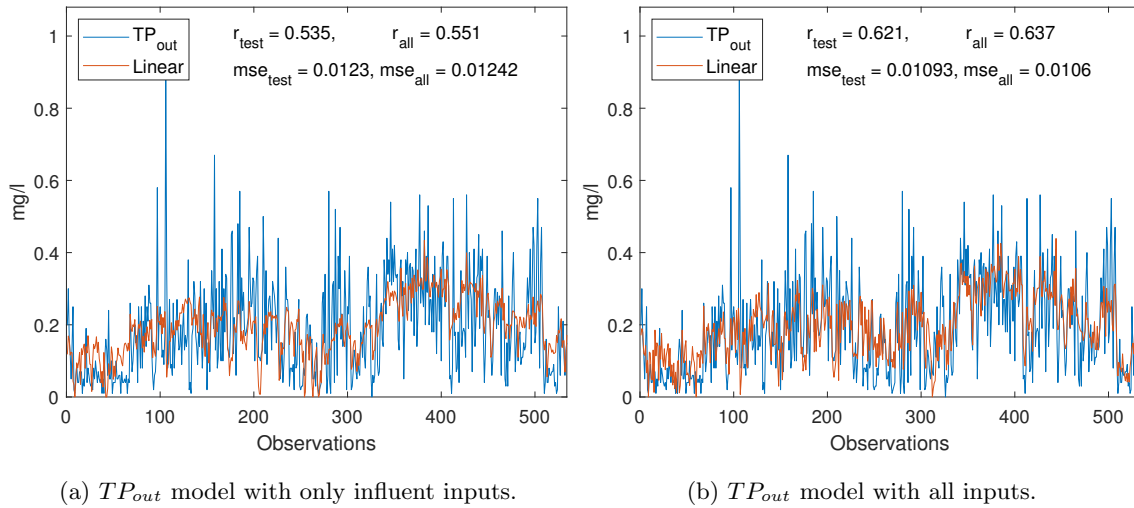


Figure 6.5: MLR models for daily data set using time periods 3-6, 02.01.17 - 13.03.19.

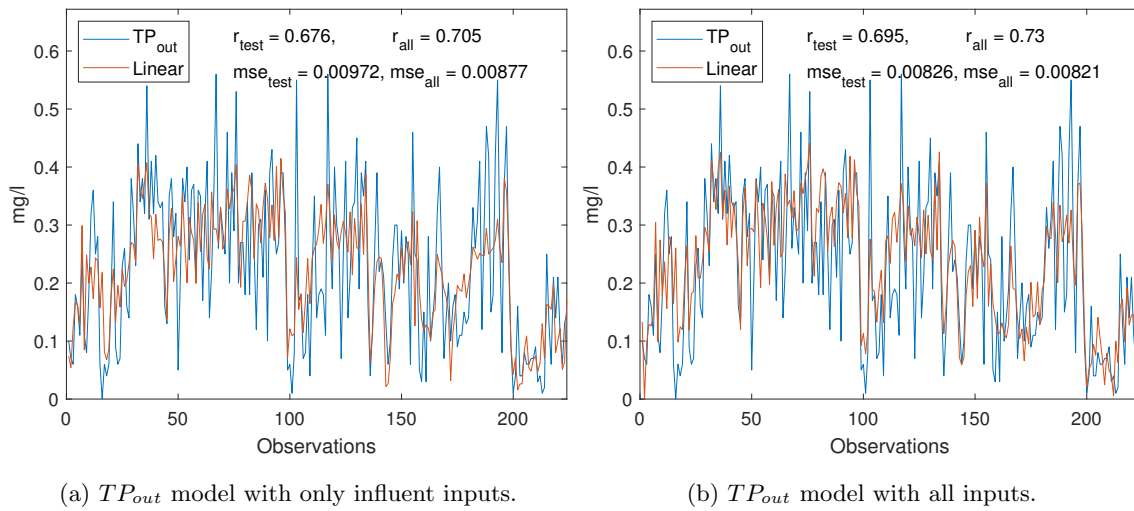


Figure 6.6: MLR models for daily data set using time period 6, 07.04.18 - 12.03.19.

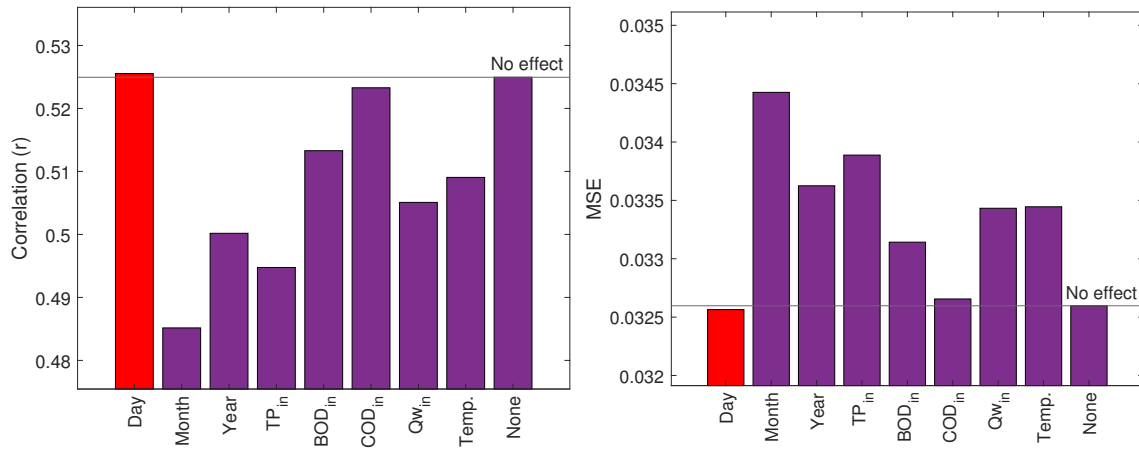
6.3 MLR effluent models sensitivity

To test which input parameters affected the results of the MLR models the most, all parameters were removed from the model one at the time and the results are compared. The analysis was based on the median result from the MLR models and are in Figures 6.7a through 6.11c compared by all data results. The graphs should be interpreted as the parameters that are furthest from the 'no effect' value (where no parameter is removed) are the ones that give the biggest effect on the result. A red bar colour has been given to parameters which give a positive effect when removed, while blue (daily data set) or purple (bi-weekly data set) is considered the opposite. In other words, parameters in a red colour should be removed from an optimal model. To get some consistency in the results, the sensitivity is compared by all data results, because test results gave different results between each test.

6.3.1 Bi-weekly data set

The results of Figures 6.7a and 6.7b closely mirror each other and show that TP_{out} is reliant on $Temp.$, Qw_{in} , TP_{in} , month, and year more than other parameters, while the day slightly worsen the result. The same can somewhat be said about MAPE in Figure 6.7c where Qw_{in} is the most significant parameter. For BOD_{out} the results in Figures 6.8a and 6.8b show that BOD_{in} is by far the most significant parameter, followed by $Temp.$, TP_{in} and COD_{in} . Figures 6.9a, 6.9b, and 6.9c show that COD_{out} is reliant on $Temp.$, BOD_{in} , COD_{in} and year.

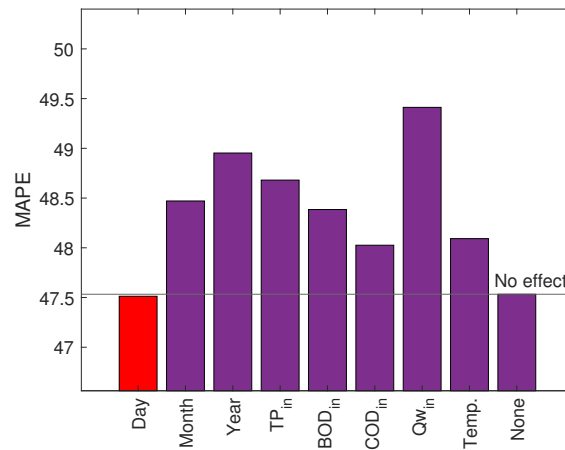
6.3. MLR EFFLUENT MODELS SENSITIVITY



(a) Sensitivity compared by correlation.

(b) Sensitivity compared by MSE.

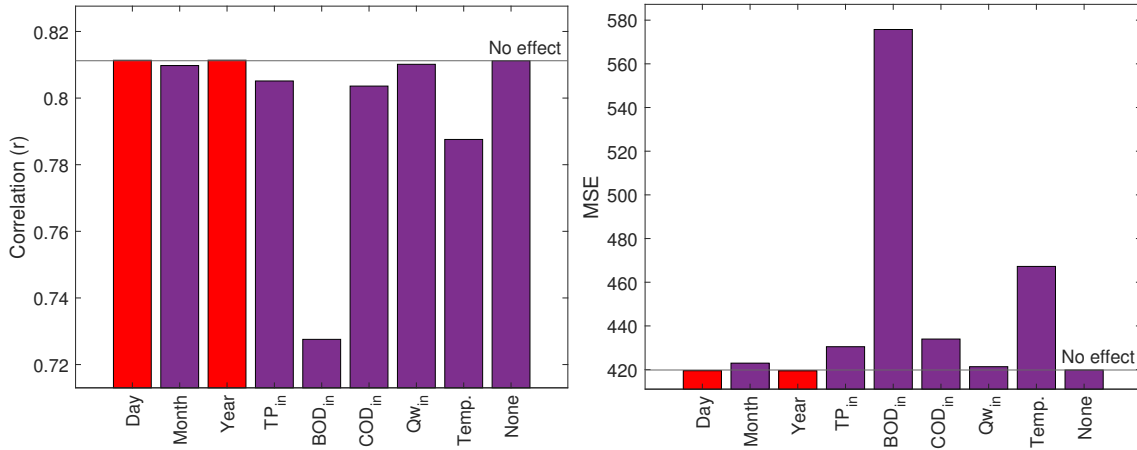
Figure 6.7: TP_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.



(c) Sensitivity compared by MAPE.

Figure 6.7: TP_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).

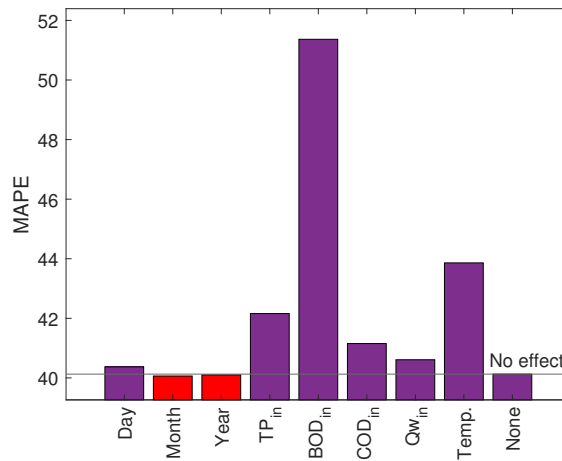
6.3. MLR EFFLUENT MODELS SENSITIVITY



(a) Sensitivity compared by correlation.

(b) Sensitivity compared by MSE.

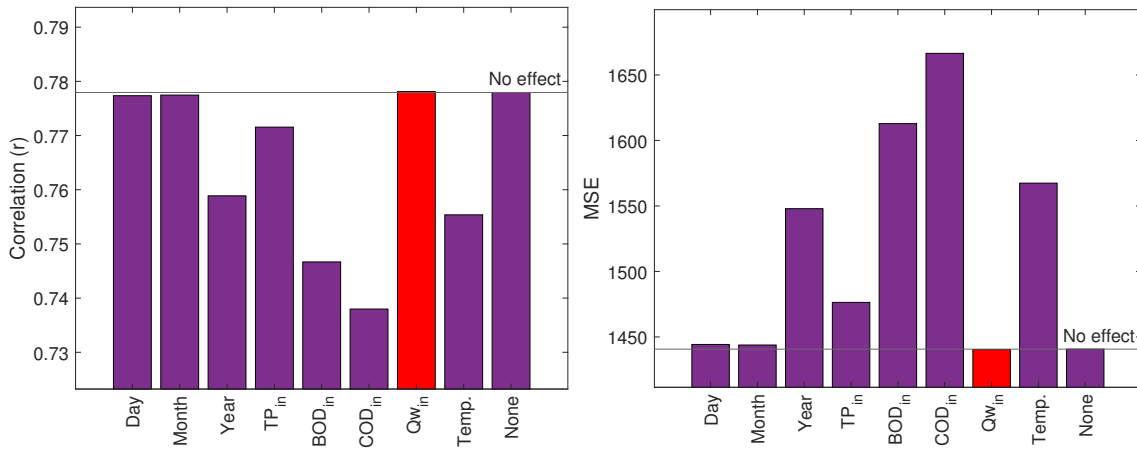
Figure 6.8: BOD_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.



(c) Sensitivity compared by MAPE.

Figure 6.8: BOD_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).

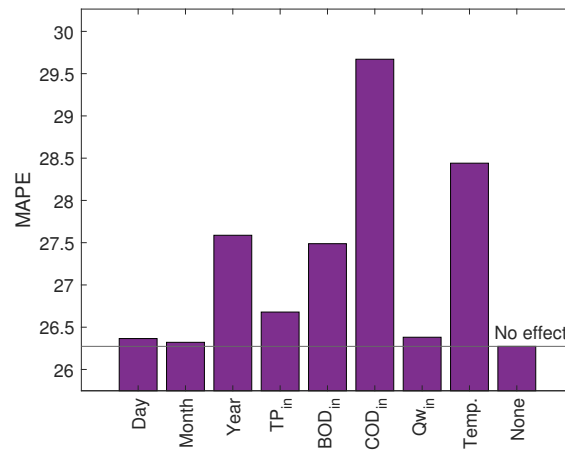
6.3. MLR EFFLUENT MODELS SENSITIVITY



(a) Sensitivity compared by correlation.

(b) Sensitivity compared by MSE.

Figure 6.9: COD_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.



(c) Sensitivity compared by MAPE.

Figure 6.9: COD_{out} sensitivity of dropping input parameters for the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).

6.3.2 Daily data set

For Figures 6.10a, 6.10b, and 6.10c TP_{in} , month, and $turb_{out}$ seems to be the most important parameters for TP_{out} . A slight negative effect is found from Q_{rw} and pH_{out} for MAPE. For Figures 6.11a, 6.11b, and 6.11c which only uses time period 6, Q_{rw} , TP_{in} , $Turb_{out}$, PIX_{add} , and $Temp$. all give consistently positive results for TP_{out} over all measurements. pH_{out} is shown to be slightly negative.

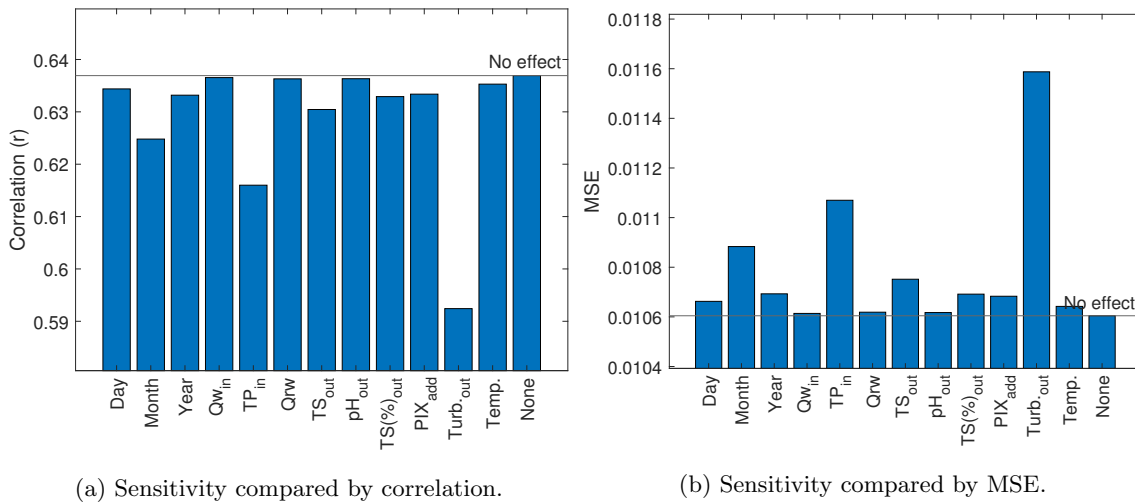
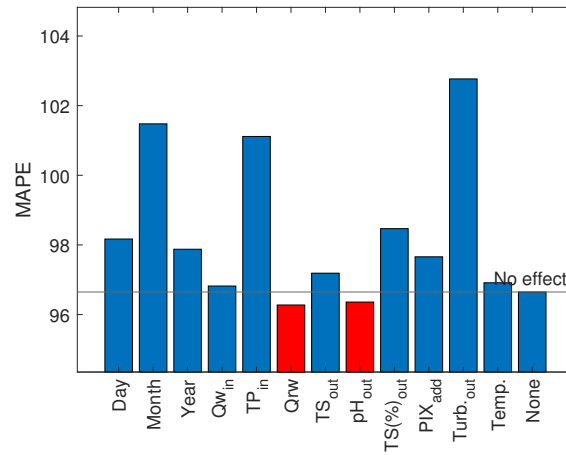


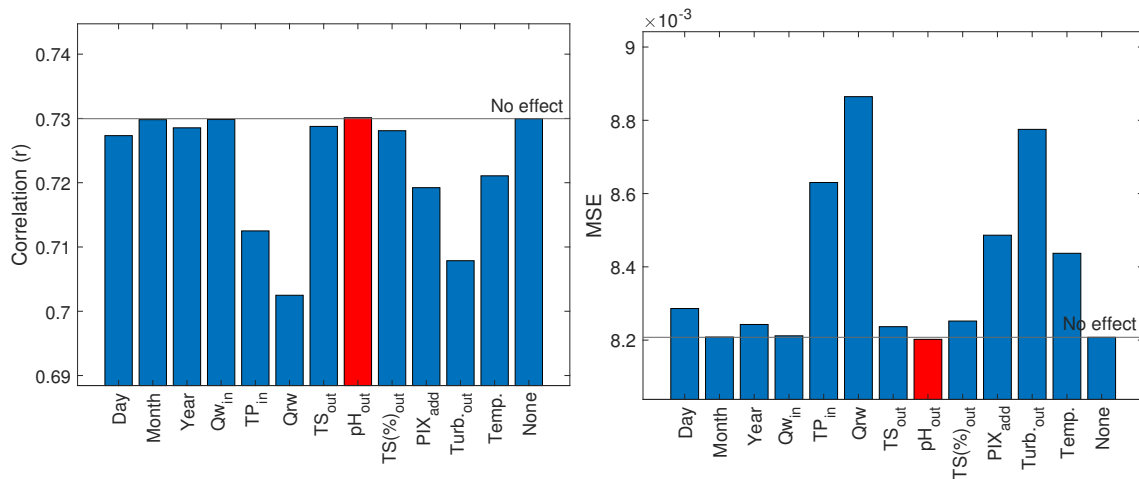
Figure 6.10: TP_{out} sensitivity of dropping input parameters for the daily data set using time periods 3-6, 02.01.17 - 12.03.19.

6.3. MLR EFFLUENT MODELS SENSITIVITY



(c) Sensitivity compared by MAPE.

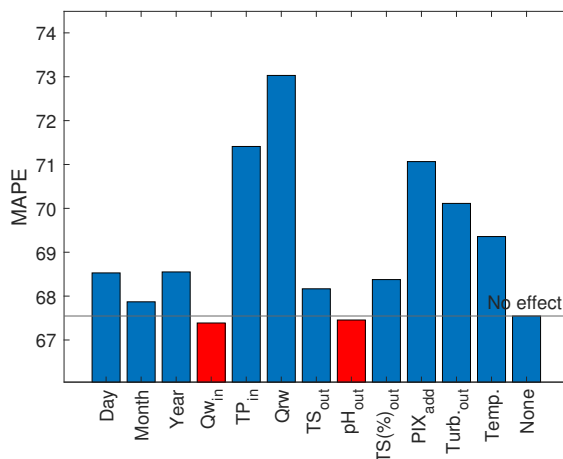
Figure 6.10: TP_{out} sensitivity of dropping input parameters for the daily data set using time periods 3-6, 02.01.17 - 12.03.19 (Cont.).



(a) Sensitivity compared by correlation.

(b) Sensitivity compared by MSE.

Figure 6.11: TP_{out} sensitivity of dropping input parameters for the daily data set using time period 6, 07.04.18 - 12.03.19.



(c) Sensitivity compared by MAPE.

Figure 6.11: TP_{out} sensitivity of dropping input parameters for the daily data set using time period 6, 07.04.18 - 12.03.19 (Cont.).

6.4 ANN model improvements

Several improvements were done to the base models described in Chapter 4.3. While many changes were performed without any noticeable effect on the result, a few changes stood out for all models. The following changes can be considered to be in an approximate order of most significant to least significant change: using specific time periods; increasing the layer size of layer 1; adding new input parameters ($Temp.$, date, and Qw_{in} for both data sets and Qrw for the daily data set); adding a second layer; changing the activation functions to ELU; increasing learning rate and adding learning rate decay; and changing the input normalising to be between 0.1 and 0.9. Additionally, some models were improved by the following changes: adding a weight decay; adding dropout between first, second and output layer; and changing the batch size. The final models that were used to produce the results shown in Chapter 6.5 are shown in Tables 6.3 and 6.4 for the bi-weekly and daily data set respectively. In these tables some abbreviations are used which are clarified in Table 6.2. These

abbreviations will not be used again and is just to more easily represent the neural network setup in this section.

Table 6.2: Abbreviations used in ANN model explanation

Abbreviation	Explanation	Abbreviation	Explanation
LS	Layer size	O	Optimiser function
NIn	Input normalisation	LR	Learning rate
AF	Activation function	LRD	Learning rate decay
WD	Weight decay (L2)	L	Loss function
E	Epochs	BS	Batch size
DO	Dropout between previous and current layer		

Table 6.3: Final ANN models for bi-weekly data set

Layer In	Layer 1	Layer 2	Layer Out	Optimiser	Other
<i>TP_{out}</i> , Time period 2, All inputs					
LS = 8 NIn = (0.1,0.9)	LS = 400, AF = ELU, WD = 0.001	LS = 60, AF = ELU,	LS = 1 AF = ELU,	O = Adam LR = 0.003 LRD = 0.002	L = MSE E = 400 BS = 8
<i>BOD_{out}</i> , Time period 2, All inputs					
LS = 8 NIn = (0.1,0.9)	LS = 1000, AF = ELU, WD = 0.1	LS = 200, AF = ELU, DO = 0.2	LS = 1 AF = ELU, DO = 0.2	O = Adam LR = 0.001 LRD = 0.0002	L = MSE E = 400 BS = 8
<i>COD_{out}</i> , Time period 2, Influent inputs					
LS = 8 NIn = (0.1,0.9)	LS = 1000, AF = ELU, WD = 0.001	LS = 200, AF = ELU,	LS = 1 AF = ELU,	O = Adam LR = 0.004 LRD = 0.002	L = MSE E = 400 BS = 8

Table 6.4: Final ANN models for daily data set

Layer In	Layer 1	Layer 2	Layer Out	Optimiser	Other
<i>TP_{out}</i> , Time period 6, Influent inputs					
LS = 8 NIn = (0.1,0.9)	LS = 800, AF = Linear,	LS = 30, AF = ELU,	LS = 1 AF = ELU,	O = Adam LR = 0.003 LRD = 0.003	L = MSE E = 400 BS = 14
<i>TP_{out}</i> , Time period 6, All inputs					
LS = 12 NIn = (0.1,0.9)	LS = 30, AF = Linear,	LS = 3, AF = ELU,	LS = 1 AF = ELU,	O = Adam LR = 0.003 LRD = 0.003	L = MSE E = 400 BS = 12
<i>TP_{out}</i> , Time period 3-6, Influent inputs					
LS = 8 NIn = (0.1,0.9)	LS = 800, AF = ELU,	LS = 30, AF = ELU,	LS = 1 AF = ELU,	O = Adam LR = 0.003 LRD = 0.005	L = MSE E = 400 BS = 12
<i>TP_{out}</i> , Time period 3-6, All inputs					
LS = 12 NIn = (0.1,0.9)	LS = 800, AF = Linear,	LS = 30, AF = ELU,	LS = 1 AF = ELU,	O = Adam LR = 0.003 LRD = 0.003	L = MSE E = 400 BS = 12

6.5 ANN effluent models

Figures 6.12a, 6.12b, and 6.12c show the results of the ANN model for the bi-weekly data set using time period 2. The correlations are rather low for TP_{out} with only 0.52 for test data, but much higher for BOD_{out} and COD_{out} at 0.82 and 0.76 respectively. For these models the day, month, year, TP_{in} , BOD_{in} , COD_{in} , Qw_{in} and $Temp.$ were used as inputs. The same inputs were used for models using time period 2-6 and 3-6 as shown in Table 6.5.

For the daily data set Figures 6.13a and 6.13b show the results of ANN models using

data from time periods 3-6. These models uses only influent parameters (day, month, year, Qw_{in} , TP_{in} , Qrw and PIX_{add}) and all parameters (month, year, Qw_{in} , TP_{in} , Qrw , TS_{out} , pH_{out} , $TS(\%)_{out}$, PIX_{add} and $Turb_{out}$) respectively. This is also the case for Figures 6.14a and 6.14b, but these models only uses data from time period 6. The results are better when using all inputs compared to using only influent inputs (with correlation increasing from 0.55 to 0.62 and 0.75 to 0.77 respectively), but improves more significantly when using only data from time period 6 (with correlation increasing from 0.55 to 0.75 and 0.62 to 0.77 respectively). In the models the correlation coefficient (r), and the mean squared error (MSE) are shown.

6.5.1 Bi-weekly data set models

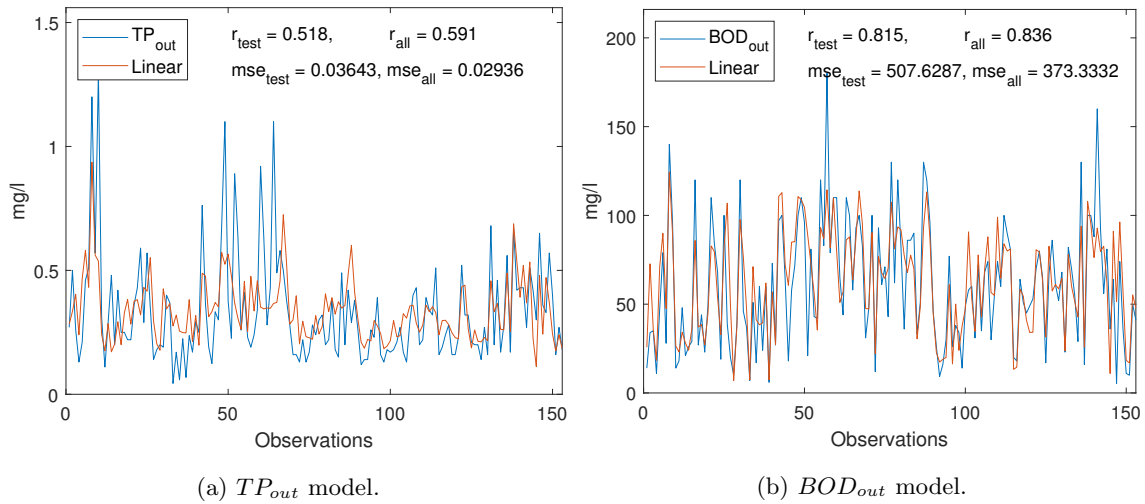
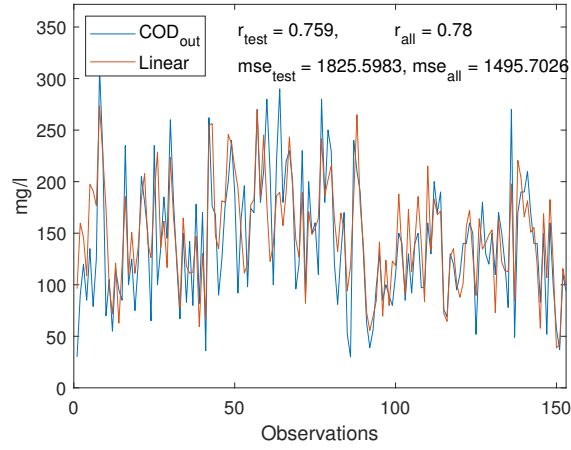


Figure 6.12: ANN models of the bi-weekly data set using time period 2, 04.04.06 - 13.08.15.

6.5. ANN EFFLUENT MODELS



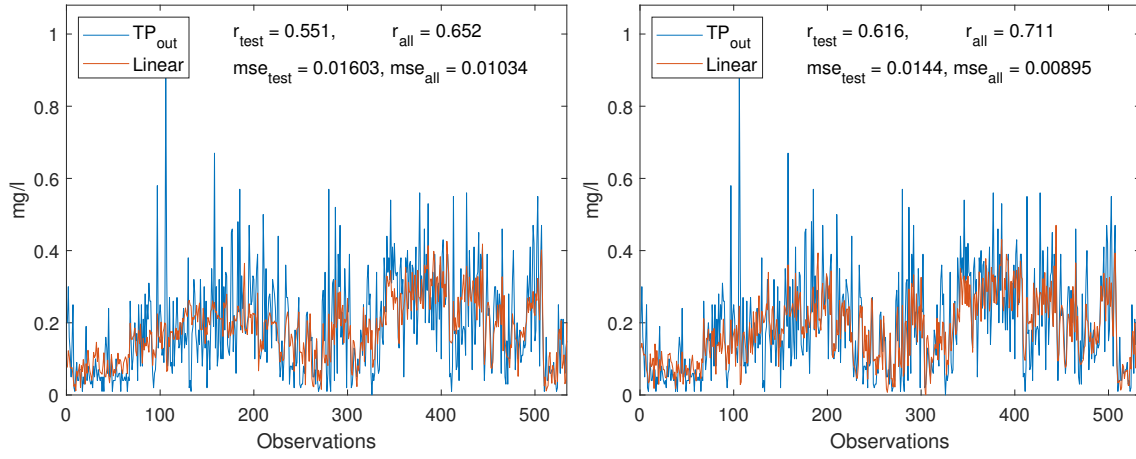
(c) COD_{out} model.

Figure 6.12: ANN models of the bi-weekly data set using time period 2, 04.04.06 - 13.08.15 (Cont.).

Table 6.5: Remaining ANN models of bi-weekly data set.

Output	Time period	Dates	r_{test}	MSE_{test}	r_{all}	MSE_{all}
TP_{out}	2-6	04.04.06 - 13.12.18	0.483	0.03315	0.485	0.0323
BOD_{out}	2-6	04.04.06 - 13.12.18	0.831	429.4	0.837	373.4
COD_{out}	2-6	04.04.06 - 13.12.18	0.761	1676	0.795	1392
TP_{out}	3-6	14.08.15 - 13.12.18	0.348	0.0315	0.524	0.02617
BOD_{out}	3-6	14.08.15 - 13.12.18	0.37	202.8	0.531	198.2
COD_{out}	3-6	14.08.15 - 13.12.18	0.21	808.9	0.581	709.2

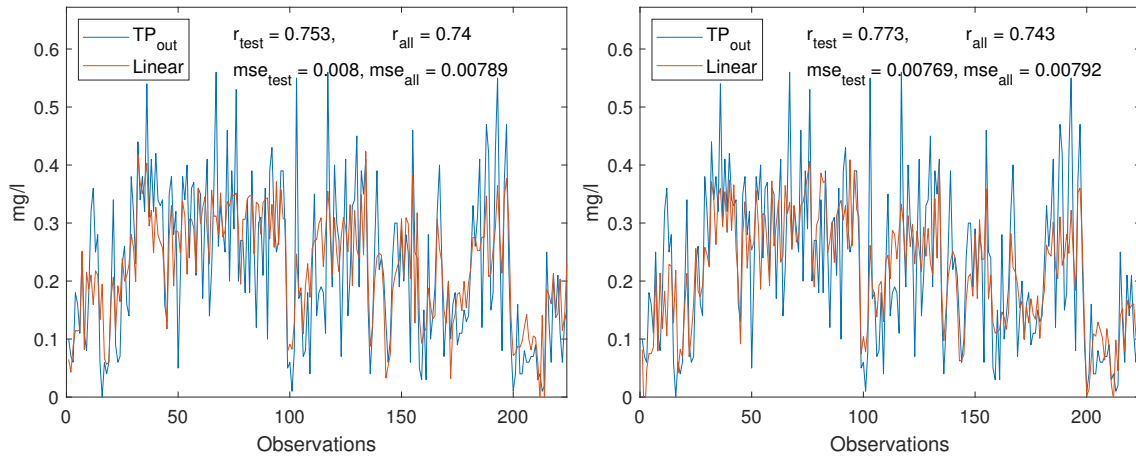
6.5.2 Daily data set models



(a) TP_{out} model with only influent inputs.

(b) TP_{out} model with all inputs.

Figure 6.13: ANN models for daily data set using time periods 3-6, 02.01.17 - 13.03.19.



(a) TP_{out} model with only influent inputs.

(b) TP_{out} model with all inputs.

Figure 6.14: ANN models for daily data set using time period 6, 07.04.18 - 12.03.19.

6.6 ANN effluent models sensitivity

To test which input parameters affected the results of the ANN models the most, all parameters were removed from the model one at a time and the results were compared. Figures 6.15a through 6.19c show how each of the parameters compares to the all data results. The graphs should be interpreted as the parameters that are furthest from the 'no effect' value (where no parameter is removed) are the ones that give the biggest effect on the result. A red bar colour has been given to parameters which give a positive effect when removed, while blue (daily data set) or purple (bi-weekly data set) is considered the opposite. In other words, parameters in a red colour should be removed from an optimal model.

6.6.1 Bi-weekly data set

For TP_{out} Figures 6.15a and 6.15b show that the year, TP_{in} , and Qw_{in} all give a positive effect on the result, and the same can be said for Figure 6.15c but here Qw_{in} gives the biggest positive effect. In Figures 6.16a through 6.16c BOD_{in} is by far the most significant parameter, with the date, TP_{in} , and $Temp.$ also giving a positive effect. Figures 6.17a through 6.17c show that COD_{out} is most reliant on the year, $Temp.$, COD_{in} , and BOD_{in} , while the month gives a slight negative effect.

6.6. ANN EFFLUENT MODELS SENSITIVITY

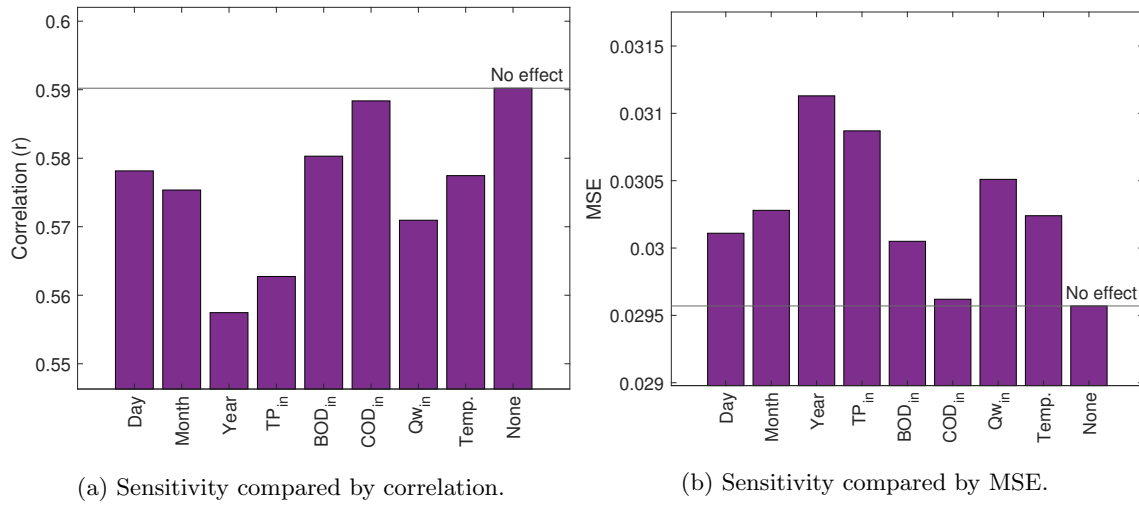


Figure 6.15: TP_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15.

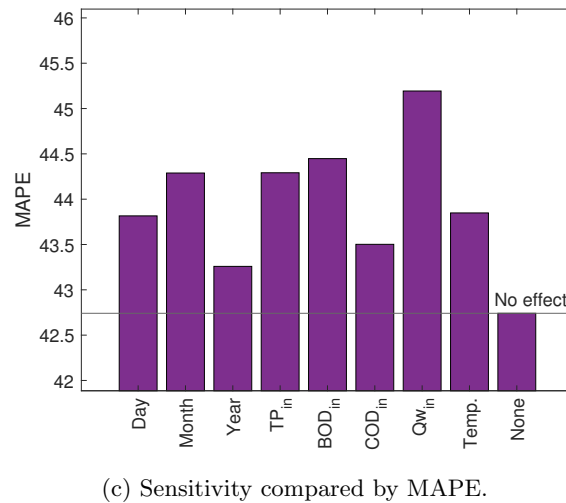
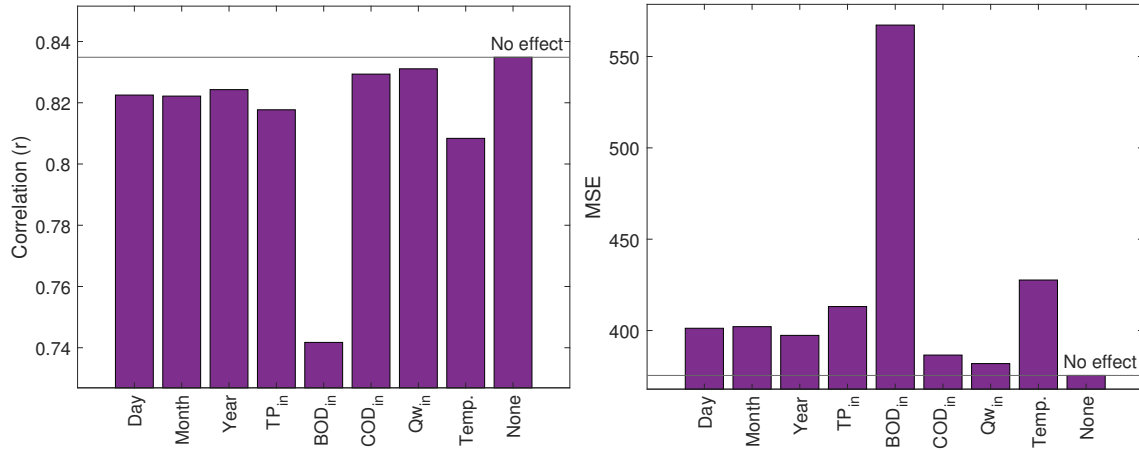


Figure 6.15: TP_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15 (Cont.).

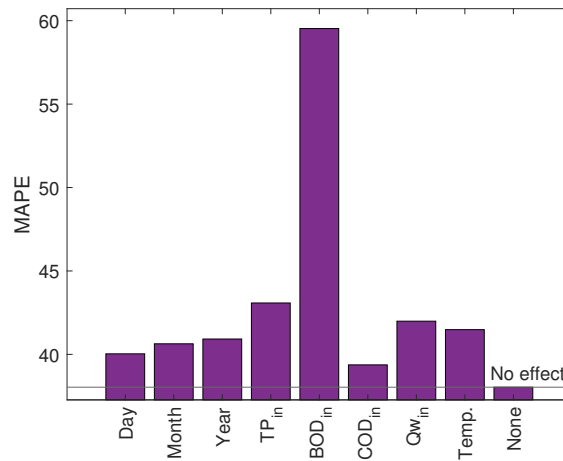
6.6. ANN EFFLUENT MODELS SENSITIVITY



(a) Sensitivity compared by correlation.

(b) Sensitivity compared by MSE.

Figure 6.16: BOD_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15.



(c) Sensitivity compared by MAPE.

Figure 6.16: BOD_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15 (Cont.).

6.6. ANN EFFLUENT MODELS SENSITIVITY

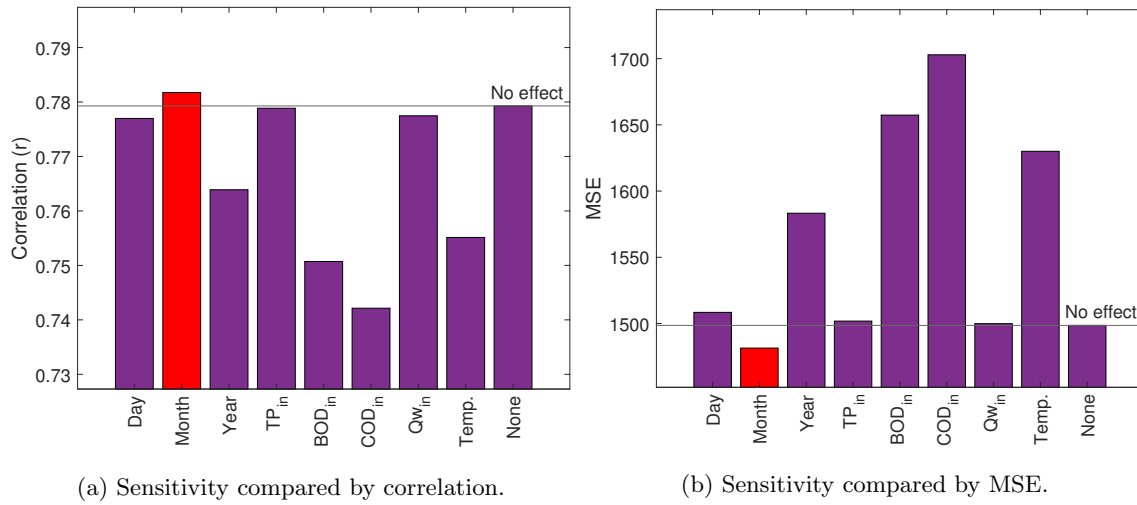


Figure 6.17: BOD_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15.

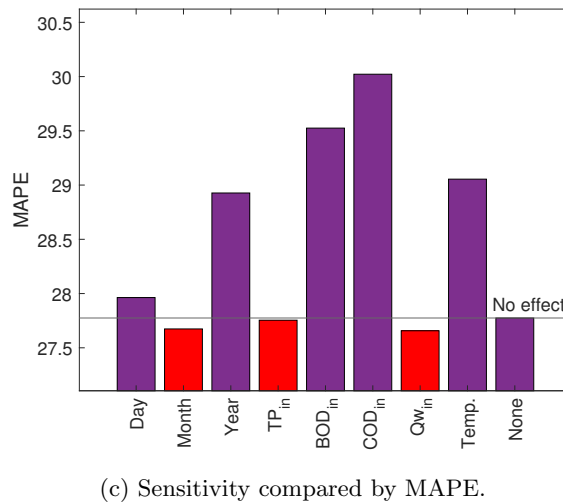


Figure 6.17: COD_{out} sensitivity of dropping input parameters for the daily data set using time period 2, 04.04.06 - 13.08.15 (Cont.).

6.6.2 Daily data set

By using time periods 3-6 for the daily data set, Figures 6.10a through 6.10c show that TP_{out} is reliant on most parameters available but mostly on the year, TP_{in} , Q_{rw} , $Turb_{out}$, and $Temp$. The day and $Q_{w_{in}}$ has a slight negative effect on MAPE. For Figures 6.11a through 6.11c, TP_{out} seems to be most reliant on TP_{in} , Q_{rw} , and $Turb_{out}$.

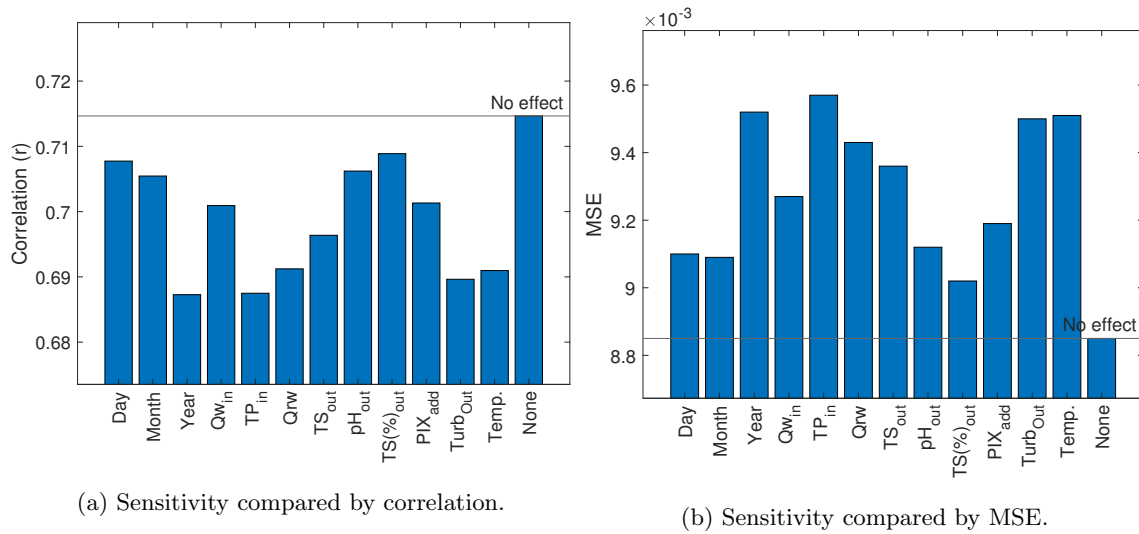
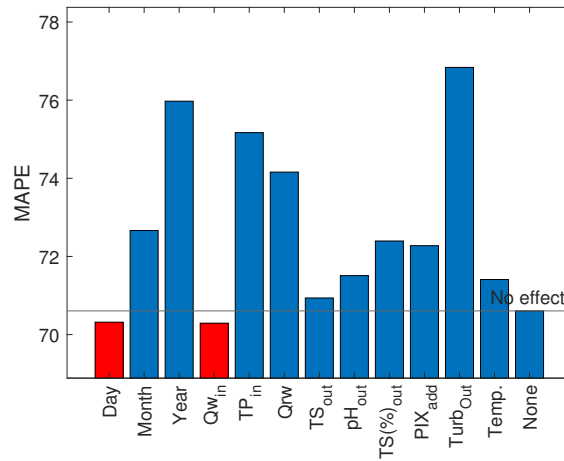


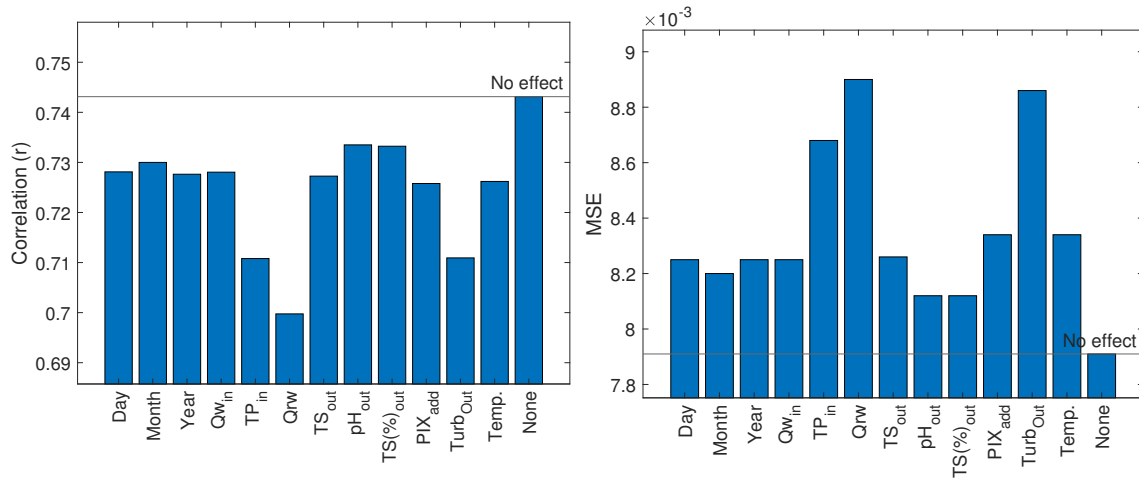
Figure 6.18: TP_{out} sensitivity of dropping input parameters for the daily data set using time periods 3-6, 02.01.17 - 12.03.19.

6.6. ANN EFFLUENT MODELS SENSITIVITY



(c) Sensitivity compared by MAPE.

Figure 6.18: TP_{out} sensitivity of dropping input parameters for the daily data set using time periods 3-6, 02.01.17 - 12.03.19 (Cont.).

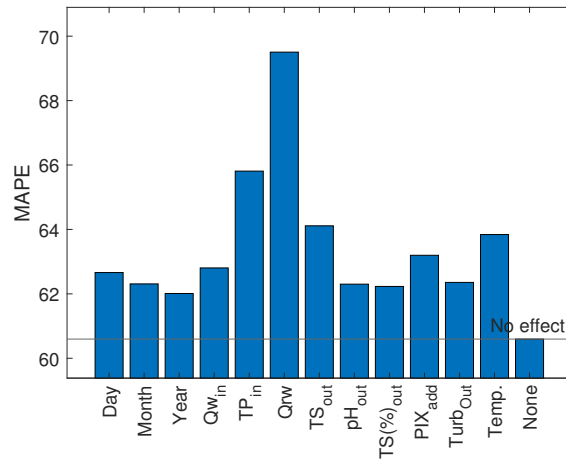


(a) Sensitivity compared by correlation.

(b) Sensitivity compared by MSE.

Figure 6.19: TP_{out} sensitivity of dropping input parameters for the daily data set using time period 6, 07.04.18 - 12.03.19.

6.6. ANN EFFLUENT MODELS SENSITIVITY



(c) Sensitivity compared by MAPE.

Figure 6.19: TP_{out} sensitivity of dropping input parameters for the daily data set using time period 6, 07.04.18 - 12.03.19 (Cont.).

Chapter 7

Discussions

This chapter includes discussions about the results of Chapter 5 and 6, and the methods that were used to accomplish these results.

7.1 Data analysis

The goal of the data analysis was to make a judgement on the available data sets. These data sets were not made to be used for data modelling and could therefore have been in the worst case either useless, inadequate or in a better case effective in describing the plant. Having only around 24 measurements per year for the bi-weekly data set was a concern as this could potentially give an inadequate explanation of the full year the data was meant to cover. The fact that the data stretched over almost 17 years could also mean that data from the beginning of the data set was not comparable to newer data. This problem was however not as prevalent for the daily data set which featured 535 measurements for 26 months in total. This meant that the data better described the different shifts that would occur through from day to day. The choice to remove data prior to when temperature and water discharge measurements begun was done because: 1) it was believed results would benefit from additional parameters and 2) the many changes in the plant would mean that data from this period would have a much lower explanatory power than newer data on the current plant state.

7.1.1 Trends of influent and effluent parameters

The results of the data analysis reflect the fact that the plant has been changing throughout the years. For the bi-weekly data set, this is seen in both BOD_{out} and COD_{out} as they have generally decreased to much lower values than from before

August 2015, as seen in Figures 5.1d and 5.1f. No obvious change in behaviour was detected in TP_{out} in neither the bi-weekly or daily data set (Figures 5.1b and 5.2i). TP_{out} does however seem to have some sort of cyclic behaviour which somewhat correlates to the temperature. The same can be seen for TP_{in} as seen in Figure 5.2b. For other effluent values like $TS(\%)_{out}$, $Turb_{out}$ and pH_{out} no obvious changes could be seen (Figures 5.2f, 5.2h, and 5.2e). The only other effluent parameter with noticeable changes was TS_{out} as seen in Figure 5.2d. The behaviour here is curious due to the linear change between time period 5 and 6 before the huge immediate increase. One possible fault for some of the data points in the daily data set could be how the data was recorded. Manual measurements, data readings and data recordings could easily lead to human errors. This was evident when going through the data set as many measurements had been misentered. Another possible fault could be some of the uncalibrated and possibly faulty sensors that were used to gather measurements. These faults are quite possibly part of the reasons for the behaviour of TS_{out} which would mean the data is not entirely reliable.

Most influent parameters were stable or continuously cyclic which is to be expected as changes in the plant would not affect the wastewater before it enters the plant. Some outliers in e.g. $TS(\%)_{out}$, pH_{out} and $Treat_{eff}$ could be due to, in addition to the previously described faults, certain machine failures etc. that was not noted in the data sets. Arguments could be made to exclude these observations as they were most likely due to events that were not in line with the ordinary operations of the plant. This could possibly contribute negatively in the MLR and ANN models. As no solid proofs for such a decision was found they were however included.

7.1.2 Time periods

The 6 time periods that were chosen seemed to represent the different changes in the data sets well. While it is curious why they do not line up well with the different changes in the plant as exact as would be expected, they still seem to represent different time periods of the plant. Due to lack of daily data from before 2017, it is possible that some change in the behaviour of other plant parameters could be the explanation of the decrease in BOD_{out} and COD_{out} . This is also the period where the bio reactor, Salsnes filter and centrifuges were installed which could all possibly have had a contribution to keeping BOD_{out} and COD_{out} values low. While the decrease happened about 7 months before the new machines were installed, it is not unlikely that they incidentally were installed at the same time as BOD_{out} and COD_{out} had some sort of cyclic low period. This could explain why the values started early and stayed low for the remaining 3.5 years. It could therefore be possible to change the beginning of time period 3 to 06.03.16 instead of 14.08.15.

It was also difficult to conclude some of the reasons for the last 3 time periods. Time period 4 was made because of a change in the PIX formula used in an attempt to increase the plant performance. Comparing TP_{out} values (as well as other effluent values) from before and after this change, it could seem like the change of formula has not had any impact. Due to a lack of daily measurements from before 2017, this can however not be concluded. Precipitation of iron begun at about the same time as TS_{out} began to decrease which was the start of time period 5. The big jump at 07.04.18 which begun time period 6 was more difficult to explain based on changes at the plant. As previously explained, the reasons for the behaviour of TS_{out} might be due to how it was recorded which would mean that the measurement could be unreliable.

7.1.3 Data correlations

The most notable findings with the data correlations was the general increase in correlations when using only newer data or only older data shown in Table 5.2 against Tables 5.1 and 5.3 for the bi-weekly data set and Table 5.4 against 5.5 of the daily data set. This could mean that older data has a low correlation with newer data, which strengthens the reason to split it into different time periods. Some correlations can also be seen to decrease significantly as well. This can be due to there not actually being a correlation between the values on a day to day basis but rather that they follow the general changes over the years similarly. Correlations between PIX_{add} , $TS(\%)_{out}$ and other values are heavily depended on their behaviour in time periods 3, 4 and 5, and therefore give more reliable results when only considering time period 6. This is also likely to be the reason there is a notable difference in model performance when using either only old, old and new, and only new data for MLR and ANN models. The decrease in correlation between BOD_{out} and COD_{out} , and influent parameters when going from using time period 2 to using any newer time period is most likely due to the same changes that decreased their values around 2015-2016.

The negative correlation between Qw_{in} and TP_{in} , BOD_{in} , and COD_{in} is likely since the amount of sewage is regarded as stable, while the increase in water may vary. More water is unlikely to be the causation of more pollutants, and it would instead decrease the amount of pollutants per litre. Note that this does not mean that the total amount of pollutants would decrease.

The increase in correlation between TP_{in} , BOD_{in} , and COD_{in} is strange as the plant should not have interfered with the wastewater before a sample is gathered for testing. A possibility would be a change in test procedures, which would not be surprising after 18 years of testing. It is also possible that some changes to sewage system or

plant inlet has slightly changed in such a way that the pollutants have become more correlated.

7.1.4 Available data parameters and differences between bi-weekly and daily data set

The available parameters were typical of what was used in existing literature, but as the data in this thesis was split into two data sets, each data set was partly lacking. Suspended solids, total nitrogen, pH, and COD can be seen as the most frequently used input parameters. In addition, total dissolved solids, BOD, TP, and total suspended solids were used by more than one paper. Out of these, the bi-weekly data set only contains COD, BOD, and TP as input parameters, while the daily data set only has pH and TP. While the parameters used in different papers does vary, and many of parameters used in this thesis has been used in one or more papers, it is evident that certain parameters with great explanatory power has not been included in this thesis. To improve the results of the models it would probably be beneficial to include measurements of suspended solids, total nitrogen and COD due to their performance in the existing literature. To make an effective model to control the additive PIX, it could also be beneficial to make measurements automated and time continuous. By measuring these additional parameters together with the ones already in use continuously over a longer time period, it is likely that an effective model could be generated to control parts of the plant processes.

A difference between the two data sets was the deviations between TP measurements. As this was a measurement of the same parameter, it should under perfect circumstances be identical. This is however not the case, and the comparison between them shows that the bi-weekly measurements were in general much higher. The histograms showing the deviations also showed that the bi-weekly data set gave higher values.

This is most likely due to the plant sensor being uncalibrated, that the tests were performed at different times, and that the tests could have been conducted differently. Note that with only 36 observations, the histograms would not be entirely accurate. Additionally, there was deviations between Qw_{in} measurements which could be due to what time of day the measurement was taken and how it was calculated. Different measurement timings could also have influenced the TP deviations.

7.2 MLR and ANN models in general

Based on the findings of the data analysis results chapter the data were modelled with the use of MLR and ANN. The models showed that the plant was possible to model accurately for certain parameters while others were not so successful. The choice of randomly splitting data into 70% training and 30% test data for the models gave the models more credibility. If all data had been used on both testing and training, it would not have been able to prove its ability to generalise for new data. The random split also meant that tests were not only performed on e.g. the newest data, which would only have represented a specific, small time period instead of getting test data from the whole time span of the data set. Testing on only new data would however give the most updated view of the plant, but with the small amount of data available, this would not be optimal.

Due to the random selection of test data, the choice of running the tests 1001 times to get a median result for the MLR models also helped generating a reliable result that would likely perform equally well on new data. Some of the outlier tests in this method gave results that were far above, or below, the median result, and they would not have been a good representation of the model. The median result would also most likely include a good representation of normal plant operation values and and some more extreme plant operation values, which would make it better at estimating

both cases. As the ANN models were based on a single random state, the test and train data were randomly chosen, but always chose the same random data every time. Implementing a method which made new random states for every iteration like with the MLR models would be beneficial but was more difficult for the ANN models as the whole model had to be re-initialised for every iteration. Some other problems with implementing this method would be: models with different test and train data could converge differently, and at times overfit for some models; the amount of time per iteration would decrease the speed of progress when working on the models; and the whole model would have to be fully re-initialised to not use weights, biases, etc. from previous models. While this would have been beneficial, it was not performed in this thesis. Some manual testing of different random states showed that results were acceptable.

The use of the date in this thesis gave some additional explanatory power as the changes to the plant and the seasonal changes could be more easily modelled. It could be possible that using the weekday instead of the day of the month would be beneficial as this could help model shifts during the week and weekends. With no data from weekends in the current data sets, this would however most likely not give any significant results currently. The date could also tell the model that the parameters available does not necessarily explain the whole process perfectly. When experiencing two equal input sets, it is possible that the output would not be equal due to some missing information. The date could in this case add some flexibility to the model.

7.3 Bi-weekly data set influent models

Influent models were made for BOD_{in} and COD_{in} with MLR which gave accurate estimations as shown in Figures 6.1b through 6.3b. The quality of each model increases by how much more newer data is used and how much less old data is used. The model

using time period 3-6 ends with a correlation of 0.86 and 0.91 for BOD_{in} and COD_{in} respectively which can be considered to be quite accurate. The differences between the different model accuracy's can be reflected in the correlations between each parameter in Tables 5.1 through 5.3. Due to the high correlation between BOD_{in} and COD_{in} , and TP_{in} , TP_{in} is by far the most significant parameter in the model, and this can be seen in the correlation tables as well. This high correlation leads to believe that a polynomial model could be made which would be purely dependant on TP_{in} . This model could potentially give even better results than the MLR models as the polynomial model could focus more on a potential non-linear relationship between the parameters. That the MLR models which only used TP_{in} as an input were similar in quality to the models using all input parameters strengthens this belief. Creating an ANN model of the influent values could also be beneficial but due to the high dependence on a single input value it is not necessarily that an ANN would be too different from a linear or polynomial model.

7.4 Bi-weekly data set models

The effluent MLR models for the bi-weekly data set gave a fairly high correlation between measured and estimated data for time period 2 for BOD_{out} and COD_{out} as seen in Figures 6.4b and 6.4c. It could however be seen that this correlation decreased when including time periods 3-6, and there was very low correlation when using only time periods 3-6 as seen in Table 6.1. This is most likely due to a larger change in the plant after 2015. It can be suspected that it is the same change that caused BOD_{out} and COD_{out} to decrease around 2015-2016. Due to not enough data for such a small time period, it is difficult to make a definite conclusion, but the most likely event is the installation of the bio-reactor, Salsnes filter and centrifuges, which all happened in between March and September of 2016. This probably changed the process in such a way that the available parameters were not sufficient in modelling

BOD_{out} and COD_{out} anymore. Due to these changes, the current models are not capable of representing the processes at the plant currently, but the results show that it has been possible. With the addition of new parameters, it is possible that a better model could be made. It is worth mentioning that due to the method of which the bi-weekly measurements are done, a model could not have been made to control the plant directly. For this a model which was only based on tests that were quickly completed (TP and somewhat COD) would have to be made. It would also have been crucial to measure these parameters every day.

For TP_{out} the bi-weekly data set seems to be incapable of generating sufficient results for both older, and newer data as seen in Figure 6.4a and Table 6.1. As the daily data set manages this better, this leads to believe that some of the parameters of the daily data set could have been used to help model the laboratory measurements of TP_{out} . However, with only 36 overlapping measurement dates, this tactic would not have been able to give reliable results yet. Given more time and measurements, this could be a possibility in the future.

The ANN models of the effluent values of the bi-weekly data set gave even higher correlation values than for the MLR models (0.42 to 0.52 for TP_{out} , 0.79 to 0.82 for BOD_{out} , and 0.75 to 0.76 for COD_{out}) as seen in Figures 6.4a through 6.4c. While the difference is not substantial, it does show that the ANN models give better results. As the ANN models could have been trained further, it is likely that even further improvements could be made. Table 6.5 show that the models of time period 3-6 are both better than their respective MLR models, but also better than the ANN models of time period 2 for BOD_{out} and COD_{out} . As the models for time period 3-6 are much worse, it would be expected that the newer data decreased the model quality. It is however possible that due to how few observations are used in time period 3-6 that these values did not impact the result significantly, and that the models for time

period 2-6 were better trained than the models for time period 2. Using more time on training these models could therefore give a more easily explained result. It is also possible that the random states used in sampling testing and training data were more beneficial for the models of time period 2-6.

7.4.1 Sensitivity analysis

The sensitivity analyses gave a good overview over which parameters were most efficient in modelling the plant as shown in Figures 6.7a through 6.9c and 6.15a through 6.17c. While it would have given a better overview by using test data for comparisons, the amount of test data was not enough to give consistent results. Two different samples of test data could give different results as to which parameters were most important. By using all data, it was always the same data that was analysed, and therefore gave more consistent results. It would have been possible to use every iteration used in the MLR models for the sensitivity analysis and consider the trend of which parameters were generally significant, but this was not done in this thesis. That there were generally few parameters which gave a negative effect means that the models are dependant on getting as much information as possible. Adding more parameters to the model would therefore be beneficial.

As the bi-weekly data set MLR model for TP_{out} had a correlation of only 0.42 for test data (Figure 6.4a), the sensitivity analysis of it might not be as reliable as for the other parameters. The parameters that did well in this test might not have performed well in a model which had better results. Something that supports this claim is that the month is the most important parameter for the result. While the month could give information about seasonal shifts, it does not give information about the plant processes. That the year is also among the most important parameters is also suspect as this mostly gives information about larger shifts throughout the years,

e.g. changes to the plant. The sensitivity analysis using the ANN model in Figure 6.12a gives somewhat similar results but relies less on the month and more on the day and year. As there should not be any significant change based on which day of the month it was, it is possible that this was just random noise that, due to the low amount of data and model correlation, coincidentally gave a positive effect.

For BOD_{out} and COD_{out} the analyses in Figures 6.8a through 6.9c for the MLR models give more reliable results which tell that they are mostly dependant on their own influent values but also each other's influent values. As they are closely related, this is to be suspected. The analyses for BOD_{out} and COD_{out} also give similar results for the ANN models (Figures 6.16a through 6.17c), which strengthens the theory that BOD_{in} and COD_{in} but also $Temp.$ are the most important parameters in modelling them. COD_{in} is seen to be less important for BOD_{out} for the ANN model compared to the MLR model with TP_{in} being more significant. This compares well to the correlation analysis in Table 5.1 and the existing literature. It must be noted that these sensitivity analyses may be inaccurate in describing the current plant state as they were based on time period 2.

7.5 Daily data set models

For the daily data set (Figures 6.5a through 6.6b and 6.13a through 6.14b), the models were better when predicting TP_{out} than the bi-weekly models, but were reliant on using only time period 6 to model TP_{out} well. It can be seen that the models are also somewhat reliant on using both influent and effluent values, mostly due to $Turb_{out}$ as was shown in the sensitivity analysis. While it is promising that the model based purely on influent parameters had a high correlation, the results does show that there is potential in a model that uses all input parameters. As all measurements used for in this thesis were measured at approximately the same time, it can be argued that

whether some measurement points were before or after PIX was added would not be decisive. It would however be interesting to explore a model which was based on time continuous measurements.

The increased correlation of the models when using only newer data can also be reflected in the correlation analysis of the data sets (Tables 5.4 and 5.5), where correlations between parameters were generally higher for time period 6. Using only time period 6 meant that only 11 months of data was used, giving the models less chance to learn special occurrences and getting a better overview over the plant. This amount of data should still be sufficient however, as most types of influent levels should be reflected. The newer data should also give a better representation of the current state of the plant.

By inspecting the graphs using only influent inputs against the ones using all inputs, it can be seen that the models with all inputs better model the noise of the measured values. This could mean that the influent only models are slightly underfitting the data and are not able to generalise as well. Yet it could also be a positive sign as the TP sensors were expected to have some errors and possibly added noise. Modelling a more average version of the noise could be beneficial. It could also be negative if the sensors are more correct than anticipated, which would mean the highest and lowest values that the plant would experience would not be modelled correctly.

The models with much higher all data correlation than test data show that the models are not generalising as well to new inputs. In the cases where the test results were higher, like with the daily data set models using time period 6 (Figures 6.14a and 6.14b), its possible that the random state that was chosen was conveniently good for the test data. The correlation values shown could therefore be slightly better than what the median result test used with the MLR models would have given.

7.5.1 Sensitivity analysis

Comparing the sensitivity of TP_{out} it can be seen that it is more dependant on just a few parameters with the MLR model as seen in Figures 6.10a through 6.11c, while it seems dependant on all parameters with the ANN model in Figures 6.18a through 6.19c. This may be due to how the MLR method finds its slope coefficients, as it sometimes lowers certain coefficients almost to zero. For the outliers, the two models are however fairly similar when considering time periods 3-6. TP_{in} and $Turb_{out}$ are for both models the most effective. $Turb_{out}$ can be an effective methods of determining the cleanliness of the water, which is why it most likely is so effective in modelling TP_{out} . It is also, together with TP_{in} , the parameter with the highest correlation with TP_{out} . As TP_{out} has not been modelled by the literature reviewed in this thesis, no comparison can be made here. For the ANN model, Q_{rw} , the year, and $Temp.$ were also significant in modelling TP_{out} . $Temp.$ has in general been successful in describing most effluent values which leads to believe that the processes are dependant on the weather and season. Since Q_{rw} also has a significant effect on the results of both models, it is possible that controlling this value with other methods than is currently used could be beneficial. Due to the changes in the plant over the 26 months the daily data set covers, the year also adds some effect. The MLR model using time period 6 is more like its equivalent ANN model with the effectiveness of the parameters being almost identical. As with the time period 3-6 model, TP_{in} and $Turb_{out}$ are crucial. The year is less important in this model, and this may be because the data only includes 11 months which have experienced no notable change in the plant. Besides this, these results are similar to the ANN model using time period 3-6 as Q_{rw} and $Temp.$ also give some positive effect. This leads to believe that the mentioned parameters have great explanatory power for TP_{out} .

7.6 ANN model improvements

The final models used in the thesis proved to be quite superior to the initial models proposed. The addition of new input parameters and dividing into different time periods gave some important structural improvements and possibly granted the largest increase in correlations. Due to the complexities of the data sets, it was also crucial to use larger layer sizes and even a second hidden layer for all models. These were the changes that were by far the most important which shows that understanding the parameters used in the models can be crucial. While using less data is normally a drawback for training and testing, as discussed earlier, the different changes to the plant made this necessary and also improved the result.

Using the Adam optimiser function proved to give generally better results than stochastic gradient descent. Together with increasing the layer sizes, increasing the learning rate helped speeding up the learning process and made convergence faster. This was also helped by using a learning rate decay which made sure the learning rate was more fitting for the current epoch. Both learning rate decay, weight decay, and dropout helped reducing overfitting when they were used.

It was difficult to determine if smaller changes actually gave a change to the result or if it was mostly based on the random nature of the training process. As a single random state was used in training, this did help reduce the noise and was important when learning how to create the optimal network. As mentioned earlier, having only a single random state for the results could make it appear better than it would be on new test data. It can therefore be expected that some correlations are slightly higher than they would have been using multiple random states.

The sigmoid and ReLU activation functions were primarily attempted as these are frequently used in ANN's. Both activation functions did however provide challenges

as they frequently produce zero-values and near zero-values for the estimations. This can be explained by the presence of the vanishing gradient problem and the bias shift problem respectively. The change of activation function to the ELU function eliminated these problems and provided for a much simpler learning process. The same argumentation can be used for the linear function which is similar to ELU but is linear also below 0. The linear function in sparse cases produced slightly better results than ELU.

Chapter 8

Conclusions

The wastewater treatment process at Saulekilen has in this thesis been modelled by the use of multiple linear regression (MLR) and artificial neural networks (ANN's). The MLR and ANN predicted values of influent and effluent measurements are promising. The ANN models prove to be superior to the MLR models in prediction of total phosphorous (TP), biological oxygen demand (BOD) and chemical oxygen demand (COD) for effluent wastewater. For influent BOD and COD measurements, only MLR models were made.

Two data sets have been used to model the wastewater treatment process at Saulekilen. The two data sets are measurements acquired bi-weekly over a total period of 17 years and daily over a total period of 26 months. Due to the changes to the wastewater treatment processes over these years, the data sets have been split into 6 time periods of different lengths in order to obtain the best possible models.

The most notable results are the predictions of effluent TP using the daily data set and the data from time period 6. The best results of the models with regards to correlation were obtained for influent BOD and COD, especially for time periods 3-6. The models gave good correlations for effluent BOD and COD for time period 2. Sensitivity analyses of the model outputs were performed. These showed that the models were in most cases improved by including all the available input parameters. Some parameters proved to be more effective in modelling the output than others. These parameters were: reject water, turbidity and influent TP for effluent TP; influent BOD for effluent BOD; influent BOD, COD and temperature for effluent COD; and influent TP for both influent BOD and COD. The choice of ANN architecture has also shown to be crucial to generate the best results.

Bibliography

- [1] Eldon Rene and M B Saidutta. “Prediction of bod and cod of a refinery wastewater using multilayer artificial neural networks”. In: *Journal of Urban and Environmental Engineering (JUEE)*; Vol 2, No 1 (2008) (Nov. 2013). DOI: 10.4090/juee.2008.v2n1.001007.
- [2] Archis Ambulkar and Jerry A. Nathanson. *Wastewater treatment*. [Online; accessed 16-May-2019]. 2018. URL: <https://www.britannica.com/technology/wastewater-treatment/Primary-treatment>.
- [3] New York City Environmental Protection. *New York City’s Wastewater Treatment System*. [Online; accessed 03-April-2019]. URL: <https://www1.nyc.gov/html/dep/html/wastewater/wssystem-process.shtml>.
- [4] Miljøstatus. *Kommunalt avløp*. [Online; accessed 04-April-2019]. 2017. URL: <https://www.miljostatus.no/tema/ferskvann/kommunalt-avlop/>.
- [5] Emrah Dogan et al. “Application of Artificial Neural Networks to Estimate Wastewater Treatment Plant Inlet Biochemical Oxygen Demand”. In: *environmental progress* 27 (Dec. 2008), pp. 439–446. DOI: 10.1002/ep.10295.
- [6] Miklas Scholz. “Chapter 4 - Organic effluent”. In: *Wetland Systems to Control Urban Runoff*. Ed. by Miklas Scholz. Amsterdam: Elsevier, 2006, pp. 15–18. ISBN: 978-0-444-52734-9. DOI: <https://doi.org/10.1016/B978-044452734-9/50007-4>. URL: <http://www.sciencedirect.com/science/article/pii/B9780444527349500074>.
- [7] Martin B. Hocking. “4 - Water Quality Measurement”. In: *Handbook of Chemical Technology and Pollution Control (Third Edition)*. Ed. by Martin B. Hocking. Third Edition. San Diego: Academic Press, 2005, pp. 105–138. ISBN: 978-0-12-088796-5. DOI: <https://doi.org/10.1016/B978-012088796-5/50007-7>. URL: <http://www.sciencedirect.com/science/article/pii/B9780120887965500077>.
- [8] B. Ráduly et al. “Artificial neural networks for rapid WWTP performance evaluation: Methodology and case study”. In: *Environmental Modelling & Software* 22.8 (2007). Bayesian networks in water resource modelling and management, pp. 1208–1216. ISSN: 1364-8152. DOI: <https://doi.org/10.1016/j.envsoft.2006.07.003>. URL: <http://www.sciencedirect.com/science/article/pii/S1364815206001939>.

- [9] Sani Isa Abba and Gozen Elkiran. “Effluent prediction of chemical oxygen demand from the astewater treatment plant using artificial neural network application”. In: *Procedia Computer Science* 120 (2017). 9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW 2017, 22-23 August 2017, Budapest, Hungary, pp. 156–163. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2017.11.223>. URL: <http://www.sciencedirect.com/science/article/pii/S1877050917324353>.
- [10] L. Hertel et al. “Deep convolutional neural networks as generic feature extractors”. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. July 2015, pp. 1–4. DOI: 10.1109/IJCNN.2015.7280683.
- [11] Michael Nielsen. “Neural networks and deep learning”. In: vol. 1. [Online; accessed 21-January-2019]. Oct. 2018. URL: <http://neuralnetworksanddeeplearning.com/index.html>.
- [12] Marek Rei. *Thoughts on Machine Learning and Natural Language Processing*. [Online; accessed 15-February-2019]. 2014. URL: <http://www.marekrei.com/blog/neural-networks-part-1-background/>.
- [13] Andrey Kurenkov. *A 'Brief' History of Neural Nets and Deep Learning*. [Online; accessed 15-February-2019]. 2015. URL: <http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/>.
- [14] Caroline Clabaugh, Dave Myszewski, and Jimmy Pang. *Neural Networks: History: The 1940's to the 1970's*. [Online; accessed 15-February-2019]. 2000. URL: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html>.
- [15] Oktay Özkan, Ozgur Ozdemir, and Sukru Taner Azgin. “Prediction of Biochemical Oxygen Demand in a Wastewater Treatment Plant by Artificial Neural Networks”. In: *Asian Journal of Chemistry* 21 (June 2009), pp. 4821–4830.
- [16] Mahmoud S. Nasr et al. “Application of Artificial Neural Network (ANN) for the prediction of EL-AGAMY wastewater treatment plant performance-EGYPT”. In: *Alexandria Engineering Journal* 51.1 (2012), pp. 37–43. ISSN: 1110-0168. DOI: <https://doi.org/10.1016/j.aej.2012.07.005>. URL: <http://www.sciencedirect.com/science/article/pii/S1110016812000518>.
- [17] Ruicheng Zhang and Xulei Hu. “Effluent Quality Prediction of Wastewater Treatment System Based on Small-world ANN”. In: *Journal of Computers* 7 (Sept. 2012). DOI: 10.4304/jcp.7.9.2136-2143.

BIBLIOGRAPHY

- [18] Hossam Zaqoot and Mazen Hamada. “Application of artificial neural networks for the prediction of Gaza wastewater treatment plant performance-Gaza strip”. In: *Journal of Applied Research in Water and Wastewater* 5 (July 2018). DOI: 10.22126/arww.2018.874.
- [19] HydroTech. *How do the wastewater treatment plants (WWTPs) work?* [Online; accessed 03-April-2019]. 2017. URL: <https://www.hydrotech-group.com/blog/how-do-the-wastewater-treatment-plants-wwtps-work>.
- [20] Wikipedia contributors. *Sewage treatment — Wikipedia, The Free Encyclopedia*. [Online; accessed 16-May-2019]. 2019. URL: https://en.wikipedia.org/w/index.php?title=Sewage_treatment&oldid=896901563.
- [21] The Editors of Encyclopaedia Britannica. *Biochemical oxygen demand*. [Online; accessed 16-May-2019]. 2011. URL: <https://www.britannica.com/science/biochemical-oxygen-demand>.
- [22] Inc. Woodard & Curran. “5 - Waste Characterization”. In: *Industrial Waste Treatment Handbook (Second Edition)*. Ed. by Inc. Woodard & Curran. Second Edition. Burlington: Butterworth-Heinemann, 2006, pp. 83–126. ISBN: 978-0-7506-7963-3. DOI: <https://doi.org/10.1016/B978-075067963-3/50007-2>. URL: <http://www.sciencedirect.com/science/article/pii/B9780750679633500072>.
- [23] Miljødirektoratet. *Årlig rapportering for avløp*. [Online; accessed 04-April-2019]. 2019. URL: https://www.miljodirektoratet.no/verktoy/skjema/arlig-rapportering-for-avlop/?_t_id=1B2M2Y8AsgTpgAmY7PhCfg%5C%3d%5C%3d&_t_q=vann+og+avl%5C%c3%5C%b8p&_t_tags=language%5C%3ano%5C%2csiteid%5C%3a3fe17408-2bcc-45f8-86af-9820a42b8e53%5C%2candquerymatch&_t_ip=158.37.216.19&_t_hit.id=Miljodirektoratet_Web_Features_Landing_Pages_LandingPage/_eafabc34-f19e-458d-a5b3-4fab24a27e4c_no&_t_hit.pos=4.
- [24] Lorns Bjerkan. *Disse grafene viser hvor ekstrem Oslo-sommeren har vært*. [Online; accessed 27-March-2019]. 2018. URL: <https://www.aftenposten.no/osloby/i/L003Q1/Disse-grafene-viser-hvor-ekstrem-Oslo-sommeren-har-vart>.
- [25] David R. Anderson, Dennis J. Sweeney and Others. *Statistics*. [Online; accessed 22-May-2019]. 2019. URL: <https://www.britannica.com/science/statistics/Experimental-design#ref60713>.

BIBLIOGRAPHY

- [26] Will Kenton. *Multiple Linear Regression – MLR Definition*. [Online; accessed 02-April-2019]. 2019. URL: <https://www.investopedia.com/terms/m/mlr.asp>.
- [27] Michelle Lacey. *Multiple Linear Regression*. [Online; accessed 02-April-2019]. 1997. URL: <http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm>.
- [28] MATLAB. *version 9.6 (R2019a)*. [Online; accessed 22-May-2019]. Natick, Massachusetts: The MathWorks Inc., 2019. URL: <https://se.mathworks.com/help/matlab/ref/polyfit.html>.
- [29] MATLAB. *version 9.6 (R2019a)*. [Online; accessed 22-May-2019]. Natick, Massachusetts: The MathWorks Inc., 2019. URL: https://se.mathworks.com/help/stats/corr.html#mw_1b19e0d5-7906-4577-a0a5-b20311da7faf.
- [30] Frank Rosenblatt. *The Perceptron, a Perceiving and Recognizing Automaton*. 1957.
- [31] Nitish Shirish Keskar and Richard Socher. “Improving Generalization Performance by Switching from Adam to SGD”. In: *CoRR* abs/1712.07628 (2017). arXiv: 1712.07628. URL: <http://arxiv.org/abs/1712.07628>.
- [32] *heuristic*. URL: <https://dictionary.cambridge.org/dictionary/english/heuristic>.
- [33] G. Cybenko. “Approximation by Superpositions of a Sigmoidal Function”. In: vol. 2. Feb. 1989, 303–314 vol.2.
- [34] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)”. In: *arXiv e-prints*, arXiv:1511.07289 (Nov. 2015), arXiv:1511.07289. arXiv: 1511.07289 [cs.LG].
- [35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- [36] The MathWorks Inc. *Neural Network Toolbox User’s Guide*. [Online; accessed 14-May-2019]. 2004. URL: http://cda.psych.uiuc.edu/matlab_pdf/nnet.pdf.
- [37] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: <http://arxiv.org/abs/1502.03167>.

BIBLIOGRAPHY

- [38] Dominic Masters and Carlo Luschi. “Revisiting Small Batch Training for Deep Neural Networks”. In: *CoRR* abs/1804.07612 (2018). arXiv: 1804.07612. URL: <http://arxiv.org/abs/1804.07612>.
- [39] Lawrence K. Wang et al. “Chemical Precipitation”. In: *Physicochemical Treatment Processes*. Ed. by Lawrence K. Wang, Yung-Tse Hung, and Nazih K. Shamas. Totowa, NJ: Humana Press, 2005, pp. 141–197. ISBN: 978-1-59259-820-5. DOI: 10.1385/1-59259-820-x:141. URL: <https://doi.org/10.1385/1-59259-820-x:141>.

Appendix A

Code to produce model results

This appendix includes two examples of the codes used to generate the results of this thesis. The first code is the code to generate the MLR models made in Matlab. Results of the code are shown as they are generated. The results are in this case for predictions of influent BOD and COD. Note that predictions of effluent TP is also generated but this result is not used. The second code is the code to generate the ANN models made in python. This example code predicts effluent TP for the bi-weekly data set.

A.1 MLR code from Matlab

Table of Contents

.....	1
This code generates BODout, CODout and TPout predictions with bi-weekly data	1
For loop to run through code 1001 times	1
Find median result	3
Plot median result	4
Sensitivity of median result	7

```
close all;
clear;
```

This code generates BODout, CODout and TPout predictions with bi-weekly data

```
%Loading data set
a = load('C:\Users\magga\Documents\MAS500 - Masteroppgave
\SaulekilenDataBWTemp.mat');
%x = normalize(a.a,1,'range'); %Normalising data set
```

For loop to run through code 1001 times

```
for t=1:1001 %1001 chosen as a large, odd number
    x = a.a;
    %x(156:end,:) = []; %Removes newer data
    x(1:155,:) = []; %Removes older data
    %Randomly sorts data
    x = [x,randperm(length(x))'];
    x = sortrows(x,12);
    %Splits data into 70% training and 30% testing
    x_train = x(1:round(size(x,1)*0.7),:);
    x_test = x(round(size(x,1)*0.7)+1:end,:);

    %Names for figures and file names
    Name2 =
    ["Day", "Month", "Year", "TP_{in}", "BOD_{in}", "COD_{in}",...
    "Qw_{in}", "Temp.", "None"];
    Name = ["TP_{in}", "BOD_{in}", "COD_{in}"];
    %D1 M2 Y3 TPin4 TPout5 BODin6 BODout7 CODin8 CODout9 Qw10 Temp11
    if size(a.a,1) == size(x,1)
        testsize = "Full";
    elseif size(a.a,1)-120 > size(x,1)
        testsize = "Partial";
    else
        testsize = "Old";
    end

    %For loop for running through each output parameter
    kk=0;
```

```

for k=[4,6,8] %Chooses outputs [TPout,BODout,CODout]
    kk=kk+1;
    ll=0;
    %Saving correct parameter for plotting
    if kk == 1
        x_saveTP(:, :, t) = x(:, :);
    elseif kk == 2
        x_saveBOD(:, :, t) = x(:, :);
    else
        x_saveCOD(:, :, t) = x(:, :);
    end

    %For loop for running through all input parameters for
    %sensitivity analysis
    for l=[1,2,3,4,10,11,12]
        ll = ll+1;
        %Remain removes one parameter for every iteration
        remain = [1,2,3,4,10,11];
        if l<12
            remain(ll) = [];
        end
        %y is output parameter, split into testing and training
        y = x(:, k); %[4,5]=TP [6,7]=BOD [8,9]=COD
        y_train = y(1:round(size(y,1)*0.7), :);
        y_test = y(round(size(y,1)*0.7)+1:end, :);
        %xp chooses input parameters based on remaining inputs
        xp = [x(:, remain), ones(size(x,1),1)];
        xp_train = [x_train(:, remain), ones(size(x_train,1),1)];
        xp_test = [x_test(:, remain), ones(size(x_test,1),1)];
        L = size(xp,2)-1;
        %Regression of training data
        b = regress(y_train, xp_train);

        %Creates prediction formulas based on regression
        y_predictTest = 0;
        y_predictAll = 0;
        for i=1:L+1
            if i==L+1
                y_predictTest = y_predictTest + b(i);
                y_predictAll = y_predictAll + b(i);
            else
                y_predictTest = y_predictTest + b(i)*xp_test(:, i);
                y_predictAll = y_predictAll + b(i)*xp(:, i);
            end
        end
        for i=1:size(y_predictTest,1)
            if y_predictTest(i,1)<0
                y_predictTest(i,1) = 0;
            end
        end
        for i=1:size(y_predictAll,1)
            if y_predictAll(i,1)<0
                y_predictAll(i,1) = 0;
            end
        end
    end
end

```

```

end
pt = [y_test,y_predictTest];
pa = [y,y_predictAll];
%Correlation of measured vs. predicted
rt = corr(y_test,y_predictTest);
ra = corr(y,y_predictAll);
%MSE of measured vs. predicted
mset = immse(y_test,y_predictTest);
msea = immse(y,y_predictAll);

%Name for figures
FigName = ["TPinBW", "BODinBW", "CODinBW"];

%MAPE for measured vs. predicted
percent=0;
for i=1:size(x_test,1)
    percent = percent+abs(y_test(i)-y_predictTest(i))/
max(y_test(i),0.01);
end
MAPEt = percent*100/size(y_predictTest,1);
percenta=0;
for i=1:size(x,1)
    percenta = percenta+abs(y(i)-y_predictAll(i))/
max(y(i),0.01);
end
MAPEa = percenta*100/size(y_predictAll,1);

%saving values for later use
rts(kk,ll,t)=rt;
msets(kk,ll,t)=mset;
MAPETs(kk,ll,t)=MAPEt;
ras(kk,ll,t)=ra;
mseas(kk,ll,t)=msea;
MAPEas(kk,ll,t)=MAPEa;

ys(:,kk,t)=y;
ys_predictAll(:,kk,t)=y_predictAll;
ys_test(:,kk,t)=y_test;
ys_predictTest(:,kk,t)=y_predictTest;

end
end
end

```

Find median result

```

%Finds median result index based on test correlation
inxTP = ceil(find(rts==median(rts(1,end,:)))/3)/11; inxBOD = 1; %inxTP
= 1...
    %to make sure the following code works. Predictions of TP is
not...
    %important in this example code.
inxBOD = ceil(find(rts==median(rts(2,end,:)))/3)/11;

```

```

inxCOD = ceil(find(rts==median(rts(3,end,:)))/3)/11;
inx = [inxTP, inxBOD, inxCOD];

%Chooses the correct data based on index
ypTP = ys(:,1,inx(1));
ypBOD = ys(:,2,inx(2));
ypCOD = ys(:,3,inx(3));
yp_predictAllTP = ys_predictAll(:,1,inx(1));
yp_predictAllBOD = ys_predictAll(:,2,inx(2));
yp_predictAllCOD = ys_predictAll(:,3,inx(3));

%Sorts data based on time instead of being randomly sorted
x_saveOutTP = x_saveTP(:, :, inx(1));
ypTP = [ypTP, x_saveOutTP(:, 1:3)];
yp_predictAllTP = [yp_predictAllTP, x_saveOutTP(:, 1:3)];
ypTP = sortrows(ypTP, [4, 3, 2]);
yp_predictAllTP = sortrows(yp_predictAllTP, [4, 3, 2]);

x_saveOutBOD = x_saveBOD(:, :, inx(2));
ypBOD = [ypBOD, x_saveOutBOD(:, 1:3)];
yp_predictAllBOD = [yp_predictAllBOD, x_saveOutBOD(:, 1:3)];
ypBOD = sortrows(ypBOD, [4, 3, 2]);
yp_predictAllBOD = sortrows(yp_predictAllBOD, [4, 3, 2]);

x_saveOutCOD = x_saveCOD(:, :, inx(3));
ypCOD = [ypCOD, x_saveOutCOD(:, 1:3)];
yp_predictAllCOD = [yp_predictAllCOD, x_saveOutCOD(:, 1:3)];
ypCOD = sortrows(ypCOD, [4, 3, 2]);
yp_predictAllCOD = sortrows(yp_predictAllCOD, [4, 3, 2]);

```

Plot median result

```

%Loop for running through each output parameter
for i=1:3
    if i == 1
        pa = [ypTP(:,1), yp_predictAllTP(:,1)];
    elseif i == 2
        pa = [ypBOD(:,1), yp_predictAllBOD(:,1)];
    else
        pa = [ypCOD(:,1), yp_predictAllCOD(:,1)];
    end
    %Printing figure
    figure;
    plot(pa);

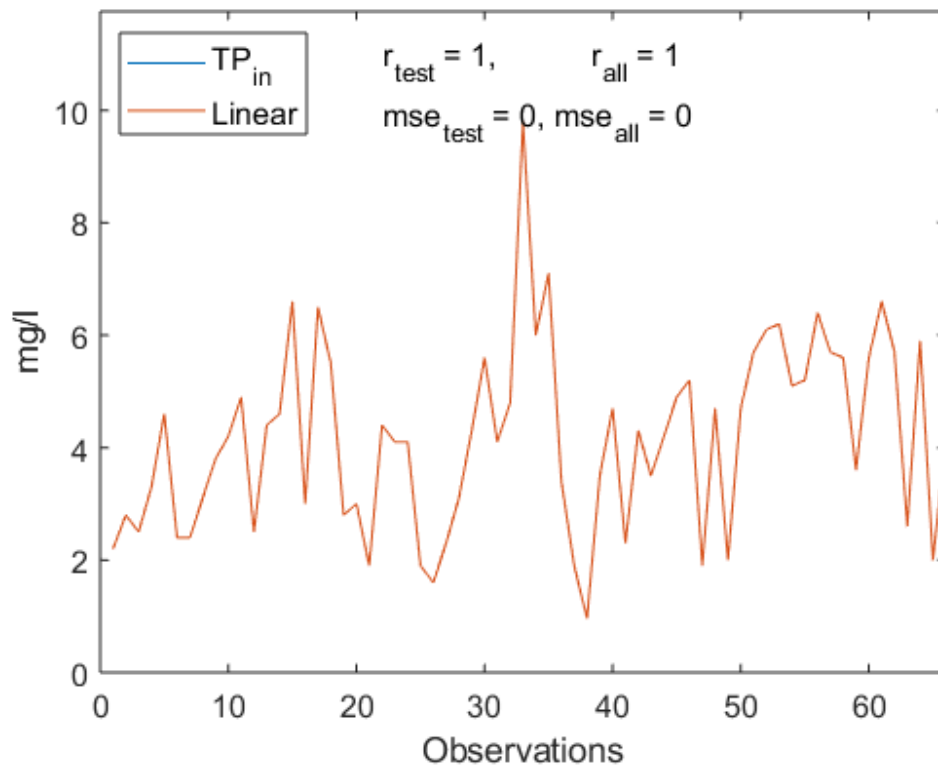
    legend(convertStringsToChars(Name(i)), 'Linear', 'Location', 'Northwest', 'FontSize',
        txt = ['r_{test} = ' num2str(round(rts(i,end,inx(i)),3)) ' ',
            r_{all} = ' num2str(round(ras(i,end,inx(i)),3))];
        txt2 = ['mse_{test} = ' num2str(round(msets(i,end,inx(i)),5)) ' ',
            mse_{all} = ' num2str(round(mseas(i,end,inx(i)),5))];
        text(size(pa,1)/3,max(pa(:,1))/0.9,txt, 'FontSize',12)
        text(size(pa,1)/3,max(pa(:,1))/1,txt2, 'FontSize',12)
        ylabel('mg/l', 'FontSize',12);

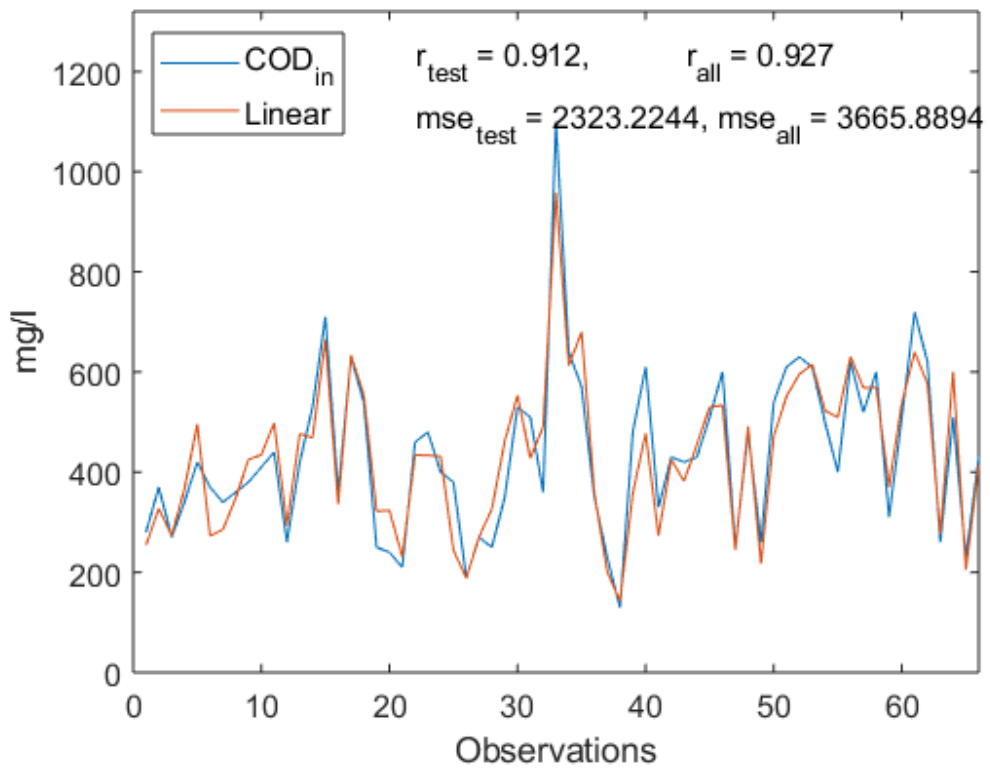
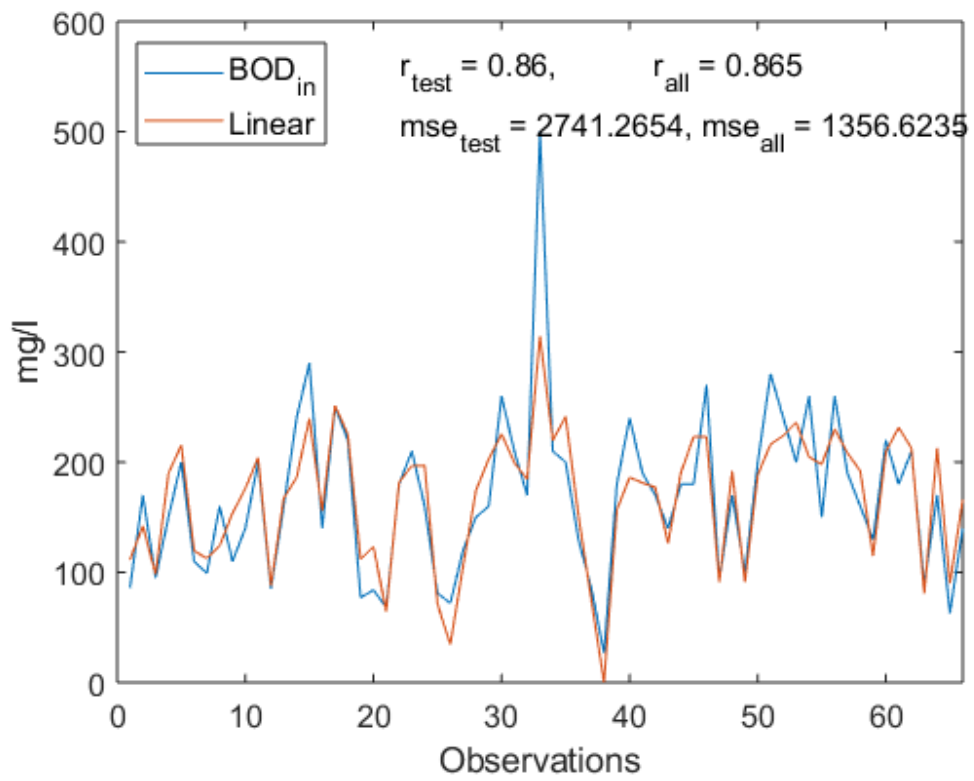
```

```

xlabel('Observations','FontSize',12);
xlim([0,size(x,1)]);
ylim([0,max(max(pa))*1.2]);
set(gca,'FontSize',12);
figpath = 'C:\Users\magga\Dropbox\Apps\Overleaf\GeneralKenobi
\Pictures\';
filename = convertStringsToChars(FigName(i)+testsize);
%print([figpath,filename], '-depsc');
end

```



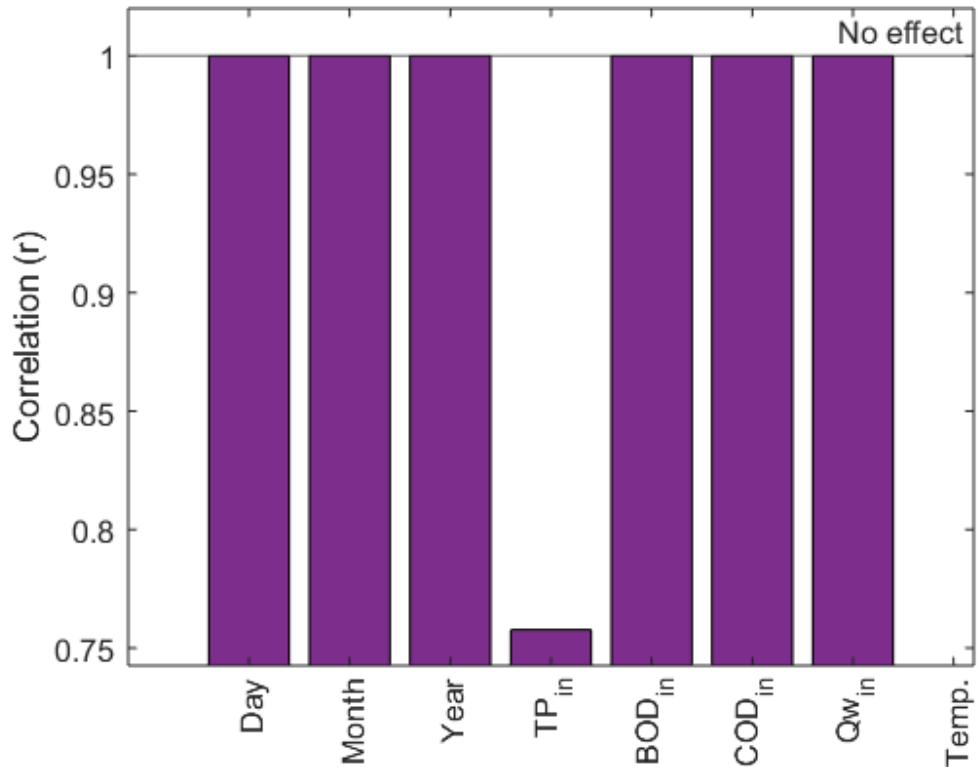


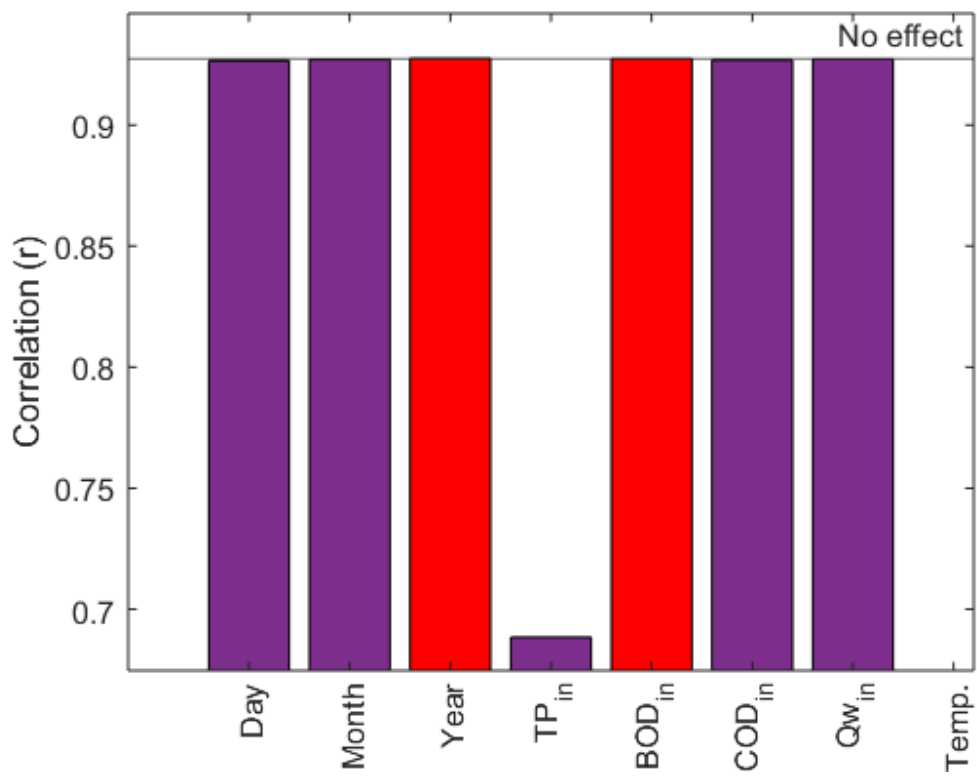
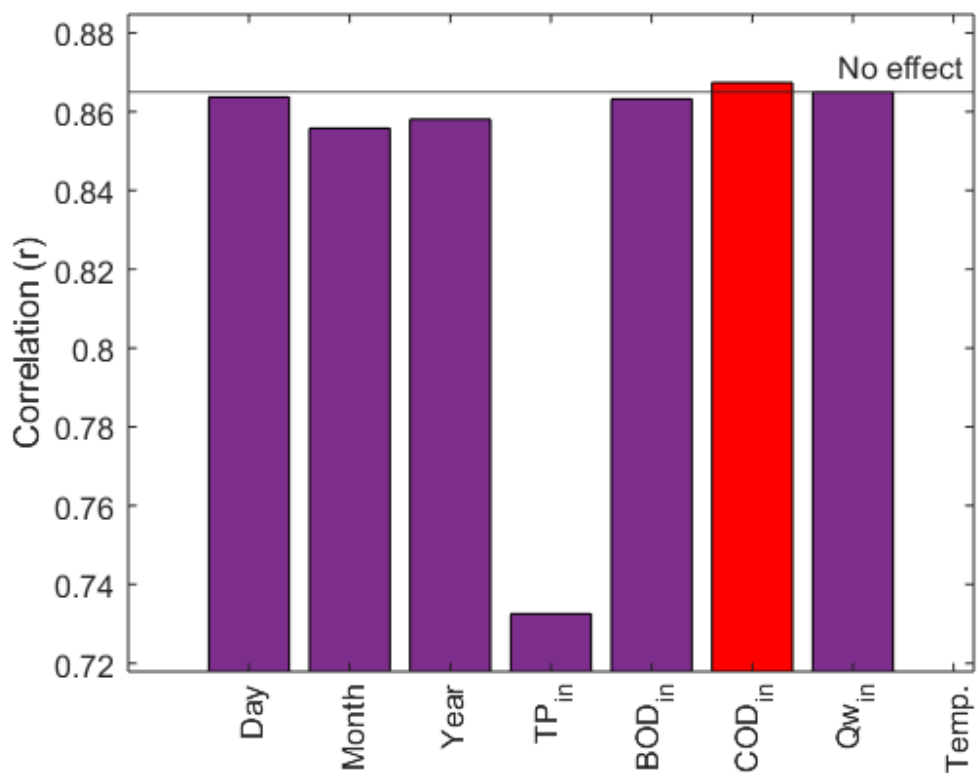
Sensitivity of median result

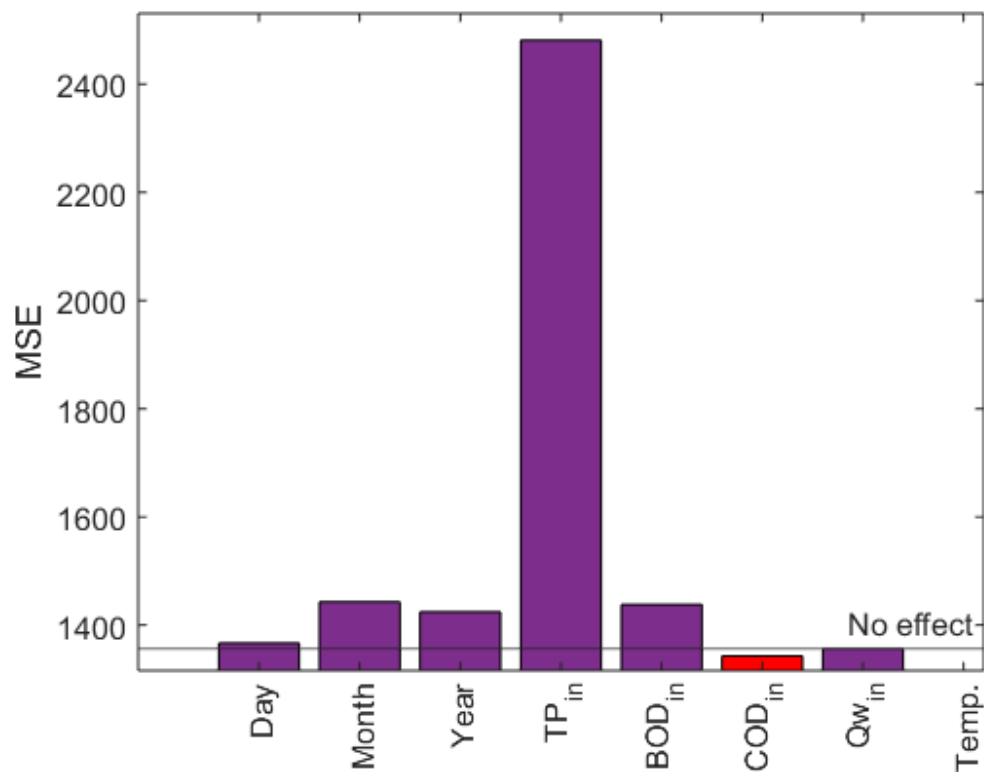
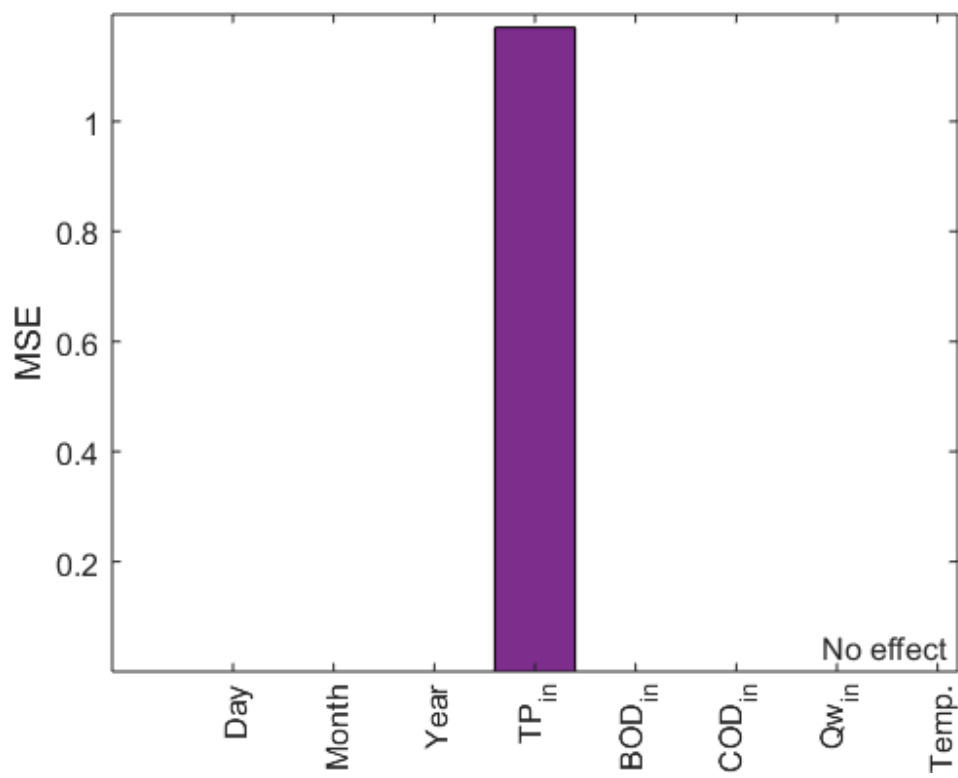
```
%Extract the correct scores
for i=1:3
    score(i,:,1) = rts(i,1:11,inx(i)); %[TP/BOD/COD, paramremoved,
    median]
    score(i,:,2) = msets(i,1:11,inx(i));
    score(i,:,3) = MAPEts(i,1:11,inx(i));
    score(i,:,4) = ras(i,1:11,inx(i));
    score(i,:,5) = mseas(i,1:11,inx(i));
    score(i,:,6) = MAPEas(i,1:11,inx(i));
end
%Figure and file names
param = ["TPin", "BODin", "CODin"];
ylabelName = ["Correlation (r)", "MSE", "MAPE"];
fig = ["rSensitivityBW", "MSESensitivityBW", "MAPEsensitivityBW"];

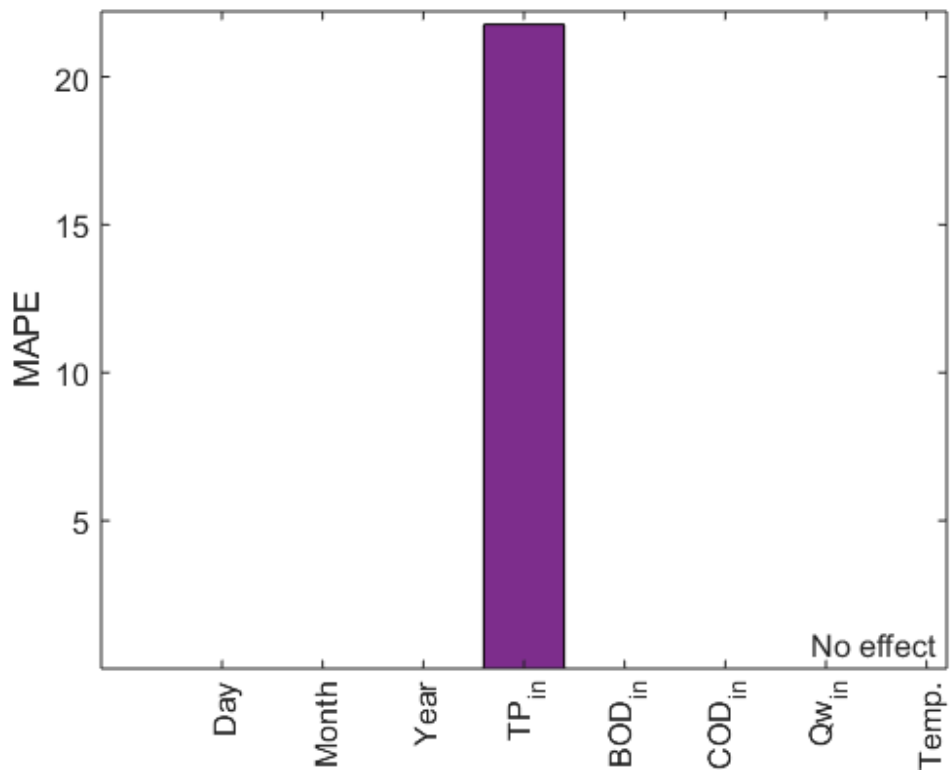
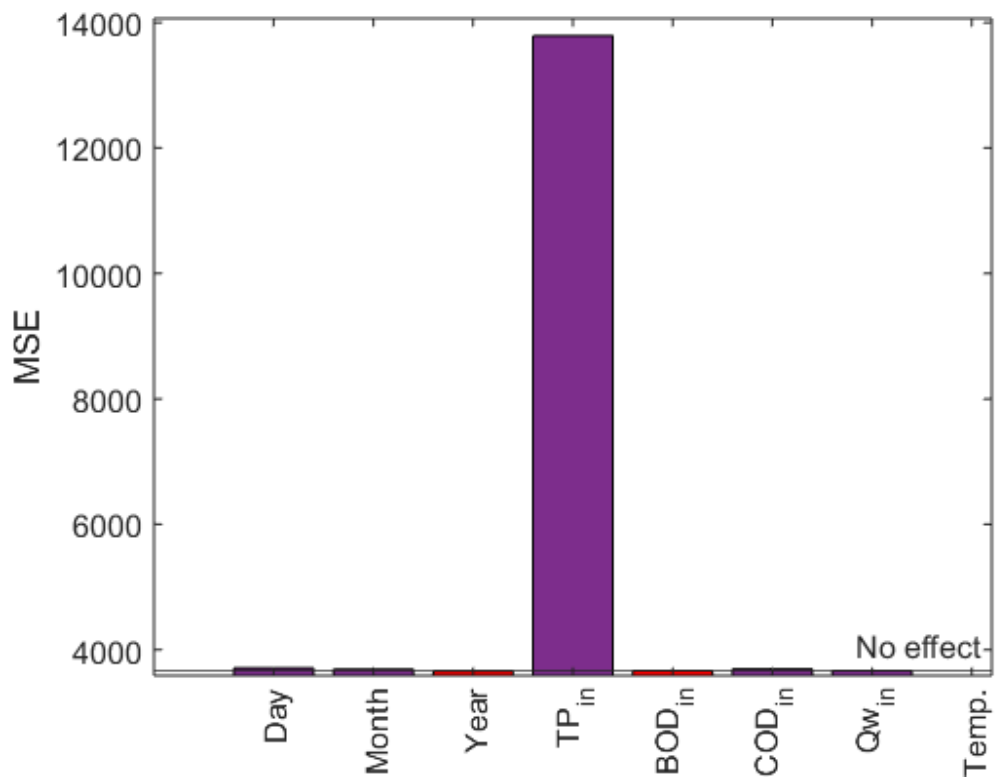
%For loops for running through results of "all data" and output
parameters
for i=4:6 %Teller for r/mse/mape
    for j=1:3 %Teller for Tp/cod/bod
        %Printing bar figures
        figure;
        b1 = bar(score(j,:,i));
        b1.FaceColor = 'flat';
        for k=1:size(score,2)
            if i==4
                if score(j,k,i) > score(j,end,i)
                    b1.CData(k,:) = [1, 0, 0];
                else
                    b1.CData(k,:) = [0.4940, 0.1840, 0.5560];
                end
            else
                if score(j,k,i) < score(j,end,i)
                    b1.CData(k,:) = [1, 0, 0];
                else
                    b1.CData(k,:) = [0.4940, 0.1840, 0.5560];
                end
            end
        end
        ylim([min(score(j,:,i))*0.98,max(score(j,:,i))*1.02])
        ylabel(convertStringsToChars(ylabelName(i-3)), 'FontSize',12)
        set(gca, 'XTick', 1:size(Name2,2), 'XTickLabel',
        Name2, 'FontSize',12);
        xtickangle(90)
        hold on;
        yline1 = yline((score(j,end,i)), '-', 'No
        effect', 'FontSize',12);
        yline1.LabelHorizontalAlignment = 'right';
        figpath = 'C:\Users\magga\Dropbox\Apps\Overleaf\GeneralKenobi
        \Pictures\';
        filename = param(j)+fig(i-3)+testsize;
```

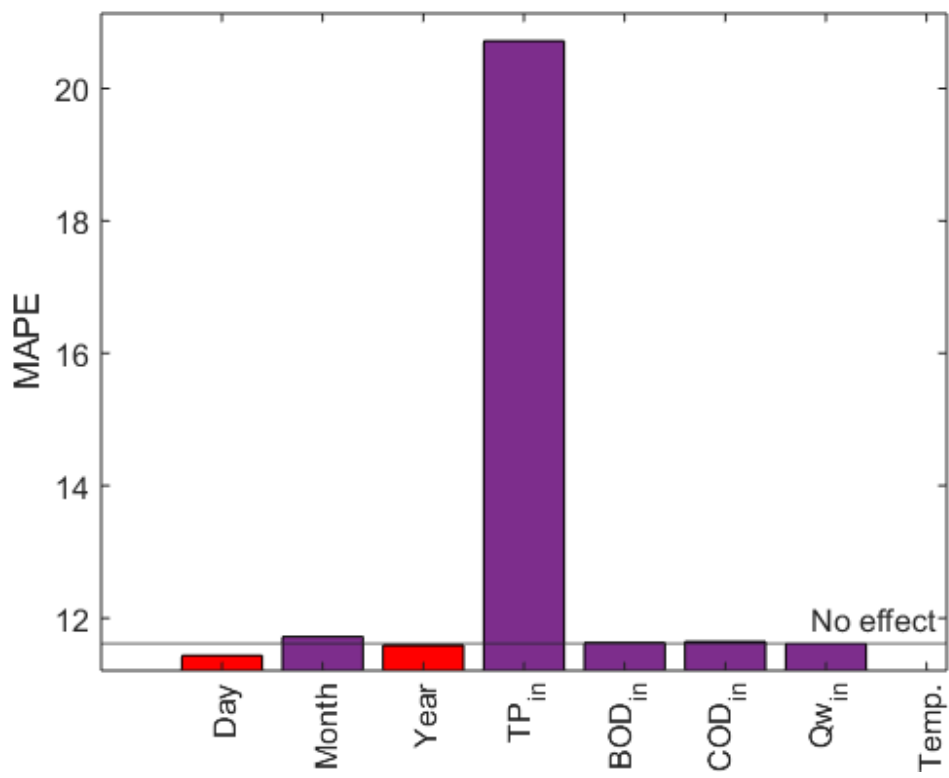
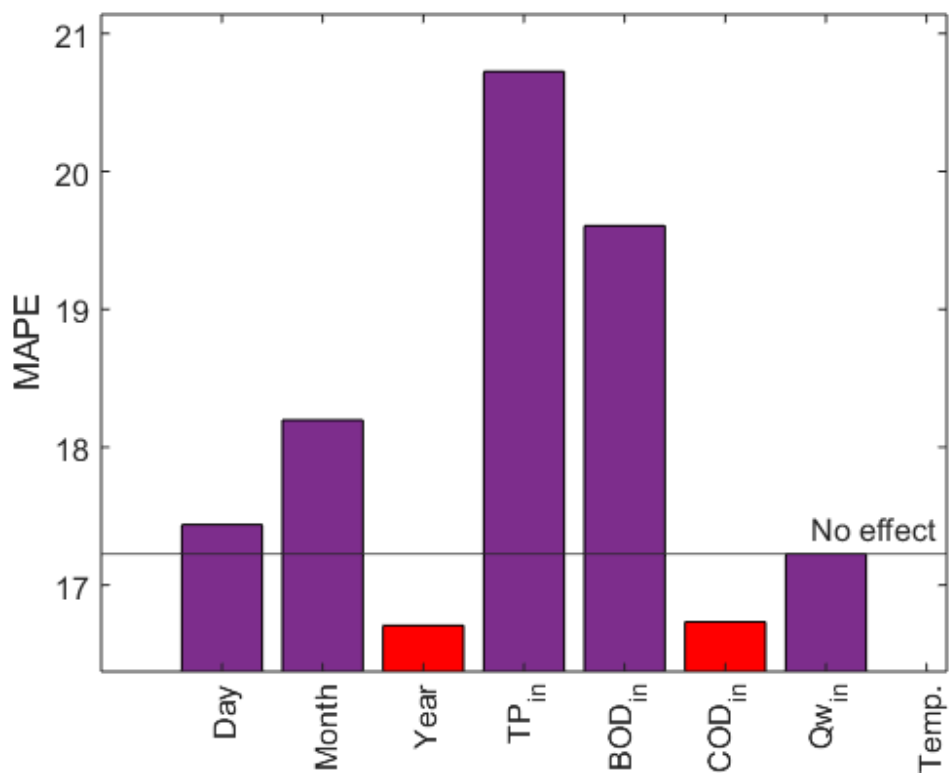
```
    %print([figpath,convertStringsToChars(figure)], '-depsc');  
end  
end
```











Published with MATLAB® R2018b

A.2 ANN code from python

```
'''Initializing code'''

import scipy.io as sio
import numpy as np
from keras.backend import epsilon
from statistics import mean
from sklearn import preprocessing
from keras.models import Sequential
from keras.layers import Dense, BatchNormalization, Activation, Dropout, \
    GaussianNoise
from keras.layers import LeakyReLU, ELU
from scipy.stats.stats import pearsonr
from keras import regularizers, optimizers
from keras.callbacks import History
import matplotlib.pyplot as plt
import scipy.io

import time
start_time = time.time()

#Chooses seed for single random state
seed = 6
np.random.seed(seed)
```

Listing 1: Example code

A.2. ANN CODE FROM PYTHON

```
'''Getting data and converting into right shape'''

#Get data from Mat file --- A1
MatFile = sio.loadmat('SaulekilenDataBWTemp.MAT')
Val_unnormalized = MatFile['a'].astype('float32')
InputIndex = [0,1,2,3,5,7,9,10] #Chooses input parameters
OutputIndex = [4] #Chooses output parameters
#Parameters are as follows with index:
#D0 M1 Y2 TPin3 TPOut4 BODin5 BODout6 CODin7 CODout8 Qw9 Temp10

#Normalizing Val matrix and chooses time period (155:221 = time period 3-6)
min_max_scaler = preprocessing.MinMaxScaler()
Val = min_max_scaler.fit_transform(Val_unnormalized[155:221,InputIndex])\
    *0.8+0.1 #1:154 , 155:221
Val = np.c_[Val,Val_unnormalized[155:221,OutputIndex]] #1:154 , 155:221

#Shuffling to make sure data is evenly spread
shuffle = np.arange(0,Val.shape[0])
np.random.shuffle(shuffle)

#Dividing into 70% training and 30% testing
TrainMax = round(Val.shape[0]*0.7)

Val_train = Val[shuffle[0:TrainMax]]
Val_test = Val[shuffle[TrainMax:Val.shape[0]]]
```

Listing 2: Example code

A.2. ANN CODE FROM PYTHON

```
#Dividing into input and output data
#D0 M1 Y2 TPin3 BODin4 CODin5 Qw6 Temp7
InputIndex = [0,1,2,3,4,5,6,7]
OutputIndex = [8]
x_train = Val_train[:, InputIndex]
x_test = Val_test[:, InputIndex]

y_train = Val_train[:, OutputIndex]
y_test = Val_test[:, OutputIndex]
```

Listing 3: Example code

A.2. ANN CODE FROM PYTHON

```
### ANN model

def baseline_model():
    model = Sequential()
    model.add(GaussianNoise(0, input_shape=(x_train.shape[1],)))
    model.add(Dense(800, activation='elu', \
                    kernel_regularizer=regularizers.l2(0.01)))
    #model.add(BatchNormalization(momentum=0.95))
    #model.add(ELU(alpha=1.0))
    #model.add(Dropout(0.2))
    model.add(Dense(30, activation='elu'))
    #model.add(ELU())
    #model.add(Dropout(0.2))
    #model.add(Dense(100, activation='exponential'))
    model.add(Dense(1, activation='elu'))
    #model.add(ELU(alpha=1.0))
    #model.add(Activation('sigmoid'))

    Adam = optimizers.adam(lr=0.0005, beta_1=0.9, beta_2=0.999, decay=0.02)
    model.compile(loss='mse', optimizer=Adam, metrics=['mean_squared_error'])
    return model

model = baseline_model()
history = model.fit(x_train, y_train, validation_data=(x_test, y_test), \
                   epochs=400, batch_size=10, verbose=0)
```

Listing 4: Example code

```
### Predictions
x_all = np.vstack((x_train, x_test))
y_predict = model.predict([x_test])
y_predictTrain = model.predict([x_train])
y_all = np.vstack((y_train, y_test))
y_predictAll= model.predict([np.vstack((x_train, x_test))])

### Printing losses
model.test_on_batch(x_test, y_test)
model.metrics_names
# list all data in history
print(history.history.keys())
# summarize history for accuracy
fig = plt.figure()
fig.subplots_adjust(left=0.125, right=1.5, bottom=0.1, top=0.5, wspace=0.2)
ax1 = fig.add_subplot(1,2,1)
#plt.plot(history.history['mean_absolute_percentage_error'])
ax1.plot(history.history['val_mean_squared_error'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['test', 'train'], loc='upper left')
# summarize history for loss
ax2 = fig.add_subplot(1,2,2)
ax2.plot(history.history['loss'])
ax2.plot(history.history['val_loss'])
```

Listing 5: Example code

```
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

### Calculate quality
percent = np.zeros((1,1))
percentA = np.zeros((1,1))

#Correlation
r = pearsonr(y_test[:,0], y_predict[:,0])
rA = pearsonr(y_all[:,0], y_predictAll[:,0])
#Mean squared error
MSE = sum((y_predict[:,0]-y_test[:,0])**2)/len(y_predict[:,0])
MSE_All = sum((y_predictAll[:,0]-y_all[:,0])**2)/len(y_predictAll[:,0])
#Mean absolute percentage error
for i in np.arange(0, len(y_predict[:,0])):
    percent = percent+abs(y_test[i,0]-y_predict[i,0])/max(y_test[i,0],0.01)
MAPE = percent*100/len(y_predict[:,0])
for i in np.arange(0, len(y_predictAll[:,0])):
    percentA = percentA+abs(y_all[i,0]-y_predictAll[i,0])/max(y_all[i,0],0.01)
MAPE_All = percentA*100/len(y_predictAll[:,0])

MSEp = "MSE: "
MAPEp = "MAPE: "
rp = "r: "
```

Listing 6: Example code

A.2. ANN CODE FROM PYTHON

```
MSEp = MSEp+str(round(np.median(MSE),5))+", "+str(round(np.median(MSE_All),5))\
    #np.median er kun for å gjøre slik at round faktisk funker
MAPEp = MAPEp+str(round(np.median(MAPE),5))+", "+\
    str(round(np.median(MAPE_All),5))
rp = rp+str(round(np.median(r[0]),5))+", "+str(round(np.median(rA[0]),5))

### Print results

print(MSEp)
print(MAPEp)
print(rp)
print(str(round(np.median(MSE),5)+" "+str(round(np.median(MAPE),5))+" "+\
    str(round(np.median(r[0]),5))+" %s" % (round(time.time() - start_time,1)))
y_plotAllSort = np.vstack((y_all[:,0] , y_predictAll[:,0], x_all[:,0],\
    x_all[:,1], x_all[:,2])).transpose()
y_plot0 = np.vstack((y_test[:,0] , y_predict[:,0])).transpose()
#y_plot0 = y_plot0[y_plot0[:,0].argsort()] #Sorts after size
y_plotAll = np.vstack((y_all[:,0] , y_predictAll[:,0])).transpose()

fig = plt.figure()
fig.subplots_adjust(left=0.125, right=1.6, bottom=0.1, top=2.3, wspace=0.2)

fig.subplots_adjust(left=0.125, right=1.6, bottom=0.1, top=0.6, wspace=0.2)
ax1 = fig.add_subplot(1,2,1)
ax1.plot(y_plot0)
plt.title('TPout test')
#plt.show()
```

Listing 7: Example code

```

ax2 = fig.add_subplot(1,2,2)
ax2.plot(y_plotAll)
plt.title('TPout train+test')
plt.show()

y_plot1 = np.vstack((y_test[:,1] , y_predict[:,1])).transpose()
#y_plot0 = y_plot0[y_plot0[:,0].argsort()] #Sorts after size
y_plotAll1 = np.vstack((y_all[:,1] , y_predictAll[:,1])).transpose()

fig = plt.figure()
fig.subplots_adjust(left=0.125, right=1.6, bottom=0.1, top=2.3, wspace=0.2)

fig.subplots_adjust(left=0.125, right=1.6, bottom=0.1, top=0.6, wspace=0.2)
ax1 = fig.add_subplot(1,2,1)
ax1.plot(y_plot1)
plt.title('BODout test')
#plt.show()
ax2 = fig.add_subplot(1,2,2)
ax2.plot(y_plotAll1)
plt.title('BODout train+test')
plt.show()

y_plot2 = np.vstack((y_test[:,2] , y_predict[:,2])).transpose()
#y_plot0 = y_plot0[y_plot0[:,0].argsort()] #Sorts after size
y_plotAll2 = np.vstack((y_all[:,2] , y_predictAll[:,2])).transpose()

```

Listing 8: Example code

A.2. ANN CODE FROM PYTHON

```
fig = plt.figure()
fig.subplots_adjust(left=0.125, right=1.6, bottom=0.1, top=2.3, wspace=0.2)

fig.subplots_adjust(left=0.125, right=1.6, bottom=0.1, top=0.6, wspace=0.2)
ax1 = fig.add_subplot(1,2,1)
ax1.plot(y_plot2)
plt.title('CODout test')
#plt.show()
ax2 = fig.add_subplot(1,2,2)
ax2.plot(y_plotAll2)
plt.title('CODout train+test')
plt.show()

#Saving models
#print("--- %s seconds ---" % (time.time() - start_time))
#scipy.io.savemat(r'C:\Users\magga\Documents\MAS500 - Masteroppgave/ANNy_plotAllTP23.mat', mdict={
#scipy.io.savemat(r'C:\Users\magga\Documents\MAS500 - Masteroppgave/ANNy_plot0TP23.mat', mdict={'y
#scipy.io.savemat(r'C:\Users\magga\Documents\MAS500 - Masteroppgave/ANNy_plotAllSortTP23.mat', mdi

'''

#Calculation of learning rate decay trend
lr = np.zeros((200,1))
lr[0] = 0.001
d=1
for i in np.arange(1,200):
    lr[i] = lr[i-1]*1/(d+1)
'''
```

Listing 9: Example code

Appendix B

Agendas and minutes of meetings

This appendix includes the agendas and minutes of meetings for the thesis. As soon as the thesis had gained some momentum, a fortnightly meeting was scheduled to have a continuous discussion between student and supervisors. Additionally the meetings with the staff of Saulekilen are included in this appendix.

B.1 Agenda #1

Location: UiA, Grimstad, room A3-095

Date: 07.02.2019

Time: 12:15-13:15

Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Issues to be discussed

- Report structure
 1. Introduction-Theory-Method-Results-Discussion-Conclusion, a good structure?
 2. Contents of theory chapter: More focus on ANN or WWTP?
 3. Contents of method chapter: More focus on ANN or WWTP?
- Gantt chart
 1. Current progress good?
 2. Add or subtract certain items?
 3. Change durations?
- What should be the upcoming focus?

B.2 Minutes of meeting #1

Location: UiA, Grimstad, room A3-095
Date: 07.02.2019
Time: 12:15-13:15
Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Table B.1: Minutes of meeting #1

Item	Issue Discussed
1	Report structure
1.1	Background: Should discuss what I should do in regards to what others have done as things could be different here.
1.2	Motivation should be after background as a follow up.
1.3	General processes of WWTP should be described in chapter 2 together with the data and processes of Sauekilen as a concrete example.
1.4	Theory chapter should be mostly ANN and the method chapter should be the methods I have used.
1.5	Advanced chemical and biological theory should not be a focus in second chapter (WWTP chapter).
2	Gantt chart
2.1	Gantt chart looks fine, but will as always change over time.
2.2	MNIST study seems to be a good place to begin with Keras.
3	Upcoming focus
3.1	Get Keras to work with MNIST.
3.2	Analyse data, plan data setup and additional data.
3.3	Sensor min/max could potentially be used as normalised min/max.

B.3 Agenda #2

Location: UiA, Grimstad, room C3-093

Date: 18.02.2019

Time: 13:00-14:30

Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Issues to be discussed

- Progress since last meeting (Quick briefing)
 1. MNIST model
 2. Universal approximation theorem
 3. Evaluation of current progress
- What inputs/outputs should be used?
 1. Wide variety between other projects.
 2. Often: TSS, BOD, COD, TP, TN and pH in, and COD, BOD and TSS out.
 3. Current data only includes TP, BOD and COD in/out.
- How many data points do we need?
 1. Other projects varies from 100 to 300 to 1200. We currently have around 350 from 2001-2018.
- How can we use the predicted BOD/COD/TSS to optimise the plant?
- What do we do with outliers and incomplete data points?
- ANN model design.
- What should be the upcoming focus?

B.4 Minutes of meeting #2

Location: UiA, Grimstad, room C3-093
Date: 18.02.2019
Time: 13:00-14:00
Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Table B.2: Minutes of meeting #2

Item	Issue Discussed
1	Saulekilen data analysis
1.1	Contact Knut Berg Larsen for data.
1.2	Amount of water might be an important input.
1.3	Make a network based on the ~ 350 data points to make sure there are no bottle-necks.
2	ANN model
2.1	Make multiple models, f.ex.: COD+BOD output, COD output, BOD output etc.
2.2	Increase the model size to include additives as well.
2.3	Use additive model as a model for the whole plant which can be used as a controller.
3	Output data analysis
3.1	Power point tracking can be used to find best values for additives.
3.2	Important to test the network well: If additives goes up, quality should go up as well.
3.3	Manually check if there are outliers in the output.

B.5 Agenda #3

Location: Saulekilen Renseanlegg
Date: 12.03.2019
Time: 9:30-10:30
Attendees: Magnus Ersdal, Saulekilen staff

Issues to be discussed

- How does Saulekilen decide additives today?
- Where does the measurement points lay in relation to the additive points?
- What do Saulekilen measure and add?
- Could some of the measurements of others be of any use for Saulekilen?
- Are there certain additives/measurements that are planned to be added?
- Do measurements happen for every settler?
- How is Saulekilen data compared to UCI data?
- How is the performance of the plant so far?
- How important is it for the performance to be as required?

B.6 Minutes of meeting #3

Location: Saulekilen Renseanlegg
 Date: 12.03.2019
 Time: 9:30-10:30
 Attendees: Magnus Ersdal, Saulekilen staff

Table B.3: Minutes of meeting #3

Item	Issue Discussed
1	Additives
1.1	Additives are decided based on amount of water and pH.
1.2	Small changes in sewage, but larger changes in additional water.
1.3	If the amount of water goes down, the additives are increased.
1.4	Additives are added during flocculation.
2	Measurement points
2.1	Measurement points are prior to the flocculation.
3	Data amounts
3.1	Water, pH and Suspended solids are measured continuously.
3.2	Phosphor is measured every day, BOD and COD 24 times a year.
3.3	PIX (based on pH and Q) and polymer (based on Q) are the additives.
3.4	Pollutants is measured a couple of times a year.
4	Saulekilen compared to other plants
4.1	No additives/measurements are planned to be added.
4.2	No obvious reason to use others measurement methods.
4.3	Measurements does not happen for every settler, does therefore not compare to UCI data.
6	Plant performance
6.1	Performance does vary a lot, but is mostly sufficient.
6.2	Performance can be way too high or not high enough.

B.7 Agenda #4

Location: UiA, Grimstad, room C3-094

Date: 13.03.2019

Time: 13:00-14:00

Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Issues to be discussed

- Progress since last meeting (Quick briefing)
 1. Wide range of models attempted.
 2. Network setups seems to increase accuracy slightly.
 3. Hyperparameters does not seem to increase accuracy enough.
 4. Current accuracy still very low.
 5. MLR model also gives low accuracy
 6. Found a new data set online (UCI) that gave same results.
- Review of Saulekilen meeting
 1. Received 2017-2019 daily data (500+ data points).
 2. Retrieving continuous data from server not easy.
 3. Saulekilen only adds PIX and polymers.
 4. Does only measure: pH, Water, SS, Phosphor, BOD and COD.
- Is the low accuracy due to data that does not cover the whole plant?
- Could there be any structural mistakes in the network?
- Should the Report still be wrote about ANN even with bad results?
- How should the introduction include sources?
- What should be the upcoming focus?

B.8 Minutes of meeting #4

Location: UiA, Grimstad, room C3-094
Date: 13.03.2019
Time: 13:00-15:00
Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Table B.4: Minutes of meeting #4

Item	Issue Discussed
1	Insufficient data
1.1	The residence time between input and output data could be crucial to account for.
1.2	Can use auto correlation on water in/out to find the residence time if Saulekilen staff does not have the information.
1.3	The amount of influent water might have a significant effect.
1.4	Possible that there is not enough available data for a sufficient model.
2	Other models
2.1	Look for another model with available data that can be repeated to check for differences.
2.2	Contact Grimstad/Kristiansand plant and check for differences in data.
3	Citation
3.1	Introduction should include citation in the same way as the rest of the text.

B.9 Agenda #5

Location: UiA, Grimstad, room C3-092

Date: 01.04.2019

Time: 11:15-12:15

Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Issues to be discussed

- Progress since last meeting (Quick briefing).
 1. Extracted data from the new data set.
 2. Many incorrect values/formulas were fixed.
 3. Did not find any new data sets to work with.
 4. Did not find any literature about the UCI database that uses ANN.
 5. Made a MLR model for the new data set with $r=0.62$.
 6. Made an ANN model for the new data set with $r=0.63$.
 7. Called Groos treatment plant and found that they used slightly different methods.
- Should the missing data be filled in or deleted?
- Continue to look for similar literature with available data?
- How detailed should the ANN theories be?
- The existing literature on the UCI data uses decision trees, is this worth comparing to?
- Should I focus more on writing now, or should I start a plan B?

B.10 Minutes of meeting #5

Location: UiA, Grimstad, room C3-092
Date: 01.04.2019
Time: 11:15-12:15
Attendees: Magnus Ersdal, Andreas Klausen

Table B.5: Minutes of meeting #6

Item	Issue Discussed
1	Data correlation
1.1	Current correlation of $r=0.62$, not good enough to use.
1.2	Due to the high correlation between T-P in and BOD/COD in, it could be possible to generate data for BOD/COD and use this in the model.
1.3	The correlation between COD/BOD out could be used to make a prediction on the BOD out data without having to wait 5 days.
1.4	Should be careful about generating new data without proof of realism.
2	Other models
2.1	Could be useful to compare the UCI ANN to the decision tree models in existing literature.
2.2	Could be useful to compare Groos data to Saulekilen data, and if it is possible to make a good model for Groos, a conclusion should be drawn to improve Saulekilen data.
3	Writing
3.1	Should focus on writing.
3.2	Not necessary to describe ANN in too full of a detail. Back propagation probably not necessary.

B.11 Agenda #6

Location: UiA, Grimstad, room C4-043

Date: 08.04.2019

Time: 13:00-14:30

Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Issues to be discussed

- Progress since last meeting (Quick briefing)
 1. Wrote about theory and method.
 2. Groos data is similar in setup to Saulekilens bi-weekly data.
 3. Made a linear and non-linear model for TP to BOD/COD
- Report setup
 1. How should the top text be?
 2. Necessary to describe MLR and polynomial regression?
 3. How large should the figures be?
 4. Is Latex' standard positioning the best option for figures and tables?
 5. Should task description be a copy of the task description given by UiA?
- Sources
 1. Is the NNADL by Nielsen enough of a source for theory?
 2. What are the rules when copying pictures from other sources?
 3. Should even the basic material be backed up better?
 4. Is a blog like HydroTech a valid source?

B.12 Minutes of meeting #6

Location: UiA, Grimstad, room C4-043
Date: 08.04.2019
Time: 13:00-14:30
Attendees: Magnus Ersdal, Andreas Klausen

Table B.6: Minutes of meeting #6

Item	Issue Discussed
1	Regression models and data
1.1	Calculate standard deviation to use as noise for TP to BOD/COD.
1.2	Include water flow for the bi-weekly model.
1.3	Get the formula for PIX from Saulekilen to include in the bi-weekly data.
1.4	Possible that some of the data should be removed as it was measured during special environments (during malfunctions etc.).
2	Report setup
2.1	Save Matlab files as "eps" to get better resolution.
2.2	Figures should fit the linewidth, but most importantly be easily readable.
2.3	The task description can stay as it is for now, but should probably be slightly changed when the full scope of the project is figured out.
3	Sources
3.1	Not a necessary improvement to have multiple sources if a single source is credible.
3.2	The HydroTech blog post could be considered credible.
3.3	Possible to refer to sources in the beginning of a section.

B.13 Agenda #7

Location: Saulekilen Renseanlegg
Date: 24.04.2019
Time: 10:00-11:30
Attendees: Magnus Ersdal, Geir Breimyr

Issues to be discussed

- Presentation of the work so far
 1. Created model that finds TP_{out} from the measurements in the daily and bi-weekly data set.
 2. PIX will be the control parameter.
 3. Accuracy is descent.
- Information about plant
 1. Where is visible depth measured compared to PIX?
 2. Can one days treatment have an effect on the next day?
 3. How large can measurement errors be?
 4. What is the reason for all the large changes in the data?
 5. What is the formula for PIX?
- Model
 1. Are there other measurement methods that could have been interesting to add to the plant?
 2. Are there other measurements that should be included in the current model?
 3. Is it alright to add and correct values that are wrong in the way it is currently done?
- Report
 1. Is the current text describing Saulekilen accurately?
 2. Is it called PIX or PAX?

B.14 Minutes of meeting #7

Location: Saulekilen Renseanlegg
Date: 24.04.2019
Time: 10:00-11:30
Attendees: Magnus Ersdal, Geir Breimyr

Table B.7: Minutes of meeting #7

Item	Issue Discussed
1	Information about plant
1.1	Visible depth, TS, TS(%) and pH are all measured after PIX is added.
1.2	It is possible that one days measurements could affect the next day. There are at least not usually large changes from day to day.
1.3	Changes in methods, repairs, rebuildings etc. are probably the reason for the data changes.
1.4	PIX formula based on measurements that are not available in the current data set.
2	Measurement errors
2.1	Visible depth is done by eyesight and could differ from one inspector to another. Expected to have some error here.
2.2	Phosphorous sensor has not been calibrated and probably gives wrong numbers. This will be fixed soon. This is probably the reason why laboratory measurements are different from plant measurements on TP.
2.3	pH_{in} has not been used for a long while and could therefore give wrong numbers.
3	Model discussion
3.1	Should check if there are more inconsistencies between the daily and bi-weekly data set. Possible that Saulekilen should replace their phosphorous sensor.
3.2	Suspended solids could have been interesting to include. Nitrogen could be noted as well.
3.3	As the temperature of the local weather is thought to affect the wastewater, this could be an interesting addition to the model.
3.4	The current method of editing and adding values seems logical.
4	Report
4.1	The current description of Saulekilen seems accurate.
4.2	PAX was the old additive, PIX is the new additive.

B.15 Agenda #8

Location: UiA, Grimstad, room C4-041

Date: 25.04.2019

Time: 10:00-11:30

Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Issues to be discussed

- Presentation of the work so far and Saulekilen meeting
 1. Adding water flow for the bi-weekly model gave minimal improvement.
 2. The data set has been divided into 6 different time periods to reflect the changes to the plant.
 3. Only Q_w and TP_{in} are measured before PIX is added.
 4. Large possibility for measurement errors.
 5. PIX formula based on measurements that are not available in the current data set.
- Model
 1. Could temperature (or any other weather measurements) be interesting to add to the model?
 2. What measures should be taken regarding the measurement errors?
 3. Is it sufficient to only use data from the last 7-9 months as this is the only time where data does not fluctuate due to plant changes?
- Report
 1. Should every test be included in the Report, and in that case, how?
 2. Should the Method chapter be setup with the basis on the scientific method? (Observation, Question, Hypothesis, Prediction, Tests, Results)
 3. If the scientific method should be used, are the current hypothesis sufficient?
- Would an automatic measurement of visible depth be an good addition for a model which uses continuous measurement data?

B.16 Minutes of meeting #8

Location: UiA, Grimstad, room C4-041
Date: 24.04.2019
Time: 10:00-11:50
Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Table B.8: Minutes of meeting #8

Item	Issue Discussed
1	Data
1.1	Correlation between lab and plant data looks better for TP_{in} as the values are higher, meaning smaller percentage errors.
1.2	Temperature could be interesting to use. A webcrawler could extract the necessary data from i.e. Yr.no.
1.3	Good to divide into time periods as the system is not equal at all times.
2	Model
2.1	For the MLR model it is possible to use some data for training and other for testing to make sure it generalises.
2.2	As the system that is being analysed is changing over time, it could be useful to use a continuous model for new data.
3	Report
3.1	It should not be assumed that the reader knows much about WWTP and the text should reflect that.
3.2	Should be mentioned in the Report that Saulekilen does not use sedimentation
3.3	Should write about the variance and correlation between the lab and plant data.
3.4	Result graph showing both MSE and correlation could be used and the larger changes should be explained.
3.5	Could have two result chapters where one is for data analysis and the other is for ANN/MLR results.
3.6	Tables and graphs that show results should not be in the method chapter.
3.7	Method chapter should explain what was done before testing and what should be done if tests showed specific results.

B.17 Agenda #9

Location: UiA, Grimstad, room D3-051

Date: 06.05.2019

Time: 13:00-14:00

Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Issues to be discussed

- Presentation of the work so far
 1. Including Qr and dates helped with the results at times.
 2. Added all current results except for ANN to the Report.
 3. BODin and CODin models are massively improved by using recent data
- Model
 1. Should the MLR model also use train vs test data?
 2. Should ANN/MLR quality be considered by test/train/all or a mix?
- Report
 1. Should formulas be written with variable first or explanation first?
 2. How are the current examples of thesis statements?
 3. How does the link between method and result look now?
 4. Are the new figures readable enough, and how should the bar graphs be?
 5. The book by Michael Nielsen uses $1/(2n)$ for MSE, but this is not used other places, which should be used?
 6. How should the sensitivity graphs look?
 7. Useful to mention all the different dates with a malfunction/error?
 8. Any more results that should be included? More time periods
 9. Should the gannt charts be included?

B.18 Minutes of meeting #9

Location: UiA, Grimstad, room D3-051
Date: 06.05.2019
Time: 13:00-14:00
Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Table B.9: Minutes of meeting #9

Item	Issue Discussed
1	Data
1.1	Check if there is any i.e. Gaussian noise on the deviation between lab and plant measurements of TP_{out} .
2	Model
2.1	Better to use test data and train data for MLR as well.
3	Report
3.1	Could have lines in the data trends to show the different time period splits.
3.2	Be more descriptive in each caption.
3.3	Caption for tables should be above the table.
3.4	Important to mention that the sensors are not calibrated and are not equal to the lab tests.
3.5	Be consistent when writing about variables: variable x is this, y is that.
3.6	Showing results for the all time periods and just the recent time periods is good.
3.7	Gannt charts not that important to include.
3.8	The bar graphs could be split up, more important that they are easy to read than there being many of them.

B.19 Agenda #10

Location: UiA, Grimstad, room C4-095

Date: 15.05.2019

Time: 14:00-15:00

Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Issues to be discussed

- Presentation of the work so far
 1. Go through last meetings MoM.
 2. Have made all tests depend on shuffled train/test data.
 3. "Finished" theory, method and WWTP chapters.
- Model
 1. Would standard deviation be an interesting comparison between y_{test} and y ? They seem to usually have different noise levels and ranges.
- Report
 1. Is any of the pictures describing the WWTP good?
 2. What could the reason for the COD/BOD ratio be, and why is it so different from Saulekilen's ratio.
 3. How should the progression to and final ANN models be described?
 4. Should the UCI database be included in the report?
 5. What is most important to include in appendix?
 6. Anything to add to the result chapter?
 7. How should the structure of the discussion chapter be?
 8. Not always simple to rewrite formula description to become "active" reading.
 9. In Equation 3.1, should it be \hat{y}_i or y_i ?

B.20 Minutes of meeting #10

Location: UiA, Grimstad, room C4-095
Date: 15.05.2019
Time: 14:00-15:30
Attendees: Magnus Ersdal, Morten Ottestad, Andreas Klausen

Table B.10: Minutes of meeting #10

Item	Issue Discussed
1	Model
1.1	Expected to have less noise in estimated data than measured data, could be written about as well.
2	Report
2.1	Motivation should be before ANN which should be before background.
2.2	Both pictures in WWTP describes it well, can use both.
2.3	The COD/BOD ratio could be misleading or based on different circumstances. Should read up on it to make sure the introduction is clear.
2.4	Get better understanding for the biological and chemical processes.
2.5	Should describe possible measurement errors in WWTP chapter.
2.6	Model comparison should be explained in theory chapter.
2.7	Variance is not necessary to show as standard deviation is more descriptive.
2.8	Should explain the main points of improvement of the ANN's.
2.9	UCI database did not lead anywhere, so it is unnecessary to write about.
2.10	Appendix could include additional results and code for MLR and ANN model.
2.11	Correct to use y_i and \hat{y}_i as it is.
2.12	Nothing obvious to add to result chapter.
2.13	Discussion should focus on what the result showed and what has been discovered through the thesis.