

https://dspace.org/

Aerodynamic Force Variation on a Trailing MotoGP Motorcycle in a Corner

Shaw, Craig Byrne

2024

University of the Witwatersrand, Johannesburg

https://hdl.handle.net/10539/45451

Shaw, Craig Byrne . (2024). Aerodynamic Force Variation on a Trailing MotoGP Motorcycle in a Corner [Masters dissertation, University of the Witwatersrand, Johannesburg]. WIReDSpace. https://hdl.handle.net/10539/45451

Downloaded from a DSpace repository. https://dspace.org/



Prediction of Water Hyacinth Coverage on Hartbeespoort Dam

Claudia D. Camacho de Gouveia

(Student number: 1600387)

Supervisor: Doctor Joke Bührmann

School of Mechanical, Industrial and Aeronautical Engineering

University of the Witwatersrand

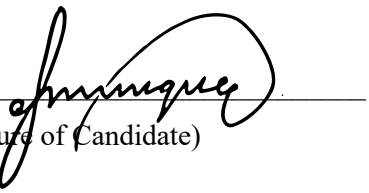
Johannesburg, South Africa.

A dissertation submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Industrial Engineering.

21 October 2024

DECLARATION

I, Claudia Dominique Camacho de Gouveia, declare that this dissertation is my own unaided work. It is being submitted for the Degree of Master of Science in Industrial Engineering (Masters in Research) to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



(Signature of Candidate)

21 day of October 2024 in Hartbeespoort

ABSTRACT

Water hyacinth is an invasive weed contributing to Hartbeespoort Dam's poor water quality. Although biological control is the most effective and sustainable method of controlling water hyacinth, the dam has unfavourable conditions for agents that the weed thrives in. Literature uses mathematical models and remote sensing to theorise growth rates or estimate coverage. However, prediction could prove beneficial as planning biological control is essential to its success. Hence, a model to predict water hyacinth coverage was developed.

This research simplified the complex relationships involved in water hyacinth growth to focus on the most influential factors: temperature and nutrients. Missing data were imputed using multiple k-nearest neighbours. Nutrient datasets had limited data, thus five scenarios were developed to extrapolate datasets, using Monte Carlo simulation and seasonal patterns. The features were used to build ensemble, decision tree, artificial neural network and support vector machine models. Ensemble using the bagging method was the best model resulting in a root mean square error of 4.01 for water hyacinth coverage predictions from 1 June 2018 to 1 May 2019.

ACKNOWLEDGEMENTS

I would like to acknowledge the following for their contribution to this dissertation, however small or indirect it might have been:

- Dr Joke Bührmann, besides your advice and support, I would not have been able to ever submit were it not for your patience and encouragement. Thank you for also guiding me in more ways than just in my masters.
- David Kinsler, Gillian McGregor and Prof Julie Coetzee, I am beyond grateful for the data you have allowed me to use. Without it I would not have been able to pursue this topic. Thank you also Prof Julie for your quick responses and expertise.
- Prof Jan Marten Huizenga and others at NWU who built the website from which I downloaded nutrient datasets.
- SAWS for suppling temperature data.
- And lastly my family, you did not understand or know what I was doing during this time but you supported me always and helped me. I will forever be grateful, thank you. Dion, I would have given up many times had it not been for you. Thank you for believing in me.

Psalm 27:14

Wait for the LORD;
be strong, and let your heart take courage;
wait for the LORD!

Thank you Lord for giving me the patience, wisdom and courage I needed.

TABLE OF CONTENTS

DECLARATION ..... **i**

ABSTRACT ..... **ii**

ACKNOWLEDGEMENTS ..... **iii**

TABLE OF CONTENTS ..... **iv**

LIST OF FIGURES ..... **viii**

LIST OF TABLES ..... **xii**

NOMENCLATURE ..... **xv**

1 INTRODUCTION **1**

1.1 Research Background 1
1.2 Research Motivation 2
1.3 Problem Statement 2
1.4 Scope 2
1.5 Research Question 3
1.6 Research Objectives 3
1.7 Ethical Clearance 4
1.8 Dissertation Overview 4

2 FACTORS INFLUENCING WATER HYACINTH COVERAGE **5**

2.1 Nutrients in Water 5
2.1.1 Total phosphorous (TP) 6
2.1.2 Total nitrogen (TN) 6
2.1.3 Relationship with water hyacinth 7
2.2 Ambient Temperature 7
2.3 Relevant Research 8
2.3.1 Remote sensing 8
2.3.2 Centre for Biological Control 9
2.4 Favourable Conditions for Water Hyacinth Growth 10
2.5 Chapter Summary 10

3 LITERATURE REVIEW: DATA HANDLING **11**

3.1 Data Analysis and Visualisation 11
3.1.1 Visualisation of the data 12
3.2 Data Cleansing 12
3.2.1 Verifying correct data 12

3.2.2 Transforming variables 12
3.2.3 Dealing with outliers 13
3.3 Handling Missing Data 13
3.3.1 Imputation 14
3.3.2 Multiple imputation methods 14
3.3.3 Relevant case studies on handling missing data 15
3.4 Validating Imputations 15
3.4.1 Comparing descriptive statistics 16
3.4.2 Comparing data graphically 16
3.4.3 Hypothesis testing 16
3.5 Extrapolating Datasets With Simulation 17
3.5.1 Fitting a PD 17
3.5.2 Determining a theoretical PD 17
3.5.3 Validating a PD with the goodness-of-fit test 19
3.5.4 Generating new data with Monte Carlo simulation 19

4 LITERATURE REVIEW: PREDICTIVE MODELLING 21

4.1 Predictive Modelling 21
4.2 Time Series Analysis 21
4.2.1 Seasonality 22
4.2.2 Autocorrelation 22
4.2.3 Time series forecasting models 23
4.2.4 Evaluating time series model performance 23
4.2.5 Time series models and analysis used in relevant literature 24
4.3 Machine Learning (ML) 24
4.4 ML Algorithms 25
4.4.1 Artificial neural network (ANN) 26
4.4.2 Support vector machine (SVM) 27
4.4.3 Random forest 28
4.4.4 k-Nearest neighbour (KNN) 29
4.5 Feature Selection 29
4.5.1 Finding new features 30
4.6 Evaluating ML Models Performance 30
4.6.1 Calculating performance 30
4.6.2 Verification and validation 31
4.6.3 Cross-validation 32
4.6.4 Overfitting and underfitting 32
4.7 Hyperparameter Tuning 32
4.8 Model Selection 33

4.9 Application of ML Models to WHC 33

5 DATA DESCRIPTION AND PRELIMINARY ANALYSIS 34

5.1 Water hyacinth coverage (WHC) 34
5.1.1 Data analysis of WHC 34
5.1.2 Percent of missing WHC data 38
5.2 Ambient temperature 39
5.2.1 Data analysis of MaxTemp and MinTemp 39
5.2.2 Percent of missing MaxTemp and MinTemp data 41
5.3 Total phosphate (TP) 42
5.3.1 Data analysis of TP and PO_4 datasets 44
5.3.2 Percent of missing TP data 47
5.4 Total nitrogen (TN) 48
5.4.1 Data analysis of NO_3+NO_2 and NH_4 datasets 49
5.4.2 Percent of missing data in NO_2+NO_3 and NH_4 datasets 51
5.5 Timelines of Each Dataset Used in Model Building 51

6 DATA HANDLING 53

6.1 Multiple KNN Imputation 53
6.2 Validating Imputed Data 54
6.2.1 Validating imputations of the input target variable: WHC 55
6.2.2 Validating imputations of the temperature features: MaxTemp and MinTemp 56
6.2.3 Validating imputations of the TP feature 58
6.2.4 Validating imputations of the nitrogen features: NO_2+NO_3 and NH_4 59
6.3 Extrapolating TP Dataset 60
6.4 Extrapolating TN Dataset 63
6.5 Finalised Datasets 65

7 MODEL BUILDING 67

7.1 Feature Analysis 67
7.1.1 Pearson's correlation 67
7.1.2 F-test hypothesis test 68
7.1.3 Possible combinations of features 69
7.2 Training Models and Feature Selection 70
7.3 Hyperparameter Tuning and Model Testing 71
7.4 Final Predictions 71

8 RESULTS 72

8.1 Model Building 72
8.2 Final Predictions 74
8.3 Discussion 76

9 CONCLUSIONS AND RECOMMENDATIONS 78

REFERENCES 79

APPENDIX A Details of Extrapolating TP Dataset 91

A.1 Time Series Analysis of TP Dataset 91
A.1.1 Summary of the time series analysis on TP dataset 93
A.2 Developing Five Scenarios for Extrapolating TP Dataset 94
A.2.1 Monte Carlo simulation 94
A.2.2 Interpolated TP data 96
A.2.3 Lagged TP dataset 98
A.3 Generating TP Dataset using Scenario 1 100
A.4 Generating TP Dataset using Scenario 2 102
A.5 Generating TP Dataset using Scenario 3 103
A.6 Generating TP Dataset using Scenario 4 104
A.7 Generating TP Dataset using Scenario 5 105
A.8 Selecting a TP Dataset 105

APPENDIX B Details of Extrapolating TN Dataset 107

B.1 Time Series Analysis of TN Dataset 107
B.1.1 Summary of time series analysis on TN datasets 110
B.2 Developing Five Scenarios for Extrapolating TN Dataset 110
B.2.1 Monte Carlo simulation 111
B.2.2 Interpolated TN data 112
B.2.3 Lagged TN dataset 114
B.3 Generating TN Dataset using Scenario 1 115
B.4 Generating TN Dataset using Scenario 2 116
B.5 Generating TN Dataset using Scenario 3 117
B.6 Generating TN Dataset using Scenario 4 118
B.7 Generating TN Dataset using Scenario 5 119
B.8 Selecting a TN Dataset 119

APPENDIX C Codes for Multiple KNN Imputation 121

APPENDIX D Code for Final Model 123

LIST OF FIGURES

Figure 1: Monthly average temperatures at the Hartbeespoort dam (NCEI, 2024) 8

Figure 2: Macrophyte Monitoring Tool developed by David Kinsler (Kinsler, 2021a) 9

Figure 3: An example of a Partial Autocorrelation Function graph (MATLAB, 2018) 23

Figure 4: Structure of ANN model (own interpretation) 26

Figure 5: Labelled structure of an SVM model (Tibrewal, 2023). 27

Figure 6: An example of a simple decision tree shown graphically (Dinov, 2018) 28

Figure 7: Time series plot of the observed WHC dataset 35

Figure 8: WHC data in 2016 (a) compared to 2020 (b) 35

Figure 9: Histogram of observed WHC dataset 36

Figure 10: Number of WHC data points recorded per month 37

Figure 11: Time series plot of WHC dataset with missing data points 38

Figure 12: Observed WHC dataset used in model building and to make final predictions 39

Figure 13: Time series plot of observed MaxTemp dataset 40

Figure 14: Time series plot of observed MinTemp dataset 40

Figure 15: Histogram of observed MaxTemp (a) and MinTemp (b) datasets 41

Figure 16: Observed temperature datasets used in model building and for final predictions 42

Figure 17: Time series plot of observed phosphorous datasets 44

Figure 18: Outliers identified and replaced with nearest TP value 45

Figure 19: Time series plot of the observed TP dataset without outliers 45

Figure 20: Time series plot of TP dataset without outliers with shaded area 46

Figure 21: Histogram of observed TP dataset 47

Figure 22: Time series plot of TP dataset with missing data points 47

Figure 23: Observed TP dataset used in model building and to make final predictions 48

Figure 24: Time series plot of NO_2+NO_3 dataset 49

Figure 25: Time series plot of NH_4 dataset 50

Figure 26: Outliers identified and replaced with nearest nitrogen value 50

Figure 27: Histogram of observed NO_2+NO_3 (a) and NH_4 (b) datasets 51

Figure 28: Timeline of each dataset 52

Figure 29: Process flow diagram for preparing the datasets for model building 53
Figure 30: Plot of imputed WHC and the observed WHC dataset 55
Figure 31: Imputed WHC and the confidence limits for each observed WHC data point 56
Figure 32: Plot of imputed and observed MaxTemp (a) and MinTemp (b) 56
Figure 33: Confidence limits with imputed MaxTemp (a) and MinTemp (b) 57
Figure 34: Plot of imputed and observed TP dataset 58
Figure 35: Confidence limits with imputed TP data 59
Figure 36: Plot of imputed and observed NO_2+NO_3 (a) and NH_4 (b) 59
Figure 37: Confidence limits of observed NO_2+NO_3 dataset with imputed data 60
Figure 38: Process flow diagram of method used to extrapolate TP dataset 61
Figure 39: Time series plot of the chosen TP dataset generated in Scenario 5 63
Figure 40: Process flow diagram of method used to extrapolate the TN dataset 64
Figure 41: Time series plot of the chosen TN dataset generated in Scenario 3 65
Figure 42: Time series plot of finalised WHC dataset used as the input target variable 65
Figure 43: Time series plot of finalised MaxTemp (a), MinTemp (b), TP (c) and TN (d) features 66
Figure 44: Process flow diagram of the methods followed in model building 67
Figure 45: Time series plot of the AvgTemp feature 68
Figure 46: Hypothesis F-test p-values of each feature 69
Figure 47: Time series plot of TNTP feature 69
Figure 48: Validation RMSE for each model trained using different feature combinations 72
Figure 49: Average validation RMSE for each feature combination 73
Figure 50: Test RMSE for each model after hyperparameter tuning 73
Figure 51: Scatter plot of the test predictions made by the ensemble (boost) model 74
Figure 52: Time series plot of final WHC predictions 74
Figure 53: Time series plot of final WHC prediction from 1 June 2018 to 30 April 2019 75
Figure 54: Comparison of RMSE of model using timeline and randomly selected unseen dataset 75
Figure 55: Final predictions of model using randomly selected unseen dataset 76
Figure 56: Time series plot of the observed TP dataset with trends 91

Figure 57: ACF plot of the observed TP dataset 92
Figure 58: Observed TP dataset and the identified cycles 92
Figure 59: Histogram of observed TP dataset (a) and itself lagged by 2220 points (b) 93
Figure 60: PACF plot of the observed TP dataset 93
Figure 61: Development of scenarios used to extrapolate the TP dataset 94
Figure 62: Plot of observed TP data and Monte Carlo simulation of TP 96
Figure 63: Daily average of the observed TP dataset (a) and simulated TP dataset (b) 96
Figure 64: RMSE of the polynomials curves fitted to the daily average TP data 97
Figure 65: Plot of interpolated TP data and the daily average observed TP data 98
Figure 66: Plot of fitted CDF with confidence interval ranges for the interpolated TP data 101
Figure 67: Plot of observed TP dataset and TP dataset generated in Scenario 1 101
Figure 68: Plot of Scenario 1 and the TP dataset simulated using Monte Carlo simulation 102
Figure 69: Plot of observed TP dataset and TP dataset generated in Scenario 2 103
Figure 70: Plot of observed TP dataset and TP dataset generated in Scenario 3 103
Figure 71: Plot of observed TP dataset and TP dataset generated in Scenario 4 104
Figure 72: Plot of observed TP dataset and TP dataset generated in Scenario 5 105
Figure 73: T-test p-values of the different TP datasets generated 106
Figure 74: Time series plot of observed TN dataset with trends 107
Figure 75: ACF plot of the observed TN dataset 108
Figure 76: ACF plot of the observed TN dataset for the first 80 lags 108
Figure 77: ACF plot of the observed TN dataset with dates in relation to the first 20 lags 109
Figure 78: PACF plot of TN data 109
Figure 79: Development of scenarios used to extrapolate the TN dataset 110
Figure 80: Scatter plot of observed TN dataset (a) and the simulated TN dataset (b) 112
Figure 81: Scatter plot of the daily average TN data 112
Figure 82: RMSE of fitted polynomial curves to the daily average TN data 113
Figure 83: Scatter plot of the interpolated TN data 113
Figure 84: Plot of CDF with confidence interval ranges for the interpolated TN data 115
Figure 85: Plot of observed TN dataset (a) and TN dataset generated in Scenario 1 (b) 116

Figure 86: Plot of observed TN dataset (a) and TN dataset generated in Scenario 2 (b) 117

Figure 87: Plot of observed TN dataset (a) and TN dataset generated in Scenario 3 (b) 117

Figure 88: Plot of observed TN dataset (a) and TN dataset generated in Scenario 4 (b) 118

Figure 89: Plot of observed TN dataset (a) and TN dataset generated in Scenario 5 (b) 119

Figure 90: KS p-values of the different generated TN datasets..... 119

LIST OF TABLES

Table 1: Levels of eutrophication (Bhusal & Devkota, 2020; DWS, 2019)..... 5

Table 2: Satellites used in remote sensing and GIS research (Khandelwal et al., 2021; Landsat Missions, 2024; Kinsler, 2021a; ESA, 2022) 9

Table 3: Conditions that influence water hyacinth growth and coverage (Rojas-Sandoval & Acevedo-Rodríguez, 2022; Owens & Madsen, 1995)..... 10

Table 4: Different types of data analyses (Taherdoost, 2020)..... 11

Table 5: Descriptive statistics definitions (Wasserman, 2010; Heiberger & Holland, 2015) 11

Table 6: Summary of graphs used to visualise data (Montgomery et al., 2016; Theobald, 2017) 12

Table 7: Categories of missing data (Cheng et al., 2019; Kang, 2013; Mohammed et al., 2021). 13

Table 8: Summary of conditions required for considering a distribution (Law, 2015; Ricci, 2005; Kissell & Poserina, 2017) 18

Table 9: Explanation of goodness-of-fit statistics (Kroese et al., 2020; Kenton, 2023)..... 19

Table 10: Types of ML algorithms (Igual & Seguí, 2017; Russell, 2018). 25

Table 11: Types of problems solved using supervised learning algorithms (Theobald, 2017; Kuhn & Johnson, 2016; Kroese et al., 2020)..... 25

Table 12: Overfitting versus underfitting (Russell, 2018; Bonaccorso, 2018; Mohri et al., 2018) 32

Table 13: Characteristics of the observed WHC dataset (Kinsler, 2021a) 34

Table 14: Descriptive statistics of observed WHC from 1 August 2015 to 28 December 2019 . 36

Table 15: Extract of WHC dataset and the list of dates required (Kinsler, 2021a) 37

Table 16: Characteristics of observed temperature dataset (SAWS, 2022)..... 39

Table 17: Descriptive statistics of the observed temperature datasets..... 40

Table 18: Station codes where nutrient data were obtained (Huizenga, 2015; DWS, 2019) 42

Table 19: Extract of TP dataset (DWS, 2019)..... 43

Table 20: Extract of PO_4 dataset obtained from NEMP (Huizenga, 2015) 43

Table 21: Characteristics of TP and PO_4 datasets (DWS, 2019; Huizenga et al., 2013)..... 43

Table 22: Characteristics of combined TP dataset..... 44

Table 23: Descriptive statistics of observed TP dataset 46

Table 24: Extract of nitrogen datasets from NEMP (Huizenga, 2015) 48
Table 25: Characteristics of dataset containing nitrogen compounds 49
Table 26: Descriptive statistics of the observed nitrogen datasets 50
Table 27: Information used to impute missing data within each dataset..... 54
Table 28: Difference in descriptive statistics of imputed and observed WHC..... 55
Table 29: Difference in descriptive statistics of imputed and observed temperature datasets 57
Table 30: Difference in descriptive statistics of imputed and observed TP dataset 58
Table 31: Difference in descriptive statistics of imputed and observed NO_2+NO_3..... 59
Table 32: Sum of NO_2+NO_3 and NH_4 to form TN dataset 60
Table 33: Pearson's correlation matrix for features..... 68
Table 34: Combinations of features used to train models 70
Table 35: Hyperparameters of each model using feature combination C..... 73
Table 36: Summary of scenarios used to extrapolate the TP dataset..... 94
Table 37: Descriptive statistics of the observed TP dataset from 1980 to 1985..... 95
Table 38: Theoretical PDs fitted to the observed TP dataset..... 95
Table 39: Goodness-of-fit results for PDs fitted to observed TP dataset 95
Table 40: Extract of the daily average TP data..... 97
Table 41: Details of the polynomial curves fitted to the daily average TP data..... 97
Table 42: Extract of TP dataset and lagged datasets 99
Table 43: Descriptive statistics of the delta datasets used in simulation scenarios 99
Table 44: Theoretical PD for TP-1506 and TP-2220 datasets..... 100
Table 45: Goodness-of-fit results for distributions fitted to TP-1506 and TP-2220 datasets.... 100
Table 46: Parameters of the distributions fitted to the deltas used in simulation scenarios 100
Table 47: Descriptive statistics of observed TP dataset compared to that of Scenario 1 102
Table 48: Descriptive statistics of observed TP dataset compared to that of Scenario 2 103
Table 49: Descriptive statistics of observed TP dataset compared to that of Scenario 3 104
Table 50: Descriptive statistics of observed TP dataset compared to that of Scenario 4 105
Table 51: Descriptive statistics of observed TP dataset compared to that of Scenario 5 105
Table 52: Summary of scenarios used to extrapolate the TN dataset..... 110

Table 53: Descriptive statistics of the observed TN dataset 111
Table 54: Theoretical PDs selected for fitting to the observed TN dataset 111
Table 55: Goodness-of-fit results for the PDs fitted to observed TN dataset 111
Table 56: Details of the polynomial fitted to the daily average TN data 113
Table 57: Extract of observed TN dataset and lagged datasets 114
Table 58: Descriptive statistics of the TN-1 and TN-2 datasets 114
Table 59: Theoretical PD selected for fitting to deltas used in simulation scenarios 114
Table 60: Goodness-of-fit results for distributions fitted to TN-1 and TN-2 datasets 115
Table 61: Parameters of the t Location Scale distribution fitted to the TN-1 and TN-2 datasets 115
Table 62: Descriptive statistics of the TN dataset generated in Scenario 1 116
Table 63: Descriptive statistics of the TN dataset generated in Scenario 2 117
Table 64: Descriptive statistics of the TN dataset generated in Scenario 3 118
Table 65: Descriptive statistics of the TN dataset generated in Scenario 4 118
Table 66: Descriptive statistics of the TN dataset generated in Scenario 5 119

NOMENCLATURE

Abbreviation	Name
ACF	Autocorrelation Function
AD	Anderson-Darling test
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
AvgTemp	Average Temperature (℃)
CBC	Centre for Biological Control
CDF	Cumulative Distribution Function
DOY	Day of Year
GIS	Geographic Information Sensing
KNN	k-Nearest Neighbour
KS	Kolmogorov-Smirnov test
MaxTemp	Maximum Temperature (℃)
MinTemp	Minimum Temperature (℃)
ML	Machine Learning
PACF	Partial Autocorrelation Function
PD	Probability distribution
RMSE	Root mean square error
SVM	Support Vector Machine
TN	Total Nitrogen (mg/L)
TN-1 (delta)	Difference between TN and itself lagged once
TN-2 (delta)	Difference between TN and itself lagged twice
TNTP	Ratio of TN to TP
TP	Total Phosphorous (mg/L)
TP-1506 (delta)	Difference between TP and itself lagged by 1506 points
TP-2220 (delta)	Difference between TP and itself lagged by 2220 points
WHC	Water hyacinth coverage (%)

1 INTRODUCTION

1.1 Research Background

The Hartbeespoort dam is situated in the North-West province of South Africa and has been eutrophic since the 1970s (WASA, 2020). When a body of water is eutrophic, it has a high concentration of nutrients, particularly total phosphorous (TP) and total nitrogen (TN), which are above the standards set out by the Department of Water Affairs and Forestry (Rossouw et al., 2008). The high concentration of nutrients in the dam is due to wastewater from surrounding areas and traces of mining effluent (Mitchelle et al., 2016). These nutrients form favourable conditions for the growth of water hyacinth (Matthews, 2016).

Water hyacinth, scientifically named Eichhornia Crassipes, is a highly invasive water weed, classified as the world's worst aquatic weed (Ghosh, 2010). It can absorb excess nutrients and toxins present in water and thus can be used to improve water quality (Auchterlonie et al., 2021; Petrucio & Esteves, 2000). However, this weed has a rapid growth rate resulting in thick mats that engulf waters, harm ecosystems, make water activities and accessibility difficult, and become a breeding place for disease-carrying organisms (Reddy & Tucker, 1983; Rojas-Sandoval & Acevedo-Rodríguez, 2022; Mzuza et al., 2015). Thus, research focuses on the eradication of water hyacinth.

The Centre for Biological Control (CBC) is doing the most research on water hyacinth control on the Hartbeespoort dam (Coetzee & Hill, 2011). The CBC researches water hyacinth control methods and has found biological control to be the cheapest and most sustainable (Coetzee & Hill, 2011; Miller, 2019). This method involves releasing insects, called biological agents, that solely feed on the targeted weed. For some weeds, more than one insect is released and/or insects are mass-released as often as once weekly (CBC, 2020a; Briese & Sindel, 2000). Biological control fails if either not enough agents are released or if the timing of release is wrong (Coetzee et al., 2022). Researchers at CBC have found the most effective biological agent to be the Megamelus Scutellaris – commonly known as the plant hopper (CBC, 2020a). Although effective, conditions at the Hartbeespoort dam are unfavourable for the agent, especially in winter, according to Coetzee and Hill (2011). As a result, the plant hoppers either kill almost all the water hyacinth, resulting in a return of high nutrients in the water, or the organisms die and struggle to repopulate, resulting in water hyacinth regrowth and coverage. It is also difficult to predict the effect the biological agents have on the weed due to the many factors influencing both (Jeffords & Hodgins, 1995).

All this is a problem as the Hartbeespoort Dam's surrounding community depends on the dam for domestic, industrial, agricultural, and recreational use (SANCOLD, 2018; Scott et al., 1980). Water hyacinth can improve water quality thus, their presence on the dam could be useful if controlled correctly. The CBC focuses on the plant hopper, its life cycles, favourable conditions, and effectiveness at controlling water hyacinth. Other researchers, such as Keebine (2019) and

Singh et al. (2020), focus on spatial mapping and remote sensing WHC using Geographic Information Sensing (GIS) and image recognition.

It is evident there are gaps in existing research. One identified gap is how water hyacinth coverage (WHC) can be predicted. By predicting WHC, similarly to sales predictions, control methods can be better planned (Maluleke & Bührmann, 2021; Gutiérrez et al., 2001). The researchers suggest biological control is more effective when planned. Predicting WHC on the dam will ensure control methods can be planned, thus improving the effectiveness of existing control methods and improving water quality in the long run. This way, this research will provide a proactive approach to the problem of poor water quality in the dam instead of the current and not as effective reactive approach.

1.2 Research Motivation

WHC is unpredictable due to the many factors influencing its growth (Jeffords & Hodgins, 1995). Control methods are not always effective due to conditions at the dam (Miller, 2019). The dam exhibits favourable conditions for water hyacinth growth and mostly unfavourable conditions for biological control populations to be able to stabilise (Coetzee & Hill, 2011). Mathematical models for estimating water hyacinth growth in relation to biological control, and visa-versa, do exist but, at the time of research, there are no models for predicting the WHC on the Hartbeespoort dam (Singh et al., 2020). Additionally, biological control is only effective if the correct number of bugs are placed on the dam at the right time (Miller, 2019; Coetzee et al., 2022). Currently, augmentative biological control is being used; this is where biological control is placed in excess, which is costly and not always sustainable (Jeffords & Hodgins, 1995). Therefore, planning biological control will make it more effective and improve water quality.

Water hyacinth is also useful in improving water quality, so some presence is beneficial (Auchterlonie et al., 2021; Petrucio & Esteves, 2000; Scott et al., 1980). WHC predictions can be used to plan biological control. Well-controlled water hyacinth will benefit surrounding communities as the water quality will improve, leading to drinkable water, little to no algae, and clear waterways. Currently, there are only theoretical, mathematical approaches to predicting water hyacinth growth. In the literature studied, no evidence of a way to predict coverage on a dam in a less theoretical way, was found.

1.3 Problem Statement

Based on the research motivation, the problem statement is as follows: There have not been any practical methods for predicting WHC on the dam, which is necessary to ensure effective control.

1.4 Scope

This research focused on determining relationships between available WHC data and the main factors influencing the coverage – temperature and nutrients in the dam water. This was to obtain

the main characteristics that could be used in developing the prediction model. Thus, focus was placed on combining the available datasets and developing models to predict WHC.

Different machine learning (ML) models were compared to determine which produced the better results, but researching new ways to build prediction models or finding the optimal model type was outside of the scope.

Existing ways to clean and handle missing data were used to achieve the purpose of this research which was to develop models that predict WHC.

Algae was not considered in this research as it does not influence water hyacinth directly. Algae is influenced by nutrients in the water and in water hyacinth, so it was seen as a side-effect of water hyacinth.

This research also did not include plant hoppers. According to Coetzee et al. (2022), the plant hoppers are the most effective method of control the water hyacinth on the Hartbeespoort dam, when introduced often and at critical times. The model build in this research predicts the WHC so as interventions, such as biological control methods, can be planned (Gutiérrez et al., 2001). This way, the model predicts WHC without any plant hopper information, using the main factors that influence the coverage as these are sufficient to do so (Coetzee, 2022; Coetzee et al., 2022). Additionally, the biological agents effect during the time they were present on the dam, being from 2018, was reflected as a seasonal trend (Coetzee, 2022; Coetzee et al., 2022).

Unlike existing mathematical models such as those in Aoyama and Nishizaki (1993) and Wilson et al. (2005), this research included more than one influencing factor and made predictions of WHC based on multiple past datasets and not just estimates based on the mass of water hyacinth.

This research was limited to the following:

- Available WHC data
- Available temperature, TN, and TP data
- The dates of the available WHC data
- Building and testing four model types with the most potential for predicting coverage, identified in the literature review, to determine a more effective model.

1.5 Research Question

The main research question addressed in this study is: How can predictive modelling be used to predict the percent coverage of water hyacinth on the Hartbeespoort dam to improve the effectiveness of biological control?

1.6 Research Objectives

The main objective of the research was to **develop a model that predicts the percent coverage of water hyacinth on the Hartbeespoort dam**. This objective was achieved by meeting the following sub-objectives, which were to:

1. Investigate and cleanse the data obtained.

2. Handle missing data within these datasets and simulate datasets where necessary.
3. Identify features used in building the predictive models.
4. Identify criteria typically used to evaluate predictive models.
5. Identify predictive model algorithms with potential to predict WHC.
6. Build at least three predictive models.
7. Compare the built predictive models using the identified criteria.
8. Choose a predictive model and make final predictions.

1.7 Ethical Clearance

The WHC dataset was obtained from research done at Rhodes University. A letter of permission to use the data was obtained from Rhodes University and approved by the Wits Ethics Committee. The South African Weather Services (SAWS) provided the temperature datasets. A form permitting the use of the data was filled out and submitted to SAWS. All other datasets were obtained from open and free online sources. Thus, the Wits Ethics Committee issued an ethics waiver for all the data provided, valid until September 2024. The waiver letter number is HRECNMW21/10/02.

1.8 Dissertation Overview

The subsequent chapters of the research are as follows:

Chapter 2 contains the research background details to determine factors influencing WHC. A literature review was done in Chapter 3 to investigate methods of handling missing data. A second literature review covering the development of prediction models, including relevant research that uses machine learning models for similar data, is shown in Chapter 4. The following chapter, Chapter 5, describes the data obtained and details the preliminary analysis of each dataset. Thereafter, Chapter 6 handles the missing data through multiple imputation and then extrapolates the nutrient datasets to obtain data for the timeline required. The development of the models and results are presented in Chapters 7 and 8. Chapter 8 also discusses the findings of the final model. Finally, Chapter 9 provides a conclusion to the research as well as recommendations for improvements and future uses of the model.

Four appendices were also added to address the following:

Appendix A and B discuss, in detail, the method used to extrapolate the TP and TN datasets, respectively. The codes developed in MATLAB used to impute the missing data in all datasets is presented in Appendix C. Lastly, Appendix D contains the code used to make predictions using the ensemble model, generated in MATLAB.

2 FACTORS INFLUENCING WATER HYACINTH COVERAGE

Water hyacinth is a very invasive weed influenced by multiple environmental factors. Water hyacinth depends on these two factors: temperatures and nutrients (Sivappa, 2017; Aoyama & Nishizaki, 1993). Any plant needs nutrients to grow (Uchida & Silva, 2000). The most vital nutrients for plant growth are nitrogen and phosphate (PO_4).

The two main factors influencing WHC, temperatures and nutrients exposed to, were researched. This chapter provides literature on the nutrients and temperatures and how these factors affect WHC. The chapter explores the definition of water quality and what it entails, followed by relevant research on water hyacinth monitoring as well as favourable conditions for water hyacinth growth.

2.1 Nutrients in Water

Nutrients found in water are essential for plant growth and aquatic ecosystem survival (Uchida & Silva, 2000). However, nutrients can also enter water through human activities or natural disasters such as floods (Richards, 2020; Chen et al., 2021). A body of water is classified as eutrophic when it has a high concentration of nutrients above the limits set by the Department of Water Affairs and Forestry (Rossouw et al., 2008). Thus, measuring the concentrations of nutrients in the water determines eutrophication (Smith, 2009).

Total phosphorous (TP) is the principal nutrient considered when determining eutrophication, but total nitrogen (TN) can also be measured to deduce eutrophication (DWS, 2019; Smith, 2009). These levels show the potential for algal and plant productivity in water (DWS, 2019). Table 1 shows the different levels and the corresponding ranges of nutrients needed in a body of water.

Table 1: Levels of eutrophication (Bhusal & Devkota, 2020; DWS, 2019)

Level of Eutrophication	TP (mg/L)	TN (mg/L)
Oligotrophic	< 0.015	< 0.4
Mesotrophic	0.015 – 0.047	0.4 – 0.6
Eutrophic	0.047 – 0.130	0.6 – 1.5
Hypertrophic	> 0.130	> 1.5

Often, eutrophication is referenced in terms of the ratio TN to TP. In a study done by Sivappa (2017), lakes with a TN to TP ratio between 10 and 30 were assumed to have a balanced nutrient status and thus are oligotrophic (Water Resources Mission Area, 2019). This ratio is mainly used to determine which nutrient, TN or TP, is the limiting nutrient causing eutrophication and algal growth (Bergström, 2010; Ptacnik et al., 2010). In lakes, dams or even fishponds, this is used to determine which nutrient to add, or remove, to improve the eutrophication level of the water (Guildford & Hecky, 2000). So, if the ratio is high, TP is considered the limiting nutrient (Savic et al., 2022) and a compound containing nitrogen can be added to clear up algae (Guildford & Hecky, 2000).

Water quality is defined by several variables such as TP, TN, and other chemical, physical and biological variables (Omer, 2020; Ballance & Bartram, 1996). Thus, the Department of Water Affairs and Forestry has the National Eutrophication Monitoring Programme (NEMP) to record these nutrients for each body of water in South Africa (DWS, 2019). The department has placed limits to determine what constitutes drinkable water and to determine eutrophication levels.

The Hartbeespoort dam is hypertrophic due to wastewater and sewage effluent leaking into the dam via rivers and traces of mining effluent (Mitchelle et al., 2016). According to Carroll and Curtis (2021), at Hartbeespoort dam, the influx of TN and TP in 2010 increased at a higher rate than the outflux when compared to 2016. Thus, this excess amount of nutrients results in overgrowth in plants and other organisms, such as cyanobacteria, known as blue-green algae (Matthews, 2016; Yang et al., 2008). The repercussions of having plant overgrowth and cyanobacteria are diseases, undrinkable water, and death of other organisms in the ecosystem, such as fish (Mzuza et al., 2015; Rojas-Sandoval & Acevedo-Rodríguez, 2022). Thus, controlling water hyacinth is vital to good quality water. To do so successfully, Coetzee and Hill (2011) proved water quality played a much greater role in water hyacinth control than biological control agents.

2.1.1 Total phosphorous (TP)

TP is the sum of all phosphorous found in a body of water, normally phosphate (PO_4) only (Water Resources Mission Area, 2019; Huizenga et al., 2013; Jung et al., 2019). Therefore, TP is the same as PO_4 and, thus, is sometimes called total phosphate.

The Department of Water Affairs and Forestry states that the legal limit of PO_4 in effluent is 1 mg/L (Coetzee & Hill, 2011; DWS, 2019; Holmes, 1993). Drinkable water should have no more than 1.5 mg/L of PO4. Additionally, a body of water is eutrophic if it has PO_4 greater than 0.047 mg/L, so the legal limit falls in the eutrophic range.

According to Coetzee and Hill (2011), controlling the concentration of nutrients entering the water is the best way to control water hyacinth and cyanobacteria. This is because inorganic nitrogen and PO_4 are the factors with significant influence on water hyacinth and algae growth (Yang et al., 2008). The best way to control PO_4 is to lower the maximum allowed concentration, especially for dams communities are dependent on for drinking and agriculture, such as the Hartbeespoort dam (Kempster et al., 1997; Coetzee & Hill, 2011).

2.1.2 Total nitrogen (TN)

Similar to TP, TN is the sum of all the compounds containing nitrogen found in a body of water (Water Resources Mission Area, 2019; Ballance & Bartram, 1996). Huizenga et al. (2013) state this is normally NO_2, NO_3 and NH_4. For water to be drinkable, it should not have more than 12 mg/L of TN. However, this falls in the hypertrophic range seen in Table 1. Thus, legal nutrient limits are too high.

Additionally, nitrogen is used in fertilisers to promote plant growth, meaning it can easily enter bodies of water (Water Resources Mission Area, 2019). A benefit of this is that some countries harvest water hyacinth and turn it into fertiliser (Andika et al., 2016).

2.1.3 Relationship with water hyacinth

Other researchers also state that one of the factors influencing water hyacinth growth is water quality, specifically TN and TP concentrations in the water (Fernando, 2003; Rojas-Sandoval & Acevedo-Rodríguez, 2022). In the studied literature, TN and WHC have a relationship that is affected by seasonal changes (Carroll & Curtis, 2021; Meyer & Rossouw, 1993).

According to Aoyama and Nishizaki (1993), the concentrations of TN and TP in a body of water both have a hyperbolic relationship with water hyacinth growth rate. This means if there is a large concentration of nutrients in the water that the water hyacinth inhibits, the rate at which the water hyacinth will grow is also high. The researchers showed this relationship mathematically:

$$Y = C_{max} \cdot \frac{X}{K_s + X} \quad (1)$$

where C_{max} and K_s are constants, Y is the percent of TN or TP in the water hyacinth, and X is the concentration of TN or TP present in the water, both in mg/L. The values for C_{max} and K_s were 4.8 and 0.3, respectively.

Using Equation (1), the percent of TN or TP in the water hyacinth, given the other variables, can be calculated. However, there are no current mathematical relationships between nutrients in water and WHC directly. The researchers showed the mass of water hyacinth can be estimated by knowing the TN and TP present in the water hyacinth. Similarly, Wilson et al. (2005) used air temperature at a specific time to estimate the mass of water hyacinth.

This literature proved it difficult to calculate WHC, but estimating a relative growth rate or mass was possible. These estimations can be used to calculate coverage as the relative growth rate can be calculated per unit of area (Freedman & Harms, 2017; Wilson et al., 2005). However, these equations still require the mass of a sample of water hyacinth from the body of water researched. Milićević (2023) in an example of how exponential and logistic models were used to estimate water hyacinth growth, in terms of its biomass and under semi-controlled environments. Therefore, it is difficult to calculate future values of WHC using a mathematical equation.

In addition, water hyacinth can absorb PO_4, nitrogen, and other nutrients and harmful chemicals (Auchterlonie et al., 2021; Petrucio & Esteves, 2000). Thus, it can be used to clean water. However, it has a limit as to how much it can absorb. Once water hyacinth dies, it releases all the nutrients it has absorbed back into the water, thus feeding the existing seed banks (Sullivan & Wood, 2012). Hence the difficulty in controlling its growth and coverage.

2.2 Ambient Temperature

Ambient temperature is the overall temperature of the environment in which the experiment or observation is being made (De Paoli et al., 2021; Coetzee, 2012). Temperature has a considerable influence on WHC. Consequently, literature represents this relationship mathematically. Reddy

and DeBusk (1984) found a parabolic relationship between ambient temperatures and water hyacinth growth rate, while Imaoka and Teranishi (1988) found an exponential one depending on the range of temperatures. Therefore, it is evident ambient temperatures directly affect WHC.

Water hyacinth does not die easily due to temperature (Rojas-Sandoval & Acevedo-Rodríguez, 2022). Only extreme temperatures affect water hyacinth thus low and high temperatures should only be considered as separate factors if it is evident there is high variance due to extreme temperatures occurring often. Otherwise, Wilson et al. (2005) use daily average temperatures to examine the relationship with WHC. However, Miller et al. (2020) assume minimum temperatures (MinTemp) are the limiting factor in water hyacinth growth and are thus crucial. Figure 1 shows the monthly average temperatures at the Hartbeespoort dam.

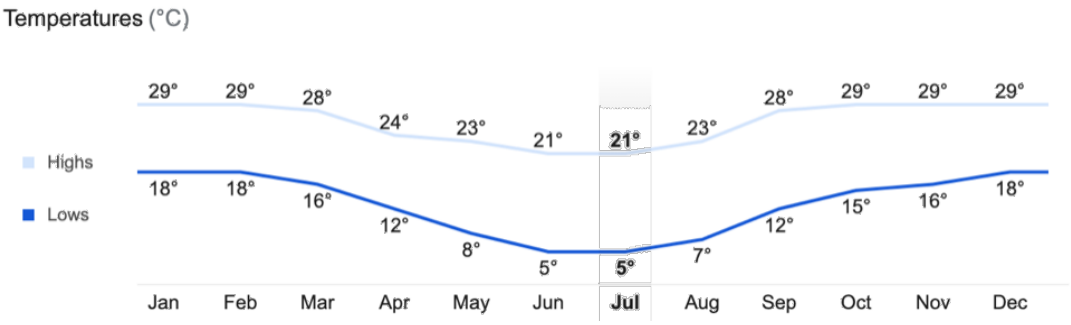


Figure 1: Monthly average temperatures at the Hartbeespoort dam (NCEI, 2024)

Hartbeespoort dam has a few weather stations nearby. Coetzee (2012) proved the temperatures from the weather station in Kosmos (station number 05125544) to be similar to the temperatures that the water hyacinth on the dam is exposed to. Hence, it was assumed to be the same.

2.3 Relevant Research

This section summarises relevant research on water hyacinth monitoring and recording whereby remote sensing and other methods and tools were used to produce WHC data in the past.

2.3.1 Remote sensing

Remote sensing is a method of scanning the earth to detect and monitor physical characteristics of interest, such as weed coverage on a body of water (USGS, 2024). Remote sensing is the most commonly used biological and geographical method of retrieving and recording weed coverage data (Chen et al., 2021). Satellite imagery distinguishes the different greenery in the image, resulting in a coverage of the specific greenery interested in. All research in this area, such as that done by Chen et al. (2021), used remote sensing and GIS to obtain WHC data, but these methods cannot predict future data.

Table 2 shows the different satellites available and used for these methods by numerous researchers.

Table 2: Satellites used in remote sensing and GIS research (Khandelwal et al., 2021; Landsat Missions, 2024; Kinsler, 2021a; ESA, 2022)

Satellite Name	Details
Sentinel-1	European Space Agency. Records images every 12 days.
Sentinel-2	European Space Agency. Records images every 2 to 5 days.
MODIS	NASA. Records images every 1 or 2 days.
Landsat-8	United States Geographical Survey. Records images every 16 days.

As shown in the table, satellites record images at different day intervals. Also, poor weather influences satellite images, so some are unusable. This means data retrieved using GIS and remote sensing has missing data.

2.3.2 Centre for Biological Control

The Centre for Biological Control (CBC) is an organisation based at Rhodes University that does research, tests and implements biological control methods for different weeds throughout South Africa (CBC, 2020b). With the help of volunteers, the CBC introduced plant hoppers to the Hartbeespoort dam in 2018, but due to lack of funding, the agents could not always be sustained (Coetzee, 2022; Coetzee et al., 2022). In 2020, many types of agents were reintroduced to the dam resulting in a very low WHC especially in the winter months (CBC, 2020).

In addition to introducing biological control agents to the Hartbeespoort dam, the organisation conducted research on mapping and recording WHC on the Hartbeespoort dam using spatial and remote sensing (Coetzee et al., 2022; Moffat et al. 2024). WHC data can be retrieved from an online Macrophyte Monitoring Tool, shown in Figure 2, built using the Earth Engine App (Kinsler, 2021a). The tool uses images from the Sentinel-1 and Sentinel-2 satellites.

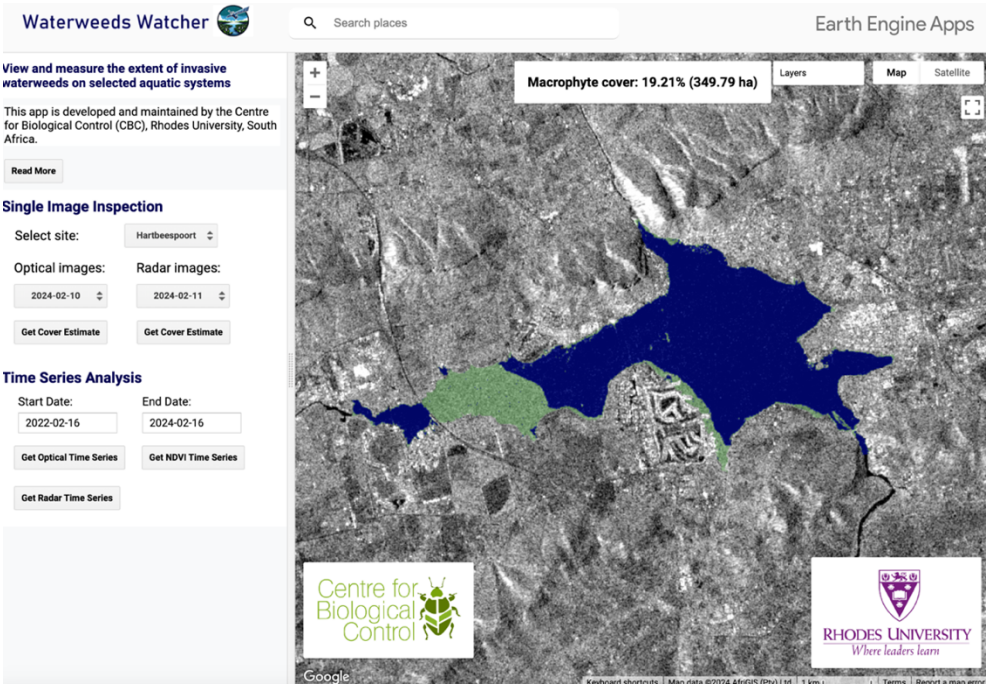


Figure 2: Macrophyte Monitoring Tool developed by David Kinsler (Kinsler, 2021a)

This research provided public access to WHC status on Hartbeespoort dam.

2.4 Favourable Conditions for Water Hyacinth Growth

Water hyacinth has conditions in which it thrives and is more favourable. On the contrary, there are conditions that kill water hyacinth if exposed to it for long periods. Water hyacinth originates in Brazil and as a result it favours freshwater, warm and humid temperatures, and growth rates change seasonally (Ghosh, 2010). Thus, the water hyacinth growth rate depends greatly on environmental conditions and is difficult to predict using traditional methods such as mathematical equations.

Table 3 shows favourable conditions for promoting water hyacinth growth, optimal growth rate as well as lethal conditions (Dersseh et al., 2019; Coetzee & Hill, 2011).

Table 3: Conditions that influence water hyacinth growth and coverage (Rojas-Sandoval & Acevedo-Rodríguez, 2022; Owens & Madsen, 1995)

Environmental Factor	Growth Range	Lethal Range	Favourable Range
TP (mg/L)	≥ 0.1	< 0.1	1.7 to 3
TN (mg/L)	≥ 0.6	< 0.6	5.5 to 20
Ambient Temperature (℃)	10 to 40	< 5 and > 40	28 to 30

When looking at Figure 1, it is evident that the conditions at the dam fall within the growth, and even favourable, ranges of WHC. Finally, Table 3 shows water hyacinth requires eutrophic ranges, in Table 1, for growth.

2.5 Chapter Summary

This chapter explained the link between WHC and water quality but first covering the definition of water quality in terms of the two main nutrients that influence WHC being TP and TN. Then, temperature and its effect on WHC was discussed. Literature showed there were mathematical relationships between water hyacinth and temperature, as well as water hyacinth and nutrients, that can be used to estimate the WHC in future periods, provided the mass of water hyacinth present on the dam is known. Other relevant research on obtaining WHC data through remote sensing methods was presented which included the work CBC has done in this regard. At the time of conducting this literature review, there were no known methods of predicting WHC percentage using ML. For this reason, predictive models and ML were researched. The chapter was closed off with more detail on favourable conditions for water hyacinth growth.

3 LITERATURE REVIEW: DATA HANDLING

Data handling refers to the methods involved in processing data, specifically data cleansing, analysis and visualisation (Grus, 2017). In this chapter, the steps for data handling are explained using relevant literature. This includes data analysis and visualisation, data cleansing, handling missing data and validating imputed data. Lastly, the simulation of data to extrapolate a dataset further is also discussed.

3.1 Data Analysis and Visualisation

The first step of data handling is to inspect the data. This can be done by analysing the data in more depth and through visualisation. There are different data analysis methods, described in Table 4.

Table 4: Different types of data analyses (Taherdoost, 2020)

Type	Description
Exploratory analysis	To uncover insights about the data.
Descriptive analysis	To understand the data in terms of what happened
Explanatory analysis	To understand the data in terms of why it happened
Predictive analysis	Using data to predict outcomes
Prescriptive analysis	To solve problems found

Makridakis et al. (2011) emphasised the importance of preliminary data analysis before predictive model building. Descriptive analysis, or statistics, provides a valuable preliminary analysis as it simplifies and summarises data, so it is intelligible (Igual & Seguí, 2017; Kroese et al., 2020). Descriptive statistics has three categories that group calculations and describe similar information about the dataset (Heiberger & Holland, 2015). These categories are (Wasserman, 2010):

- Measures of central tendency – includes the mean, median, and mode.
- Measures of variability – is variance, standard deviation, and coefficient of variance.
- Measures of asymmetry – describes the skewness and calculates the kurtosis.

Table 5 explains the different statistics in each category.

Table 5: Descriptive statistics definitions (Wasserman, 2010; Heiberger & Holland, 2015)

Calculation	Details
Mean	The average of the dataset.
Median	Middle data point of an ordered dataset; used to determine quartiles.
Mode	The data point that appears most often.
Variance	Describes the spread of data in terms of its deviation from the mean.
Standard Deviation	The square root of the variance.
Coefficient of variance	Used to compare variables. Calculated by: $\frac{standard\ deviation}{mean}$
Skewness	Measures asymmetry of a dataset in terms of the direction of its tail.
Kurtosis	Measures asymmetry of a dataset in terms of how wide its tail is; used to suggest a probability distribution and explains how often a data point falls near the centre or in its tails. If $= 3$, then it is similar to a normal distribution. If > 3, then there are many outliers.

The tails of a dataset are where the extreme values, such as outliers and seasonal peaks, are found – the wider the tail, the more of these points there are (Wasserman, 2010; Heiberger & Holland, 2015). It is important to do this preliminary data analysis so there is something to reference once predictions or inferences are made, or missing data is handled (Taylor, 2013). In addition, preliminary data analysis includes time series analyses which is useful for identifying trends and patterns required for predictions or simulations (Makridakis et al., 2011) – refer to Section 4.2.

3.1.1 Visualisation of the data

It is difficult to understand data even after calculating descriptive statistics. Representing the data visually through plotting graphs, is often used for interpreting what has been calculated. Several graphs exist however, research has shown the graphs in Table 6 to be useful for regression data (Afrifa-Yamoah et al., 2020; Heiberger & Holland, 2015; Kelleher et al., 2020).

Table 6: Summary of graphs used to visualise data (Montgomery et al., 2016; Theobald, 2017)

Graph or Plot	Description
Histogram	Shows data distribution shape and skewness.
Scatter plot	Shows relationships between two variables. It is best for continuous data.
Time series plot	To find trends, patterns, cycles, or seasonality.
Seasonal plot	Overlapping data from each season to find underlying patterns.

These graphs can also identify missing data points in a dataset that require handling.

3.2 Data Cleansing

Data cleansing involves the transformation of a dataset for it to enter a model (Kuhn & Johnson, 2016). The three main tasks of data cleansing are verifying correct data, transforming variables and dealing with outliers (Kelleher et al., 2020). Once data is cleansed, the data can be handled.

3.2.1 Verifying correct data

The data should always be checked for inconsistent data for the performance of the model to improve (Grus, 2017). In other words, it is crucial to ensure data is incorrupt and complete, so all datasets are consistent and of good quality (Russell, 2018). Data should be cleansed before proper and in-depth analysis can be done (Igual & Seguí, 2017). Inconsistencies include any formatting errors, spelling mistakes, and numbers or text in the wrong places. Any of these and other straightforward fixes are done is this step so that the data is in a suitable condition for visualisation and pre-analysis for errors such as missing data or outliers to be identified.

3.2.2 Transforming variables

Transforming variables refer to whether formats and labelling are correct. If not, the data needs to be labelled into numerical, date and/or categorical. Other transformations include standardising of data which can be useful for when comparing data of different scales. Standardising data means centring and scaling the data so that it has a standard deviation of 1, mean of 0 and a range from 0 to 1 (Kuhn & Johnson, 2016). This means standardisation transforms the data into a dataset with a normal distribution using Equation (2) (Kelleher et al., 2020):

$$z = \frac{x - \mu}{\sigma} \quad (2)$$

where z is the normalised data point, x is the original data point, μ is the population mean and σ is the population standard deviation. Some ML models, such as ANN, perform better when data is standardised.

3.2.3 Dealing with outliers

Identifying and replacing or removing outliers is key when it comes to cleaning data (Kelleher et al., 2020). This is because some ML algorithms are sensitive to outliers and thus perform better when there are none (Müller & Guido, 2016). A traditional method of calculating outliers is the inter-quartile range (IQR) method which uses the following equations:

$$upper\ outlier\ threshold = Q_1 - (threshold\ factor) * IQR \quad (3)$$

$$lower\ outlier\ threshold = Q_3 + (threshold\ factor) * IQR \quad (4)$$

where IQR is the difference between the first quartile, Q_1, and third quartile, Q_3. A threshold factor of 1.5 is common practice unless greater sensitivity is required (Kelleher et al., 2020). Any data points that are above the upper limit or below the lower limit are outliers. It is always better to replace outliers than remove them so that data points are not lost (Newman, 2014). An effective way to handle outliers is to replace each outlier with its nearest neighbour (Igual & Seguí, 2017).

3.3 Handling Missing Data

Missing data is when there is an observation but no recorded value. Little and Rubin (2002) categorised missing data according to three categories, namely Missing Completely at Random (MCAR), Missing at Random (MAR), and, Missing not at Random (MNAR). Table 7 compares these categories (Jakobsen et al., 2017; Mack et al., 2018; Marino et al., 2021; Newman, 2014)

Table 7: Categories of missing data (Cheng et al., 2019; Kang, 2013; Mohammed et al., 2021).

	MCAR	MAR	MNAR
Assumption of probability of missing values	The same for all cases of missing values.	Random but has a known reason.	Not random and has a known reason.
Relation or dependence of missing values	No relation between missing values and observations.	Different for all cases of missing values.	Missing values depend on the missing values.
Reason for missing values	Due to the design of the experiment or data collection method.	Due to the observations.	Due to the nature of research – mostly found in medical trial research.
Example	Remote sensor malfunction or poor-quality satellite image.	Men giving their weight while women tend not to.	A person drops out of a medical trial due to an allergic reaction.

Dong and Peng (2013) researched methods for handling missing data and discovered that missing data impacted the accuracy of predictions, because it potentially contains vital information that affects forecasts or other results. Some missing values are irrelevant. The rule of thumb is missing values are ignorable when the total percentage of missing data is less than 5% (Jakobsen et al., 2017). Nonetheless, Newman (2014) advised using all available data, as it is better to have more

information than a lack thereof. Therefore, all missing values should be handled rather than ignored or deleted (van Buuren, 2018; Kelleher et al., 2020).

On the other hand, inferences drawn from a dataset with more than 40% missing data are considered hypothetical because there is not enough data to confirm its reliability (Jakobsen et al., 2017; Marino et al., 2021). Thus, should a variable have a high percentage of missing data, the variable should be replaced if possible (Kelleher et al., 2020). As a result, researchers have tested many methods of handling missing data throughout the years (Dong & Peng, 2013; Mohammed et al., 2021). Next, a description of imputation is provided followed by multiple imputation. Lastly the implementation of imputation in relevant case studies is provided.

3.3.1 Imputation

The best method of handling MCAR data is through imputation as it is effective and reliable (Dinov, 2018; Kuhn & Johnson, 2016). Imputation replaces missing values with a reasonable value, such as a mean or mode (van Buuren, 2018). There are many imputation methods; Mohammed (2021) compared a few to determine which is best for handling missing data in a generic dataset with different percentages of missing data. This research showed that the imputation results were independent of the dataset or percent of missing data, meaning this research applies to any dataset with any percent of missing data within reason. The imputation methods compared were mean, median, k-Nearest Neighbour (KNN), sample, and Multiple Imputation using Chained Equations (MICE). Out of these, the best-performing imputation method was MICE. However, KNN imputation performed second best. Jakobsen et al. (2017) also used MICE for datasets with greater complexity, as it involves more computation, compensating for that complexity.

Other literature using KNN imputation, such as Batista and Monard (2002) and Gao et al. (2011), found this method performed consistently well regardless of the dataset or percent of missing data, supporting Mohammed (2021). Furthermore, KNN imputation performed well on small datasets with high dimensionality – a large number of features (Kuhn & Johnson, 2016). However, Mohammed (2021) showed multiple imputation methods, such as MICE, performed better than single-imputation methods, such as KNN imputation.

3.3.2 Multiple imputation methods

There are many different multiple imputation methods (van Buuren, 2018). This is because any imputation method can be made into a multiple imputation method, as it involves imputing a missing value with multiple probable values and then combining those probable values, using statistics, to obtain a final value (Cheng et al., 2019; Newman, 2014). Ekeu-wei et al. (2018) compared handling missing data methods to replace missing peak river discharge and found multiple imputation to be competent in filling in missing data usually obtained from satellites.

To impute a missing value multiple times, random samples of the data are taken based on the assumption that all samples have the same statistical weight (van Buuren, 2018). In general, the steps for multiple imputation, according to Jakobsen et al (2017), are as follows:

1. Randomly split the dataset into multiple subsets.
2. Impute the missing values found in each subset.
3. Repeat steps 1 and 2 so that there are different subsets.
4. Combine all the imputations for each missing value.

This method of imputation is popular across disciplines (Newman, 2014). Thus, Cheng et al. (2019) regarded multiple imputation as robust. Little and Rubin (2002) proved this method to be less biased than other methods. Consequently, Madley-Dowd et al. (2019) found multiple imputation to perform better on datasets with missing data percentage of 10% or greater. Similarly, in a study conducted by Dong and Peng (2013), multiple imputation performed well for datasets with 20%, 40% and 60% missing data. This study also showed that as the percent of missing data increased, more imputations were required for each missing value, with the number of imputations determined through trial and error.

A popular multiple imputation method is to use KNN (Kuhn & Johnson, 2016). This KNN method is simple as it replaces the missing data point with the 'k' nearest neighbour in the same way as before (see Section 4.4.4). The data is first split into groups and then the missing data in each group is imputed with its nearest neighbour within its group. This process is repeated multiple times meaning that each data point is imputed with a multiple different values. Hence, the missing values are replaced with the average of all the values it was imputed with during the multiple imputation process. A study done by Faisal and Tutz (2021) used this method as it is less intricate compared to other multiple imputation methods, such as MICE, but still yields accurate enough results for any situation due to its robustness.

3.3.3 Relevant case studies on handling missing data

Literature on the remote sensing field used the Senitile-2 satellite to obtain WHC data through GIS or remote sensing (Pádua et al., 2022; Coetzee et al., 2022; Cheng et al., 2019). These researchers handled missing data using geographic tools, and artificial intelligence when missing data was due to poor-quality satellite images.

In research using water quality datasets, single KNN imputation proved to generate the most accurate results (Yang, 2022). The method was quick and effective and thus was best suited for small water quality datasets, specifically a dataset of sulphate readings with 24% missing data. Other researchers used KNN imputation to impute missing meteorological data, such as temperature, and obtained good accuracy (Diouf et al., 2022; Ferrari & Ozaki, 2014). Lastly, Eskelson et al. (2009) conducted research in forestry inventory, where missing forest coverage data were imputed using KNN imputation successfully – it is common practice to use this method in the forestry industry.

3.4 Validating Imputations

Literature has different ways of validating imputed data, such as those summarised below (Li et al., 2015; Hoogland et al., 2020; Nguyen et al., 2017):

- Comparing descriptive statistics
- Comparing data graphically
- Hypothesis testing

Literature studied on each of these points are discussed in subsequent sections.

3.4.1 Comparing descriptive statistics

Nguyen et al. (2017) compared the descriptive statistics of the observed data to that of the imputed or predicted data to validate it. The main statistics these researchers compared were the mean, standard deviation, minimum, and maximum data points. The allowable differences depended on the industry and the problem. However, this information is often difficult to find, especially for WHC and other datasets in this research. It is standard practice to use 95% confidence levels, thus a 5% difference was considered acceptable (Kroese et al., 2020; Freedman et al., 2007).

3.4.2 Comparing data graphically

Comparing data graphically means plotting the two datasets on the same set of axes and seeing if they are similar in shape, range and other notable features. Literature plotted histograms and scatter plots (Nguyen et al., 2017; Kelleher et al., 2020; Ricci, 2005). Descriptions of these plots are made in Section 3.2.1.

3.4.3 Hypothesis testing

Hypothesis testing determines statistical significance, where the null hypothesis is the sceptical view being tested (Igual & Seguí, 2017). If the calculated p-value is less than the set alpha, depending on the test, then the view being tested is rejected, thus meaning there is substantial statistical significance. The p-value is the probability of rejecting the null hypothesis if it is, in fact, true (Freedman et al., 2007). It is standard practice to set alpha to 5% because, at this value, the acceptance interval is narrow enough to ensure the accuracy of the model is not compromised (Grinstead & Snell, 2006; Kroese et al., 2020). Both Law (2015) and Taha (2017) always choose a confidence level of 95%.

There are many hypothesis tests to choose from depending on the data and type of test required. To validate the imputed datasets, the Kolmogorov-Smirnov (KS) test was used as it is commonly used for that purpose (Nguyen et al., 2017). Additionally, the KS test is sensitive to location and shape and is, thus, advantageous for testing data where location and shape are important, such as data with seasonal influences as with the data in this research (Kenton, 2023). Thus, the test showed how likely the two datasets are drawn from the same unknown probability distribution (PD) (Kenton, 2023; Ricci, 2005). This is useful as it showed that the observed and imputed datasets came from the same PD and, thus, were statistically similar. If the calculated p-value was greater than 0.05, then the two datasets did belong to the same unknown PD and the closer to 1, the better the fit (Nguyen et al., 2017). However, the test only looks at one PD. For data with more than one PD, the KS p-value could wrongly reject the null hypothesis (Ricci, 2005). Another instance where this can occur is when there are a large number of observations in the dataset being tested. The extra data points could mean the sensitive KS test picks up small differences otherwise

insignificant and then rejects the null hypothesis even though the PD fits the observed data, generally.

There are other tests similar to the KS test such as the t-test which was used in cases where the KS test did not provide a useful result. A two-sample t-test determines whether two datasets come from a PD with the same mean and variance (Al-Kassab & Majeed, 2022; Curran, 2013). If the t-test results in a p-value is greater than 0.05, assuming 95% significance, then the two datasets cannot be distinguished and come from the same source or population. This is a useful test to do when the KS test cannot be used and is often used in cases where old processes need to be compared to new.

3.5 Extrapolating Datasets With Simulation

For some of the datasets in this study, data was available for a different time period than the required time period. This section reviews how regression data can be extrapolated by means of simulation. The steps involved are: fitting a PD, determining a theoretical PD, validating the PD with the goodness-of-fit test and using Monte Carlo simulations to generate new data.

3.5.1 Fitting a PD

A PD shows the possible values a variable can have, and how frequently those values can occur (Grinstead & Snell, 2006). This is useful for making inferences and assumptions about missing data, even in a different period. Before fitting a PD to a variable, one must be chosen and the types of PDs, namely discrete and continuous, must be understood. As the names suggest, discrete distributions are for discrete data which is categorical, while continuous for continuous or regression data. Only continuous PDs were considered because all the datasets in this research were continuous. Law (2015) and Ricci (2005) explained how to choose and fit a PD to a dataset with an unknown distribution. In summary, the steps are as follows:

1. Determine a theoretical distribution.
2. Fit the theoretical distribution.
3. Plot the theoretical distribution.
4. Determine the goodness-of-fit.

The following sections explain these steps in more detail.

3.5.2 Determining a theoretical PD

The steps to determine the best theoretical PD include selecting a theoretical PD, fitting the theoretical PD and plotting the theoretical PD. Thereafter, the goodness-of-fit of each PD can be evaluated.

Selecting a theoretical PD

Firstly, a distribution family is chosen, which depended on the type of data and the descriptive statistics of the dataset (Law, 2015; Kelleher et al., 2020). The three most notable distribution families are the exponential, Poisson, and normal distributions (Grinstead & Snell, 2006). Only the exponential and normal distributions were considered because Poisson is for discrete data

(Heumann et al., 2016). Although there are many other distributions, these families encompass the most popular distributions that tend to fit data – exponential distributions include the gamma and Weibull distributions, while the normal distribution family include the lognormal and logistic distributions. An additional distribution often used when distributions are similar to the normal PD is the Student's t Location Scale. MATLAB has the t Location-Scale PD, which is equivalent to the Student's t Location Scale (MATLAB, 2024).

The descriptions and possible applications of each distribution function are then used to suggest a distribution family (Ricci, 2005). This is because some distributions have specific applications; exponential is used to determine waiting time (Heumann et al., 2016). According to Law (2015), a Weibull distribution is a good starting PD, especially for datasets with missing data, when following a trial-and-error approach. However, descriptive statistics is the better method of choosing a theoretical PD. Table 8 shows the statistics for each PD family.

Table 8: Summary of conditions required for considering a distribution (Law, 2015; Ricci, 2005; Kissell & Poserina, 2017)

PD Family	Condition
Normal • Logistic	Population mean must be almost equal to the median Skewness = 0 if between -0.5 and 0.5 then considered normal Kurtosis = 3 or close to 3 if Kurtosis is high, but looks like Normal then it is logistic
• Lognormal	Coefficient of Variance ≥ 0.3 Skewness > 0
Exponential	Coefficient of Variance = 1 or close to 1 Skewness > 0 but close to 2
• Gamma • Weibull	Skewness > 0

It is best to choose more than one PD and then fit those to the distribution so that the PD that represents the dataset best can be chosen.

Fitting a theoretical PD

First, the parameters of each chosen PD are calculated to fit them to the dataset. Calculating these parameters can involve complex mathematics and statistics (Ricci, 2005). However, MATLAB has a distribution fitter app which allows the user to choose from several PDs and fit them to the data (MATLAB, 2006). The app calculates the parameters for each fitted distribution and makes plots to help identify which PD fits the best.

Plotting the theoretical PD

A histogram is the best graph to plot because it is a convenient way of seeing the shape of the data and comparing it to the fitted shape (Law, 2015; Ricci, 2005). If the graph shows more than one peak, this suggests it has more than one underlying distribution, so the data is separated into different distributions. Plotting a cumulative distribution function (CDF) is also helpful in determining whether the data belong to the fitted theoretical distribution (Heumann et al., 2016). The CDF is then used in simulating the dataset when using the Monte Carlo simulation method.

3.5.3 Validating a PD with the goodness-of-fit test

Goodness-of-fit refers to how well the theoretical PD or model fits or, in other words, represents the dataset. Hypothesis testing is the statistical approach to comparing how well a PD represents a dataset as explained in Section 3.4.3 (Freedman et al., 2007; Law, 2015).

Many different hypothesis tests exist to test various scenarios as shown in Section 3.4.3, with ones specific to goodness-of-fit. Law (2015) and Taylor (2013) use the Chi-Square to test goodness-of-fit for categorical data, meaning the range of the fitted distributions first need to be divided into consecutive intervals. However, Kuhn and Johnson (2016) advise against converting continuous data into categorical as it can negatively impact performance. The Anderson-Darling (AD) test is a better goodness-of-fit test for continuous data (Law, 2015; Kenton, 2023). This test can be done in MATLAB but does not work for comparing two datasets of unknown PD, as with the KS test (Kenton, 2023). In addition, Ricci (2005) stated that additional statistical calculations aid in determining goodness-of-fit. Such calculations are the log-likelihood, AD statistic, AD p-value and Akaike Information Criterion (AIC), which are explained in Table 9.

Table 9: Explanation of goodness-of-fit statistics (Kroese et al., 2020; Kenton, 2023).

Statistic	Explanation
Log-likelihood	Can be obtained when fitting the distribution to the dataset in MATLAB. It is used to calculate the AIC.
AD Statistic	A hypothesis test often used for testing goodness-of-fit of continuous data. The null hypothesis is that the data does fit the specified PD. It is used instead of other tests due to its sensitivity to extreme values and tails, thus being appropriate for datasets where extreme values are critical (seasonal points).
AD p-value	This is the p-value for the AD test. It should be as high as possible for the chosen PD to be a good fit. For 95% confidence, the null hypothesis is accepted at 0.05 or higher. If it is close to 1 then it is a very good fit.
AIC	Compares the quality of each fitted PD. The PD with the smallest AIC value is assumed to be the better-quality PD. This does not indicate whether the PD fits but can be used to determine which of the accepted PDs fits better. It is calculated using the log-likelihood of each PD using: $$AIC = 2k - 2 \times loglikelihood$$ where k is the number of parameters the PD has.

3.5.4 Generating new data with Monte Carlo simulation

Monte Carlo simulation was used by Meyer and Rossouw (1993) to simulate a 20% decrease in PO_4 to test its effect on the eutrophication level on Hartbeespoort dam. This showed that Monte Carlo simulation could accurately simulate a nutrient dataset and those simulations could be used to accurately predict or make deductions, such as eutrophication levels or WHC.

In the literature studied, Monte Carlo simulation was used in different problems such as temperature or sales demand (Gentle, 2009). It is a mathematical technique of experimentation that estimates a PD by simulating random numbers and takes risk, uncertainty and variability into account by producing a probabilistic forecast based on past data (Kroese et al., 2020). As Monte Carlo simulation is experimental, multiple experiments can be done using different scenarios to choose the best option. Research conducted by Tüzüntürk et al. (2015) is an example of this. The

researchers included seasonality by making different scenarios that adjust the intervals to or/and include seasonality in their simulations.

Monte Carlo simulation uses the PD of the dataset to make those probabilistic forecasts (Yıldırım, 2021). Once an appropriate theoretical PD has been determined, the cumulative distribution function (CDF) for each variable in the dataset can be built (Taha, 2017; Tüzüntürk et al., 2015). Thereafter, the following steps are done:

1. Create a range with fixed intervals for which the data are to be simulated – for example, from 0 to 10 minutes if looking at time.
2. Generate random numbers for those intervals using a random number generator such as the function in MS Excel.
3. Search for the CDF values that match the random numbers generated in step 2.

The values found for each random number form the simulated dataset.

4 LITERATURE REVIEW: PREDICTIVE MODELLING

This chapter explores predictive modelling, focusing on modelling methods and algorithms. It includes a discussion on traditional predictive modelling as well as ML, its types and methods.

4.1 Predictive Modelling

Predictive analytics is the method of interpreting data using mathematics, statistical algorithms and tools. It aims to forecast trends, predict patterns and identify relationships between variables, achievable through predictive modelling (Dinov, 2018; Rayner & Bolhuis, 2020).

Predictive modelling develops a model by taking known or historical data to predict new or future occurrences (Kuhn & Johnson, 2016). It aims to make predictions with the lowest possible error (Kelleher et al., 2020; Rayner & Bolhuis, 2020). This is achieved by analysing patterns, relationships and trends in the data and using this information to predict outcomes. Predictive modelling falls into either the traditional methods or ML categories, with the latter being the most popular (Makridakis et al., 2011).

Traditional methods include regression and time series analysis (Makridakis et al., 2011; Kelleher et al., 2020). Such methods are outdated and tedious but are still commonly used today (Igual & Seguí, 2017; Dinov, 2018). Firstly, regression analysis finds a line to fit all the data points as closely as possible, whereby a mathematical equation represents the line (Theobald, 2017; Mohri et al., 2018). This method predicts a continuous target variable, called a response or output variable, from independent variables, called predictors or features (Kroese et al., 2020). Time series analysis tracks observations over a period of time and has time as an independent variable (Montgomery et al., 2016). On the contrary, ML methods are current and convenient as they can be used in multiple disciplines and self-learn (Bonaccorso, 2018).

The most popular predictive modelling algorithms fall under traditional and ML methods, with most in the latter. According to multiple sources, these algorithms are Random Forests, Decision Trees, Support Vector Machines (SVM), Artificial Neural Networks (ANN), k-Nearest Neighbour (KNN), Linear Regression, and Logistic Regression (Bonaccorso, 2018; Kelleher et al., 2020; Theobald, 2017; Kroese et al., 2020).

Predictively modelling was used to fulfil the aim of predicting WHC percentage on the Hartbeespoort dam. ML was chosen specifically due to its popularity and diversity. Time series analysis and models were also considered as WHC is influenced by seasonal changes (see Chapter 2).

4.2 Time Series Analysis

A time series is any data or event measured or recorded over time, with time being the focus (Brockwell & Davis, 2016). Most often, time series data is recorded at fixed intervals, such as weekly, daily or hourly, but can also have irregular time intervals. Patterns within this data are made evident in time series plots, where time is plotted on the x-axis (Montgomery et al., 2016).

In addition, statistics are used to analyse time series and select a model, but first, patterns in the data need to be identified (Taylor, 2013). These different patterns are (Cryer & Chan, 2008; Makridakis et al., 2011):

- Horizontal – data fluctuating around a constant mean, otherwise known as stationary.
- Trend – a smooth pattern that persists for more than one year and in one direction, such as a positively increasing straight line.
- Seasonal – occurs frequently and at regular intervals within one year. For example, an increase during the summer months.
- Cyclical – a repeated pattern that occurs frequently over many years. For example, a combination of a seasonal pattern and a trend.
- Random – cannot be classified as one of the other patterns.

If any patterns are identified the data is considered to have seasonality.

4.2.1 Seasonality

The data in this research was treated as seasonal as WHC is influenced by seasonal changes. Seasonality refers to those changes that are predictable and reoccurring (Brockwell & Davis, 2016). In other words, the data is not stationary and exhibits a pattern. However, the dataset must be stationary to forecast data using some models, such as an autoregression (AR) model.

In some cases, a dataset can have an underlying seasonality, only seen when plotting data from each season over each other, called a seasonal plot (Makridakis et al., 2011; Montgomery et al., 2016). When this seasonality is removed, the dataset becomes stationary (Brockwell & Davis, 2016). Normally, seasonality is removed by calculating the difference between a data point and its corresponding data point one season before so that the difference between the dataset and its lagged data becomes the new stationary data (Dinov, 2018). Lagged data refers to a dataset that is moved forward the same number of times as it is 'lagged'; if 'lagged' once, the first data point is left out to form the new dataset (Montgomery et al., 2016). Time series statistics determine the number of times a dataset should be 'lagged'.

4.2.2 Autocorrelation

Autocorrelation is a time series statistic helpful in determining the extent of the linear relationship between two variables, similar to correlation (Montgomery et al., 2016). The difference between the two is that for autocorrelation, the second variable is the first variable lagged by a specific time period. However, data gets lost when doing this, so if there is limited data, it becomes impossible to lag it multiple times (Makridakis et al., 2011). Nonetheless, the point of lagging data is to compare the time series to itself and then plot correlograms, known as autocorrelation function (ACF) and partial autocorrelation function (PACF), to check for seasonality and other patterns (Brockwell & Davis, 2016). A correlogram, such as Figure 3, has the number of lags on the x-axis and the calculated partial autocorrelation or autocorrelation on the y-axis.

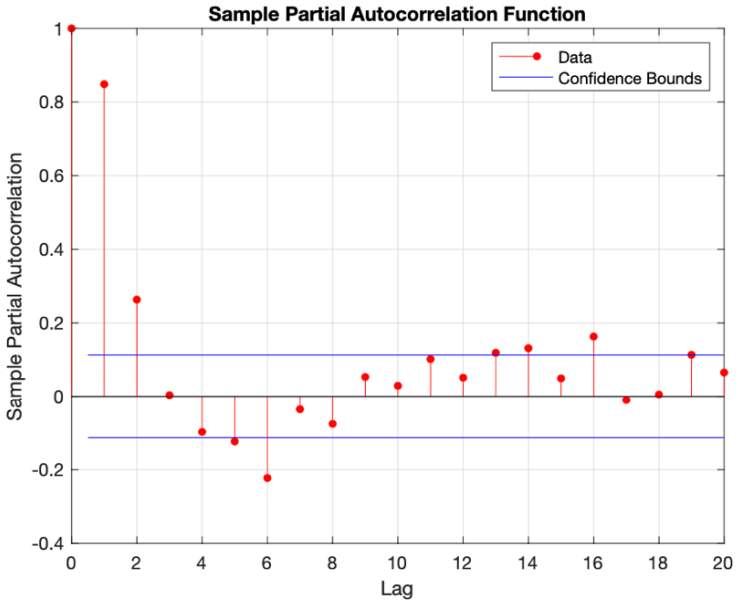


Figure 3: An example of a Partial Autocorrelation Function graph (MATLAB, 2018)

A lag is significant when it falls outside of the confidence bounds, blue lines shown in Figure 3. However, autocorrelations can be difficult to interpret. Makridakis et al. (2011) suggested applying transformations, such as taking the square root or natural logarithm of each data point, to the dataset so that results are interpretable. In addition, the intervals must then have the same transformation (Kelleher et al., 2020).

4.2.3 Time series forecasting models

Forecasting quantitative data means data is predicted based on the patterns seen in the past (Pourahmadi, 2001). Hence, continuity is assumed, meaning the patterns seen in the past are assumed to continue into the future. Once these patterns are found, a model fitting the data is chosen (Montgomery et al., 2016). According to Makridakis et al. (2011), the descriptive statistics and patterns present in the data dictate which model to choose. Autocorrelation can also determine a forecasting model as an ACF and PACF show the direct and indirect effects a lag has on the original data, respectively (Montgomery et al., 2016). An AR model is used if the PACF tends to zero (Taylor, 2013). The studied literature states that an autoregressive integrated moving average (ARIMA) model is good to use for any problem as it combines both moving average (MA) and AR models (Dinov, 2018; Montgomery et al., 2016; Pourahmadi, 2001). The peaks of the PACF show the number of lags that have a significant effect and, thus, suggest the number of lags required for the AR part of the model, while the ACF is that for the MA part of the model. MATLAB has an econometric modeller app that plots ACFs and PACFs, fits the models to the data and calculates their performance (MATLAB, 2018).

4.2.4 Evaluating time series model performance

Once a model is chosen and fitted, its performance, or forecast accuracy, is evaluated by doing a goodness-of-fit test, which tests how well the model can reproduce past data (Makridakis et al.,

2011). The best way is to calculate the error – the difference between past and future data for the same time period (Cryer & Chan, 2008).

Pourahmadi (2001) clearly showed forecasts become unreliable the further away they get from the original dataset because, for instance, only one 'lag' into the future is accurately forecasted when one lag is used to remove seasonality.

4.2.5 Time series models and analysis used in relevant literature

Mouta et al. (2023) combined remote sensing, time series and ML algorithms to map water hyacinth in Portugal. Firstly, the researchers used time series and remote sensing images, from the Sentinel-2, to determine spectral information of the water hyacinth and then used classification ML algorithms to map and predict the spatial distribution of water hyacinth in the examined area. They concluded that including other environmental factors, such as temperature and soil occupation, will ensure a deeper understanding of WHC, thus improving prediction and control. Similar research was done by Pádua et al. (2022), whereby the researchers compared ML algorithms to GIS and remote sensing methods. Although they proved the ML approach was less effective, the predictions were not actual WHC values. Instead, the study predicted potential sites where water hyacinth could be present.

The studied literature, such as Mukarugwiro et al. (2020) and Pádua et al. (2022), did not apply time series forecasting models. This is because, for any plant coverage, remote sensing is the primary method of mapping hence, GIS or image recognition algorithms are the standard practice methods. Nonetheless, the ARIMA time series model has been used to predict the population decline of a fish species in Lake Tana, Ethiopia, while considering the effects of water hyacinth (Yesuf et al., 2023).

4.3 Machine Learning (ML)

ML is an algorithm and predictive modelling tool that fits the data as best as possible for predictions to be made after it has self-learnt details about the data (Bonaccorso, 2018; Raschka, 2015). ML always consists of data and a model based on an algorithm with parameters, and its development follows the following steps (Müller & Guido, 2016; Theobald, 2017):

1. Collect data – determine data needed and collect it ensuring the data is reliable for accurate predictions.
2. Pre-processing data – includes data cleansing, handling missing data, visualising, and analysing the data.
3. Choose an ML model – determining which algorithm/s should be used based on the data, its type and the aim of the model.
4. Train model
5. Evaluate model – involves determining the performance of the by testing it and looking at its accuracy, then verifying and validating it.

6. Hyperparameter tuning – changing the parameters of the model to optimise its performance.
7. Making final predictions on unseen data

ML has three categories of algorithms called supervised learning, unsupervised learning, and reinforced learning (Bonaccorso, 2018; Raschka, 2015). Table 10 summarises the differences between the three categories.

Table 10: Types of ML algorithms (Igual & Seguí, 2017; Russell, 2018).

Category	Description
Supervised learning	Algorithms learn from a training dataset with labels. It has the correct answers for the output to learn from and will know if it is wrong when it makes the prediction.
Unsupervised learning	Algorithms analyse unlabelled data to group similar data points together by making predictions and then looking at the predefined output labels. It is mainly used for pattern recognition.
Reinforced learning	Algorithms make predictions by trial and error. After each trial, it gets a reward if it predicts correctly or a penalty if it predicts incorrectly. This is used for more complex problems.

Given that the aim of the research was to predict WHC, supervised learning was used. There are several types of problems that supervised ML can be used to solve (Kelleher et al., 2020). Table 11 provides a short explanation of each type.

Table 11: Types of problems solved using supervised learning algorithms (Theobald, 2017; Kuhn & Johnson, 2016; Kroese et al., 2020).

Problem Type	Explanation
Regression	Problems require predictions to be numerical values.
Classification	Problems need to classify data into different classes.
Time-Series Forecasting	Problems need to predict a number based on time series data.

The data in the research was all numerical and continuous data – refer to Chapter 5. Thus, the problem addressed, predicting WHC on the Hartbeespoort dam, was a regression problem. Next, supervised regression learning algorithms are discussed.

4.4 ML Algorithms

According to Kuhn and Johnson (2016) and Theobald (2017), the most used ML algorithms for a supervised learning regression problem are:

- Artificial Neural Networks (ANN)
- Support Vector Machines (SVM)
- Random Forests also called ensemble
- k-Nearest Neighbour (KNN)

ANN, SVM, KNN and ensemble algorithms were continuously used in multiple literature for predicting plant coverage or biomass, thus supporting the above-listed algorithms (Liu et al., 2023; van Klompenburg et al., 2020; Bayable et al., 2023). According to van Klompenburg et al.

(2020), ML is a dominant tool for predicting crop yields. The researchers identified ANN as being the most popular algorithm in this field. Additionally, temperature and nutrients were the most common features used for crop yield predictions.

Bayable et al. (2023) used ML algorithms to detect water hyacinth in a body of water and to estimate its seasonal spatial coverage. The algorithms tested in this study were SVM, random forest and classification and regression trees. Similarly to other literature studied, this study only made estimations of, or mapped, past water hyacinth coverage (Mouta et al., 2023; Pádua et al., 2022).

4.4.1 Artificial neural network (ANN)

ANN searches for correlations in data (Igual & Seguí, 2017; Theobald, 2017). Dinov (2018) considers it a black-box ML algorithm as it can solve almost any problem. In short, ANN mimics how the human brain works; it simulates how the brain uses networks of neurons to make decisions. The model is trained using artificial neurons that pass through the different layers of the model until the output is generated (Bonaccorso, 2018).

Each ANN model consists of input layers containing the input data called predictors or features, hidden layers and neurons and finally, output layers containing the output which is the predictions (Theobald, 2017). Figure 4 shows the structure of a simple ANN model.

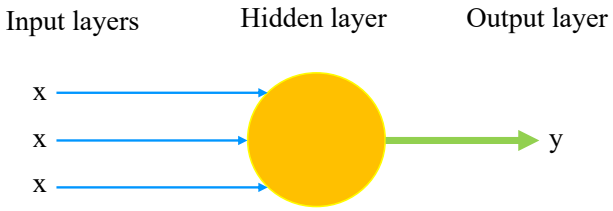


Figure 4: Structure of ANN model (own interpretation)

If an ANN model consists of more than three hidden layers, it is a deep learning model as it is more complex (Müller & Guido, 2016). Each part of the ANN model structure has parameters:

- The number of hidden layers and hidden neurons – the total number of hidden neurons is the same as the total number of features and outputs (Kroese et al., 2020). Rachmatullah et al. (2021) researched a few methods of determining the number of hidden layers and hidden neurons, such as the Sartori Method, but these methods are outdated. The researchers thus suggest using trial and error to determine the size of the hidden layer, as this has been the preferred method since 2017. Furthermore, in a study predicting forest biomass, an ANN with a single hidden layer was used (Liu et al., 2023).
- Weight of features – a weight is associated with each feature based on the features' importance; the training algorithm determines the weights (Dinov, 2018). The activation function of the model is found in the hidden layer and transforms the weights and inputs into an output for neurons to pass through each layer (Raschka, 2015). This function can be linear or non-linear and activates a neuron only if its output, the sum of the inputs and

their weights multiplied together, is greater than the predetermined threshold - a neuron may only pass on to the next layer when it is activated (Kelleher et al., 2020).

These parameters can be adjusted to optimise the model and make predictions with minimum error (Müller & Guido, 2016). To build an ANN model, Kuhn and Johnson (2016) and Dinov (2018) use the default values for the parameters and then manually adjust the parameters later to optimise the model; this is called hyperparameter tuning. Furthermore, ANNs work best with homogenous data, meaning their features need to be on a similar scale, meaning features should be standardised (Kelleher et al., 2020; Müller & Guido, 2016).

4.4.2 Support vector machine (SVM)

SVM is an algorithm based on statistics that determines a boundary line between data points of different classes (Igual & Seguí, 2017; Mohri et al., 2018). The algorithm aims to find a boundary that splits the classes as far as possible while still ensuring the distance is the same on both sides of the boundary line, called hyperplanes, and thus classify points (James et al., 2017). Figure 5 shows the structure of an SVM model and its different parts.

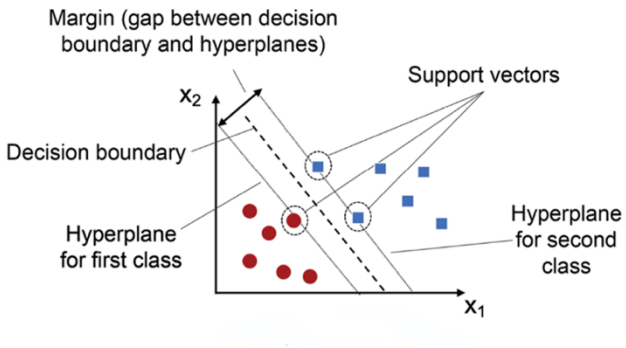


Figure 5: Labelled structure of an SVM model (Tibrewal, 2023).

As seen in Figure 5, support vectors are the points closest to the hyperplane (James et al., 2017). These points influence the position and orientation of the hyperplane (Müller & Guido, 2016). Data is plotted in n-dimensional space to split the data by a line (Mohri et al., 2018). The number of features determines the number of dimensions (Bonaccorso, 2018). In other words, should the data be non-linear, the data is converted to a higher dimension using quadratic equations to find a line. These quadratic equations are called kernels and when dimensions are converted, it is known as the kernel trick (Mohri et al., 2018). The kernel trick reduces features as the number of dimensions is equal to the number of features.

Furthermore, the kernel function depends on the data type (Kuhn & Johnson, 2016). Steinwart and Scovel (2005) tested different kernels on a tall dataset and found the Gaussian kernel function to be the most effective; tall data refers to data with more observations or data points than features. In addition, Rahman (2012) proved SVM to work well for small datasets, as it is not sensitive to outliers.

Like ANN, Dinov (2018) considers it a black-box algorithm, as well, because it is versatile. It is also powerful and flexible, thus resulting in reliable predictions (Kuhn & Johnson, 2016). For these reasons, SVM is often used for regression problems (Mohri et al., 2018).

4.4.3 Random forest

Random forest is an ensemble learning model, meaning it combines multiple algorithms to increase the robustness of the model as well as combat overfitting; in the case of a random forest, it is an ensemble of multiple decision trees (Igual & Seguí, 2017). In short, decision trees split data into branches of the possible outcomes of decisions based on past decision outcomes (Kroese et al., 2020). An example of a simple decision tree is shown in Figure 6.

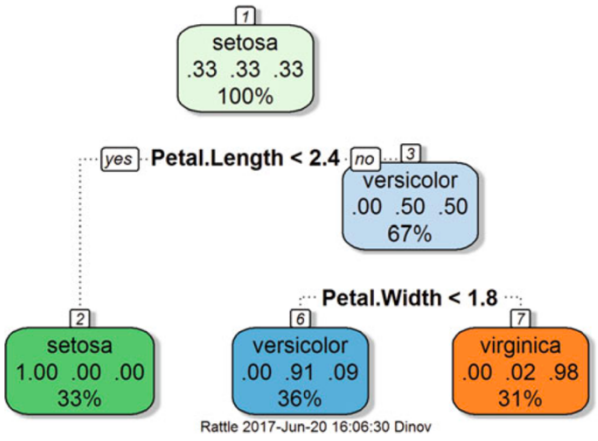


Figure 6: An example of a simple decision tree shown graphically (Dinov, 2018)

The outcomes can be binary when there are only two outcomes, generally categorical (Kroese et al., 2020). Nonetheless, decision trees, and thus random forests, can also be used for regression problems such as in Figure 6. The data must fit into a range the machine sets and learns for it to make decisions for a regression problem (Theobald, 2017). For instance, data points less than x, where x is a predetermined value dependent on the problem, are put in one branch, while those greater than are put in another. This is done until the machine cannot split or place the data into a branch.

Decision trees tend to overfit due to their high variance (Raschka, 2015; Müller & Guido, 2016). To prevent overfitting, model parameters, such as the maximum depth of the tree, can be set; this method is called pruning. However, the more effective method is combining many decision trees and creating an ensemble (Kelleher et al., 2020; Theobald, 2017).

A random forest takes a bootstrap sample, meaning it repeatedly draws a random sample, with replacement, the same number of times as there are data points (Müller & Guido, 2016). Then, a tree is built on each sample set and makes predictions. In a random forest used for regression problems, the outcome of each decision tree is averaged to obtain the final outcome of the model (Bonaccorso, 2018). The greatest advantage of using a random forest is that the model is less sensitive to the training data due to the multiple trees, thus improving predictions (Kroese et al.,

2020). This is achieved by ensuring each tree is different, with even different features thus, features with different significance are beneficial (Müller & Guido, 2016).

Ensembles can combine other types of models but in general, the ensemble has two algorithms: bagging and boosting (Dinov, 2018; Kroese et al., 2020). A random forest is a case of bagging with a decision tree learning algorithm (Raschka, 2015). Additionally, Roy (2021) proved that, for forecasting vegetation, ensemble algorithms, specifically random forest, were the best-performing algorithms compared to SVM and other regressions.

4.4.4 k-Nearest neighbour (KNN)

The KNN algorithm is easy to understand but effective and can be used on most classification and regression problems with acceptable accuracy (Kelleher et al., 2020). The algorithm determines the output by looking at the number of nearest neighbours (k) values and calculating it based on those values (Bonaccorso, 2018). A distance function determines which data points are the 'nearest neighbours' and the Euclidean distance formula is the most commonly used function for continuous data, according to Hu et al. (2016).

The k affects the performance of the model because if one more 'nearest neighbour' was considered when classifying a point, it could change the class it belongs to (Kelleher et al., 2020). Mohammad et al. (2014) used the rule of thumb for choosing the k:

$$k = \sqrt{n} \tag{5}$$

where n is the total number of data points in the dataset.

4.5 Feature Selection

As the name suggests, feature selection is the process of choosing features. In most cases, datasets contain a lot of data with columns of information that do not help predict the output variable (Kelleher et al., 2020). Irrelevant data can influence what patterns the model learns, confusing the model and influencing the predictions and accuracy. It is for this reason choosing good features, with relevant information, is vital to ensuring high accuracy. In addition, Marini (2009) also proved that reducing the number of features can increase training speed.

There are many methods for selecting features. Selecting those with high importance is an effective method (Freedman et al., 2007). The predictive importance of a feature is determined by performing an F-test and calculating a p-value for each feature (Kelleher et al., 2020). The F-test tests the significance of features or even models, and the closer the p-value is to zero, the more significant the feature is (Kroese et al., 2020). All features with a p-value less than 0.05 are significant and should be selected. MATLAB has an F-test function which calculates the p-values required (MATLAB, 2017).

Another method for selecting features involves correlation calculations (Igual & Seguí, 2017; Kelleher et al., 2020). The most used correlation calculation is the correlation coefficient, known as Pearson's correlation, which is the degree of linear relationship between two variables (Grinstead & Snell, 2006; Heumann et al., 2016). This number is between -1 and +1, meaning

interpretations are effortless; the closer it is to zero, the weaker the linear relationship between the two variables is, while the direction of the relationship depends on whether the number is positive or negative. A values greater than 0.7 is highly correlated (Nguyen et al., 2017; Raschka, 2015).

The main purpose of feature selection is to ensure that the chosen features do not have the same underlying information, which correlation can determine (Müller & Guido, 2016). If the features of a model and the output variable are both numerical, Pearson's correlation can be used to select features (Taylor, 2013). Pearson's correlation is easier to interpret than other correlations due to its range of -1 to 1 (Grinstead & Snell, 2006). Features that are closely correlated measure the same underlying information that can slow and mislead training (Kuhn & Johnson, 2016). This is an effective way to determine redundant features and remove them. However, Igual and Seguí (2017), and Kubát (2017) stated combining or transforming redundant features, rather than removing them, could result in a feature the model can learn patterns from (see Section 4.5.1).

A more time-consuming but effective method is to use hyperparameter tuning. Hyperparameter tuning can help with feature selection and improving the model (Müller & Guido, 2016; Theobald, 2017). This method uses iterative feature selection which builds a few models using different combinations of features and chooses the model and features with the best performance. This is not to be confused with the kernel trick that is used to reduce the number of features for SVM.

4.5.1 Finding new features

According to Igual and Seguí (2017) and Kubát (2017), it is better to produce or calculate new features using the existing ones rather than deleting similar ones. New features can be found by analysing the relationships between features, using correlation calculations (Kelleher et al., 2020). For example, two closely correlated features can be made into a ratio, where one is divided by the other, similar to the kernel trick, instead of randomly removing one. A scatter plot can also help identify relationships or patterns that can be made into a feature (Kelleher et al., 2020).

4.6 Evaluating ML Models Performance

To evaluate an ML model, the performance of the model must be calculated. Next, the model should be verified and validated for predictions to be reliable and usable. This section discusses the methods for calculating performance and determining the verification and validity of a model.

4.6.1 Calculating performance

The performance of a model can be evaluated in different ways depending on the type of problem. For classification problems, the accuracy of the model, or misclassification, is used to calculate performance (Mohri et al., 2018). The problem addressed in this research was a regression problem and, thus, the best way to evaluate performance was by calculating the mean square error (MSE) (Müller & Guido, 2016). MSE is calculated using the following equation:

$$MSE = \frac{1}{n}\sum(predicted\ mean - \ population\ mean) \quad (6)$$

Kuhn and Johnson (2016) suggest calculating the RMSE as well as the correlation coefficient between the observed and predicted data. RMSE is the square root of the calculated MSE, and the correlation coefficient shows how well the model fits or represents the observed dataset, similar to a goodness-of-fit test, by calculating how much the two correlate (Makridakis et al., 2011; Ricci, 2005). According to Liemohn et al. (2021), RMSE cannot sufficiently validate a model alone, so additional methods exist. These researchers used histograms, statistical tests, efficiencies, and correlation.

When calculating the error of a model, it is necessary to calculate it on the validation dataset (Russell, 2018). Hence, cross-validation is powerful because it calculates the error multiple times during training. In this way, the model is optimised on the test dataset and verified once final testing on new data is complete. Calculating the error at different stages of model development can reveal errors, such as overfitting, thus verifying the model. For instance, if the validation error is lower than the training error, the model is probably overfitted – refer to Section 4.6.4 for a detailed explanation.

4.6.2 Verification and validation

The chosen model and the predictions made by the model must be verified and validated throughout model building. If not, predictions cannot be considered for reliable decision-making and thus, it is critical to understand what verification and validation are and how to achieve them. Verification determines whether the model meets its main objective/s, meaning the model is implemented correctly and without errors (Tran, 1999). Overfitting and underfitting are the two most common errors that can occur in ML model building. Hence, a model is verified if it meets the set objectives without errors, especially those errors identifiable through debugging and checking the code and method (Thacker et al., 2004).

Validation determines if the model meets its objective/s well and has high accuracy (Thacker et al., 2004). Hence, a model requires validation to ensure good training and consistent model accuracy. The performance of a model depends on industry requirements and determines whether a model is validated (Tran, 1999). It is standard to expect classification models to have an accuracy greater than 70% (Kuhn & Johnson, 2016). However, if the model is proved to represent the problem or observed data well, then the model is valid. To prove this, performance calculations are normally used (see Section 4.6.1.).

When building a model, the dataset is split into training and testing datasets which are used to train and validate the model, respectively (Bonaccorso, 2018). The recommendation is to split the dataset into a testing dataset, at 20%, and a training dataset, at 80% (Raschka, 2015; Oancea & Ciucu, 2014). The dataset splits can be adjusted to improve model performance, but 20:80 is sufficient for most problems. The training dataset is used to train the model during development, while the testing dataset is kept aside until the very end when the final model is tested. In some cases the dataset is split into three datasets being training, testing and unseen – a ratio of 70:15:15 is commonly used. This is done so that there is an extra dataset that can be used during

hyperparameter tuning (see Section 4.7). The model is first trained using the training dataset, then the parameters of the model are adjusted using the testing dataset, and finally the performance of the model is evaluated using the unseen dataset. The unseen dataset is kept aside and is only seen by the model for the first time once the final model has been development. Validation should be done throughout the development of the model (Kroese et al., 2020; Mohri et al., 2018). Cross-validation is the most effective method for ensuring this (Raschka, 2015).

4.6.3 Cross-validation

Cross-validation is a statistical method of evaluating the general performance of the model more than once (Müller & Guido, 2016). There are a few different ways to perform cross-validation. The most common method is the k-fold cross validation (Russell, 2018; Dinov, 2018; Kuhn & Johnson, 2016). K-fold cross-validation randomly splits the training dataset into 'k' number of subsets called folds, where the standard practice is to use 5 or 10 folds. The model is fitted to one of those subsets, trained, and then its performance is calculated (Kroese et al., 2020). This process is repeated for the following subset until all subsets have been used to train the same model that has the same parameters (Mohri et al., 2018; Raschka, 2015). The average of each calculated performance is the true performance of the model (Müller & Guido, 2016). Randomly splitting the training dataset ensures all folds are homogenous – all folds are a good statistical representation of the training set. The number of folds influences the bias and robustness of the model, meaning the more folds there are, the smaller the bias and the more robust the model is.

4.6.4 Overfitting and underfitting

Overfitting and underfitting are model errors affecting performance and ultimately the predictions. Table 12 lists the differences between overfitting and underfitting.

Table 12: Overfitting versus underfitting (Russell, 2018; Bonaccorso, 2018; Mohri et al., 2018)

Overfitting	Underfitting
High variance.	High bias.
Over-training the model so it does not perform well on new unseen data.	Over-simplifying the problem with too many assumptions.
Training error is high, so the model cannot catch the noise of the data.	Training error is low, so the model cannot catch the underlying logic of the data.

Overfitting is the biggest concern for all prediction models, as it is often seen in practice (Kuhn & Johnson, 2016).

4.7 Hyperparameter Tuning

Hyperparameter tuning is when the parameters of the model are altered iteratively to optimise or improve performance (Müller & Guido, 2016). There are no rules for choosing the best parameters, so trial and error are used. The grid search method of trial and error is common in literature, such as Müller and Guido (2016).

Examples of parameters of some ML models are the number of hidden layers in an ANN model, the k in a KNN model, the number of trees in an ensemble, and finally, the kernel of an SVM

model. Russel (2018) and Raschka (2015) suggest using the default parameters for the initial model building and then altering them one by one until performance has improved. Models should always be optimised on test data after the model has been trained and validated (Müller & Guido, 2016). When using the regression learner app in MATLAB, all fitted and trained models are optimisable by the Bayesian optimiser, which builds multiple models but changes a parameter each time and then chooses the one that performs the best (MATLAB, 2017).

4.8 Model Selection

Müller and Guido (2016) explain that the best practice when choosing a model is to build more than one model and choose one that performs best. However, other factors could be of greater importance in reaching the objective of the model (Kroese et al., 2020). Such factors include training speed, which is influenced by the complexity of the model (Kuhn & Johnson, 2016; Kubát, 2017). Often, performance evaluations, such as RMSE, are used to choose a model (Raschka, 2020; Müller & Guido, 2016).

Literature shows that error is used to choose the better model (Yang, 2022). However, the RMSE can be difficult to interpret or compare, especially if the expected RMSE are unknown. It is common knowledge that the closer the RMSE is to zero, the better the fit (Singh et al., 2024). In addition, an F-test can be used to select a model with the highest significance – a p-value close to zero (Kroese et al., 2020).

4.9 Application of ML Models to WHC

Although regression ML algorithms were not used to predict WHC, Abebe et al. (2023) used images from multiple satellites, remote sensing and image classification ML algorithms to map water hyacinths on Lake Tana in Ethiopia. The researchers analysed the relationships between water hyacinth and climate variability, such as the evaporation of water in the lake, by analysing trends by means of Pearson's correlation. However, this study did not predict WHC in future; instead it used algorithms to enhance poor quality images and handle missing images.

In another study, regression trees, including random forest algorithms, were used to predict the efficacy of the weevil as a biological control agent for controlling water hyacinth. Agjee et al. (2018) reported that the random forest algorithm performed the best with a validation accuracy range of 55% to 88%. The accuracy was measured as Pearson's correlation as this was easier to compare and make conclusions compared to RMSE, and could be calculated because the observed data was known.

5 DATA DESCRIPTION AND PRELIMINARY ANALYSIS

To achieve the objectives of this research, data on WHC on the Hartbeespoort dam was obtained, as well as data on the variables that influence WHC considered in this research. These are ambient temperature, TP, and TN. This chapter describes the datasets obtained and the results of the preliminary analysis including identified outliers.

5.1 Water hyacinth coverage (WHC)

WHC is the output variable that was predicted. A researcher conducting a study with the CBC at Rhodes University provided access to data of WHC on the Hartbeespoort dam (see Chapter 1). The full dataset provided has not been published to date, but summaries of the data were posted online and can be accessed on the CBC website (Coetzee et al., 2022; Kinsler, 2021a). The researcher stated that the data was about 97% accurate compared to real-time WHC (Kinsler, 2021). Also, the standard deviation compared to the observed was 2.84, which the researchers considered accurate enough to be used.

The dataset contained WHC in percentage, along with the date and other information useful to the researchers. The characteristics of the dataset, relevant to this research, are shown in Table 13. This data was the most recent data available at the time of data collection.

Table 13: Characteristics of the observed WHC dataset (Kinsler, 2021a)

Characteristic	Details
Total number of data points	387
Total number of missing data points	0
File type	spreadsheet
Data type	numerical
Units measured in	date and % coverage
Number of decimal places	4
Timeline	1 August 2015 to 27 March 2021

A preliminary analysis was done on the data to identify any patterns and outliers in the dataset, as well as have descriptive statistics available for comparisons later.

5.1.1 Data analysis of WHC

Figure 7 shows the time series plot of the WHC dataset.

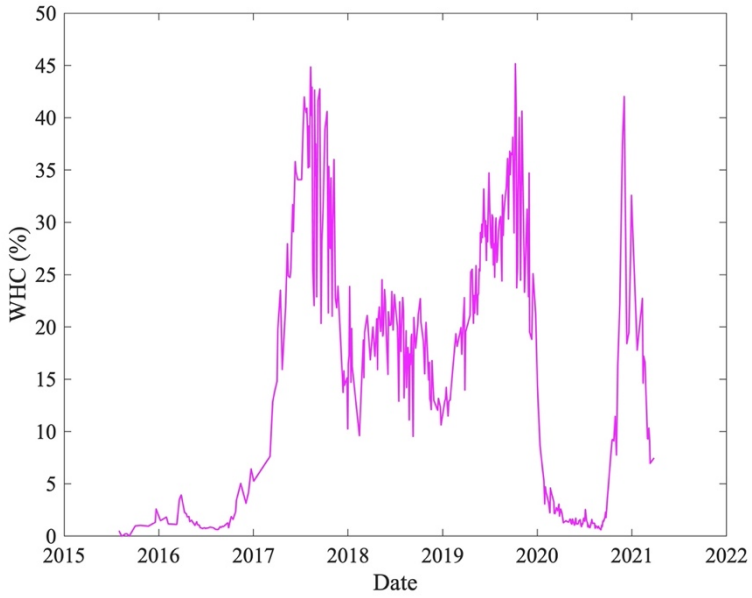
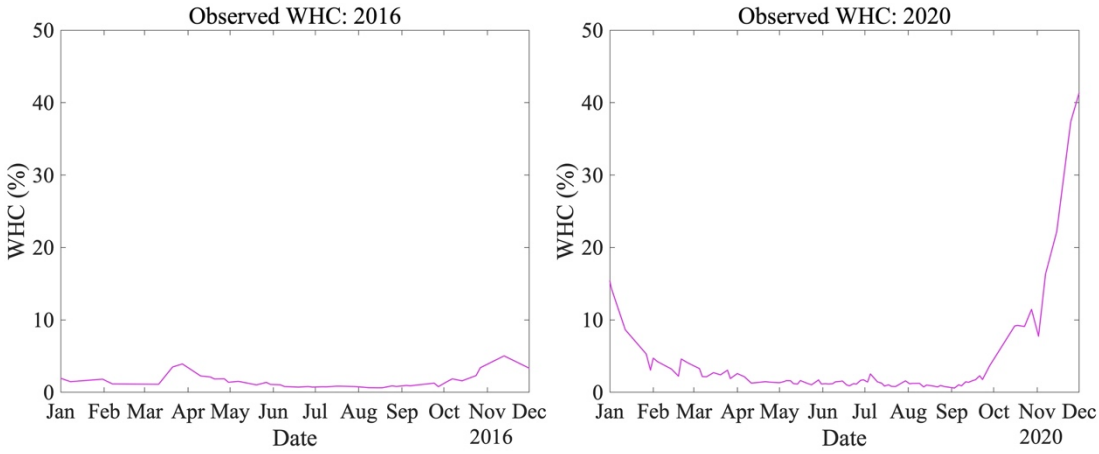


Figure 7: Time series plot of the observed WHC dataset

The time series plot shows increasing and decreasing trends every year and an apparent cycle occurring from the beginning of 2017 to the beginning of 2020. The plot suggested that the WHC data had seasonality. The seasonality was caused by environmental factors, mainly temperature and eutrophication which is determined by nutrients concentration levels (Smith, 2009). Figure 8 compares the years 2016 and 2020.



(a) (b)

Figure 8: WHC data in 2016 (a) compared to 2020 (b)

The low observed WHC in 2020 was due to the biological control efforts made by the Centre for Biological Control (CBC, 2020a). The WHC in 2020 and 2021 was also influenced by restrictions on human movement during the COVID-19 pandemic and did not reflect normal WHC levels. These reasons explain the volatility seen in 2020 compared to 2016. Therefore, data after 28 December 2019 were excluded from the study.

No outliers were apparent in Figure 7, and after calculating outliers using Equations (3) and (4), and a threshold factor of 1.5, no outliers were identified. Descriptive statistics were calculated to

obtain a better understanding of the data and to refer to after missing data was handled. Table 14 shows the calculated descriptive statistics of WHC.

Table 14: Descriptive statistics of observed WHC from 1 August 2015 to 28 December 2019

Standard Deviation	Mean	Median	Range	Kurtosis	Skewness
11.94	20.00	20.56	45.15	2.24	-0.11

The statistics show that the dataset cannot be described by a theoretical normal PD as the statistics are not equal to those outlined in Section 3.5.2 – the skewness is not equal to 0. The histogram in Figure 9 shows the PD of the dataset more clearly.

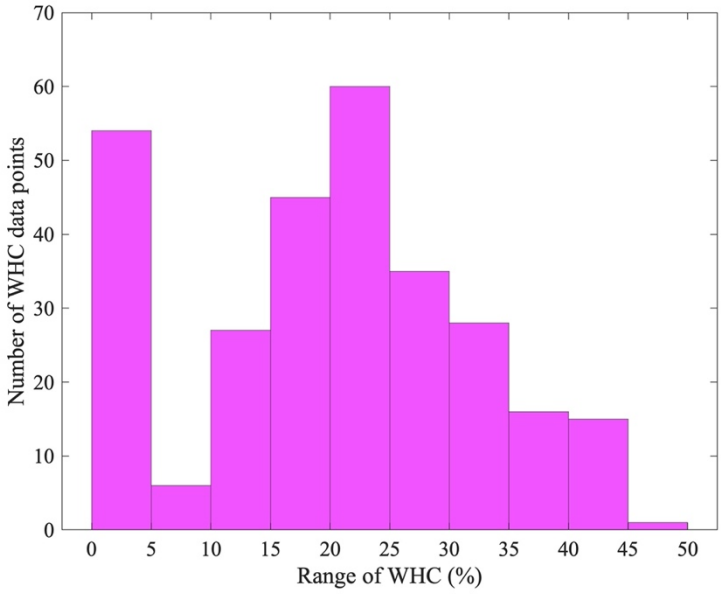


Figure 9: Histogram of observed WHC dataset

The figure shows that about 60 data points, out of 387, fall within the 20% to 25% WHC range and has two peaks, at 0% to 5% and 20% to 25%, insinuating two PDs.

Although there was not any missing data within the dataset given, the research required a more consistent interval and more data points from which the predictive model can learn. The number of data points recorded per month was counted to evaluate the frequency of the data. Figure 10 shows some of the number of data points per month, for January 2016 to January 2019.

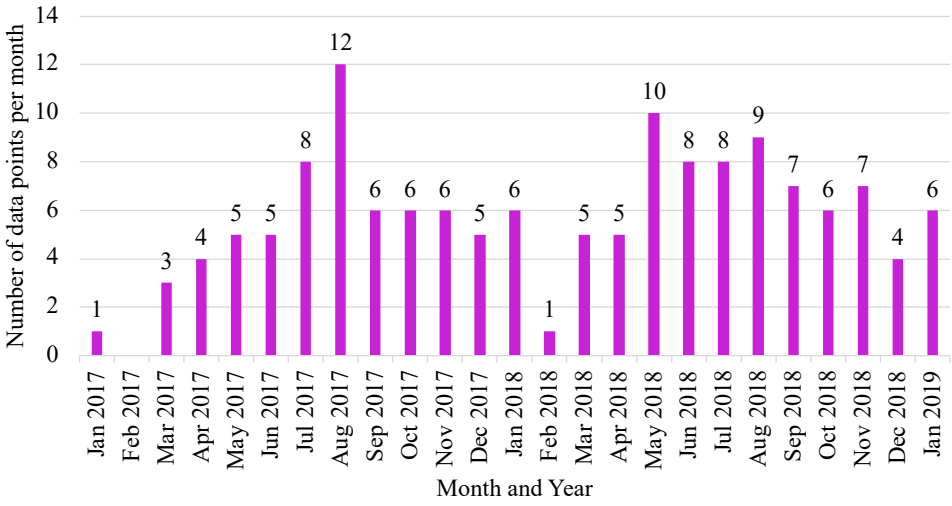


Figure 10: Number of WHC data points recorded per month

The figure shows the frequency of data points was not consistent as some months, such as February 2017, did not have any recorded data points, while others, such as August 2017, had 12. Therefore, a fixed time interval was required so that all datasets could be consistent and comparable. As WHC was the output variable, it was decided that the other datasets must match the WHC dates.

To determine the frequency of data points, the average time interval between recorded WHC dates was calculated. Five-day intervals were chosen because the average was 5.63. The satellite used, Sentinel-2, recorded data every five days, further justifying choosing a five-day interval (Khandelwal et al., 2021; Kinsler, 2021a; ESA, 2022). Dates were interpolated every five days from 1 August 2015 until the last date in its dataset and its corresponding WHC value was filled with the term "NaN", which is often used to represent unknown data values. A short extract of the list of desired dates is shown in Table 15.

Table 15: Extract of WHC dataset and the list of dates required (Kinsler, 2021a)

Row Number	Date	WHC (%)
1	2015/08/01	0.5137
2	2015/08/06	NaN
3	2015/08/11	0.0060
4	2015/08/16	NaN
5	2015/08/21	0.1301
6	2015/08/24	0.2020
7	2015/08/26	NaN
8	2015/08/31	0.2311

Row 6 in the table had a date three days after its previous date, instead of the chosen five-day interval. These extra dates, such as row 6, were kept so that more data were considered when handling missing data. In addition, row 6 was only 2 days away from the next missing point implying that a smoothing method may be applied to fill missing data points. However, as is apparent in Figure 10, this is not always the case – February 2016, for example, had no observed

data and thus the whole month would depend on observed data points that are too far away to use the smoothing method.

5.1.2 Percent of missing WHC data

The percent of missing data was calculated for using the following equation:

$$percent\ of\ missing\ data = \frac{total\ number\ of\ NaN}{total\ number\ of\ data\ points} \times 100 \quad (7)$$

There were 461 data points after ensuring there was an entry for every 5 days, starting from 1 August 2015 and ending on 28 December 2019. The total number of "NaN" was 174 and the calculated percent of missing data was 37.7%. This value was below 40% meaning the results after handling the missing data would be valid and useable (Jakobsen et al., 2017). Moreover, some data were available on other days as discussed above. Figure 11 shows this missing data more clearly.

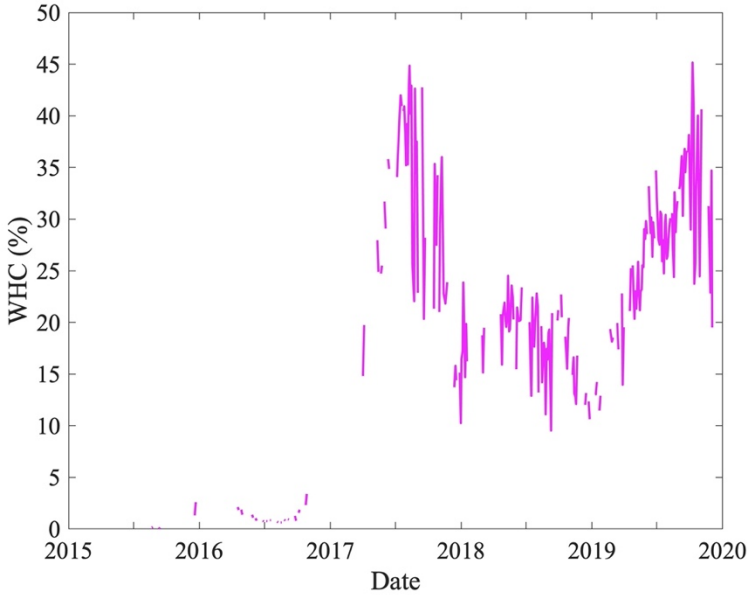


Figure 11: Time series plot of WHC dataset with missing data points

Although 37.7% of the data were missing, the shape of the dataset could still be distinguished in the figure. Lastly, Figure 12 shows the portion of the dataset used to build the predictive model, while the rest was used for comparing to final predictions.

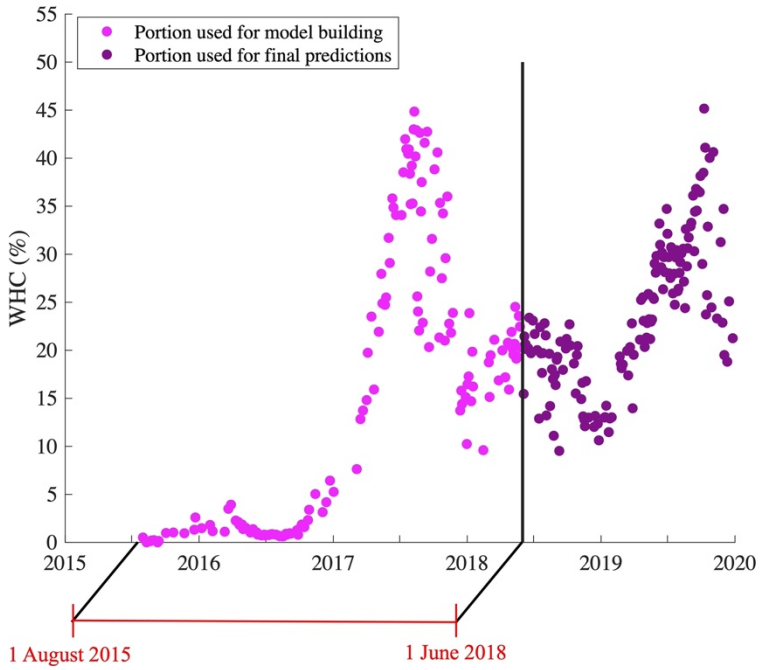


Figure 12: Observed WHC dataset used in model building and to make final predictions

The figure shows that a year and a half, from 1 June 2018 to 28 December 2019, was kept aside for final predictions, while the rest of the data was used to build the model.

5.2 Ambient temperature

Research shows weather conditions, especially temperature, was the major influence of the WHC. Therefore, the minimum and maximum daily temperatures were obtained from SAWS. A request was submitted to SAWS – obtained from their website (SAWS, 2022). The supplied dataset contained the daily minimum (MinTemp) and maximum (MaxTemp) temperature from the weather station (05125544) stipulated in Coetzee (2012). MaxTemp and MinTemp were kept as two separate features because WHC is sensitive to extreme temperatures (Reddy & DeBusk, 1984; Coetzee, 2012). Table 16 outlines the characteristics of the dataset.

Table 16: Characteristics of observed temperature dataset (SAWS, 2022)

Characteristic	MaxTemp and MinTemp Details
Total number of data points	1611 each
Total number of missing data points	20 each
File type	spreadsheet
Data type	numerical
Units measured in	date and °C
Number of decimal places	1
Timeline	1 August 2015 to 28 December 2019

The data was analysed in the same way as for WHC.

5.2.1 Data analysis of MaxTemp and MinTemp

Figures 13 and 14 display the time series plot of the MaxTemp and MinTemp data, with missing data points, respectively.

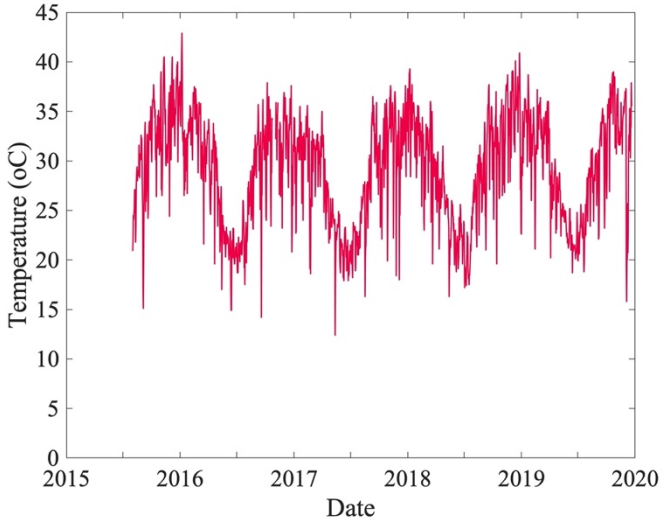


Figure 13: Time series plot of observed MaxTemp dataset

The missing data points were 13 consecutive points from October 2016 to November 2016, and 7 consecutive points in December 2019.

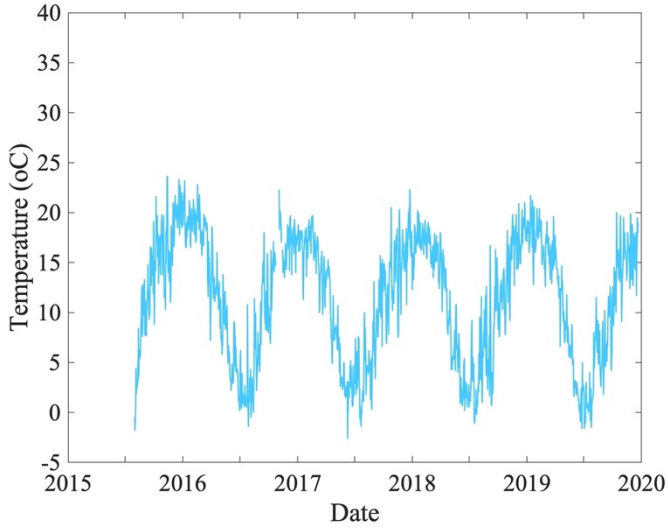


Figure 14: Time series plot of observed MinTemp dataset

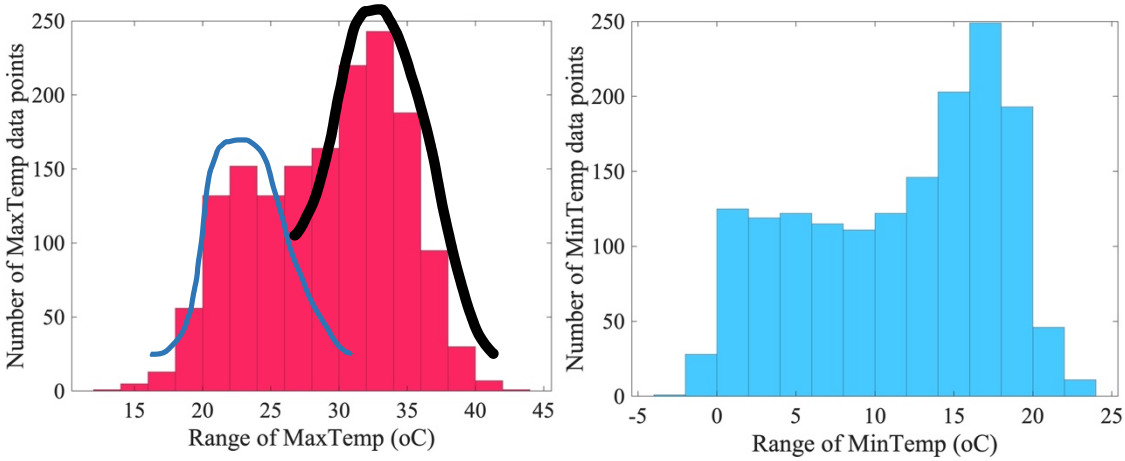
The missing points for MinTemp were on the exact same dates as for MaxTemp. The figures also display clear cycles that occur yearly and are consistent. Although global warming is resulting in an increase in temperatures, the temperatures patterns in 2015 to 2020 were consistent.

Before descriptive statistics were calculated for MaxTemp and MinTemp for later comparisons, outliers were analysed. The figures did not show any obvious outliers, but further calculations showed that both datasets did not have any outliers when using a threshold of 1.5. Table 17 presents the descriptive analysis results.

Table 17: Descriptive statistics of the observed temperature datasets

Dataset	Standard Deviation	Mean	Median	Range	Kurtosis	Skewness
MaxTemp	5.44	29.02	29.90	30.50	2.16	-0.25
MinTemp	6.22	11.41	12.60	26.20	1.88	-0.33

The descriptive statistics show that the MinTemp range falls within the range of the MaxTemp. This is because Hartbeespoort dam has high temperatures throughout the year, with winters reaching lows between 0 and 10, and only sometimes reaching below 0. Furthermore, the values are similar for each statistic. Finally, the PD of the datasets are shown in Figure 15.



(a) (b)

Figure 15: Histogram of observed MaxTemp (a) and MinTemp (b) datasets

The figure shows that temperature datasets do not follow one theoretical PD, but instead might have two PDs as both histograms have two peaks each. This makes sense as ambient temperature has different temperature ranges in winter and summer, further identifying a presence of seasonality. In other words, the first hill, the thin blue line in Figure 15a, are maximum temperatures experienced in winter months, while the black thick line is those in summer months. Furthermore, the datasets appear to have similar shapes, as the descriptive statistics suggested.

5.2.2 Percent of missing MaxTemp and MinTemp data

Equation (7) was used to calculate the percent missing data which was only 1.2% for both MaxTemp and MinTemp. All missing data points were handled and used, regardless of it being less than 5%, to take advantage of the information it provided for model building (Newman, 2014). Data was recorded daily, thus data points for every 5 days already existed.

Finally, the portion of data that was used for model building is the same as for WHC, 1 August 2015 to 1 June 2018, which can be seen in Figure 16.

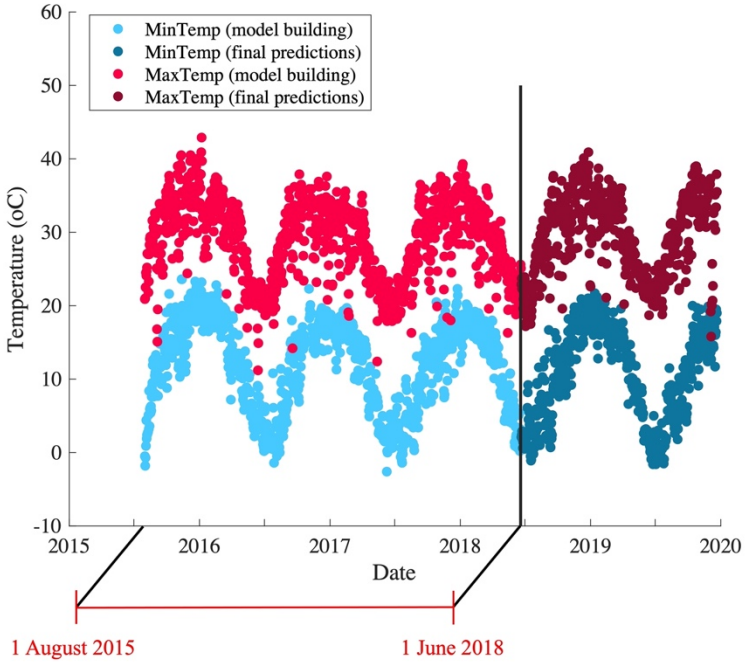


Figure 16: Observed temperature datasets used in model building and for final predictions

The temperature dates match all the dates in WHC dataset.

5.3 Total phosphate (TP)

Water hyacinth growth is greatly dependent on eutrophic levels because of the hyperbolic relationship that exists between water hyacinth growth and TP (Aoyama & Nishizaki, 1993). Consequently, a water quality dataset was downloaded from the online and publicly accessible website of the National Eutrophication Monitoring Programme (NEMP) (DWS, 2019). This dataset contained TP from many locations.

A second dataset was obtained to get more TP data. This dataset was downloaded from the Centre for Water Sciences and Management website at North-West University (Huizenga, 2015; Huizenga et al., 2013). It contained the chemical and physical properties of many South African dams. The data extracted from the dataset was recordings of phosphate (PO_4) from stations on the Hartbeespoort dam. Table 18 lists the station codes referring to locations where recordings took place along the Hartbeespoort dam, used to extract data from both datasets obtained.

Table 18: Station codes where nutrient data were obtained (Huizenga, 2015; DWS, 2019)

Dataset	Station Codes
TP from NEMP	A2H081Q01, A2H082Q01, A2H083Q01, A2R001Q01 to Q10
PO_4	A2R001Q01

The datasets had instances of multiple recordings for one day or repeated dates. The average of these recordings was taken for each day so that there was only one recording per date. An extract of the TP dataset obtained from NEMP is shown in Table 19.

Table 19: Extract of TP dataset (DWS, 2019)

Date	Time	TP	Place
2015-11-30	11:20:00	0.82400	A2H081Q01 HARTBEESPOORT DAM ON CROCODILE RIVER: LEFT CANAL
2015-12-02	11:15:00	0.76100	A2H083Q01 HARTBEESPOORT DAM ON CROCODILE RIV: DOWN STREAM WEIR
2015-12-02	10:14:00	0.38800	A2R001Q01 HARTBEESPOORT DAM ON CROCODILE RIV: NEAR DAM WALL
2015-12-14	13:35:00	0.75300	A2H081Q01 HARTBEESPOORT DAM ON CROCODILE RIVER: LEFT CANAL
2015-12-17	11:20:00	0.77600	A2H083Q01 HARTBEESPOORT DAM ON CROCODILE RIV: DOWN STREAM WEIR
2016-01-12	15:30:00	1.11500	A2H081Q01 HARTBEESPOORT DAM ON CROCODILE RIVER: LEFT CANAL
2016-01-19	12:39:00	1.03300	A2H081Q01 HARTBEESPOORT DAM ON CROCODILE RIVER: LEFT CANAL
2016-01-19	13:00:00	1.23200	A2H083Q01 HARTBEESPOORT DAM ON CROCODILE RIV: DOWN STREAM WEIR

Similarly, an extract of the PO_4 data from Huizenga (2015) is shown in Table 20.

Table 20: Extract of PO_4 dataset obtained from NEMP (Huizenga, 2015)

Station	Date	PO_4
A2R001Q01	1999/01/12	0.041
A2R001Q01	1999/01/12	0.035
A2R001Q01	1999/01/12	0.193
A2R001Q01	1999/01/12	0.157
A2R001Q01	1999/01/12	0.262
A2R001Q01	1999/02/02	0.027
A2R001Q01	1999/02/02	0.035
A2R001Q01	1999/02/02	0.032

Both datasets had missing data either recorded as blank or -9999, which were replaced with the term “NaN”. Characteristics of the two datasets, after ensuring one recording per day, are presented in Table 21.

Table 21: Characteristics of TP and PO_4 datasets (DWS, 2019; Huizenga et al., 2013)

Characteristic	TP Details	PO_4 Details
Total number of data points	2668	315
Total number of missing data points	0	1
File type	csv	spreadsheet
Data type	numerical	numerical
Units measured in	date and mg/L	date and mg/L
Number of decimal places	5	3
Timeline	11 March 1980 to 18 July 2018	1 December 1999 to 14 December 2011

Although different in name, both TP and PO_4 were recordings of the same data being the total phosphate found in the Hartbeespoort dam (DWS, 2019; Huizenga et al., 2013).

5.3.1 Data analysis of TP and PO_4 datasets

The TP and PO_4 datasets are plotted on the same set of axes in Figure 17.

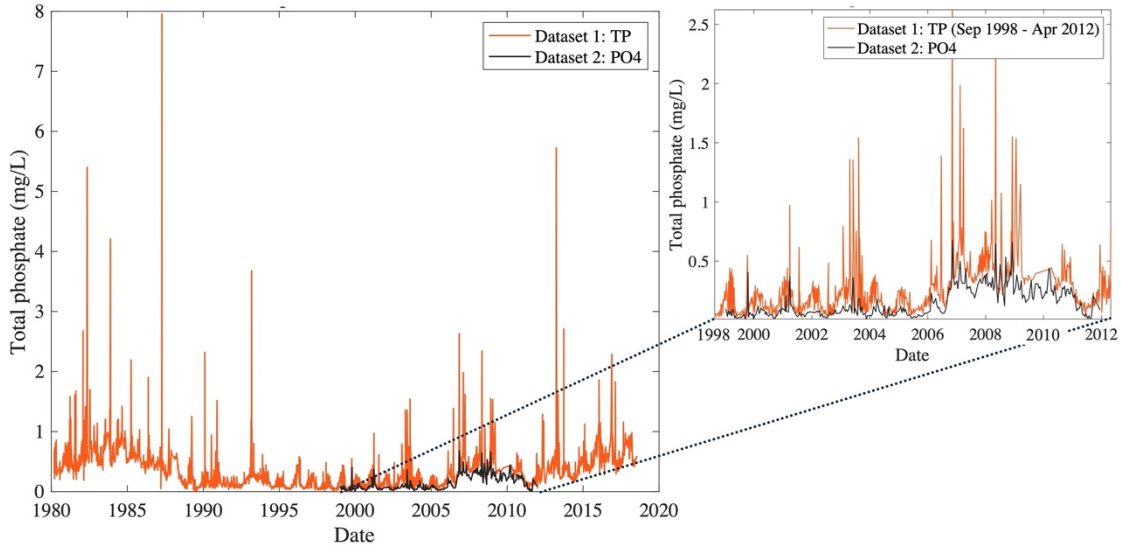


Figure 17: Time series plot of observed phosphorous datasets

In the figure it is clear dataset 1, in orange, had outliers, otherwise both datasets were similar. TP and PO_4 were combined to form one TP dataset. To combine both datasets, the daily average of recordings was taken and then outliers were handled. In some cases where the date had "NaN" values in one dataset but not the other, the existing recording was kept. After this was done, the new TP dataset had the characteristics shown in Table 22.

Table 22: Characteristics of combined TP dataset

Characteristic	Combined TP Details
Total number of data points	2706
Total number of missing data points	0
Number of decimal places	4
Timeline	11 March 1980 to 18 July 2018

The number of decimal places was changed to 4 to match the WHC data. Also, the total number of data points increased by 38 data points as the PO_4 dataset had extra dates that the TP dataset from NEMP did not. Next, outliers were identified then replaced with the nearest neighbour, as shown in Figure 18.

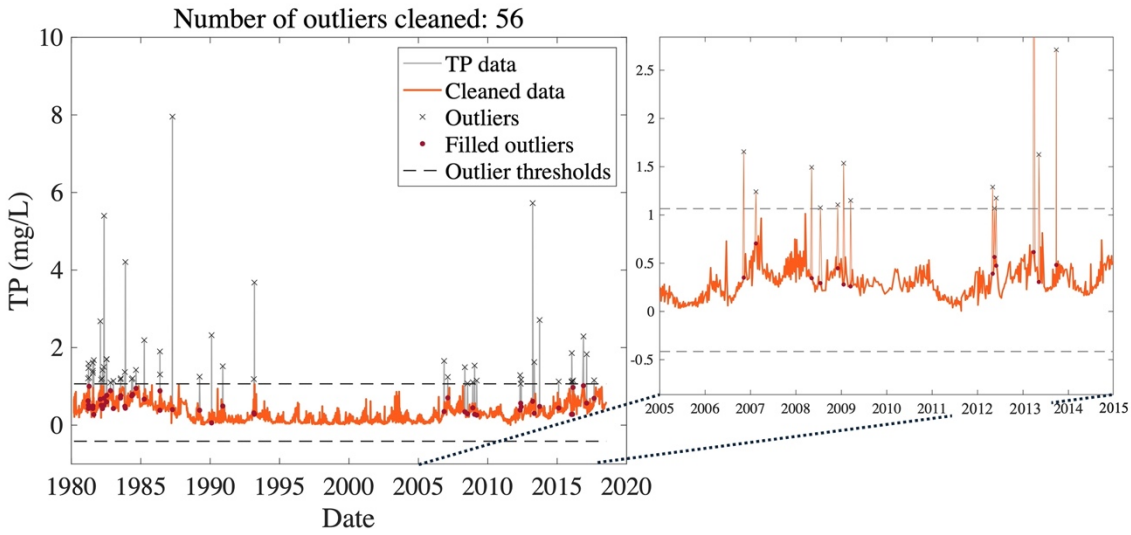


Figure 18: Outliers identified and replaced with nearest TP value

A total of 56 outliers were identified, using a threshold factor of 1.5 and Equations (3) and (4), then cleaned to form the combined TP dataset. Points on or slightly above the outlier threshold line were considered outliers and also replaced. A time series of this dataset without outliers is shown in Figure 19.

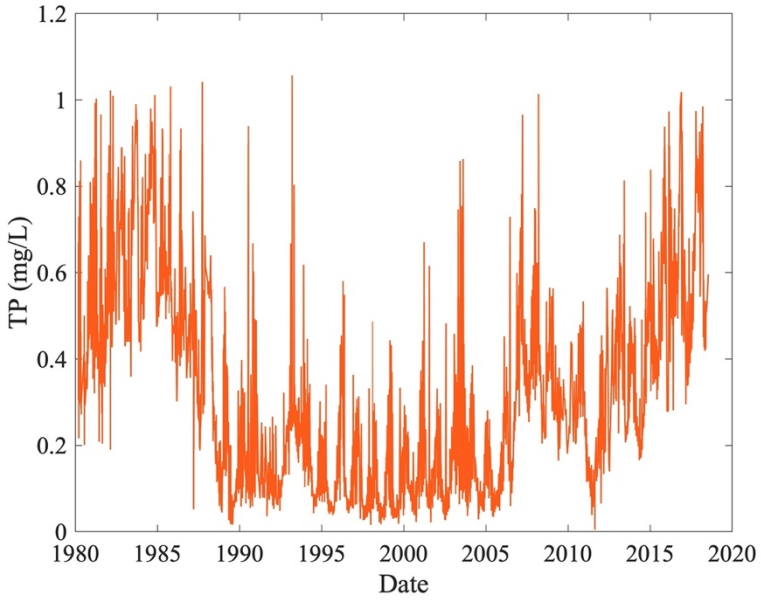


Figure 19: Time series plot of the observed TP dataset without outliers

The figure shows patterns present in the data and possible seasonal changes, but the patterns and seasonality were not as clear as it was in the other datasets. Therefore, these patterns were investigated further in Figure 20, which has the area below the graph filled to reveal the shape of the curve.

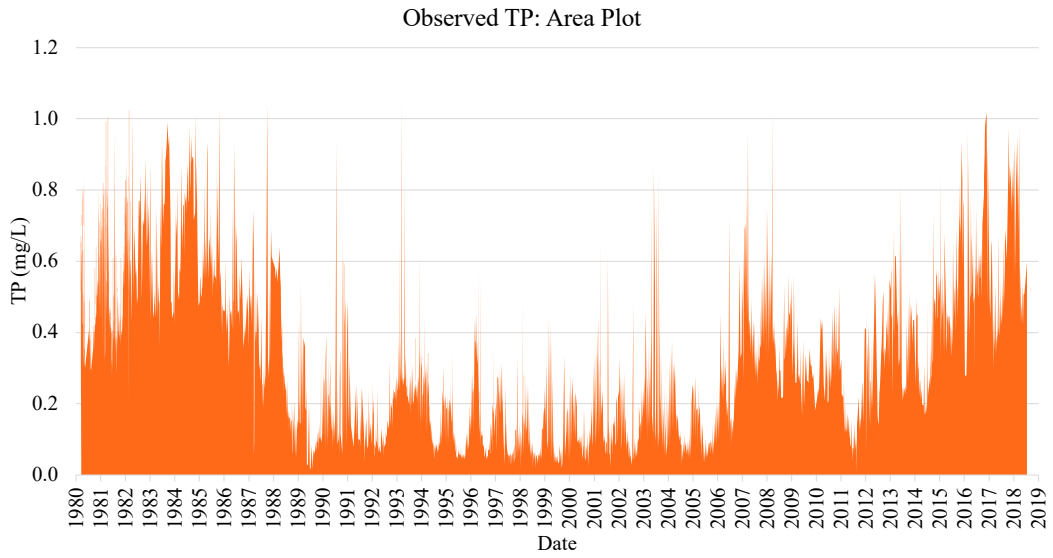


Figure 20: Time series plot of TP dataset without outliers with shaded area

The figure made the seasonal patterns that occur yearly more visible. The peaks occurred at the beginning of the year and the dataset appeared to have cycles where the yearly patterns increase for a number of years and then decrease again, such as in 1980 to 1990. Lastly, it is unclear whether the increasing trend seen from 2010 continued to increase or not, but from past data, such as from 1980 to 1990, it appears the increasing trend was followed by a decreasing one. This is further investigated in Chapter 6. The descriptive statistics of the TP dataset were calculated as done for the other datasets, and is displayed in Table 23.

Table 23: Descriptive statistics of observed TP dataset

Standard Deviation	Mean	Median	Range	Kurtosis	Skewness
0.2307	0.3449	0.3111	1.0517	2.6100	0.6107

The kurtosis of the TP dataset is close to 3 which implies a PD from the normal family could fit the data (Law, 2015; Ricci, 2005; Kissell & Poserina, 2017). A histogram was plotted to visualise the PD of the dataset.

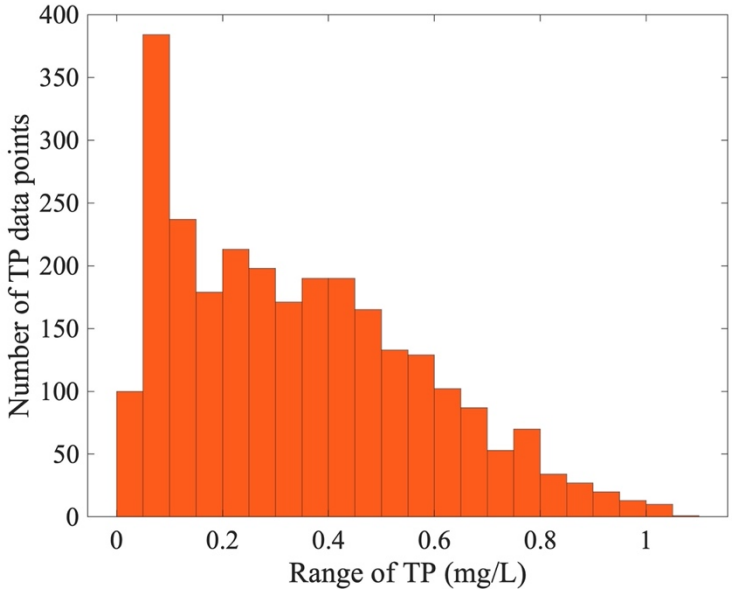


Figure 21: Histogram of observed TP dataset

The histogram appears to follow one PD as evident in the highest peak of data between 0.05 and 0.1 mg/L, which has the most data points. The TP dataset had some dates that matched WHC but data points were interpolated every 5 days starting from 1 August 2015 until the end of the TP dataset, 18 July 2018, to ensure the dates match the WHC dates (see Section 5.1.1).

5.3.2 Percent of missing TP data

Figure 22 presents the time series plot of the TP dataset with the missing values that existed after interpolating the dates needed to match the WHC dataset until 18 July 2018, where the TP dataset ended.

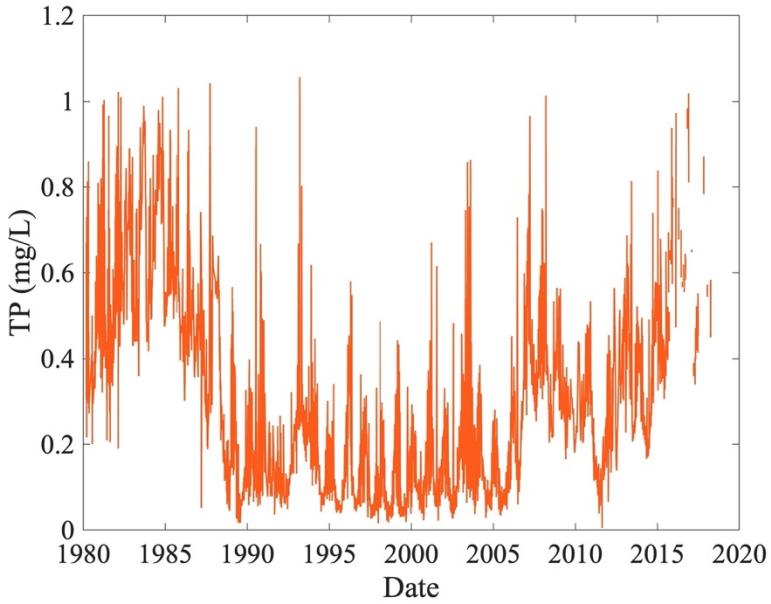


Figure 22: Time series plot of TP dataset with missing data points

The missing data forms 8.45% of the total data; the dataset had 2956 data points with 250 missing data points representing dates that match WHC. Finally, the scatter plot in Figure 23 indicates the portion of data that was used for model building.

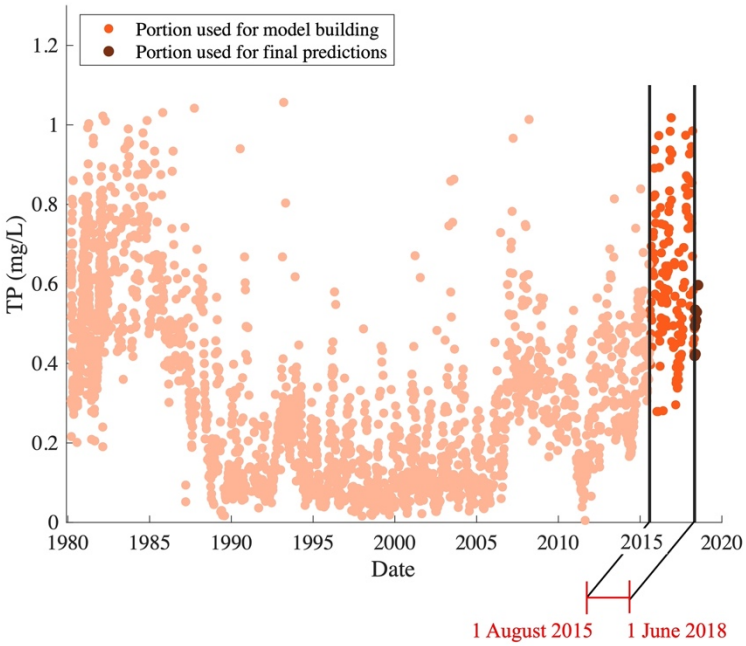


Figure 23: Observed TP dataset used in model building and to make final predictions

The TP dataset only partially matched the WHC. Consequently, TP data were generated so that there could be data for final predictions. A more detailed time series analysis was done on the TP dataset after its missing data was handled so as to obtain information that could be used to extrapolate the dataset for the period required to match the WHC dataset.

5.4 Total nitrogen (TN)

As with TP, TN has a relationship with water hyacinth growth (Aoyama & Nishizaki, 1993). Therefore, TN data was also obtained for the purpose of this research. The dataset obtained from Huizenga (2015) also contained recording of nitrogen compounds that, when added together can form a TN dataset (Huizenga et al., 2013). Recordings of the sum of nitrite and nitrate (NO_2+NO_3), and ammonium (NH_4) were extracted from the dataset. An extract of the data is shown in Table 24.

Table 24: Extract of nitrogen datasets from NEMP (Huizenga, 2015)

Station	Date	NO_3+NO_2	NH_4
A2R001Q01	1999/01/12	1.438	0.020
A2R001Q01	1999/01/12	1.484	0.020
A2R001Q01	1999/01/12	1.100	0.695
A2R001Q01	1999/01/12	1.809	0.425
A2R001Q01	1999/01/12	0.360	0.657
A2R001Q01	1999/02/02	0.321	0.106
A2R001Q01	1999/02/02	0.474	0.020
A2R001Q01	1999/02/02	0.389	0.020

The average per day was calculated, as done for TP, so there was only one recording of NO_3+NO_2 and NH_4 each per day. Characteristics of the two compounds are presented in Table 25.

Table 25: Characteristics of dataset containing nitrogen compounds

Characteristic	**NH_4 Details**	
Compound name	NO_2+NO_3	NH_4
Total number of data points	315	315
Total number of missing data points	24	1
File type	spreadsheet	
Data type	numerical	
Units measured in	date and mg/L	
Number of decimal places	3	
Timeline	1 December 1999 to 14 December 2011	

The datasets did not coincide with the WHC dataset time line and thus interpolating dates at five-day intervals was not necessary as none of the dates match those of the WHC dataset. TN is equivalent to the sum of all the compounds containing nitrogen (Water Resources Mission Area, 2019). However, the two compounds were kept separate and only added together to form the TN dataset after missing data was handled in each dataset individually.

5.4.1 Data analysis of NO_3+NO_2 and NH_4 datasets

Figures 24 and 25 display time series plots of the NO_3+NO_2 and NH_4 datasets, respectively.

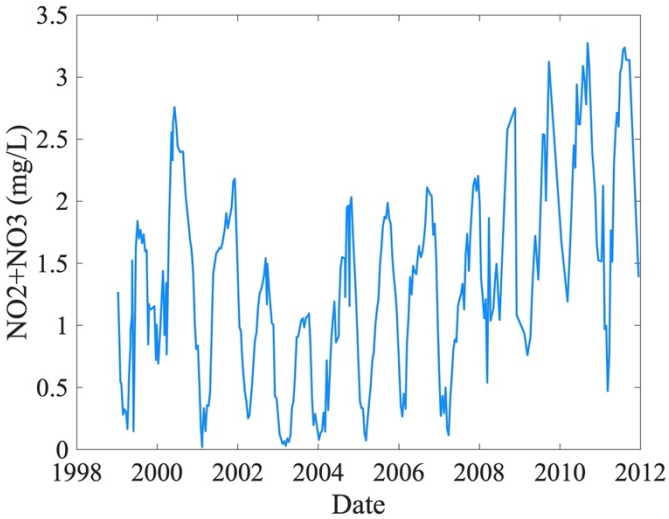


Figure 24: Time series plot of NO_2+NO_3 dataset

It is evident that NO_2+NO_3 had seasonal patterns with peaks occurring in the middle of the year. The peaks appeared to be increasing from 2010 onwards, but the rest of the data suggested that the NO_2+NO_3 remained similar across the years.

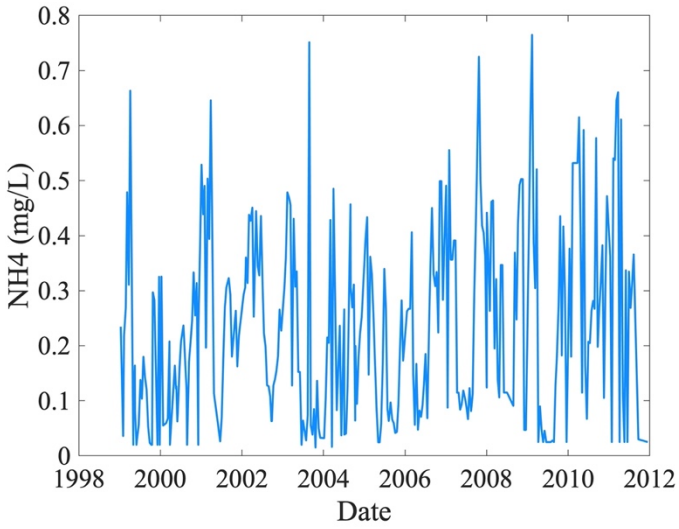
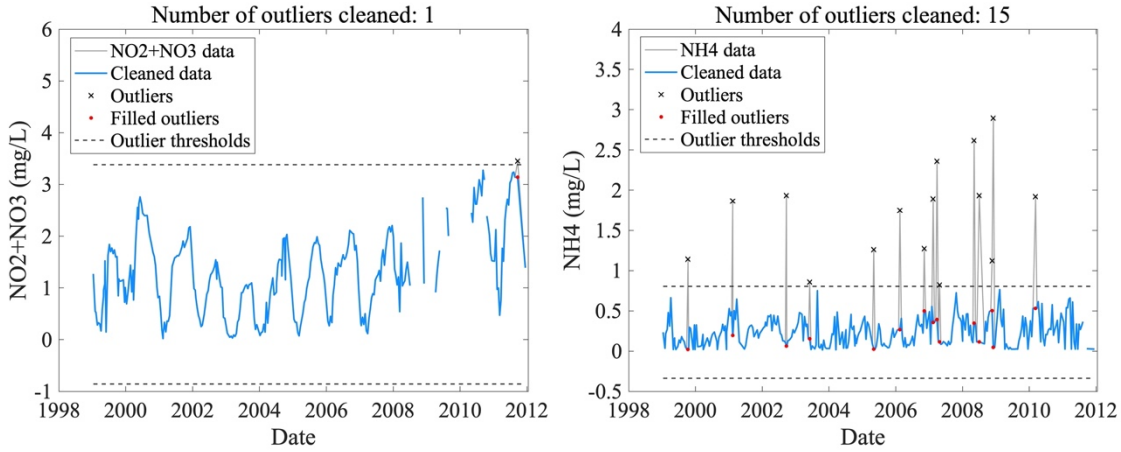


Figure 25: Time series plot of NH_4 dataset

NH_4 displayed similar seasonal patterns with peaks occurring in the middle of the year, but the patterns are not as clear as with NO_2+NO_3. The data remained consistent throughout the years except for a few outliers. All outliers were identified in the nitrogen datasets and then replaced with the nearest neighbour, as shown in Figure 26.



(a) (b)

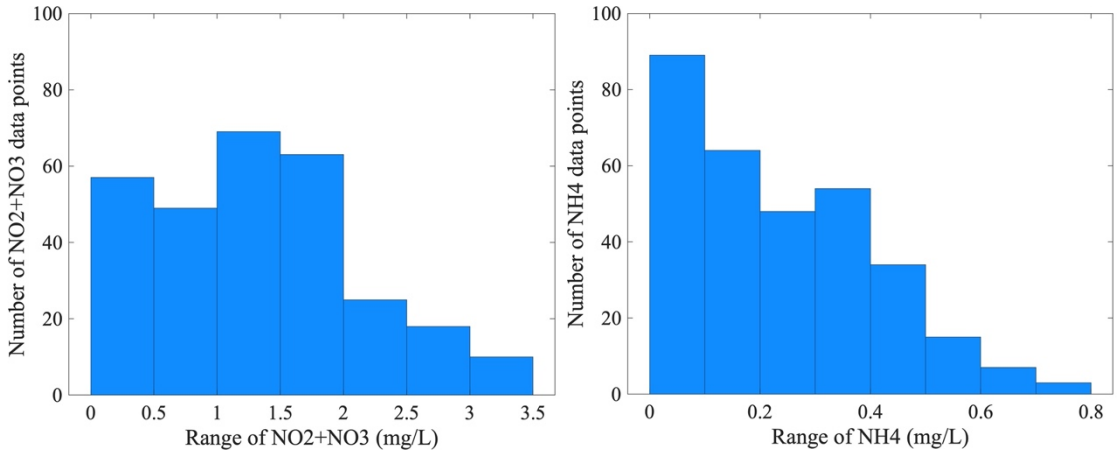
Figure 26: Outliers identified and replaced with nearest nitrogen value

Outlier thresholds were calculated using Equations (3) and (4) with a threshold factor of 1.5. Any data points on or near the outlier threshold lines, the upper and lower limits in the equations were also replaced to ensure consistency. Figure 26(a) and (b) show that the NO_2+NO_3 and NH_4 datasets had 1 and 15 outliers, respectively, and without the outliers, the data were more consistent across the years observed. To further analyse the data, descriptive statistics were calculated – shown in Table 26.

Table 26: Descriptive statistics of the observed nitrogen datasets

Dataset	Standard Deviation	Mean	Median	Range	Kurtosis	Skewness
NO_2+NO_3	0.79	1.31	1.25	3.26	2.56	0.39
NH_4	0.17	0.24	0.21	0.75	2.68	0.64

It is clear from the time series plots and descriptive statistics that the datasets are not the same which is expected as both datasets were measures of different nitrogen compounds – time series analysis was done on the full TN dataset after the missing data in NO_2+NO_3 and NH_4 were handled. This is explored in Chapter 6. However, the statistics suggest the datasets could be described by the same PD, so histograms were plotted and shown in Figures 27(a) and (b) for NO_2+NO_3 and NH_4, respectively.



(a) (b)

Figure 27: Histogram of observed NO_2+NO_3 (a) and NH_4 (b) datasets

The plots had a similar curve shape, as expected from the statistics. The histograms and descriptive statistics were used to validate the datasets after missing data was handled.

5.4.2 Percent of missing data in NO_2+NO_3 and NH_4 datasets

The number of missing data for the nitrogen datasets, NO_2+NO_3 and NH_4, was calculated based on the information in Table 25. NO_2+NO_3 had 7.6% missing values, while NH_4 had 0.3% missing values as only one data point was missing in the observed dataset. In addition, the average number of days between data points was 15 meaning that there were about only 2 recordings per month, with some months missing completely such as October and November 2011. This meant that missing data could not be interpolated from its curve to complete the dataset.

5.5 Timelines of Each Dataset Used in Model Building

It is clear that the timeline of both the nitrogen datasets, as well as the TP dataset, did not align with the timeline of the WHC dataset. Figure 28 shows the timeline of each dataset compared to that of WHC and temperature datasets.

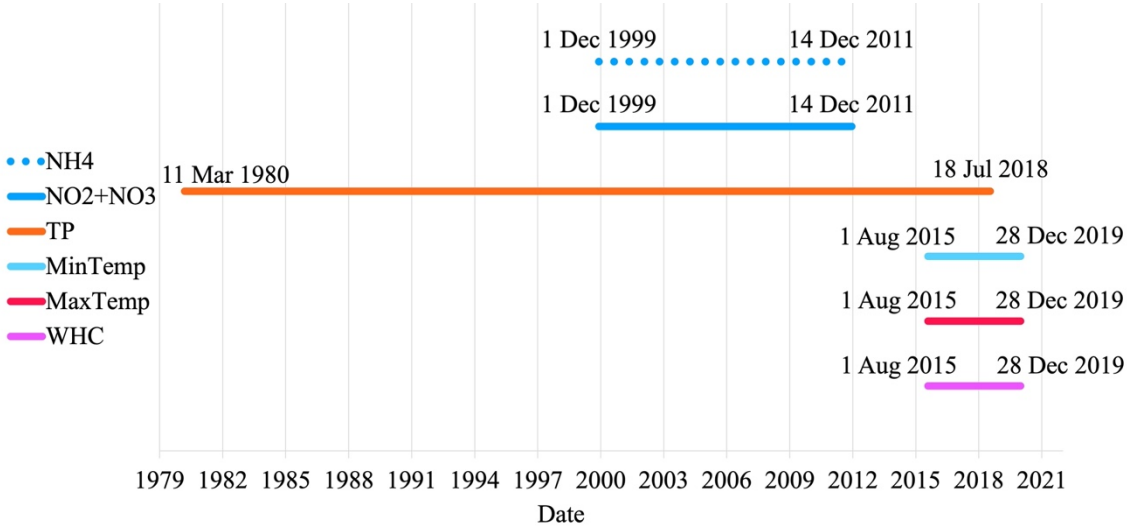


Figure 28: Timeline of each dataset

As evident in the figure, the temperature datasets, MaxTemp and MinTemp, matched the WHC timeline, unlike the other datasets. The timeline of the TP dataset overlapped that of the WHC dataset partly but the nitrogen datasets did not match at all. However, it meant that the nitrogen TP datasets required simulation to generate data so that the datasets could coincide with the WHC dataset.

In conclusion, the datasets used for model building after missing data was handled were: WHC dataset as the input target variable, and MaxTemp and MinTemp datasets as features. The TP, NO_2+NO_3 and NH_4 datasets were extrapolated so that the TP and resultant TN datasets could be used as features in model building. Hence, Chapter 6 explains how these datasets were handled.

6 DATA HANDLING

The preceding chapter showed the extent of missing data that each dataset had. This chapter describes the method used to prepare the datasets for model building. Figure 29 outlines the method which involves four steps, all using MATLAB and Excel.

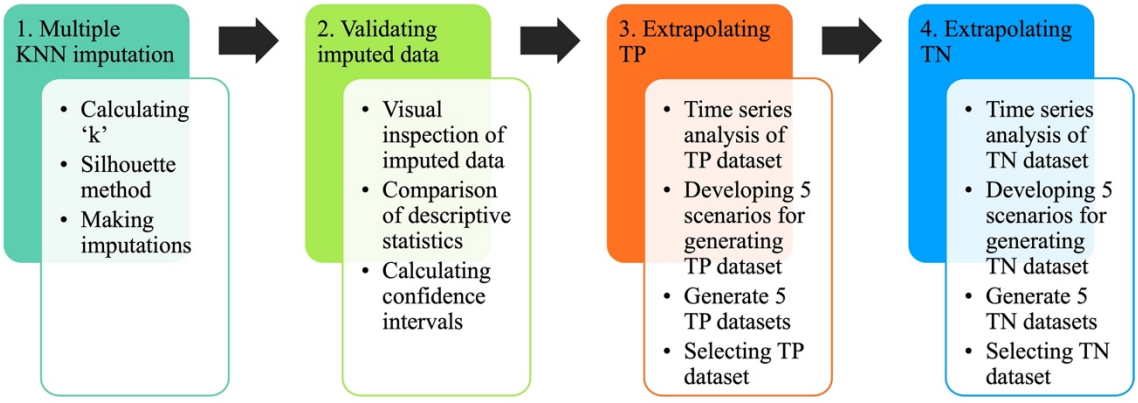


Figure 29: Process flow diagram for preparing the datasets for model building

These four steps are discussed in the sections to follow. First, the missing data in each dataset is imputed by means of multiple KNN imputation. This method of handling missing data was chosen as literature with similar datasets proved it to be effective (Diouf et al., 2022; Eskelson et al., 2009; Ferrari & Ozaki, 2014; Yang, 2022). Thereafter, the imputed data was validated by ensuring the dataset with imputed data was visually similar to the observed dataset, as well as had similar descriptive statistics, all while falling within the calculated confidence interval limits.

Next, the TP and TN datasets were extrapolated to the required timeline. Not only did the TP and TN datasets not have data for dates matching the WHC dataset, they were also inaccessible due to the lack of continuous research and reporting in the area of water quality in Hartbeespoort – refer to Section 5.3. Thus, to extrapolate the datasets, five scenarios were developed to generate the TP and TN datasets which used a combination of Monte Carlo simulation and the results of the time series analysis performed on both datasets. A dataset for TP and one for TN was chosen based on results of a statistical test.

6.1 Multiple KNN Imputation

All datasets had missing data that required imputation before it could be used in model building. As explained in Chapter 3, multiple KNN imputation is one of the best-performing and most effective methods of handling missing data for any dataset (Batista & Monard, 2002; Dong and Peng, 2013; Gao et al., 2011; Mohammed, 2021). This section describes the multiple KNN imputation method and contains the details the imputations made for each dataset: WHC, MaxTemp, MinTemp, TP and TN.

A function coded in MATLAB performed the imputation on all datasets – refer to Appendix C for the code. The function randomly placed the data points into x number of groups, where x was automatically determined in MATLAB using the silhouette method.

Both the silhouette and elbow methods are popular ways of determining the optimal number of clusters. The silhouette method is known to be more effective than the elbow method because it evaluates the distances within and between clusters (Bonaccorso, 2018; Dangeti, 2017; Raschka, 2015). MATLAB determines the optimal number of clusters using the *evalcluster* function, which returns it as a whole number (MATLAB, 2013).

Next, the missing points were imputed with KNN using the *knnimpute* function (MATLAB, 2006a). This function calculated the number of nearest neighbours, k, using Euclidean distance and thus required the conversion of dates to a serial number. The k value was calculated as the nearest integer found with Equation (5).

This is done iteratively with variations in the grouping so that the missing point is imputed with multiple values. Each missing point was then finally filled with the average of all the imputed values per point. The number of iterations was chosen through trial and error; the number of iterations was initially set to 100 and increased in increments of 100. More than 500 iterations did not result in large differences between imputations as the differences were less than 0.001 and, therefore, 500 iterations were deemed the best.

The details of the multiple KNN imputations for each dataset are displayed in Table 27.

Table 27: Information used to impute missing data within each dataset

Dataset	Total data points	'k' nearest neighbours	Number of clusters
WHC	461	21	3
MaxTemp	1611	40	2
MinTemp	1611	40	2
TP	2956	54	2
NO_2+NO_3	315	18	4
NH_4	315	18	2

6.2 Validating Imputed Data

In this section the imputed data for each dataset is validated by: first, inspecting the imputations visually; second, comparing the descriptive statistics of the datasets with imputed data to that of the observed dataset and; lastly, determining whether the points fall within the calculated confidence intervals. The visual inspections reveal whether the imputations coincide with the observed points and complete the curve of the dataset. The descriptive statistics calculated were the standard deviation, mean, median, kurtosis and skewness of each observed dataset and dataset with imputed data. The percent difference of the two descriptive statistics was calculated using Equation (8) to evaluate the extent at which the imputed data changed the observed dataset.

$$percent\ difference\ of\ a\ statistic = |x_{observed} - x_{imputed}| \times 100 \quad (8)$$

where x is the statistic of the observed and imputed datasets. The absolute value of the difference was taken because whether the difference is higher or less than 0 was not important. Any differences below 5% indicate the imputed data made insignificant changes to the statistics of the observed dataset and are thus fit the observed dataset well.

Finally, a 95% confidence level was chosen, as is standard practice, and was calculated using Equation (9) (Kroese et al., 2020; Freedman et al., 2007).

$$Confidence\ Intervals = \bar{x} \pm 1.96\left(\frac{s}{\sqrt{n}}\right) \quad (9)$$

where $\bar{x}$ is the mean of the observed dataset, s is the standard deviation of the observed dataset, and n is the number of observed data points. For 95% confidence in a normal distribution, 1.96 is used. Although the distributions of each dataset was not normal, a normal distribution was used to calculate the confidence intervals to ensure consistency in validating the datasets. The confidence intervals were added and subtracted to each of the observed data points to form the upper and lower confidence limits, respectively.

6.2.1 Validating imputations of the input target variable: WHC

To validate the imputed WHC data, the imputations were plotted for visual inspection. Figure 30 shows the imputed data points on the same set of axes as the observed WHC datasets.

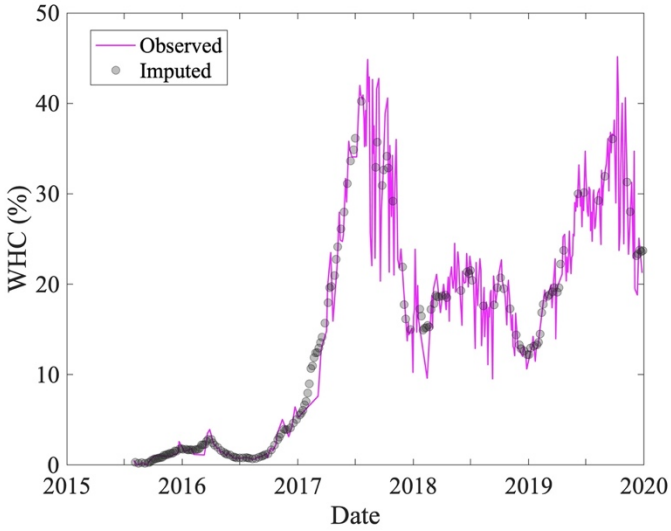


Figure 30: Plot of imputed WHC and the observed WHC dataset

The imputations correspond with the observed WHC line plot and thus the imputations appeared to be a good fit. Table 28 compares the descriptive statistics of the observed dataset and the dataset with imputed data.

Table 28: Difference in descriptive statistics of imputed and observed WHC

WHC	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	11.94	20.00	20.56	2.24	-0.11
Imputed	12.15	16.97	18.53	2.04	0.15
% difference	0.21	3.03	2.03	0.20	0.26

All the differences were below 3.5% and close to zero indicating that the imputations did not change the descriptive statistics of the observed WHC significantly. The mean and median have higher differences, still below 5%, which was expected due to the increase in total number of data points and majority of missing data points falling in the 0% to 10% WHC range.

To further validate the imputed data, the points should fall within the confidence interval limits calculated for each point. Figure 31 plots the imputed WHC and the confidence limits of the observed WHC dataset.

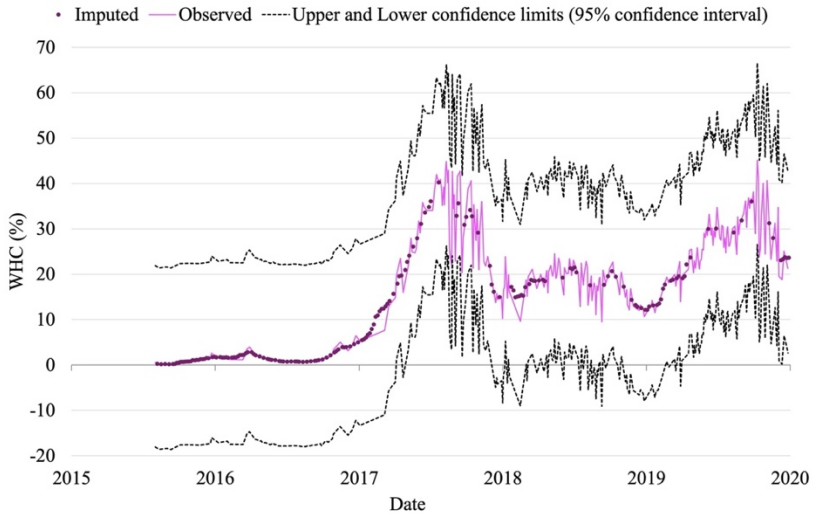


Figure 31: Imputed WHC and the confidence limits for each observed WHC data point

The figure shows that all the imputed data points fell within the confidence limits. To further validate the WHC imputations, the descriptive statistics of the data with imputed points were compared to that of the observed WHC dataset. In conclusion, the imputations were a good fit.

6.2.2 Validating imputations of the temperature features: MaxTemp and MinTemp

Figure 32 shows the MaxTemp and MinTemp dataset with the imputed data for visual inspection.

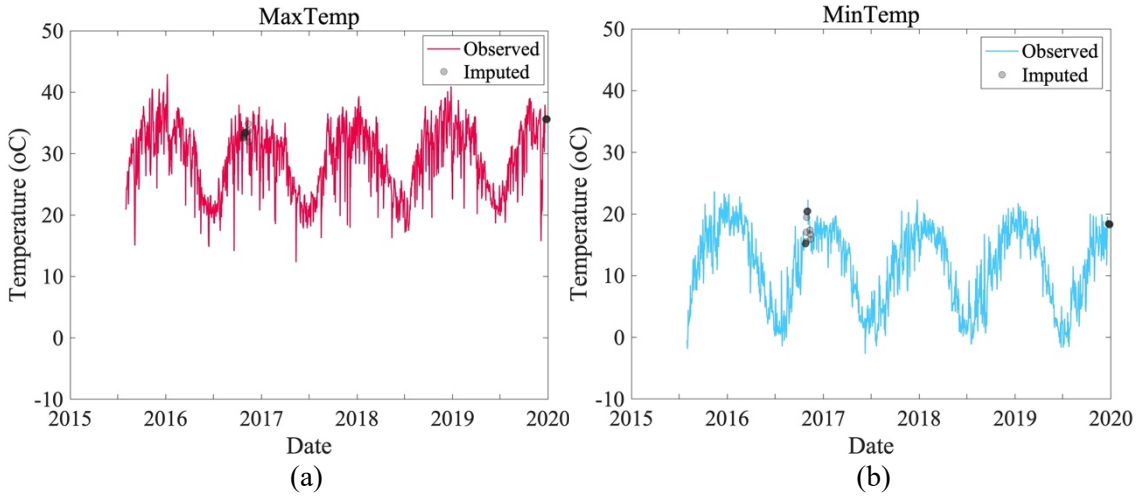


Figure 32: Plot of imputed and observed MaxTemp (a) and MinTemp (b)

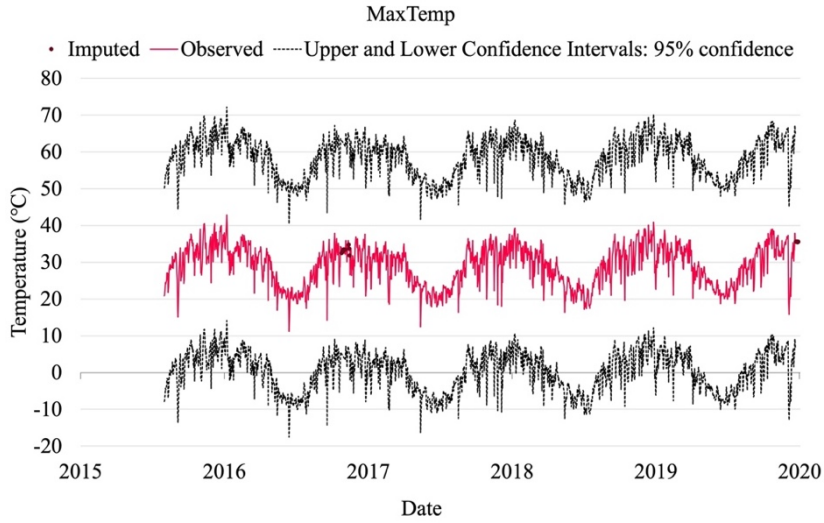
Both temperature datasets had only a few missing points that were imputed as shown in the figure. The imputed points seem to coincide with the observed data and did not influence the shape of the data. To further examine the extent to which the imputed data changed the observed dataset, the descriptive statistics were compared as shown in Table 29.

Table 29: Difference in descriptive statistics of imputed and observed temperature datasets

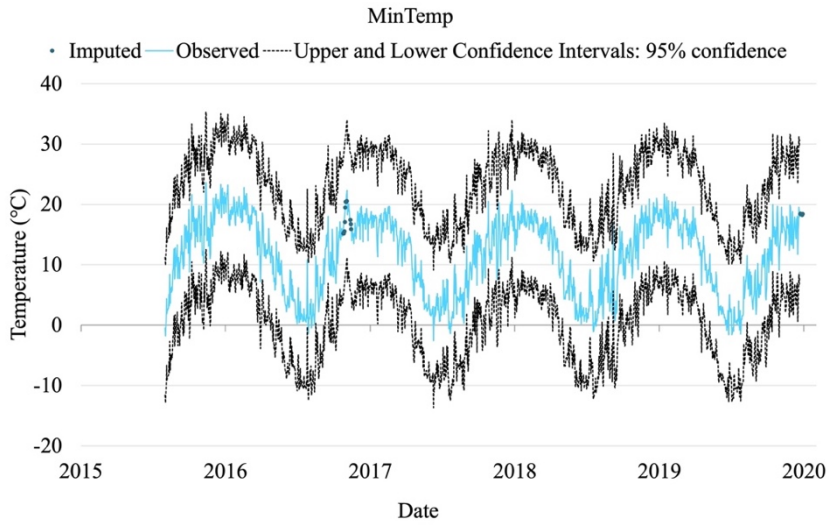
	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed MaxTemp	5.44	29.02	29.90	-0.25	5.44
Imputed MaxTemp	5.44	29.08	29.90	-0.27	5.44
% difference	0.00	0.06	0.00	0.02	0.00
Observed MinTemp	6.22	11.41	12.60	-0.33	6.22
Imputed MinTemp	6.23	11.49	12.80	-0.35	6.23
% difference	0.01	0.08	0.20	0.02	0.01

The statistics show that the imputed points changed the descriptive statistics of the observed datasets by less than 0.1%, with an average percent difference of 0.02% and 0.06% for MaxTemp and MinTemp, respectively. This proved that the imputations were a good fit.

Finally, Figure 33 displays the temperature datasets imputations with the calculated confidence limits for each dataset, in the same way as for WHC, to ensure the imputed data were valid.



(a)



(b)

Figure 33: Confidence limits with imputed MaxTemp (a) and MinTemp (b)

All the imputed data points, for both the MaxTemp and MinTemp datasets, fell well within the confidence limits. Therefore, the imputed values for both temperature datasets were a good fit and used in model building.

6.2.3 Validating imputations of the TP feature

The imputed TP data is plotted on the same set of axes as the observed TP dataset in Figure 34.

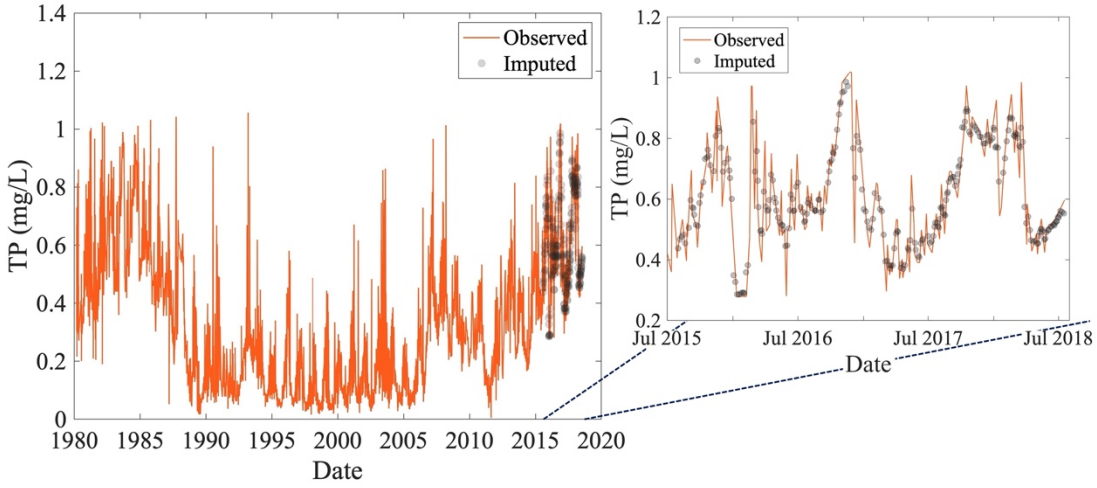


Figure 34: Plot of imputed and observed TP dataset

TP only had missing data from August 2015 to July 2018 but all data before then was used in multiple KNN imputation of the data to improve results. From visual inspection, the imputed data were a good fit as the imputed data followed the curves of the observed TP. To investigate the effect the imputed TP data had on the observed TP dataset, the descriptive statistics were compared. Table 30 shows this comparison.

Table 30: Difference in descriptive statistics of imputed and observed TP dataset

TP	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.23	0.34	0.31	2.61	0.61
Imputed	0.24	0.37	0.35	2.38	0.48
% difference	0.01	0.02	0.04	0.23	0.13

The statistics confirmed that imputed values were a good fit as all differences were below 0.25%, with an average difference of 0.09% which is very close to zero indicating almost no difference. Figure 35 displays imputed TP data with the confidence limits that were calculated based on 95% confidence on the observed TP dataset from 1 August 2015 to 18 July 2018.

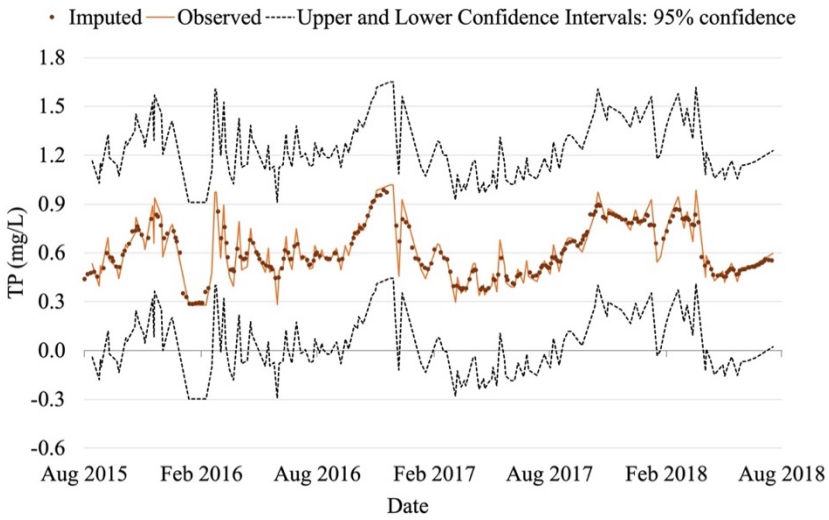


Figure 35: Confidence limits with imputed TP data

From the figure, it is clear all the imputed points fell within the confidence limits which indicate the imputed data were valid. Therefore, the imputed TP data were used to extrapolate the TP dataset to match the WHC timeline.

6.2.4 Validating imputations of the nitrogen features: NO_2+NO_3 and NH_4

As done with the other datasets, the imputed data for both nitrogen datasets was plotted and inspected. Figure 36 plots the data.

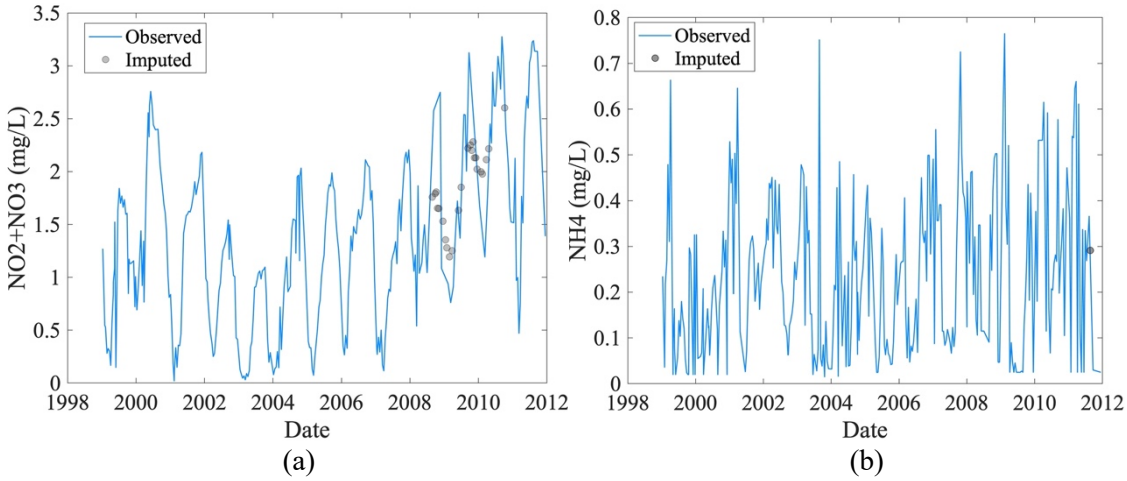


Figure 36: Plot of imputed and observed NO_2+NO_3 (a) and NH_4 (b)

NH_4 only had one missing data point to impute while NO_2+NO_3 had more between 1 to 3 mg/L in 2008 to 2011. The imputation of both datasets appear to fit the observed data similarly to the other datasets. In addition, the descriptive statistics shown in Table 31 proved that the imputed NO_2+NO_3 data did not changed the observed dataset as the average difference was 0.07%.

Table 31: Difference in descriptive statistics of imputed and observed NO_2+NO_3

NO_2+NO_3	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.79	1.31	1.25	2.56	0.39
Imputed	0.78	1.35	1.34	2.48	0.27
% difference	0.01	0.04	0.09	0.08	0.12

The statistics of NH_4 are not shown as only one data point was imputed meaning the statistics were identical when looking at two decimal places – the average difference was 0%.

The confidence intervals were calculated for both the nitrogen datasets in the same way as for the WHC dataset. The limits for the NO_2+NO_3 dataset is shown in Figure 37. It confirms that the imputed data did fall within the confidence limits and, thus, the imputed data were validated.

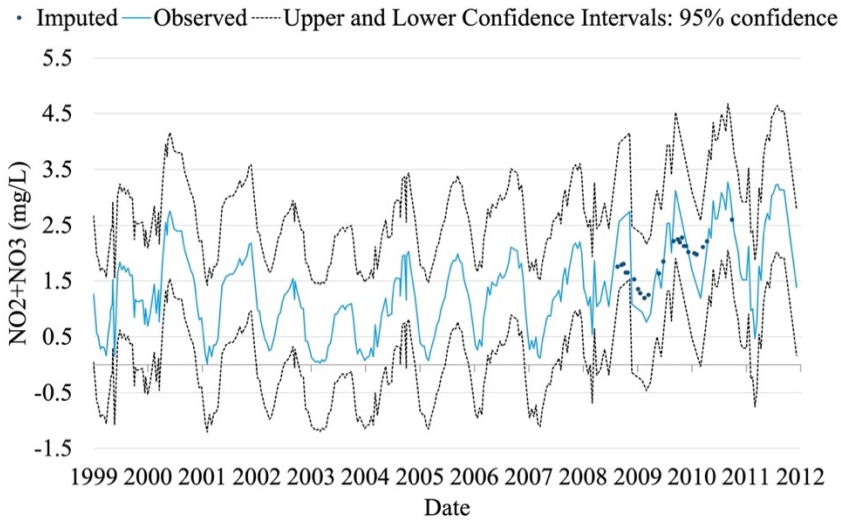


Figure 37: Confidence limits of observed NO_2+NO_3 dataset with imputed data

Next, the TN dataset was formed by adding the two datasets, as both nitrogen datasets were validated. TN is the sum of all nitrogen compounds found in the body of water (Water Resources Mission Area, 2019). Therefore, the two nitrogen datasets were added to get a TN dataset with 315 data points starting from 12 January 1999 to 14 December 2011. An example of how the datasets were added is shown in Table 32.

Table 32: Sum of NO_2+NO_3 and NH_4 to form TN dataset

Date	NO_2+NO_3 (mg/L)	NH_4 (mg/L)	TN (mg/L)
2010/06/02	2.9409	0.1648	3.1057
2010/06/17	2.6208	0.0671	2.6879
2010/06/30	2.6189	0.2074	2.8263

This TN dataset was used to extrapolate the TN dataset for the dates required for model building.

6.3 Extrapolating TP Dataset

This section summarises the method used to extrapolate the TP dataset to the timeline required to match the WHC dataset. The process flow diagram in Figure 38 outlines the steps followed. The final generated TP dataset used in model building is then displayed at the end of this section.

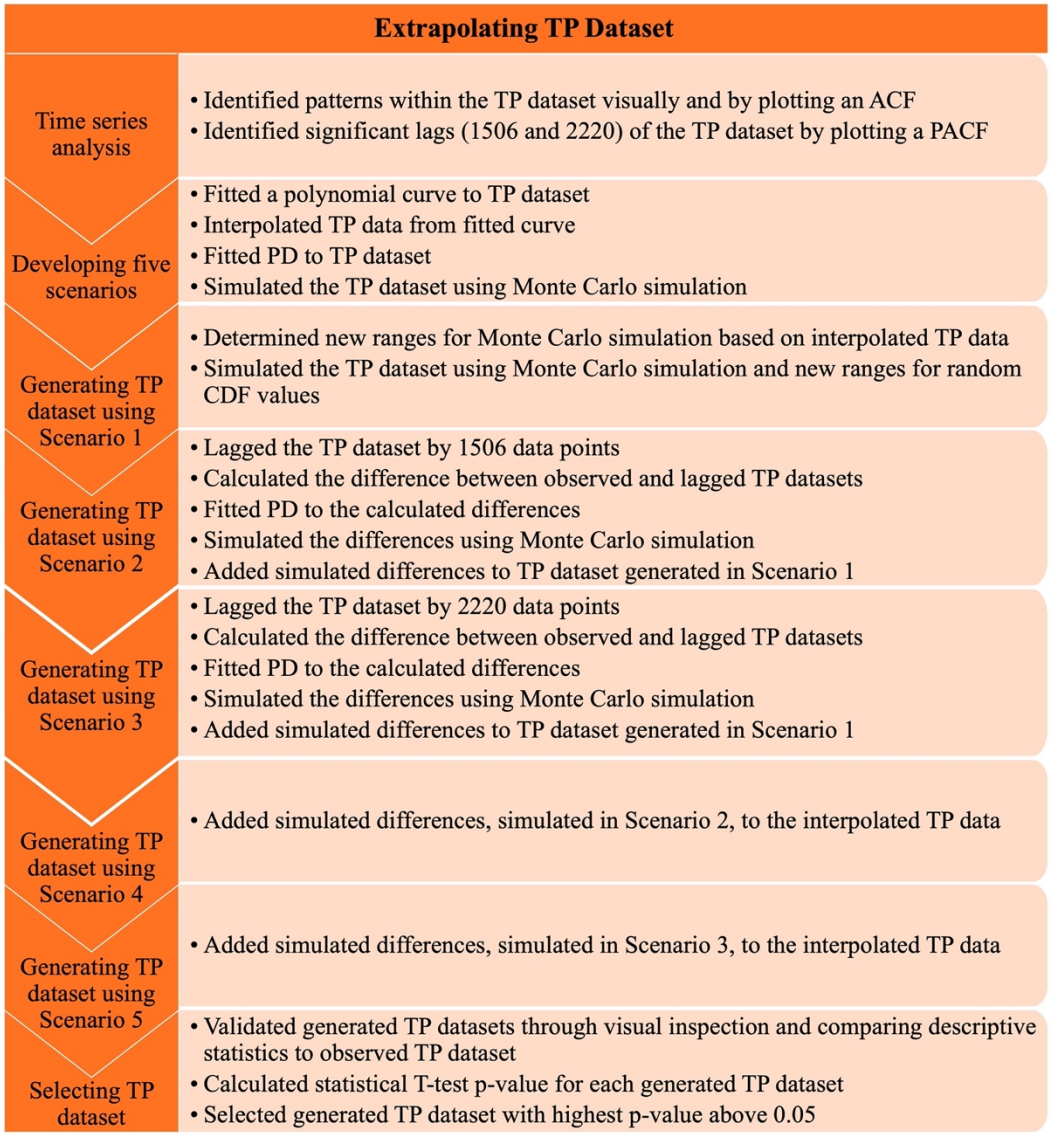


Figure 38: Process flow diagram of method used to extrapolate TP dataset

The details of the process outlined in the figure are discussed in full in Appendix A. To summarise the time series analysis, the TP dataset displayed annual seasonal patterns with peaks occurring at the beginning of each year. Cycles were also evident whereby TP data from 2014 to 2019 had a cycle similar to that seen in 1980 to 1985. The plotted ACF confirmed these trends and a lag of 2220 was identified as the point at which the two cycles matched. This was important as the cycle seen in 1980 to 1985 had a decreasing trend that was not clearly visible at the end of the dataset in July 2018. Carroll and Curtis (2021) compared years from 2010 to 2016 and noted an increase in nutrients. However, it is clear that, although there is an increase in TP from 2010 onwards, the ACF proved that there was a cycle present, and the TP is expected to decrease again. Therefore, it was assumed that the TP remained the same for the period 1 August 2015 to 28 December 2019 as seen in the observed TP from 1980 to 1985.

The PACF identified the most significant lag to be 1506, which was then used in generating the TP dataset in Scenario 2, as stated in the process flow diagram. Although the TP data is a time series dataset with seasonality, literature suggested to avoid using time series models due to its

unreliability the further away the forecast is from the known data (Pourahmadi, 2001). Therefore, time series models were not used to extrapolate the TP dataset.

Two other methods were tested, and then combined to develop the five scenarios: Monte Carlo simulation and interpolation from a curve. The first depends on randomness and thus the simulated TP data was too random and did not reflect any trends evident in the observed TP dataset. The latter was used to generalise a curve that represents the annual seasonal curve evident in the observed TP dataset. The daily average TP data was calculated but only using data from 1980 to 1985 as this was the cycle that matched the timeline being generated (2015 to 2019). Polynomial curves were fitted to this data and a polynomial with 4 degrees was chosen as it had the lowest RMSE. Next, the daily average TP data was interpolated using the equation of the curve for the dates required being from 1 August 2015 to 28 December 2019. These interpolated values were used to determine the range for each day in the year from which random values for Monte Carlo simulation could be within.

Using the information and established methods shown in Figure 38, five TP datasets were generated. Five scenarios used Monte Carlo simulation or interpolating from a curve, with randomness being added or removed through lagging, to compare which of the two methods resulted in the best possible TP dataset.

Scenario 1 generated a TP dataset by first fitting the dataset to a Lognormal PD, which was the best PD according to the goodness-of-fit AD test. Random numbers were then generated for each date in the required timeline but within the range of the calculated CDF based on the interpolated TP values for each date. Monte Carlo simulation was then used to simulate the TP dataset. Scenario 2 used the generated TP dataset from Scenario 1 and added simulated differences to include more variability based on the seasonal trend found. The differences were calculated by subtracting the TP dataset lagged by 1506 data points from the observed TP dataset, and then simulated using Monte Carlo simulation. Similarly, Scenario 3 lagged the TP dataset by 2220 data points and generated a TP dataset in the same way. Scenario 4 and 5 used the differences simulated in Scenarios 2 and 3, respectively, and added those differences to the interpolated TP data for the required timeline.

The TP dataset generated in Scenario 5 had the highest statistical test p-value of 0.9 and was thus the best dataset. Figure 39 shows the dataset. The statistical test used was the T-test which determined if the generated TP dataset came from a PD with the same mean and variance as the PD of the observed TP dataset (Curran, 2013). Although Nguyen et al. (2017) suggested using the KS test, it was not used as the p-values were too similar for one dataset to be chosen.

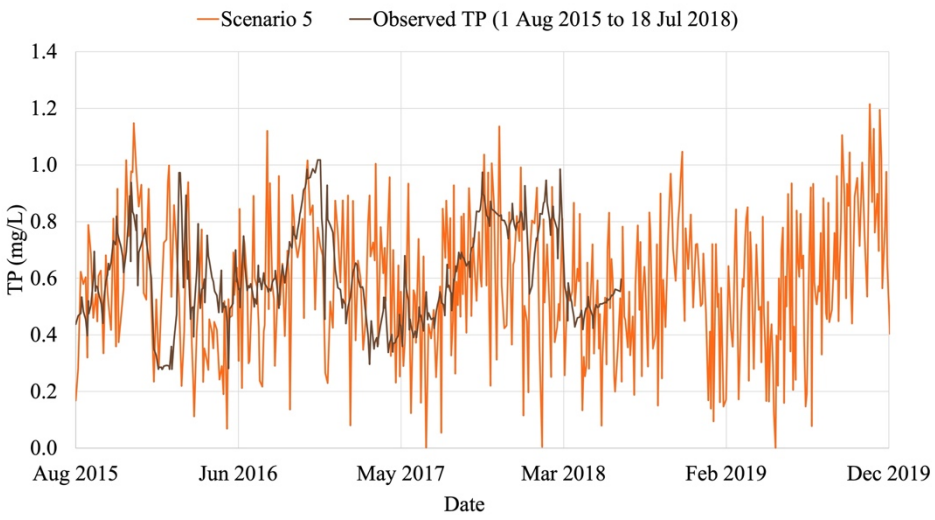


Figure 39: Time series plot of the chosen TP dataset generated in Scenario 5

6.4 Extrapolating TN Dataset

The method used to extrapolate the TN dataset consisted of the same steps as followed when extrapolating the TP Dataset, with only a few differences. Figure 40 summarises the steps in a process flow diagram that are discussed in detail in Appendix B. This section will summarise the findings of the time series analysis as well as present the generated TN dataset that was later used in model building.

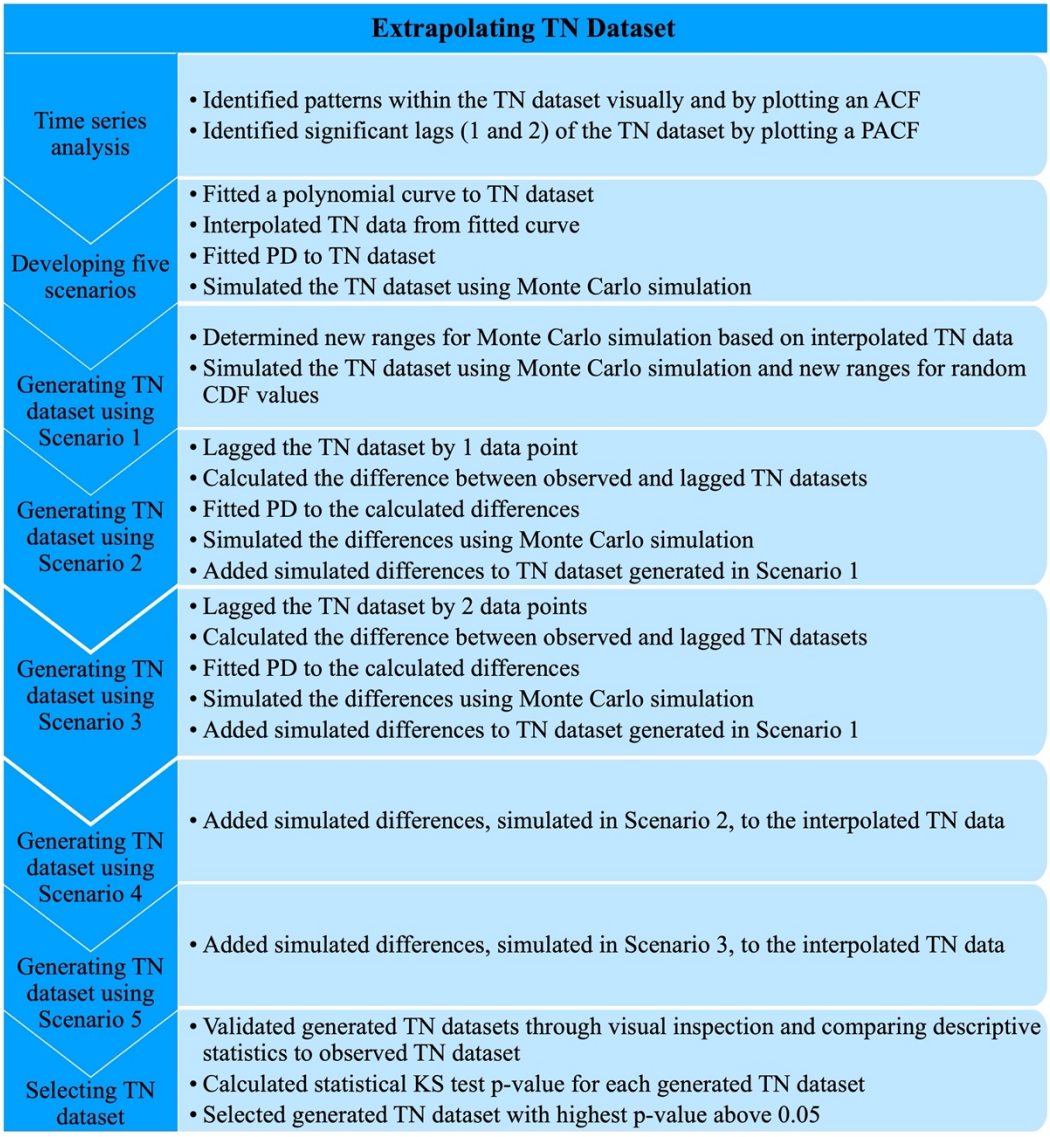


Figure 40: Process flow diagram of method used to extrapolate the TN dataset

The time series analysis revealed annual seasonal trends with peaks occurring in the middle of each year, that was also confirmed using the plotted ACF. In addition to the seasonal curves, there was also a cycle that started its decreasing trend towards the end of the dataset in 2011. Therefore, the same assumption as in Section 6.3 was made for extrapolating the TN dataset – the TN data can be simulated using the past data as it is expected to be the same and not worse than the past observed data.

The PACF identified lag 225 to be the most significant, however the TN dataset only had 315 data points. Thus, this lag was not used as the remaining data points after lagging would be too little to use for generating the TN dataset. Instead, lags 1 and 2 were used as these were also significant.

Similarly to TP, the fitted polynomial curve was one with 4 degrees. The Weibull PD was fitted to the observed TN dataset as chosen using the goodness-of-fit AD test. Using the fitted polynomial curve and PD, five scenarios were developed and used to generate TN datasets, using

the same method as for the TP dataset, with the only difference being the number of lags used in Scenarios 2 and 3.

The KS test was used to determine which scenario generated the best possible TN dataset. The chosen TN dataset was generated in Scenario 3 as it had the highest statistical p-value. Figure 41 shows this dataset which was later used in model building.

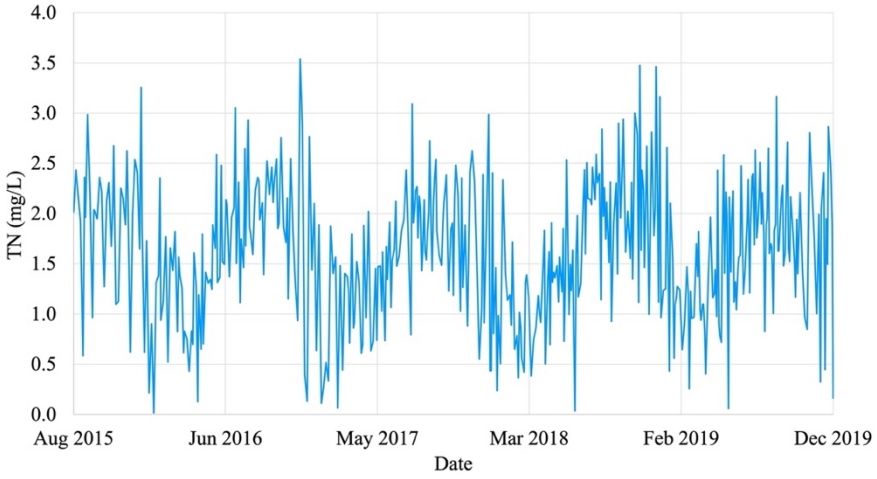


Figure 41: Time series plot of the chosen TN dataset generated in Scenario 3

6.5 Finalised Datasets

The features were split into train, test and unseen datasets during model building. The unseen data was used to make final predictions and was from 1 June 2018 to 28 December 2019. This data was 1.5 years of the total dataset. The purpose of the unseen dataset was to evaluate the model from the user's point of view as the user will input a dataset similar to the unseen dataset. The remaining data being 275 data points out of the 461 in total, was split randomly into training and testing datasets using a 80:20 ratio (Raschka, 2015; Oancea & Ciucu, 2014). Figure 42 shows the input target variable, WHC, and which data points formed part of the subsets created, while Figure 43 shows the finalised features.

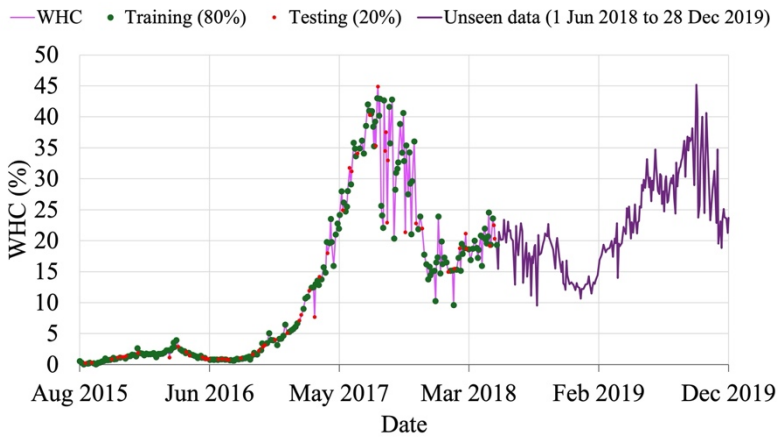


Figure 42: Time series plot of finalised WHC dataset used as the input target variable

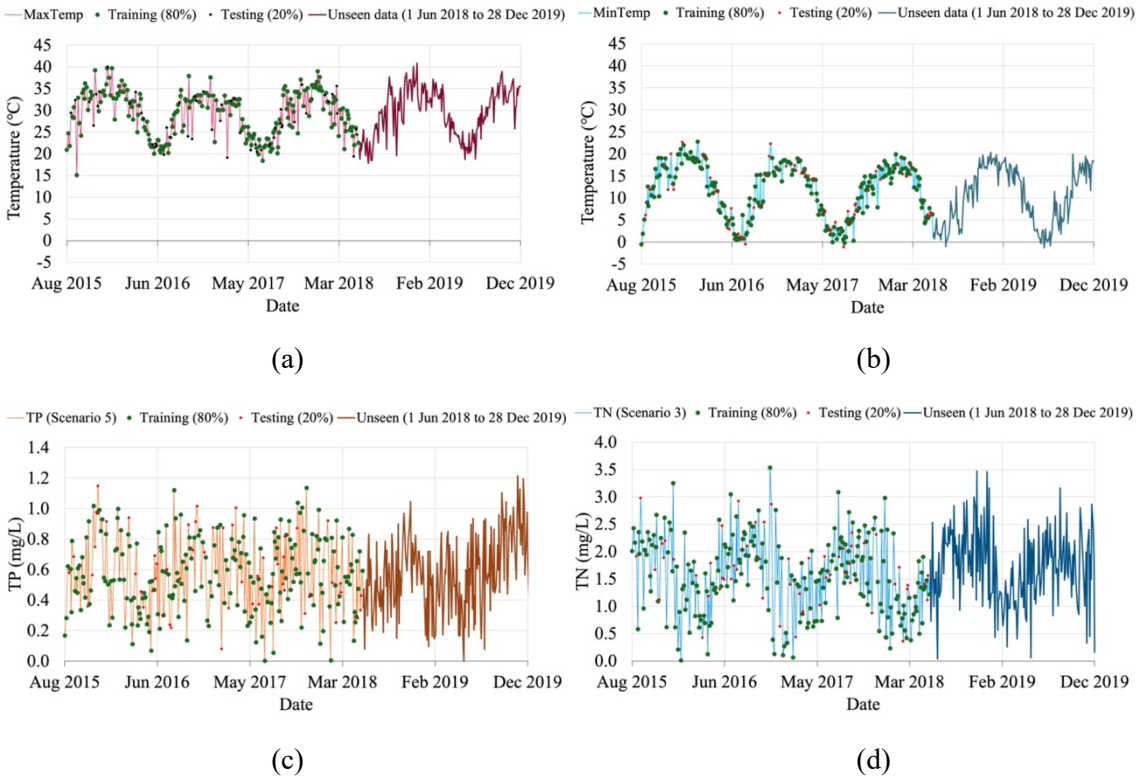


Figure 43: Time series plot of finalised MaxTemp (a), MinTemp (b), TP (c) and TN (d) features

7 MODEL BUILDING

This chapter presents the method used to create an ML model to predict WHC. The methods utilised are based on the literature review in Chapter 4. Figure 44 outlines the sections discussed in this chapter.

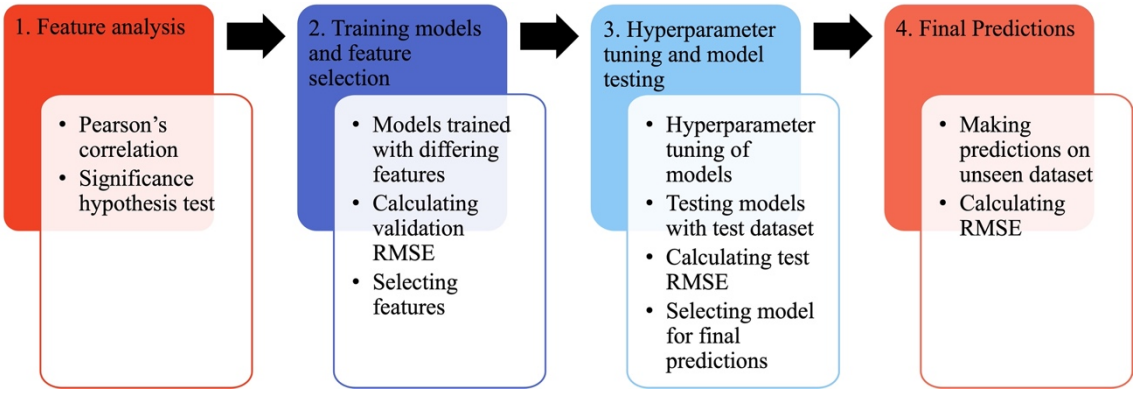


Figure 44: Process flow diagram of the methods followed in model building

Feature analysis is discussed first. Two analyses were performed being Pearson's correlation, which identified features with similar information, and a hypothesis test that tested features for significance. Next, the ML models were developed by training the five models a few times, each with varying features. This way, the best possible features could be selected based on the lowest resulting RMSE. Using the selected features, models were built using the test dataset so that hyperparameters could be tuned. The model with the lowest test RMSE was then selected and used to make final predictions on the unseen dataset. The final model and its predictions were evaluated by means of confidence interval calculations and the RMSE.

All the models built throughout the model building process were verified. Verification is knowing the models performed as they were designed to perform (Tran, 1999; Thacker et al., 2004). The models were verified by ensuring no errors in the codes and that the predictions were as expected – within range of the observed WHC.

7.1 Feature Analysis

The features used in building the predictive models to predict WHC were Date, MaxTemp, MinTemp, and the generated TP and TN datasets. Dates were converted to serial numbers so that the analyses could be made. Although there were only five features in total, Marini (2009) proved that having the best possible features is vital for good model performance. This meant that some features could have required transformation or adjustment to result in good final predictions. Two feature analyses were done to determine potential features that would positively influence the performance of the model; the first was Pearson's correlation followed by a hypothesis F-test.

7.1.1 Pearson's correlation

The first analysis done was Pearson's correlation, where the correlations between each feature, were calculated using MATLAB. The high correlated features meant they shared similar

information, thus making them redundant. Kuhn & Johnson (2016) and Kubat (2017) suggest avoiding redundant features as they influence the trends the model learns. Thus, to replace both correlating features, new features were created which combine the ones that correlate.

The correlations are shown in the matrix in Table 33. The light pink shaded cells represent variables that were strongly correlated, being a correlation greater than 70% (Raschka, 2015).

Table 33: Pearson's correlation matrix for features

	Date	MaxTemp	MinTemp	TP	TN
Date		0.02	-0.14	0.04	0.01
MaxTemp	0.02		0.73	0.27	-0.05
MinTemp	-0.14	0.73		0.21	-0.20
TP	0.04	0.27	0.21		0.03
TN	0.01	-0.05	-0.20	0.03	

MaxTemp and MinTemp correlate 73% meaning that the two features shared 73% of the same or similar information. This makes sense as both features are temperature datasets. Thus, the average of MaxTemp and MinTemp per point, called AvgTemp, was calculated to replace both features. Figure 45 shows the AvgTemp feature and how it was split into training and testing datasets for model building and the portion for final predictions.

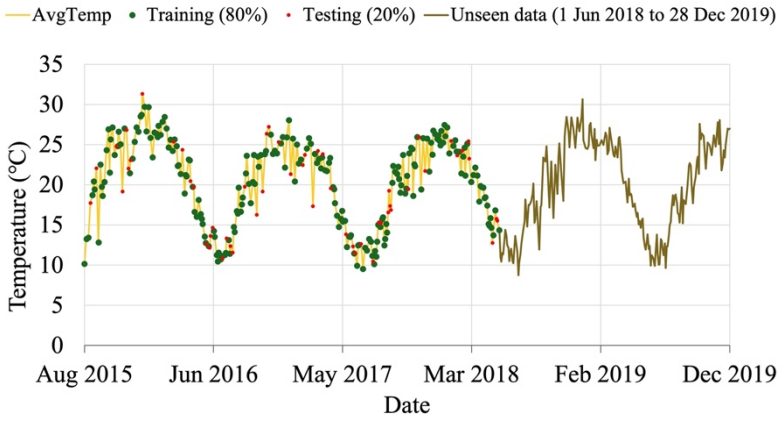


Figure 45: Time series plot of the AvgTemp feature

7.1.2 F-test hypothesis test

MATLAB was used to perform an F-test to determine if all the features should be used to describe the dataset and were significant. MATLAB returns the F-test statistic which is equal to the $-log(p)$, where p is the p-value (MATLAB, 2017). Thus, the p-values were calculated using Equation (10).

$$p = 10^{-(Ftest\ statistic)} \tag{10}$$

Features with a p-value below 0.05 were considered significant, with the most significant being close to 0 (Kroese et al., 2020). Figure 46 illustrates that Date, MaxTemp, MinTemp and AvgTemp were all significant as the p-values were 0. This makes sense as the Date and temperature datasets are influenced by seasonal changes like WHC. The TN and TP features had

p-values of 0.67 and 0.98, respectively, indicating that both were insignificant features. However, literature proved nutrients to play a vital role in WHC (Fernando, 2003; Rojas-Sandoval & Acevedo-Rodríguez, 2022). Hence the addition of a feature that combined both TP and TN. This additional feature is the ratio of TN to TP (TNTP) which is often used in determining the limiting factor in eutrophication (Bergström, 2010; Ptacnik et al., 2010). A time series plot of this new feature is shown in Figure 47. The TNTP feature had a p-value of 0, as shown in Figure 46, and was thus significant.



Figure 46: Hypothesis F-test p-values of each feature

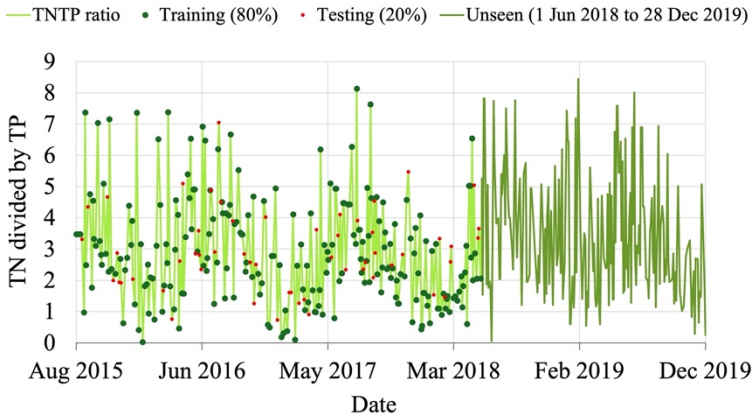


Figure 47: Time series plot of TNTP feature

Using this information and the correlation results, four combinations of features were established and used to build the supervised ML models.

7.1.3 Possible combinations of features

The feature analysis implied that the original features were not the best possible features due to redundant or insignificant features. The four feature combinations were used to train each of the five models to determine the best combination of features. Various representations of the Date feature were tested, such as having month, day and year separate, but representing this feature as a serial number resulted in the best RMSE values and the feature was therefore left as a serial number. Table 34 lists the combinations and the reason each option was chosen.

Table 34: Combinations of features used to train models

		Features							Reason for Choice
		Date	Max-Temp	Min-Temp	Avg-Temp	TP	TN	TNTP	
Combination	A	X	X	X		X	X		Original features
	B	X			X	X	X		Replacing correlating features with one that combines both
	C	X	X	X				X	Replacing insignificant features with one that combines both
	D	X			X			X	Replacing correlating and insignificant with the new features.

7.2 Training Models and Feature Selection

As explained in Chapter 6, the datasets were split into three sub-datasets called training, testing and unseen data. As a reminder, the training dataset was used to train the models while the testing dataset was used to test the models after hyperparameter tuning. The unseen dataset was used to test the model from the user point of view. The user would input data similar to that in the unseen dataset and expect accurate WHC predictions for the dates inputted. The user would not train the model with part of their dataset, thus the unseen dataset was kept aside to test the model in this way.

To start training the models, the training datasets was imported into the regression learner app in MATLAB, where cross-validation with 5 folds was used to further split the training dataset during training to ensure validation was carried out throughout training (Russell, 2018). Five types of supervised regression ML models were chosen based on literature: ensemble (boost), ensemble (bag), decision tree, ANN and SVM (Kuhn & Johnson, 2016; Theobald, 2017). The ensemble (boost) model used the boosting algorithm, which had a default learning rate of 0.1. The ensemble (bag) used the bagging algorithm, which makes use of multiple decision trees. Even though the ensemble (bag) was expected to perform better than the decision tree model, the decision tree was still included in the study for completeness.

Each of the five models were then trained four times, each time with a difference feature combination. The hyperparameters of the models were the default values so that the effects of the different feature combinations could be evaluated. The average RMSE of the models using the same combination of features was calculated to determine which of the four combinations resulted in the lowest average RMSE. This combination was selected and the models were further developed.

7.3 Hyperparameter Tuning and Model Testing

To further develop the models after the best feature combination was chosen, the hyperparameters of each model were tuned. This was done in MATLAB using the Bayesian Optimiser which iteratively altered the parameters of the model until it completed 30 iterations. For each iteration, the validation RMSE was calculated and the parameters of the iteration that resulted in the lowest RMSE were chosen. Once the hyperparameters were tuned, the models were tested using the test dataset. The model with the lowest test RMSE was the finalised model from which the final predictions were made.

7.4 Final Predictions

The unseen dataset containing data from 1 June 2018 to 28 December 2019 was used to make final predictions in the same way as a user of the model would. This way, the model was tested from the user's point of view. The RMSE was calculated in MATLAB as done previously and then discussed in Chapter 9.

8 RESULTS

This chapter presents the results obtained during model building as well as the final predictions made. The results include: the RMSE of each model trained, the best combination of features that was chosen based on the average validation RMSE of those models, the hyperparameters chosen, the test RMSE of the models tested, and the details and RMSE of the final model. The chapter ends with the final predictions being plotted and then compared to an alternative model to determine ways to improve the model; this alternative model is discussed in Chapter 9 as well.

8.1 Model Building

The five models were trained using the different combination of features. MATLAB calculated a validation RMSE for each model trained as shown in Figure 48. It is evident that the SVM models did not perform as well as the other models as on average, its validation RMSE was 8.94. The decision trees models performed slightly worse than the ensemble (bag) models, as expected. Thus, both the SVM and decision tree models were excluded from further development.

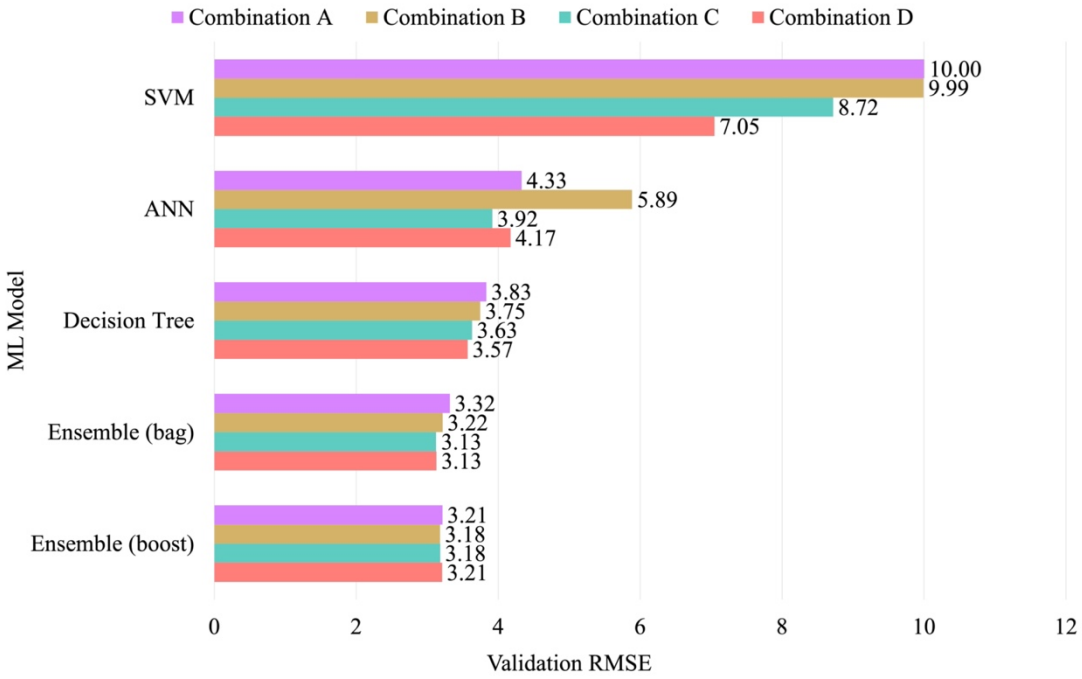


Figure 48: Validation RMSE for each model trained using different feature combinations

The remaining three models being the ensemble (boost), ensemble (bag) and ANN models, were used to determine which of the four feature combinations resulted in the best performance overall. To do so, the average validation RMSE of each model using the same feature combination was calculated. Figure 49 compares the average validation RMSE of each feature combination.

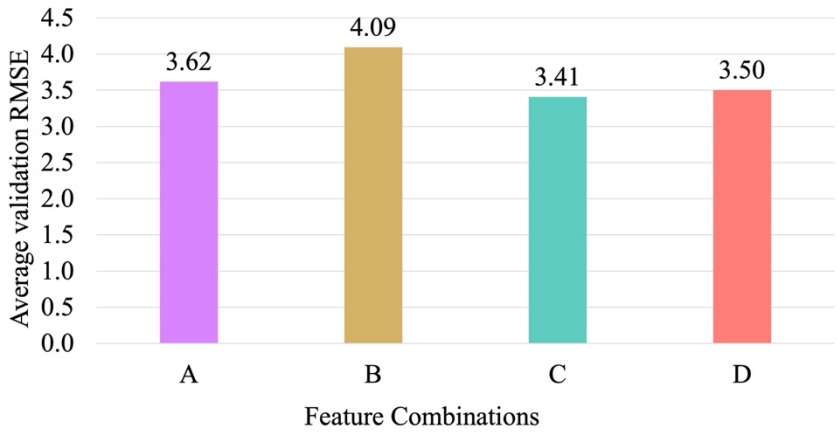


Figure 49: Average validation RMSE for each feature combination

The feature combination that resulted in the lowest validation RMSE of 3.41 was combination C which consisted of the Date, MaxTemp, MinTemp and TNTP features. These features were used to further develop the three models.

Next, the hyperparameters of these three models were tuned then tested using the test dataset. The selected hyperparameters are listed in Table 35.

Table 35: Hyperparameters of each model using feature combination C

Model Type	Hyperparameters
Ensemble (boost)	Minimum leaf size: 9 Number of learners: 500 Learning rate: 0.10266
Ensemble (bag)	Minimum leaf size: 2 Number of learners: 13
ANN	Number of hidden layers: 1 Size of hidden layer: 182 Activation: ReLU Regularization strength (Lambda): 0.01097 Iteration limit: 1000 Standardize data: Yes

The test RMSE for each model are compared in Figure 50. The ensemble (bag) had the lowest test RMSE of 2.73 and, thus, was the chosen final model.

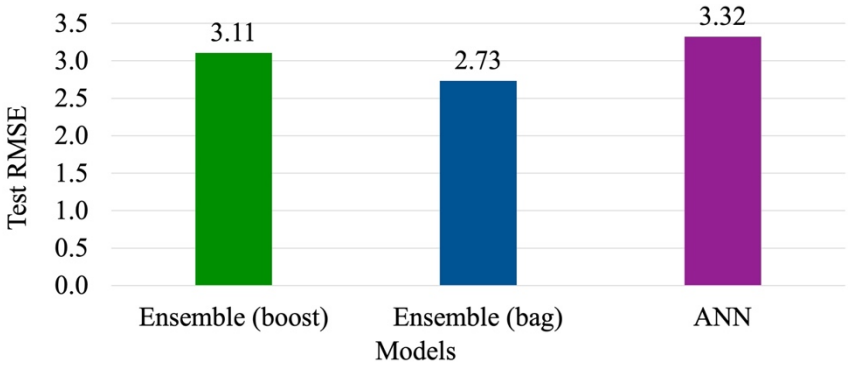


Figure 50: Test RMSE for each model after hyperparameter tuning

The predictions were plotted together with the observed WHC dataset put aside in the test dataset in Figure 51. The predictions were close to the observed WHC as the low test RMSE suggested.

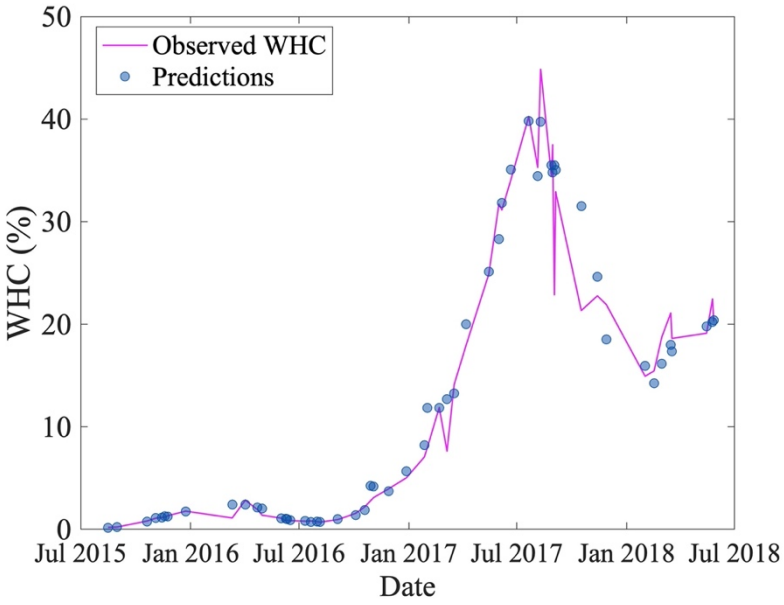


Figure 51: Scatter plot of the test predictions made by the ensemble (boost) model

Final predictions were then made using the unseen dataset to imitate how a user would use the model.

8.2 Final Predictions

The final ensemble (bag) model made predictions shown in Figure 52 with a RMSE of 7.31.

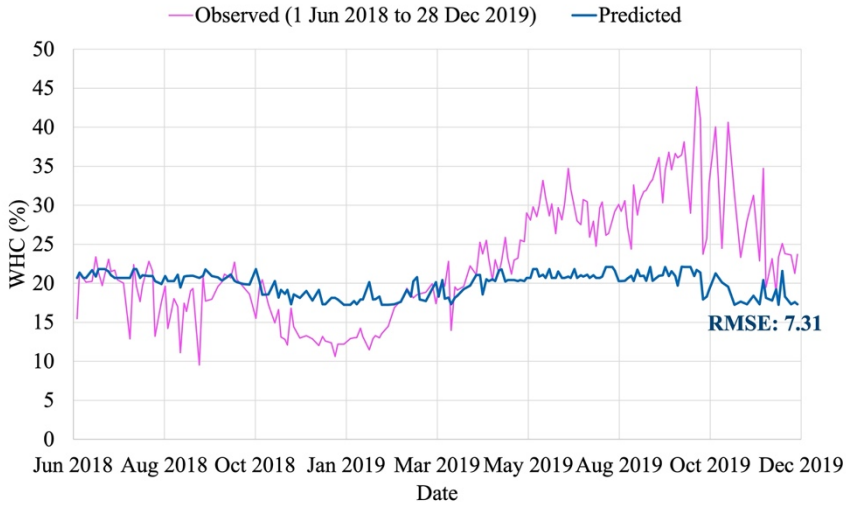


Figure 52: Time series plot of final WHC predictions

The model performed worse than it did on the test dataset with a difference of 4.58. However, it is clear that the model struggles to make predictions from May 2019 as the observed WHC from May 2019 was much more volatile than before (see Chapter 6). Therefore, data after 1 May 2019 was removed and the RMSE was recalculated. Figure 53 focuses on the data before May 2019.

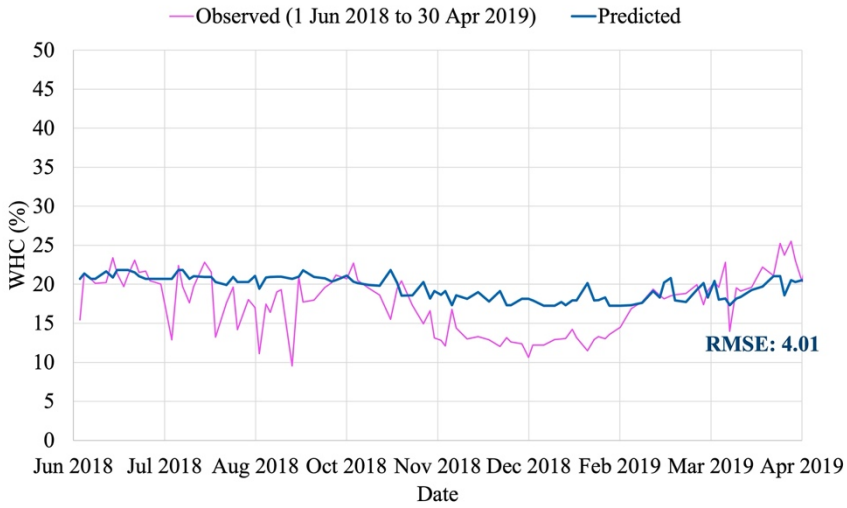


Figure 53: Time series plot of final WHC prediction from 1 June 2018 to 30 April 2019

The figure shows that the predictions before May 2019 were close to the observed WHC data with a RMSE of 4.01, which was only 1.28 higher than the test RMSE. This proves that the volatility seen in the WHC data from May 2019 made it difficult for the model to make accurate predictions for those dates.

This was compared to an ensemble (bag) model that was built using training, testing and unseen datasets that were split randomly as is normally done in ML (Bonaccorso, 2018). This way, the data was not treated as a time series dataset with timelines indicating the split between the training and unseen datasets. The unseen dataset was 1.5 years of data which consisted of 186 data points out of the total 461 data points. Thus, 186 datapoints were randomly selected to make-up the new unseen dataset. Then the remaining data was split into training and testing dataset using a ratio of 80:20. This was the same ratio used for the first models developed (see Chapter 6).

The ensemble (bag) model with the same hyperparameters was then trained, tested and re-tested using the unseen dataset. The RMSE results are compared in Figure 54.

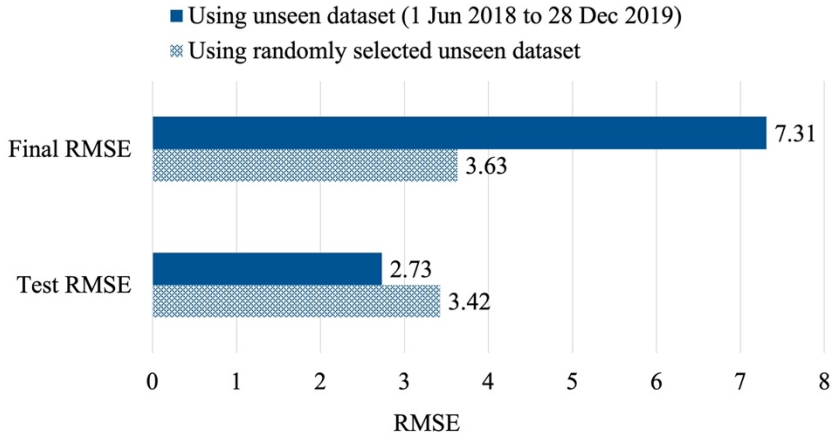


Figure 54: Comparison of RMSE of model using timeline and randomly selected unseen dataset

When the dataset was split randomly, its performance was more consistent and low with a final RMSE of 3.63. This was much lower than the 7.31 obtained by the model that treated the dataset as a time series with fixed timelines. Figure 55 plots the predictions and show how close the predictions were to the observed WHC compared to when the dataset was split by time.

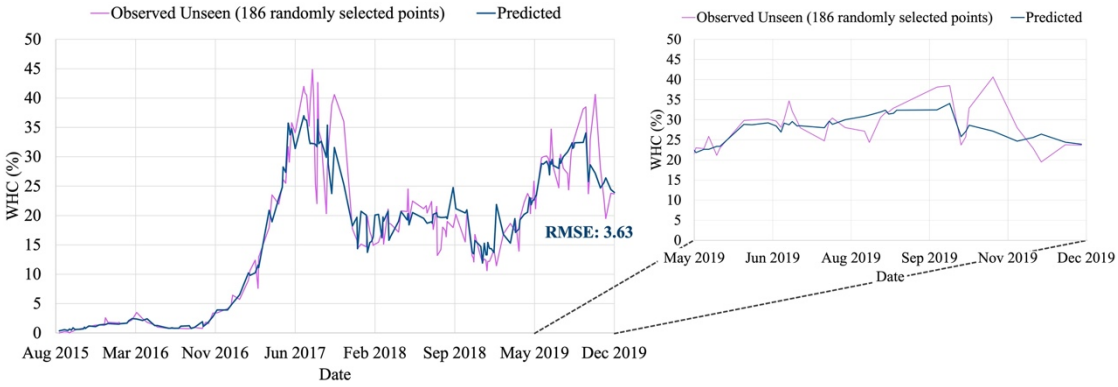


Figure 55: Final predictions of model using randomly selected unseen dataset

The figure also shows that the model could actually make accurate predictions for the dates after May 2019. This is because the training dataset consisted of some random points from May 2019 to December 2019, thus the volatility seen could be learnt. In conclusion, the model made good predictions up to 11 months after the training dataset, when fixed timelines were used, but made better predictions when the dataset was split randomly.

8.3 Discussion

WHC predictions for 1 June 2018 to 28 December 2019 were made using the ensemble (bag) model that used the Date, MaxTemp, MinTemp and TNTP features. The chosen features reflected the feature analysis results as the features chosen were those that were significant. The ratio of TN to TP allowed the model to learn patterns from both nutrients that were not evident when kept separately. This was seen in the feature analysis as both TP and TN were insignificant individually, and the models that used these features performed worse in terms of RMSE. The RMSE of the predictions made by the final ensemble model was 7.31. This was higher than the test RMSE of 2.73, however, the ensemble (bag) model captured the seasonality evident in the observed WHC relatively well up to about May 2019. The RMSE without predictions after 1 May 2019 was 4.01. This is almost half of the RMSE for all the predictions as a whole which is significantly lower. This implied that the model could not learn the patterns required to accurately predict WHC for those months. The data seen in May 2019 onwards were very volatile compared to the rest of the dataset. This could explain why the predictions were less accurate as the training dataset did not consist of data from those months.

New predictions were made on the same model that was trained and tested using different training, testing and unseen datasets. This was done to establish whether the model would perform better if the training, testing and unseen datasets consisted of random points from across the whole

dataset, instead of being stipulated according to a timeline. The model resulted in final predictions with a RMSE of 3.63. This proved that the model could accurately predict the whole dataset when the dataset was split randomly. However, WHC data is a time series dataset with seasonal patterns and thus splitting the training, testing and unseen according to a timeline is more appropriate. The ensemble (bag) model developed made good predictions and was close to its performance during testing up to May 2019. This meant that the model could make accurate predictions up to 11 months, from 1 June 2018 to 1 May 2019, after the known data.

9 CONCLUSIONS AND RECOMMENDATIONS

This research aimed to accurately predict WHC for 1 June 2018 to 28 December 2019. Data from 2020 to 2021 were excluded from this research due to the impact of the COVID-19 pandemic and consequent limitations on human movement that indirectly influence WHC on the dam. In summary, due to the many factors that influence the percent coverage of water hyacinth on the Hartbeespoort dam, the problem was simplified to only include those factors considered as the most influential. Those factors were date, maximum and minimum temperatures, and the nutrients in the dam in the form of TP and TN. The data obtained was limited and required cleaning and handling of missing data before they could be used in model building.

Multiple KNN imputation was used to impute the missing data within all the datasets. The quality of imputations was validated visually, with all imputations representing the observed datasets well within a 95% confidence interval range. Thereafter, ways to extrapolate the TN and TP datasets were developed and tested, as the datasets did not have data for dates that coincided with the other datasets. The generated TP and TN datasets, chosen based on statistical test p-values, were used to build the supervised regression ML models.

Five different types of models were built using different feature combinations based on correlation and significance. The features chosen for the final model were the Date, MaxTemp, MinTemp and TNTP features. The chosen model with the lowest test RMSE was the ensemble (bag) model. This model resulted in a RMSE of 4.01 for predictions made from 1 June 2018 to 1 May 2019. The model made accurate predictions for dates up to 11 months in advance and thus, planning biological control based on these predictions is beneficial. In conclusion, the model can be used in future on new unseen data for decision making.

Recommended uses of the model include using predictions for:

- biological control planning,
- tourism and water sport planning,
- fertiliser manufacturers to plan harvesting WHC based on potential WHC yields, and
- ecosystem studies including algae growth and fish death.

For improved predictions, it is recommended to incorporate plant hoppers as a feature for predictions to closer resemble observed WHC data as it could be a reason for the volatility seen. Additionally, should data be available in the future for more recent dates, a repeat of the predictions could be made to investigate more recent results.

REFERENCES

Abebe, T., Awoke, B.G. and Nega, W. (2023) 'Spatiotemporal patterns of water hyacinth dynamics as a response to seasonal climate variability in Lake Tana, Ethiopia', *Applied Water Science*, 13(170). doi:10.1007/s13201-023-01975-6.

Afrifa-Yamoah, E., Mueller, U.A., Taylor, S.M. and Fisher, A.J. (2020) 'Missing data imputation of high-resolution temporal climate time series data', *Meteorological Applications*, 27(1), pp. 1-18. doi:10.1002/met.1873.

Agjee, N.H., Mutanga, O., Gebreselasie, M. and Ismail, R. (2018) 'A comparison of regression tree approaches to modelling the efficacy of water hyacinth biocontrol using multitemporal spectral datasets', *Journal of Spectroscopy*, 2018(6), pp. 1–11. doi:10.1155/2018/7960314.

Al-Kassab, M.M. and Majeed, A.H. (2022) 'The use of two-sample t-test in the real data', *Advances and Applications in Statistics*, 81, pp. 13–22. doi:10.17654/0972361722071.

Andika, D.O., Ogada, J.A. and Hayombe, P.O. (2016) 'Producing liquid organic fertilizer from water hyacinth; a case of Lake Victoria, Kenya', *International Journal of Science and Research (IJSR)*, 5(2), pp. 1229–1238. doi:10.21275/v5i2.nov161013.

Aoyama, I. and Nishizaki, H. (1993) 'Uptake of nitrogen and phosphate, and water purification by water hyacinth Eichhornia crassipes (Mart.) Solms', *Water Science and Technology*, 28(7), pp. 47–53. doi:10.2166/wst.1993.0141.

Auchterlonie, J., Eden, C.-L. and Sheridan, C. (2021) 'The phytoremediation potential of water hyacinth: A case study from Hartbeespoort Dam, South Africa', *South African Journal of Chemical Engineering*, 37, pp. 31–36. doi:10.1016/j.sajce.2021.03.002.

Ballance, R. and Bartram, J. (1996) Water quality monitoring a practical guide to the design and implementation of freshwater quality studies and monitoring programmes, iris WHO. London: UNEP/WHO. Available at: https://iris.who.int/handle/10665/41851 (Accessed: 28 February 2024).

Batista, G.E. and Monard, M.C. (2002) 'A Study of K-Nearest Neighbour as anImputation Method', in *Proceedings of Soft Computing Systems: Design, Management and Applications, HIS 2002*, December 1-4, Santiago, Chile. Available at: https://www.researchgate.net/publication/220981745_A_Study_of_K-Nearest_Neighbour_as_an_Imputation_Method (Accessed: 28 February 2024).

Bayable, G. et al. (2023) 'Detection of water hyacinth (eichhornia crassipes) in Lake Tana, Ethiopia, using machine learning algorithms', *Water*, 15(5), p. 880. doi:10.3390/w15050880.

Bergström, A.K. (2010) 'The use of TN:TP and DIN:TP ratios as indicators for phytoplankton nutrient limitation in oligotrophic lakes affected by n deposition', *Aquatic Sciences*, 72(3), pp. 277–281. doi:10.1007/s00027-010-0132-0

Bhusal, A. and Devkota, A. (2020) 'Physico-Chemical characteristics of Lakes of Chitwan National Park, Central Nepal', *Biological Forum - An International Journal*, 12(1), pp. 33–39. ISSN 2249-3239.

Bonaccorso, G. (2018) Machine learning algorithms: Popular algorithms for Data Science and Machine Learning. 2nd edn. Birmingham, United Kingdom: Packt Publishing.

Briese, D.T. (2000) 'Classical Biological Control', in Australian Weed Management Systems. 1st edn. Melbourne: R.G. and F.J. Richardson Publishers (B.M. Sindel), pp. 161–186.

Brockwell, P.J. and Davis, R.A. (2016) Introduction to Time Series and Forecasting. 3rd edn, Springer Link. 3rd edn. Cham, Switzerland: Springer Cham, pp.2197-4136.

Carroll, A. and Curtis, C. (2021) 'Increasing nutrient influx trends and remediation options at Hartbeespoort Dam, South Africa: A mass-balance approach', *Water SA*, 47(2), pp. 210–220. doi:10.17159/wsa/2021.v47.i2.10917.

CBC (2020) *CBC releases status and future expectations for its Hartbeespoort Dam Water Hyacinth Project, Rhodes Latest News*. Available at: https://www.ru.ac.za/latestnews/archives/2020/cbcreleasesstatusandfutureexpectationsforitshartbeespoortdamwaterhyac.html (Accessed: 28 February 2024).

CBC (2020a) *Bugs are winning the war against Hartbeespoort's hyacinths, Rhodes Latest News*. Available at: https://www.ru.ac.za/latestnews/archives/2020/bugsarewinningthewaragainsthartbeespoortshyacinths.html#:~:text=Billions%20of%20tiny%20insects%20are,on%20the%20dying%20water%20plants. (Accessed: 28 February 2024).

CBC (2020b) *The Centre for Biological Control, Centre for Biological Control*. Available at: https://www.ru.ac.za/centreforbiologicalcontrol/ (Accessed: 28 February 2024).

Chen, J. et al. (2021) 'Simulation of Water Hyacinth Growth Area based on multi-source Geographic Information Data: An integrated method of woe and AHP', *Ecological Indicators*, 125, pp.107574. doi:10.1016/j.ecolind.2021.107574.

Cheng, Q., Yuan, Q., Ng, M., Shen, H. and Zhang, L. (2019) 'Missing data reconstruction for remote sensing images with weighted low-rank tensor model', *IEEE Access*, 7, pp. 142339–142352. doi:10.1109/access.2019.2944577.

Coetzee, J.A. and Hill, M.P. (2011) 'The role of eutrophication in the biological control of water hyacinth, eichhornia crassipes, in South Africa', *BioControl*, 57(2), pp. 247–261. doi:10.1007/s10526-011-9426-y.

Coetzee, J.A., Miller, B.E., Kinsler, D., Sebola, K. and Hill, M.P. (2022) 'It's a numbers game: Inundative biological control of water hyacinth (pontederia crassipes), using megamelus scutellaris (Hemiptera: Delphacidae) yields success at a high elevation, hypertrophic reservoir in South Africa', *Biocontrol Science and Technology*, 32(11), pp. 1302–1311. doi:10.1080/09583157.2022.2109594.

Coetzee, J.A. (2012) 'Meteorological weather station data can be used in climate matching studies of Biological Control Agents', *Biocontrol Science and Technology*, 22(4), pp. 419–427. doi:10.1080/09583157.2012.662477.

Coetzee, J.A. (2022) *Water Hyacinth Prediction Masters at Wits*.

Cryer, J.D. and Chan, K. (2008) Time Series Analysis with applications in R. 2nd edn. New York: Springer New York, pp.2197-4136.

Curran, J.M. (2013) 'The frequentist approach to forensic evidence interpretation', *Encyclopedia of Forensic Sciences*, 1(2), pp. 286–291. doi:10.1016/b978-0-12-382165-2.00194-x.

Dangeti, P. (2017) Statistics for Machine Learning: Techniques for exploring supervised, unsupervised, and reinforcement learning models with python and R. Birmingham, United Kingdom: Packt Publishing Ltd.

De Paoli, F., Bishara, R.H. and van Asselt, E.J. (2021) 'How to define the right ambient temperature range for storage and distribution of Pharmaceutical Raw Materials', *Biologicals*, 69, pp. 66–69. doi:10.1016/j.biologicals.2020.12.001.

Dersseh, M.G., Melesse, A.M., Tilahun, S.A., Abate, M. and Dagnew, D.C. (2019) 'Water hyacinth: Review of its impacts on hydrology and Ecosystem Services—lessons for management of Lake Tana', *Extreme Hydrology and Climate Variability*, pp. 237–251. doi:10.1016/b978-0-12-815998-9.00019-1.

Dinov, I.D. (2018) Data Science and Predictive Analytics: Biomedical and health applications using R. Cham, Switzerland: Springer International Publishing.

Diouf, S., Dème, E. and Dème, A. (2022) 'Imputation methods for missing values: The case of Senegalese meteorological data', *African Journal of Applied Statistics*, 9(1), pp. 1245–1278. doi:10.16929/ajas/2022.1245.267.

Dong, Y. and Peng, C.-Y.J. (2013) 'Principled missing data methods for researchers', SpringerPlus, 2(1), p. 222. doi:10.1186/2193-1801-2-222.

DWS (2019) *National Eutrophication Monitoring Programme phosphorus and chlorophyll summaries, National Eutrophication Monitoring Programme*. Available: https://www.dws.gov.za/iwqs/eutrophication/NEMP/report/NEMPyears.aspx. (Accessed: February 28, 2024).

Ekeu-wei, I., Blackburn, G. and Pedruco, P. (2018) 'Infilling missing data in hydrology: Solutions using satellite radar altimetry and multiple imputation for data-sparse regions', *Water*, 10(10), pp. 1–22. doi:10.3390/w10101483.

ESA (2022) *Applications: Sentinel-1, ESA*. Available at: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-1#:~:text=The%20Copernicus%20Sentinel%2D1%20mission,for%20Sentinel%2D1B%20in%202022. (Accessed: 28 February 2024).

Eskelson, B.N. et al. (2009) 'The roles of nearest neighbor methods in imputing missing data in Forest Inventory and monitoring databases', *Scandinavian Journal of Forest Research*, 24(3), pp. 235–246. doi:10.1080/02827580902870490.

Faisal, S. and Tutz, G. (2021) 'Multiple imputation using nearest neighbor methods', *Information Sciences*, 570, pp. 500–516. doi:10.1016/j.ins.2021.04.009.

Fernando, P. (2003) *Modelling of growth cycle of water hyacinth : an application to Bolgoda Lake*. dissertation. Available at: http://dl.lib.uom.lk/handle/123/1317 (Accessed: 28 February 2024).

Ferrari, G.T. and Ozaki, V. (2014) 'Missing data imputation of climate datasets: Implications to modeling extreme drought events', *Revista Brasileira de Meteorologia*, 29(1), pp. 21–28. doi:10.1590/s0102-77862014000100003.

Freedman, J. and Harms, N. (2017) 'Rearing and release of Megamelus scutellaris Berg (Hemiptera: Delphacidae) for biological control of Waterhyacinth in 2015', *Engineer Research and Development Center - U.S. Army* [Preprint], (APCRP-BC-40). doi:10.21079/11681/22762.

Freedman, P.P., Pisani, R. and Purves, R. (2007) Statistics. 4th edn. New York: W.W. Norton and Company Inc.

Gao, G. et al. (2011) 'Practical consideration of genotype imputation: Sample size, window size, reference choice, and untyped rate', *Statistics and Its Interface*, 4(3), pp. 339–351. doi:10.4310/sii.2011.v4.n3.a8.

Gentle, J.E. (2009) 'Monte Carlo Methods for statistical inference', *Statistics and Computing*, pp. 417–433. doi:10.1007/978-0-387-98144-4_11.

Ghosh, D. (2010) 'Water Hyacinth Befriending the Noxious Weed', *Science Reporter*, 47(12), pp. 46–48. ISSN 0036-8512.

Grinstead, C.M. and Snell, J.L. (2006) Grinstead and Snell's Introduction to Probability. 2nd edn. American Mathematical Society.

Grus, J. (2017) Data Science from scratch: First principles with python. 1st edn. Sebastopol, California: O'Reilly.

Guildford, S.J. and Hecky, R.E. (2000) 'Total nitrogen, total phosphorus, and nutrient limitation in lakes and oceans: Is there a common relationship?', *Limnology and Oceanography*, 45(6), pp. 1213–1223. doi:10.4319/lo.2000.45.6.1213.

Gutiérrez, E.L., Ruiz, E.F., Uribe, E.G. and Martínez, J.M. (2001) 'Biomass and Productivity of Water Hyacinth and Their Application in Control Programs', in *ACIAR Proceedings 102*, pp. 109–119. Available at: https://www.researchgate.net/publication/251805567_Biomass_and_Productivity_of_Water_Hyacinth_and_Their_Application_in_Control_Programs. (Accessed: 28 February 2024).

Heiberger, R.M. and Holland, B. (2015) Statistical Analysis and Data Display: An intermediate course with examples in R. 2nd edn. New York: Springer New York.

Heumann, C., Schomaker, M. and Shalabh (2016) Introduction to Statistics and Data Analysis - With Exercises, Solutions and Applications in R . Cham, Switzerland: Springer International Publishing.

Holmes, S. (1993) South African Water Quality Guidelines: Domestic Use. 2 edn. Pretoria: Department of Water Affairs and Forestry.

Hoogland, J. et al. (2020) ‘Handling missing predictor values when validating and applying a prediction model to new patients’, *Statistics in Medicine*, 39(25), pp. 3591–3607. doi:10.1002/sim.8682.

Hu, L., Huang, M., Ke, S. and Tsai, C. (2016) ‘The distance function effect on K-nearest neighbor classification for medical datasets’, *SpringerPlus*, 5(1). doi:10.1186/s40064-016-2941-7.

Huizenga, J., Silberbauer, M., Dennis, R. and Dennis, I. (2013) ‘Technical note: An inorganic water chemistry dataset (1972–2011) of rivers, dams and lakes in South Africa’, *Water SA*, 39(2). doi:10.4314/wsa.v39i2.18.

Huizenga, J.M. (2015) 'An inorganic water chemistry dataset of rivers, dams and lakes in South Africa', *Centre for Water Sciences and Management*. Available at: https://www.waterscience.co.za/waterchemistry/data.html. (Accessed 24 May 2024).

Igual, L. and Seguí, S. (2017) Introduction to data science: A python approach to concepts, techniques and applications. Cham,Switzerland: Springer International Publishing.

Imaoka, T. and Teranishi, S. (1988) ‘Rates of nutrient uptake and growth of the water hyacinth [eichhornia crassipes (Mart.) solms]’, *Water Research*, 22(8), pp. 943–951. doi:10.1016/0043-1354(88)90140-6.

Jakobsen, J.C., Gluud, C., Wetterslev, J. and Winkel, P. (2017) ‘When and how should multiple imputation be used for handling missing data in randomised clinical trials – A practical guide with flowcharts’, *BMC Medical Research Methodology*, 17(1). doi:10.1186/s12874-017-0442-1.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2017) An introduction to statistical learning with applications in R. New York: Springer.

Jeffords, M.R. and Hodgins, A.S. (1995) Pests have enemies too: Teaching young scientists about biological control. Champaign, Illinois: Illinois Natural History Survey.

Jung, D.G., Jung, D. and Kong, S.H. (2019) ‘Characterization of total-phosphorus (TP) pretreatment microfluidic chip based on a thermally enhanced photocatalyst for portable analysis of eutrophication’, *Sensors*, 19(16), pp. 3452. doi:10.3390/s19163452.

Kang, H. (2013) ‘The prevention and handling of the missing data’, *Korean Journal of Anesthesiology*, 64(5), pp. 402–406. doi:10.4097/kjae.2013.64.5.402.

Keebine, G. (2019) *Mapping and monitoring the spatial distribution of Eichhornia crassipes (water hyacinth) in the Hartbeespoort Dam, South Africa, using remote sensing data*. dissertation. Available: https://wiredspace.wits.ac.za/server/api/core/bitstreams/34e318e2-438c-4145-a385-008da30cd99f/content (Accessed: 28 February 2024).

Kelleher, J.D., Namee, B.M. and D’Arcy, A. (2020) Fundamentals of machine learning for Predictive Data Analytics: Algorithms, worked examples, and case studies. 2nd edn. Cambridge, MA: The MIT Press.

Kempster, P.L., van Vliet, H.R. and Kühn, A. (1997) ‘The need for guidelines to bridge the gap between ideal drinking-water quality and that quality which is practically achievable and acceptable’, *Water SA*, 23(2), pp. 163–167. ISSN 0378-4738.

Kenton, W. (2023) *Goodness-of-Fit, Investopedia*. Available at: https://www.investopedia.com/terms/g/goodness-of-fit.asp#:~:text=The%20Anderson%2DDarling%20(A%2DD)%20test%20is%20a%20variation%20on,variations%20observed%20in%20the%20tails. (Accessed: 29 February 2024).

Khandelwal, P., Rammer, D., Pallickara, S. and Pallickara, S.L. (2021) 'Mind the gap: Generating imputations for satellite data collections at myriad spatiotemporal scopes', *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)* [Preprint]. doi:10.1109/ccgrid51090.2021.00019.

Kinsler, D. (2021) *Master's Project Query - Satellite Image Recognition*.

Kinsler, D. (2021a) *Water Hyacinth status, Rhodes University – Where leaders learn*. Available at: https://www.ru.ac.za/centreforbiologicalcontrol/resources/waterhyacinthstatus/ (Data Provided: 21 May 2021; Website Accessed: 29 February 2024).

Kissell, R. and Poserina, J. (2017) 'Advanced math and statistics', *Optimal Sports Math, Statistics, and Fantasy*, pp. 103–135. doi:10.1016/b978-0-12-805163-4.00004-9.

Kroese, D.P., Botev, Z.I., Taimre, T. and Vaisman, R. (2020) Data Science and Machine Learning: Mathematical and Statistical Methods. Boca Raton, Florida: Chapman & Hall/CRC.

Kubát, M. (2017) An Introduction to Machine Learning. 2nd edn. Cham, Switzerland: Springer International Publishing.

Kuhn, M. and Johnson, K. (2016) Applied predictive modeling. New York: Springer.

Landsat Missions (2024) *Landsat 8*. Available at: https://www.usgs.gov/landsat-missions/landsat-8 (Accessed: 29 February 2024).

Law, A.M. (2015) Simulation Modeling and Analysis. 5th edn. New York: McGraw-Hill.

Li, P., Stuart, E.A. and Allison, D.B. (2015) 'Multiple Imputation: A Flexible Tool for Handling Missing Data', *JAMA*, 314(18), pp. 1966–1967. doi:10.1001/jama.2015.15281.

Liemohn, M.W. et al. (2021b) 'RMSE is not enough: Guidelines to robust data-model comparisons for Magnetospheric Physics', *Journal of Atmospheric and Solar-Terrestrial Physics*, 218, p. 105624. doi:10.1016/j.jastp.2021.105624.

Little, R.J.A. and Rubin, D.B. (2002) Statistical analysis with missing data. Hoboken, New Jersey: John Wiley & Sons, Inc.

Liu, J., Yue, C., Pei, C., Li, X. and Zhang, Q. (2023) 'Prediction of regional forest biomass using Machine Learning: A case study of beijing, China', *Forests*, 14(5), pp. 1008. doi:10.3390/f14051008

Mack, C., Weistreich, D. and Su, Z. (2018) Managing missing data in patient registries: Addendum to registries for evaluating patient outcomes: A user's guide. Rockville (MD): Agency for Healthcare Research and Quality (US).

Madley-Dowd, P., Hughes, R., Tilling, K. and Heron, J. (2019) 'The proportion of missing data should not be used to guide decisions on multiple imputation', *Journal of Clinical Epidemiology*, 110, pp. 63–73. doi:10.1016/j.jclinepi.2019.02.016.

Makridakis, S.G., Hyndman, R.J. and Wheelwright, S.C. (2011) Forecasting: Methods and applications. 3rd edn. New York: Wiley.

Maluleke, V.E., Bührmann, J.H (2021) 'Corporate Failure Prediction of JSE Listed South African Firms Using Machine', in *SAIIE32 Proceedings*, 4 - 6 October 2021, Glenburn Lodge, Gauteng, South Africa, pp545-555.

Marini, F., Bucci, R., Magrì, A.L. and Magrì, A.D. (2008) 'Artificial Neural Networks in chemometrics: History, examples and perspectives', *Microchemical Journal*, 88(2), pp. 178–185. doi:10.1016/j.microc.2007.11.008.

Marino, M., Lucas, J., Latour, E. and Heintzman, J.D. (2021) 'Missing data in primary care research: Importance, implications and approaches', *Family Practice*, 38(2), pp. 199–202. doi:10.1093/fampra/cmaa134.

MATLAB (2006) *Distribution Fitter, Fit probability distributions to data*. Available at: https://www.mathworks.com/help/stats/distributionfitter-app.html (Accessed: 24 February 2024).

MATLAB (2006a) *knnimpute, Impute missing data using nearest-neighbor method*. Available at: https://www.mathworks.com/help/bioinfo/ref/knnimpute.html (Accessed: 23 March 2024).

MATLAB (2013) *evalclusters, Evaluate clustering solutions*. Available at: https://www.mathworks.com/help/stats/evalclusters.html (Accessed: 24 February 2024).

MATLAB (2017) *Regression learner app, Regression Learner*. Available at: https://www.mathworks.com/help/stats/regressionlearner-app.html (Accessed: 24 February 2024).

MATLAB (2018) *Econometric Modeller, Analyze and model econometric time series*. Available at: https://www.mathworks.com/help/econ/econometricmodeler-app.html (Accessed: 24 February 2024).

MATLAB (2024) *t Location-Scale Distribution, MathWorks*. Available at: https://www.mathworks.com/help/stats/t-location-scale-distribution.html (Accessed: 29 February 2024).

Matthews, S. (2016) 'Satellite technology keeping an eye on South Africa's dams', *The Water Wheel January/February 2016*, January, pp. 24–26. ISSN 0258-2244. Available at: https://www.wrc.org.za/wp-content/uploads/mdocs/The%20Water%20Wheel%20Jan-Feb%202016.pdf (Accessed: 28 February 2024).

Meyer, D.H. and Rossouw, J.N. (1993) 'Development of the reservoir eutrophication model (REM) for South African reservoirs', *Water SA*, 18(3), pp. 155–164. ISSN 0378-4738.

Milicevic, D. (2023) 'Modeling Water Hyacinth Growth Dynamics', *Archives of Biological Sciences*, 75(2), pp. 165–185. doi:10.2298/abs230222014m.

Miller, B.E., Coetzee, J.A. and Hill, M.P. (2020) 'Mind the gap: The delayed recovery of a population of the biological control agent megamelus scutellaris Berg. (Hemiptera: Delphacidae) on Water Hyacinth after winter', *Bulletin of Entomological Research*, 111(1), pp. 1–9. doi:10.1017/s0007485320000516.

Miller, B.E. (2019) *Post-release evaluation of Megamelus scutellaris Berg. (hemiptera: delphacidae): a biological control agent of water hyacinth Eichhornia crassipes (Mart.) Solms-Laub (Pontederiaceae) in South Africa*. dissertation. South East Academic Libraries System (SEALS). Available: https://vital.seals.ac.za/vital/access/manager/Repository?view=list&f0=sm_citation_publication_date%3A%222019%22&f1=sm_type%3A%22MSc%22&sort=sort_ss_sm_creator+asc&f2=sm_type%3A%22Masters%22&f3=sm_language%3A%22English%22&f4=sm_subject%3A%22Biological+pest+control+agents%22. (Accessed: 5 June 2024).

Mitchell, S.A., Crafford, J.G., van Veelen, M., Maree, J. and Reyneke, E. (2016) 'Review of the Hartbeespoort Dam integrated biological remediation programme (Harties Metsi a Me)', *Water Research Committee report*. KV 357/16. Available: https://www.wrc.org.za/wp-content/uploads/mdocs/KV%20357.pdf. (Accessed: 5 June 2024).

Moffat, R., Weaver, K., Ngxande-Koza, S., Sebola, K., English, K., Kinsler, D. and Coetzee, J. (2024) 'Bridging boundaries: Six years of community engagement with biological control implementation and monitoring of Water Hyacinth on Hartbeespoort Dam, South Africa', *Biological Control*, 194, p. 105544. doi:10.1016/j.biocontrol.2024.105544.

Mohammad, A.B., Abbadi, M.A. and Altarawneh, G.A. (2014) 'Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach', *International Journal of Computer Science and Information Security (IJCSIS)*, 12(8), pp. 33–39. ISSN 1947-5500.

Mohammed, M.B. et al. (2021) 'Comparison of five imputation methods in handling missing data in a continuous frequency table', *PROCEEDINGS OF SCIEMATHIC 2020* [Preprint]. doi:10.1063/5.0053286.

Mohri, M., Rostamizadeh, A. and Talwalkar, A. (2018) Foundations of Machine Learning. 2nd edn. Cambridge, MA: The MIT Press.

Montgomery, D.C., Jennings, C.L. and Kulahci, M. (2016) Introduction to time series analysis and forecasting. 2nd edn. Hoboken, New Jersey: John Wiley & Sons.

Mouta, N. et al. (2023) 'Sentinel-2 time series and classifier fusion to map an aquatic invasive plant species along a river—the case of water-hyacinth', *Remote Sensing*, 15(13), p. 3248. doi:10.3390/rs15133248.

Mukarugwiro, J.A. et al. (2020) 'Mapping spatio-temporal variations in water hyacinth (eichhornia crassipes) coverage on rwandan water bodies using multispectral imageries', *International Journal of Environmental Science and Technology*, 18(2), pp. 275–286. doi:10.1007/s13762-020-02824-8.

Müller, A.C. and Guido, S. (2016) Introduction to machine learning with python: A guide for data scientists. 1st edn. Sebastopol, California: O'Reilly Media.

Mzuza, M.K., Chapola, L., Kapute, F., Chikopa, I. and Gondwe, J. (2015) 'Analysis of the impact of aquatic weeds in the Shire River on generation of electricity in Malawi: A case of Nkula Falls Hydro-electric power station in Mwanza District, Southern malawi', *International Journal of Geosciences*, 06(06), pp. 636–643. doi:10.4236/ijg.2015.66051.

NCEI (2024) *Hartbeespoort, South Africa Weather averages, National Centers for Environmental Information. NOAA*. Available at: https://www.ncei.noaa.gov/ (Accessed: 28 February 2024).

Newman, D.A. (2014) 'Missing Data: Five Practical Guidelines', *Organizational Research Methods*, 17(4), pp. 372–411. doi:10.1177/1094428114548590.

Nguyen, C.D., Carlin, J.B. and Lee, K.J. (2017) 'Model checking in multiple imputation: An overview and case study', *Emerging Themes in Epidemiology*, 14(1), 8. doi:10.1186/s12982-017-0062-6.

Oancea, B. and Ciucu, Ş.C. (2014) 'Time series forecasting using neural networks', *Challenges of the Knowledge Society. IT in Social Sciences*, pp. 1402–1408. Available at: https://arxiv.org/pdf/1401.1333 (Accessed: 28 February 2024).

Omer, N.H. (2020) 'Water quality parameters', *Water Quality - Science, Assessments and Policy* [Preprint]. doi:10.5772/intechopen.89657.

Owens, C.S. and Madsen, J.D. (1995) 'Low Temperature Limits of Waterhyacinth', *J. Aquat. Plant Manage.*, 33, pp. 63–68.

Pádua, L. et al. (2022) 'Water Hyacinth (eichhornia crassipes) detection using coarse and high resolution multispectral data', *Drones*, 6(2), pp. 1–14. doi:10.3390/drones6020047.

Petrucio, M.M. and Esteves, F.A. (2000) 'Uptake rates of nitrogen and phosphorus in the water by Eichhornia crassipes and Salvinia Auriculata', *Revista Brasileira de Biologia*, 60(2), pp. 229–236. doi:10.1590/s0034-71082000000200006.

Pourahmadi, M. (2001) Foundations of time series analysis and prediction theory. New York: John Wiley & Sons (Probability and Statistics).

Ptacnik, R., Andersen, T. and Tamminen, T. (2010) 'Performance of the redfield ratio and a family of nutrient limitation indicators as thresholds for phytoplankton N vs. P limitation', *Ecosystems*, 13(8), pp. 1201–1214. doi:10.1007/s10021-010-9380-z.

Rachmatullah, M.I., Santoso, J. and Surendro, K. (2021) 'Determining the number of hidden layer and hidden neuron of neural network for wind speed prediction', *PeerJ Computer Science*, 7. doi:10.7717/peerj-cs.724.

Rahman, A.R.A. (2012) *Support Vector Machine for solving small dataset problem*. dissertation. Semantic Scholar. Available at: https://www.semanticscholar.org/paper/Support-vector-machine-for-solving-small-dataset-Rahman-Rijal/f1b62425889b57744c33a3f16d86d9853a2fea3a (Accessed: 29 February 2024).

Raschka, S. (2015) Python machine learning: Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics, Birmingham, United Kingdom: Packt Publishing Ltd.

Raschka, S. (2018) *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. thesis. Available at: https://doi.org/10.48550/arXiv.1811.12808. (Accessed: 28 February 2024)

Rayner, B. and Bolhuis, M.A. (2020) Deus Ex Machina? A framework for macro forecasting with machine learning. IMF Working Papers (2020/045).

Reddy, K.R. and Debusk, W.F. (1984) 'Growth characteristics of aquatic macrophytes cultured in nutrient-enriched water: I. Water Hyacinth, water lettuce, and Pennywort', *Economic Botany*, 38(2), pp. 229–239. doi:10.1007/bf02858838.

Reddy, K.R. and Tucker, J.C. (1983) 'Productivity and nutrient uptake of water hyacinth,Eichhornia crassipes I. Effect of nitrogen source', *Economic Botany*, 37(2), pp. 237–247. doi:10.1007/bf02858790.

Ricci, V. (2005) Fitting Distributions with R. 0.4-21. Free Software Foundation. Available at: https://www.researchgate.net/publication/228791072_Fitting_Distributions_with_R. (Accessed: 19 June 2024).

Richards, N. (2020) *Ridding hartbeespoort dam of hyacinths 'highly unlikely', New Study reveals, The Citizen*. Available at: https://www.citizen.co.za/news/ridding-hartbeespoort-dam-of-hyacinths-highly-unlikely-new-study-reveals/ (Accessed: 29 February 2024).

Rojas-Sandoval, J. and Acevedo-Rodríguez, P. (2022) 'Eichhornia crassipes (water hyacinth)', *CABI Compendium*. Available: https://www.cabidigitallibrary.org/doi/10.1079/cabicompendium.20544 (Accessed: 5 June 2024).

Rossouw, J.N., Harding, W.R. and Fatoki, O.S. (2008) 'A guide to catchment-scale eutrophication assessments for rivers, reservoirs and lacustrine wetlands', in *Water Research Commission report, TT352/08*, Pretoria.

Roy, B. (2021) 'Optimum machine learning algorithm selection for forecasting vegetation indices: Modis Ndvi & Evi', *Remote Sensing Applications: Society and Environment*, 23(1), pp. 100582. doi:10.1016/j.rsase.2021.100582.

Russell, R. (2018) Machine learning step-by-step guide to implement machine learning algorithms with python. Columbia, SC: CreateSpace Independent Publishing Platform.

SANCOLD (2018) *Register of Large Dams, SANCOLD*. Available at: https://sancold.org.za/register-large-dams/ (Accessed: 28 February 2024).

Savic, R. et al. (2022) 'Nitrogen and phosphorus concentrations and their ratios as indicators of water quality and eutrophication of the hydro-system Danube–Tisza–Danube', *Agriculture*, 12(7), p. 935. doi:10.3390/agriculture12070935.

SAWS (2022) *Hartebeespoort, SAWS Home - WeatherSA* Portal. Available at: https://www.weathersa.co.za/ (Data Extracted: 31 October 2022; Accessed: 29 February 2024).

Scott, W.E., Ashton, P.J., Walmsley, R.D. and M. T. Seaman, M.T. (1980) 'Hartbeespoort Dam: A case study of a hypertrophic, warm, monomictic impoundment', *Hypertrophic Ecosystems*, 2, pp. 317–322. doi:10.1007/978-94-009-9203-0_35.

Singh, G., Reynolds, C., Byrne, M. and Rosman, B. (2020) 'A remote sensing method to monitor water, aquatic vegetation, and invasive water hyacinth at national extents', *Remote Sensing*, 12(24), p. 4021. doi:10.3390/rs12244021.

Singh, S.K., Jha, S.K. and Gupta, R. (2024) 'Enhancing the accuracy of wind speed estimation model using an efficient hybrid deep learning algorithm', *Sustainable Energy Technologies and Assessments*, 61, p. 103603. doi:10.1016/j.seta.2023.103603.

Sivappa, R.V. (2017) 'Calculating growth rate of water hyacinth pollution wise (in relation to trophic state)', *International Journal of Engineering and Technical Research (IJETR)*, 7(7), pp. 85–90. ISSN 2321-0869.

Smith, V.H. (2009) 'Eutrophication', *Encyclopedia of Inland Waters*, pp. 61–73. doi:10.1016/b978-012370626-3.00234-9.

Steinwart, I. and Scovel, C. (2005) 'Fast rates for support Vector Machines', *Learning Theory*, 3559, pp. 279–294. doi:10.1007/11503415_19.

Sullivan, P.R. and Wood, R. (2012) 'Water hyacinth (Eichhornia crassipes (Mart.) Solms) seed longevity and the implications for management', *Eighteenth Australasian Weeds Conference*, pp. 37–40.

Taha, H.A. (2017) Operations research: An introduction. 10th edn. Upper Saddle River: Pearson.

Taherdoost, H. (2020) 'Different Types of Data Analysis; Data Analysis Methods and Techniques in Research Projects', *International Journal of Academic Research in Management (IJARM)*, 9(1), pp. 1–9. ISSN: 2296-1747.

Taylor, B.W. (2013) Introduction to Management Science. 11th edn. Boston: Pearson.

Thacker, B.H. et al. (2004) *Concepts of model verification and validation* [Technical Report]. doi:10.2172/835920.

Theobald, O. (2017) Machine Learning For Absolute Beginners: A Plain English Introduction Vol 3. 2nd edn. Scatterplot Press.

Tibrewal, T.P. (2023) *Support Vector Machines (SVM): An Intuitive Explanation, Medium*. Available at: https://medium.com/low-code-for-advanced-data-science/support-vector-machines-svm-an-intuitive-explanation-b084d6238106 (Accessed: 29 February 2024).

Tran, E. (1999) *Verification/Validation/Certification*. dissertation. Dependable Embedded Systems. Available at: https://users.ece.cmu.edu/~koopman/des_s99/verification/. (Accessed: 19 June 2024).

Tüzüntürk, S., Şenaras, A.E. and Sezen, H.K. (2015) 'Forecasting Water Demand By Using Monte Carlo Simulation', *Akademik Görünüm Dergisi*, (49), pp. 25–44. ISSN:1694-528X.

Uchida, R.S. and Silva, J.A. (2000) ‘Essential Nutrients for Plant Growth: Nutrient Functions and Deficiency Symptoms’, in *Plant Nutrient Management in Hawaii’s Soils, Approaches for Tropical and Subtropical Agriculture*. 4th edn. Honolulu, Hawaii: College of Tropical Agriculture and Human Resources (CTAHR) , pp. 31–55.

USGS (2024) *What is remote sensing and what is it used for?, U.S. Geological Survey*. Available at: https://www.usgs.gov/faqs/what-remote-sensing-and-what-it-used (Accessed: 29 February 2024).

van Buuren, S. (2018) Flexible imputation of missing data. 2nd edn. Vancouver, Canada: Chapman & Hall/CRC (Interdisciplinary Statistics Series).

van Klompenburg, T., Kassahun, A. and Catal, C. (2020) ‘Crop yield prediction using Machine Learning: A Systematic Literature Review’, *Computers and Electronics in Agriculture*, 177, p. 105709. doi:10.1016/j.compag.2020.105709.

Water and Sanitation South Africa (WASA) (2020), 'Bugs beat back water hyacinth', *Water & Sanitation Africa March/April 2020*, 15, pp. 32–33. ISSN 1990-8857.

Wasserman, L. (2010) All of statistics: A concise course in statistical inference. 1st edn. New York, New York: Springer. ISBN 978-0-387-21736-9

Water Resources Mission Area (2019) *Nutrients and eutrophication active, Nutrients and Eutrophication*. Available at: https://www.usgs.gov/mission-areas/water-resources/science/nutrients-and-eutrophication (Accessed: 28 February 2024).

Wilson, J.R., Holst, N. and Rees, M. (2005) ‘Determinants and patterns of population growth in water hyacinth’, *Aquatic Botany*, 81(1), pp. 51–67. doi:10.1016/j.aquabot.2004.11.002.

Yang, X., Wu, X., Hao, H. and He, Z. (2008) ‘Mechanisms and assessment of water eutrophication’, *Journal of Zhejiang University SCIENCE B*, 9(3), pp. 197–209. doi:10.1631/jzus.b0710626.

Yang, R. (2022) 'Analyses of approaches to deal with missing data in water quality data set' in *Proceedings of the 2022 7th International Conference on Social Sciences and Economic Development (ICSSED 2022)*. [Preprint]. doi:10.2991/aebmr.k.220405.184

Yesuf, B.M. et al. (2023) ‘Time-series Arima modelling of the (Cyprinidae) fishery in water hyacinth-infested and non-infested sites in Lake Tana, Ethiopia’, *Journal of Freshwater Ecology*, 38(1). doi:10.1080/02705060.2023.2216218.

Yıldırım, S. (2021) ‘Monte Carlo: Simulation Methods for Statistical Inference’, *IE 58001: Simulation methods for Statistical Inference*. Available at: https://people.sabanciuniv.edu/sinanyildirim/teaching.html (Accessed: 5 June 2024).

APPENDIX A Details of Extrapolating TP Dataset

This appendix discusses the steps involved in extrapolating the TP dataset, that were not shown in detail in Chapter 6. Figure 38 in Chapter 6 outlines these steps. Time series analysis was done to identify any patterns that should be present in the generated TP dataset. Thereafter, the five scenarios were developed. Scenarios were developed and tested similarly to the way in which Tüzüntürk et al. (2015) did. The scenarios were then used to generate five different TP datasets which were compared, by means of a statistical test, and the best dataset was used in model building.

A.1 Time Series Analysis of TP Dataset

Figure 56 shows the time series plot of the observed TP dataset which contained data from 11 March 1980 to 18 July 2018. The dataset was that after missing data was imputed using multiple KNN imputation in Chapter 6.

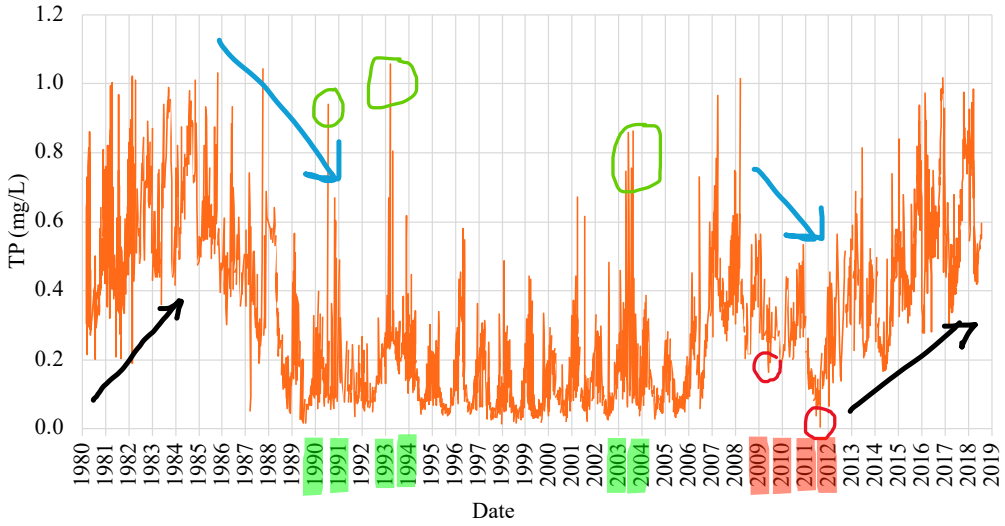


Figure 56: Time series plot of the observed TP dataset with trends

It is evident TP had a seasonal pattern shown by the hills that repeated every year, with peaks and troughs occurring at the beginning and middle of each year. The areas circled in green indicate years that had extremely higher peaks than its surrounding years but were not classified as outliers when using Equations (3) and (4).

There were a few instances of increasing and decreasing trends. The first increasing trend lasted 6 years, from 1980 to 1986, after which a decreasing trend was seen for another 6 consecutive years. This pattern was evident again in 2011 where the observed TP increased for another 6 years. However, there was a period where the observed TP only had yearly seasonal patterns and a straight-line trend from 1990 to 2006, and then another increasing and decreasing pattern for 3 consecutive years each. This was an apparent cycle assumed to have continued after 2018. An ACF was plotted to confirm this identified cycle and is shown in Figure 57.

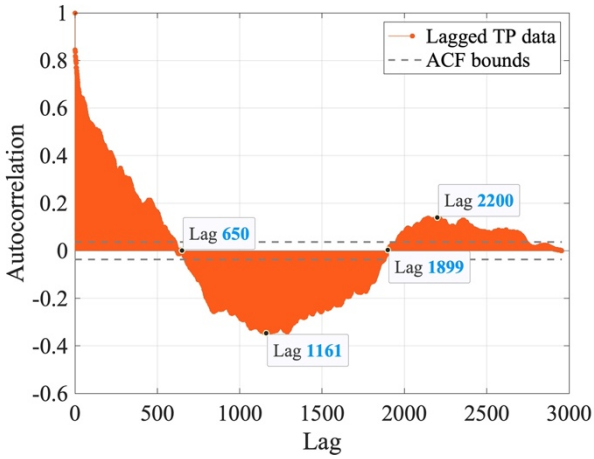


Figure 57: ACF plot of the observed TP dataset

The TP dataset had 2956 data points and thus the ACF had 2955 lags. The ACF shows that there were both seasonal patterns, shown as little hills, and a cycle whereby the peaks of the hills increased and decreased for consecutive years. The turning points of the cyclic trends are highlighted in the figure. The TP dataset did not have daily recordings and thus the number of data points between each trough and peak did not relate to the number of days. For instance, point 649, which relates to lag 650, was 5 July 1983 and point 1160, which relates to lag 1161, was 9 February 1993. Thus, the difference between the first and second turning points of the cycle was about 9 years. Similarly, the difference between lag 1161 and 1899 was also about 9 years. This confirms that there was a cycle in the TP dataset.

Carroll and Curtis (2021) compared nutrient recordings from 2010 to 2016 and noted an increasing trend. However, it is clear from the graph that, although there was an increase from 2010, a was cycle present that meant the TP is expected to decrease again after 2018. Therefore, this cycle was further explored.

It was found that parts of the cycle match; the ACF showed the section with the dates required, being 2015 onwards, had a peak at point 2200, thus the data was lagged by 2200 points and then other lags after 2200 was tested until the data lined-up the best. Figure 58 shows the sections of the observed TP that line up.

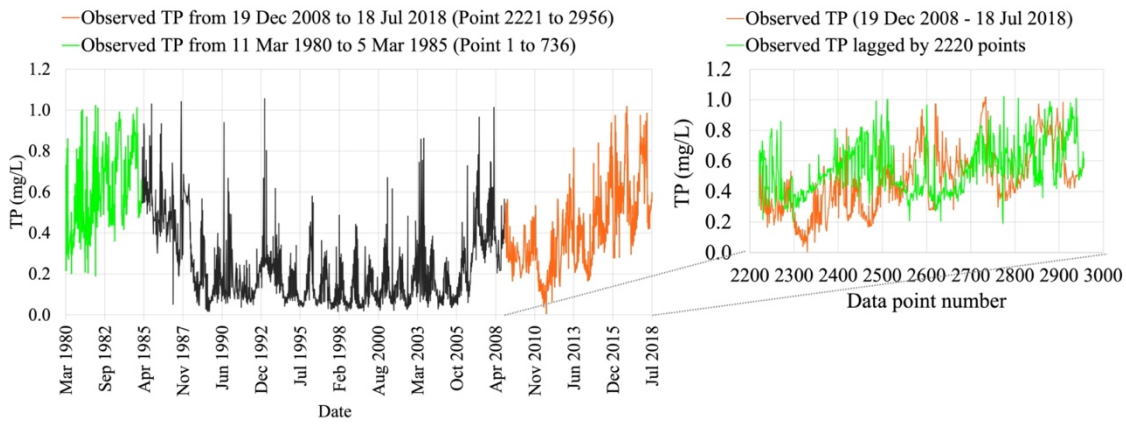
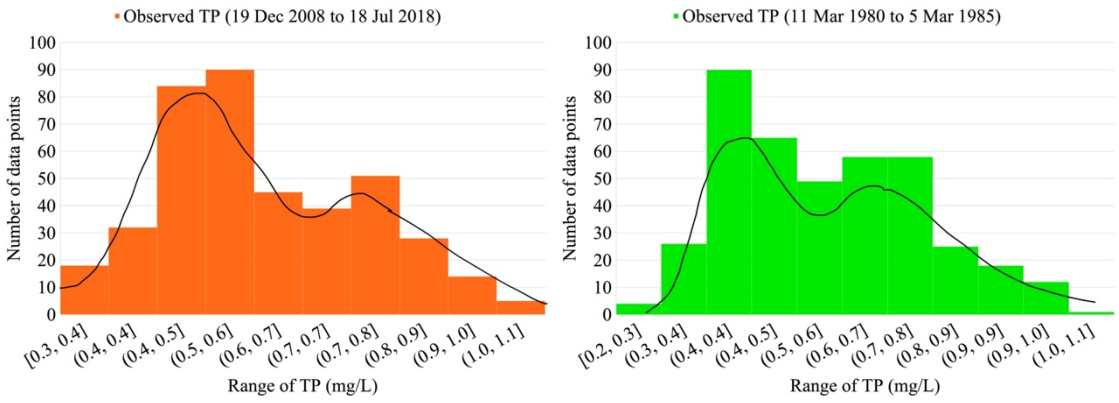


Figure 58: Observed TP dataset and the identified cycles

Therefore, it was assumed that the TP data would remain the same and not worse than the TP data from the past. This meant that the light orange section, point 1 to 736, could be used to generate the period required which fell in the orange section from point 2221 to 2956. To further prove that the two portions of TP data in the figure were very similar, histograms were plotted and compared in Figure 59.



(a) (b)

Figure 59: Histogram of observed TP dataset (a) and itself lagged by 2220 points (b)

The histograms were very similar thus implying the two portions of the TP dataset had very similar PDs. This meant both portions were statistically similar and thus the TP data from point 1 to 736 (11 March 1980 to 5 March 1985) was used to extrapolate the TP dataset for 1 August 2015 to 28 December 2019.

The PACF was also plotted for TP to identify significant lags that could be used in simulating the dataset instead of the cycle above. Figure 60 shows the PACF and identifies lag 1506 to be the most significant as it had the highest partial autocorrelation value above the PACF bounds.

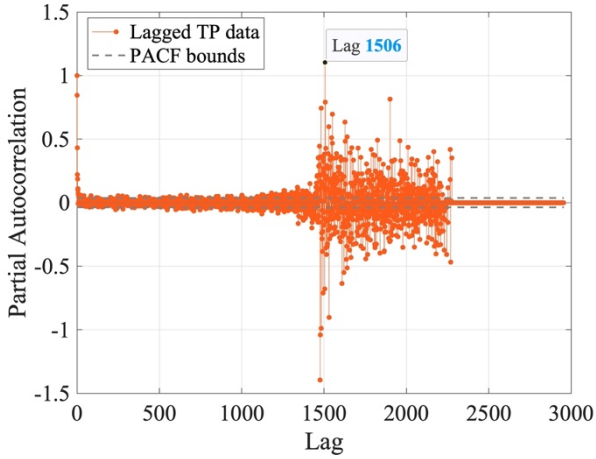


Figure 60: PACF plot of the observed TP dataset

A.1.1 Summary of the time series analysis on TP dataset

In summary, the analysis confirmed that TP had seasonal patterns and cycles of increasing and decreasing trends. Although the trends seemed to continue into future years, the assumption that the past data will remain similar in future was made. This is because the ACF showed cycles whereby the increasing trend started to decrease again in the period being generated being from 1 August 2015. The yearly seasonal patterns were more crucial than the increasing and decreasing

trends seen as WHC is greatly dependent on seasonal factors (Ghosh, 2010). Therefore, based on the assumption that the seasonal curve remained the same for every year, simulations incorporated the seasonal curve represented by a polynomial curve fitted to the daily average TP calculated.

A.2 Developing Five Scenarios for Extrapolating TP Dataset

Three methods were explored: Monte Carlo Simulation, Interpolation and Time Series Forecasting. However, it was found that these methods could not be used on their own due to reasons stipulated later in this section. Therefore, five scenarios were developed in which a combination of those methods were used. Table 36 describes the methods used in each scenario.

Table 36: Summary of scenarios used to extrapolate the TP dataset

Scenario	Summary of method
1	Interpolated TP data were used to calculate the ranges each random value should fall into in the Monte Carlo simulation.
2	Monte Carlo simulation of the difference between the TP dataset and itself lagged by 1506 data points (called TP-1506) was added to the TP dataset generated in Scenario 1.
3	Monte Carlo simulation of the difference between the TP dataset and itself lagged by 2220 data points (called TP-2220) was added to the TP dataset generated in Scenario 1.
4	The TP-1506 dataset, simulated in Scenario 2, was added to the TP data interpolated from the polynomial curve representing the annual seasonal pattern.
5	The TP-2220 dataset, simulated in Scenario 2, was added to the TP data interpolated from the polynomial curve representing the annual seasonal pattern.

The number of times the datasets were lagged were determined by the PACF in Section A.1. For each scenario, a number of calculations or simulations were completed before the scenario could be done. These are outlined in Figure 61 which lists the work done that helped develop the scenarios.

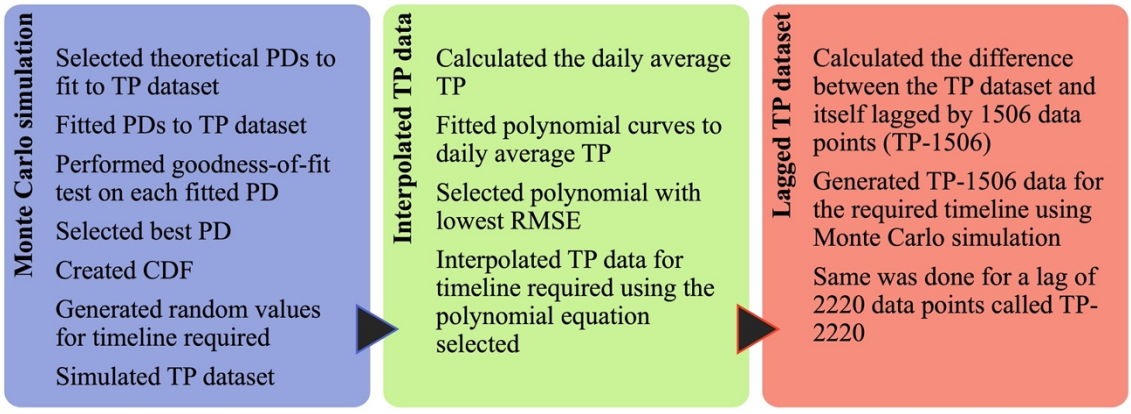


Figure 61: Development of scenarios used to extrapolate the TP dataset

These are discussed in the subsequent sections.

A.2.1 Monte Carlo simulation

This step involved fitting three different PDs to the TP dataset chosen based on calculated descriptive statistics. The theoretical PDs were fitted to a section of the observed TP dataset from 5 March 1980 to 5 March 1985, shown in green in Figure 58, based on the identified cycle. Next, a PD was fitted to the dataset after being selected based on its goodness-of-fit test result.

Fitting PD to observed TP dataset

The descriptive statistics of the observed TP dataset are shown in Table 37.

Table 37: Descriptive statistics of the observed TP dataset from 1980 to 1985

Standard Deviation	Mean	Median	Kurtosis	Skewness	Coefficient of Variance
0.18	0.57	0.56	2.34	0.39	0.31

Note that the descriptive statistics was calculated using only the section of the observed TP dataset that was used for simulations. The statistics suggested that Log-Logistic, Lognormal and Weibull PDs could fit the TP dataset for the reasons listed in Table 38.

Table 38: Theoretical PDs fitted to the observed TP dataset

Theoretical PD	Reason for Selection
Log-Logistic	is similar to lognormal
Lognormal	skewness > 0 coefficient of variance > 0.3
Weibull	skewness > 0 kurtosis not = 2 but close to 2 so could be similar family to exponential

After fitting the theoretical distributions to the datasets using the distribution fitter app in MATLAB, the goodness-of-fit test was done to determine which of the PDs fitted the observed dataset the best.

Goodness-of-fit test

For each distribution fitted, the loglikelihood, the AD test p-value, and the AIC were calculated in MATLAB. The distribution with the highest p-value above 0.05, and the lowest AIC was the best fitting PD. The results are shown in Table 39.

Table 39: Goodness-of-fit results for PDs fitted to observed TP dataset

Dataset	Theoretical PD	AD p-value	AIC
Observed TP	Lognormal	0.070	**-270**
	Log-Logistic	0.023	-248
	Weibull	0.015	-256

From the table, the best fitting distribution was the Lognormal with location and scale parameters of -0.60384 and 0.31619, respectively. The parameters were calculated in MATLAB. Using these parameters and distribution, the CDF was calculated and used to simulate the TP dataset using Monte Carlo simulation. However, the simulated TP dataset was not a good representation of the observed TP dataset as it was very random and lacked the trends evident in the observed dataset.

Problems with Monte Carlo simulation

Monte Carlo simulation depends on randomness; thus, the results are random. This means different random numbers are generated when the simulation is run again, resulting in a completely different dataset. Figure 62 displays the observed and simulated TP dataset.

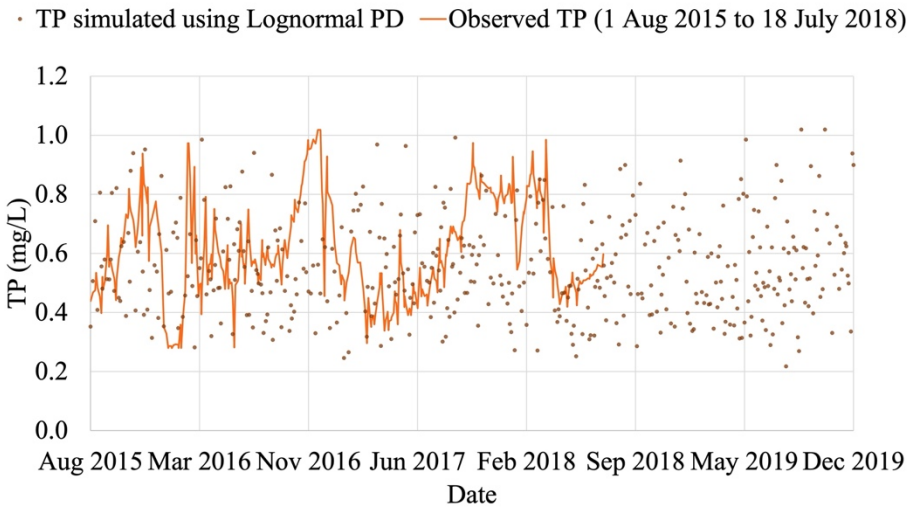
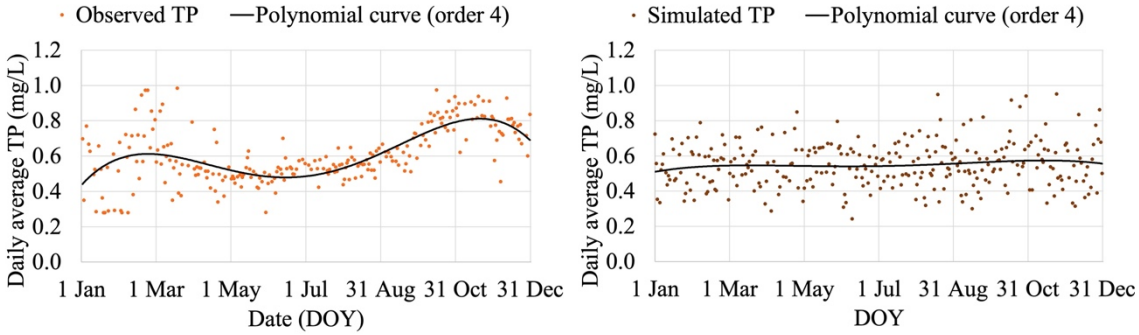


Figure 62: Plot of observed TP data and Monte Carlo simulation of TP

It is clear that the simulated TP dataset did not capture the seasonal patterns present in the observed TP dataset. Although the simulated and observed TP datasets were similar in range, patterns and seasonality in the data were lost. Figure 63 compares the daily average observed and simulated TP to highlight the lack of seasonal pattern in the simulation.



(a) (b)

Figure 63: Daily average of the observed TP dataset (a) and simulated TP dataset (b)

Although the simulated values in Figure 63(b) do fall within the same range as the observed in Figure 63(a), from 0.2 to 1, it is more random which is expected due to the nature of Monte Carlo simulations. The trend lines show that the simulated TP dataset has almost no curve whereas the observed daily average TP has a defined one. This means that the average seasonal pattern on the observed TP could be defined by a polynomial curve which the simulated values did not capture. Therefore, Monte Carlo simulation could not be used as is to simulate the nutrient datasets due to this problem.

A.2.2 Interpolated TP data

A curve was fitted to the daily average TP using the curve fitter app in MATLAB. To do so, the daily average TP was calculated so there was a generalised year of data. An extract of the TP data is shown in Table 40.

Table 40: Extract of the daily average TP data

Date	Daily Average TP	DOY
02 Jan	0.6979	2
03 Jan	0.3494	3
05 Jan	0.7702	5
07 Jan	0.6666	7

DOY refers to the Day of Year which is for example, 365 for 31 December 2022 because there are 365 days in a non-leap year, and 32 for 1 February 2023 because it is the 32nd day in the year. A scatter plot of the daily average TP is shown in Figure 63(a). The daily average TP has a polynomial trend, represented by the black solid line, and thus polynomial curves were fitted to the TP dataset.

Fitting polynomial curves to the observed TP dataset

Three polynomial curves were fitted. For the app to work, the dates were converted to DOY, so that the app could plot numbers instead of dates. The RMSE, shown in Figure 64, was calculated so that the curve with the lowest RMSE could be chosen.

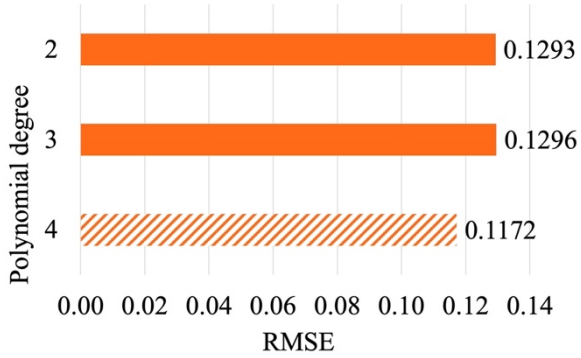


Figure 64: RMSE of the polynomials curves fitted to the daily average TP data

A polynomial with 4 degrees had the lowest RMSE and the equation that describes this polynomial is shown in Table 41.

Table 41: Details of the polynomial curves fitted to the daily average TP data

Polynomial Equation	Details
$y = p_1 * x^4 + p_2 * x^3 + p_3 * x^2 + p_4 * x + p_5$ where: x is DOY, normalized by mean 183.9 and standard deviation 104.9 y is daily average TP, in mg/L	Polynomial with 4 degrees. Coefficients (with 95% confidence intervals): $p1 = -0.0801$ $(-0.1013, -0.059)$ $p2 = -0.0038$ $(-0.0223, 0.0146)$ $p3 = 0.2681$ $(0.2111, 0.3251)$ $p4 = 0.0836$ $(0.0475, 0.1197)$ $p5 = 0.4857$ $(0.4582, 0.5133)$

The equation was used to interpolate TP data which was used to influence the random numbers generated during Monte Carlo simulation.

Interpolating TP data from a curve

The fitted curves represented the daily average TP for any given year on any given day. This meant that, assuming the years 2015 to 2019 followed the same curve as the daily average TP, any dates in those years could be interpolated from this curve. For example, if the average TP for the 1 June is 1.760 mg/L, the TP for 1 June 2015 and 2016, and every year thereafter, will be the

same. The dates needed in the years 2015 to 2019 were interpolated from the fitted curves using the following method:

- A list of the dates needed for the years 2015 to 2019, that match the other datasets, was made.
- These dates were then converted to DOY.
- The dates needed where interpolated using the equation of the chosen fitted curve in Table 41. The DOY were normalised using the mean (μ) and standard deviation (σ) and the formula: $x = \frac{(DOY-\mu)}{\sigma}$.
- To get the TP for the corresponding dates, x was substituted with the DOY needed by means of a loop in MATLAB.

The curve shows the trend that the data had every year on average.

Problems with interpolating TP data from a curve

A scatter plot was made of the interpolated TP data and is shown in Figure 65.

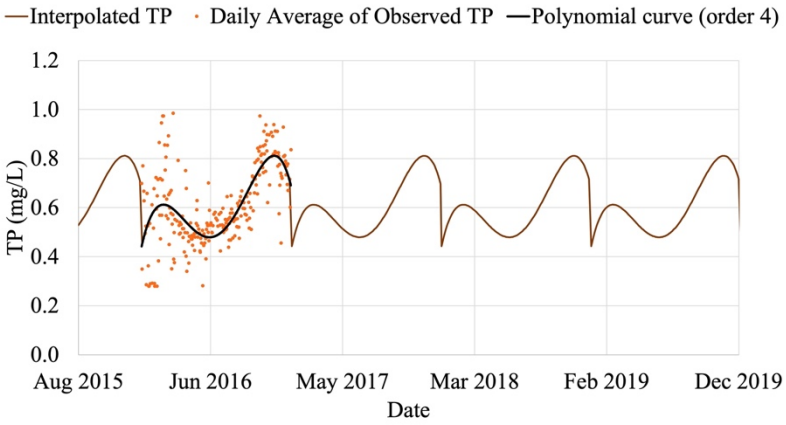


Figure 65: Plot of interpolated TP data and the daily average observed TP data

The daily average observed TP was plotted for the year 2016 for ease of comparison. It is clear the interpolated values had a limited range from about 0.4 to 0.8 mg/L compared to the observed daily average TP which had a much wider range. Interpolation generalised the data and thus could not be used to generate the TP dataset for the require timeline on its own. Thus, the curve was used to influence the randomness of Monte Carlo simulation so the simulations could reflect the seasonal pattern each year experiences.

A.2.3 Lagged TP dataset

The nutrient datasets were time series datasets, however traditional time series forecasting models was not used to simulate the datasets for the dates required. Time series forecasting is dependent on past data to predict/forecast future values (Brockwell & Davis, 2016). This means that the accuracy of the predictions is also dependent on the past data because if one lag is being used then only one "lag" into the future can be forecasted accurately. Nevertheless, predictions can also be used to predict further into the future – if data exists for year one, it can be used to predict year two and those predictions can then be used to predict year 3 and so on. According to

literature, this should be avoided as it becomes very unreliable (Pourahmadi, 2001). Hence, traditional time series methods were avoided as there was not enough available nutrient data to reliably forecast for the required timeline.

Details from the time series analysis was useful in developing the scenarios used to extrapolate the TP dataset. Such details included the significant number of lags being 1506 and 2220. As explained in Chapter 4, lagging is a method of removing seasonality from a dataset (Brockwell & Davis, 2016). The difference between a dataset and itself lag results in a horizontal dataset. These differences is used in Scenarios 2 to 5. Therefore, the TP dataset was lagged as shown in Table 42.

Table 42: Extract of TP dataset and lagged datasets

			Lags/Deltas			
Data Point	**Date**	**Observed TP**	**1506**	**TP-1506**	**2220**	**TP-2220**
1	1980/03/11	0.3055				
2	1980/03/12	0.6400				
3	1980/03/13	0.6700				
…	…	…				
1506	1996/08/13	0.0775				
1507	1996/08/14	0.0580	0.3055	0.2475		
1508	1996/08/21	0.0430	0.64	0.597		
1509	1996/08/27	0.0480	0.67	0.622		
…	…	…	…	…		
2220	2008/12/09	0.4475	0.8830	0.4355		
2221	2008/12/16	0.4950	0.7140	0.2190	0.3055	-0.1895
2222	2008/12/18	0.3540	0.7350	0.3811	0.6400	0.2861
2223	2008/12/22	0.4060	0.7170	0.3110	0.6700	0.2640
2224	2008/12/30	0.5460	0.7510	0.2050	0.5170	-0.0290

The deltas refer to the calculated differences between the observed TP dataset and the lagged dataset, shown in the columns labelled TP-2220 and TP-1506, in grey. These differences were calculated for the dates shown in the table and thus differences for the required timeline were simulated using Monte Carlo simulation. Therefore, a distribution was fitted to deltas, TP-1506 and TP-2220.

Fitting distributions to TP-1506 and TP-2220

The descriptive statistics of the deltas were calculated so that theoretical distributions could be selected and fitted to the delta datasets, in the same way as done previously. Table 43 shows the statistics of the delta datasets.

Table 43: Descriptive statistics of the delta datasets used in simulation scenarios

Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness	Coefficient of Variance
TP-1506	0.20	-0.43	-0.44	3.71	0.38	-0.46
TP-2220	0.23	-0.03	-0.04	2.81	0.07	-6.84

The descriptive statistics revealed that a Normal, t Location Scale or Logistic distribution could fit both deltas for reasons stipulated in Table 44.

Table 44: Theoretical PD for TP-1506 and TP-2220 datasets

Dataset	Theoretical PD	Reason for Selection
TP-1506	Normal	Skewness close to 0 kurtosis close to 3 mean almost equal to median
	t Location Scale	similar to normal
	Logistic	kurtosis > 3 so not perfect normal as has wider tails
TP-2220	Normal	Skewness between -0.5 and 0.5 kurtosis close to 3 mean almost equal to median
	Logistic	similar to normal for trial

Once fitted, a goodness-of-fit test was performed with results shown in Table 45.

Table 45: Goodness-of-fit results for distributions fitted to TP-1506 and TP-2220 datasets

Dataset	Theoretical PD	AD p-value	AIC
TP-1506	Normal	0.82	**-163.3**
	t Location-Scale	0.79	-162.7
	Logistic	0.50	-158.7
TP-2220	Normal	0.94	**-53.0**
	Logistic	0.64	-42.1

The distributions fitted to TP-1506 and TP-2220 resulted in p-values greater 0.05 and thus the best fitting distribution for both deltas was the normal distribution as it had the lowest AIC value. The parameters of the selected distributions were calculated in MATLAB and are shown in Table 46.

Table 46: Parameters of the distributions fitted to the deltas used in simulation scenarios

		Parameters	
Dataset	**PD**	**Location**	**Scale**
TP-1506	Normal	-0.4289	0.1973
TP-2220	Normal	-0.0330	0.2258

The parameters were used to calculate a CDF for each dataset and differences for the required timeline were simulated using Monte Carlo simulation. The simulated differences were then used in the scenarios as described.

A.3 Generating TP Dataset using Scenario 1

In this scenario, Monte Carlo simulation was done as detailed in Section 3.5.4, but with a slight difference; the random numbers were generated within the range calculated for each date. The interpolated nutrient datasets represented the annual seasonal curve the data was supposed to follow every year. Thus, the simulated nutrient datasets should have fallen in the ±95% confidence intervals of this curve. The curve fitter app generated coefficients for ±95% confidence intervals (see Section A.2.2). These coefficients were used to interpolate the nutrient datasets twice more – one to represent the lower range and the other to represent the higher range. The CDF for these interpolations were then calculated. Random numbers for Monte Carlo simulation were then generated within the CDF range each date now had. The following equation was used to achieve this:

$$random\ number = \left(rand() * (x_{max} - x_{min})\right) + x_{min} \quad (11)$$

where $rand()$ is an Excel function that generates random numbers, x_{max} is the +95% CDF point for that date and x_{min} is the -95% CDF point for that date. Figure 66 displays the CDFs and interpolated TP.

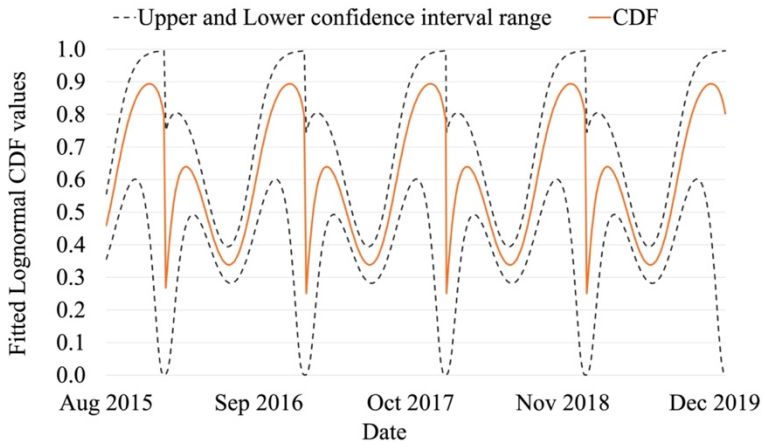


Figure 66: Plot of fitted CDF with confidence interval ranges for the interpolated TP data

The graph shows that the random number generated for 1 August 2015 fell in the range of about 0.4 and 0.6 mg/L. The simulated TP using Monte Carlo simulation fell within the limits of the curve shown as black dotted lines in the figure. Figure 67 shows the generated TP dataset using the method in Scenario 1.

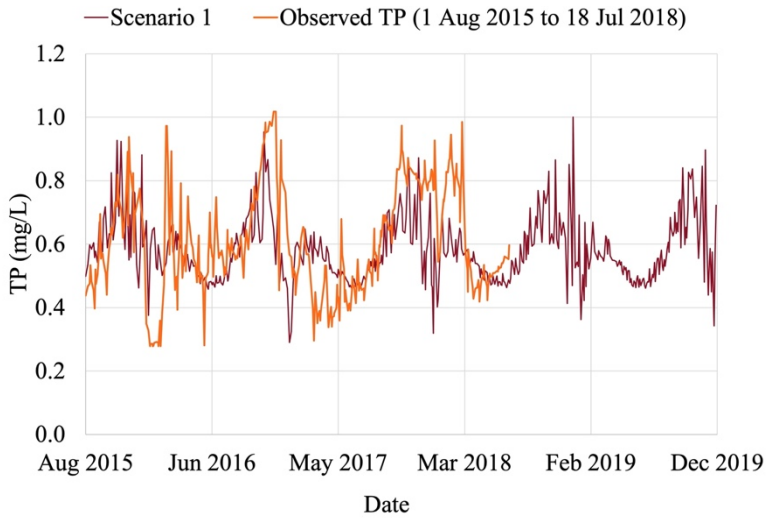


Figure 67: Plot of observed TP dataset and TP dataset generated in Scenario 1

The generated TP dataset closely represented the observed TP dataset and better displayed the seasonal patterns compared to the simulated TP dataset using Monte Carlo simulation in Section A.2. The descriptive statistics were compared to that of the observed TP for the period 1 August 2015 to 18 July 2018. This was done to analyse the extent to which the generated TP dataset represented the observed TP dataset and so that the scenarios could be compared for the best fitting dataset to be chosen. Table 47 shows the descriptive statistics.

Table 47: Descriptive statistics of observed TP dataset compared to that of Scenario 1

TP Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.1654	0.6069	0.5696	2.4518	0.3279
Scenario 1	0.1038	0.5793	0.5605	4.4835	0.9759
% difference	6.2	2.8	0.9	203.2	64.8

The percent difference was calculated using Equation (8) and refers to the difference between the statistics. The green cells indicate those with differences less than or equal to 5%. The red cells indicate the statistic with the highest percent difference. The statistics of the generated TP dataset were close to the observed, with the mean and median having a less than 5% percent difference. However, the kurtosis and skewness had large differences meaning the tails were simulated poorly. The tails are where the extreme values of a dataset are found and, thus, the peaks of the seasonal changes are found in the tails (Wasserman, 2010; Heiberger & Holland, 2015). Therefore, if the tails are not simulated well, then it implies that the seasonal patterns are not simulated well.

However, it is clear when comparing the graphs, that Scenario 1 significantly improved the simulation of TP compared to using ordinary Monte Carlo simulation because it captured the yearly trend whereas the ordinary Monte Carlo simulation in Figure 68 was random.

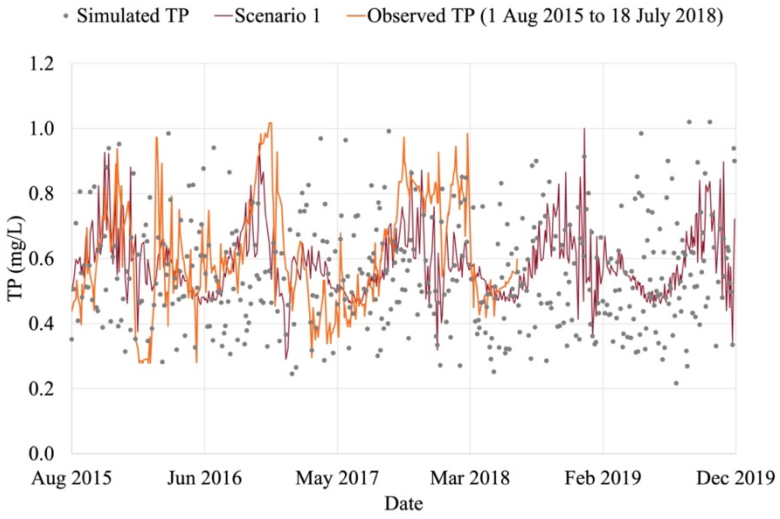


Figure 68: Plot of Scenario 1 and the TP dataset simulated using Monte Carlo simulation

A.4 Generating TP Dataset using Scenario 2

In this Scenario, the simulated differences for TP-1506 were added to the generate TP dataset in Scenario 1. This was done to include more variability as the generated TP dataset in Scenario 1 had very similar data for each year. By using the lagged differences, the seasonal patterns seen in the observed TP dataset as still included. Figure 69 shows the observed TP dataset on the same set of axes as the generated TP dataset in Scenario 2.

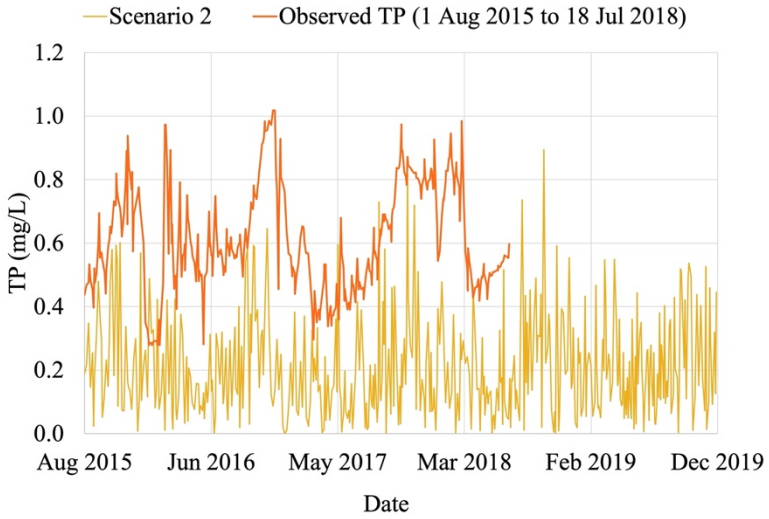


Figure 69: Plot of observed TP dataset and TP dataset generated in Scenario 2

Scenario 2 did not represent the observed TP dataset well as it fell in a much lower range. The descriptive statistics were compared, in Table 48, to reveal how different the datasets were. The statistics revealed that only the standard deviation remained similar. The other statistics had differences greater than 40 with kurtosis having the highest difference of 164%.

Table 48: Descriptive statistics of observed TP dataset compared to that of Scenario 2

TP Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.17	0.61	0.57	2.45	0.33
Scenario 2	0.16	0.21	0.17	4.10	1.09
% difference	0.7	40.0	40.2	164.7	76.0

There was an average difference in statistics of 64% which is very high. This makes sense as the generated TP dataset does not visually represent the observed TP dataset well.

A.5 Generating TP Dataset using Scenario 3

This scenario is similar to Scenario 2 with the only difference being that TP-2220 was used instead of TP-1506. To generate the TP dataset in Scenario 3, the differences were added to the generated TP dataset in Scenario 1. This dataset is plotted in Figure 70.

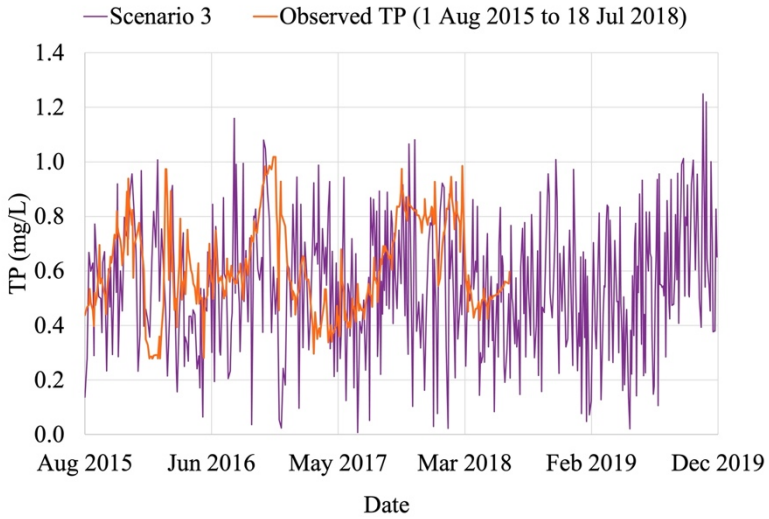


Figure 70: Plot of observed TP dataset and TP dataset generated in Scenario 3

The generated TP dataset fell in a similar range to the observed TP dataset unlike in Scenario 2. Scenario 3 also followed the seasonal pattern and so seemed to fit the observed TP dataset well. To confirm this, the difference descriptive statistics was calculated in Table 49.

Table 49: Descriptive statistics of observed TP dataset compared to that of Scenario 3

TP Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.17	0.61	0.57	2.45	0.33
Scenario 3	0.25	0.55	0.54	2.46	0.07
% difference	8.1	6.1	2.6	0.9	26.2

From the table, it is clear that the generated TP dataset in Scenario 3 was closer to the observed TP dataset, in statistics, than in Scenario 2, with an average difference of 9%. This low average difference indicated that the generated TP dataset was a good representation of the observed TP dataset. In addition, the difference in kurtosis and skewness was 0.9% and 26.2%, respectively which was significantly lower than in Scenarios 1 and 2. This meant that the generated TP dataset in Scenario 3 simulated the tails of the observed TP well. This is important as the seasonal patterns and extreme values are found in the tails, and thus the seasonal patterns were well represented.

A.6 Generating TP Dataset using Scenario 4

In Scenario 4, the TP dataset was generated by simply added the TP-1506 dataset to the interpolated TP data shown in Section A.2. The absolute value was taken of each point to ensure TP values did not fall below zero. This scenario aimed to bring more variability to the smoothed and generalised curve that the interpolated TP data make, while still including the seasonal patterns. Figure 71 plots the generated and observed TP datasets.

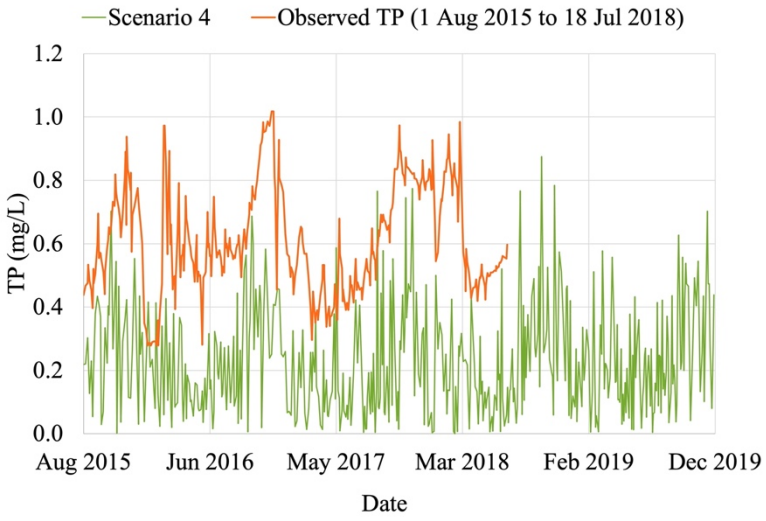


Figure 71: Plot of observed TP dataset and TP dataset generated in Scenario 4

Scenario 4 clearly had the yearly seasonal pattern evident in the observed TP dataset, however its range was lower, similarly to in Scenario 2. As done in other scenarios to validate the dataset, the descriptive statistics were also calculated in Table 50.

Table 50: Descriptive statistics of observed TP dataset compared to that of Scenario 4

TP Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.17	0.61	0.57	2.45	0.33
Scenario 4	0.17	0.22	0.19	3.55	0.92
% difference	0.4	38.3	37.9	109.8	59.4

The average difference was 49% which was lower compared to in Scenarios 1 and 2. However, the percent difference in kurtosis was very high, 109.8%, which suggests that the generated TP dataset did not capture the tails of the observed TP dataset.

A.7 Generating TP Dataset using Scenario 5

Scenario 5 was the same as Scenario 4 in method with TP-2220 added to the generated TP dataset in Scenario 1 instead of TP-1506. Figure 72 shows this dataset.

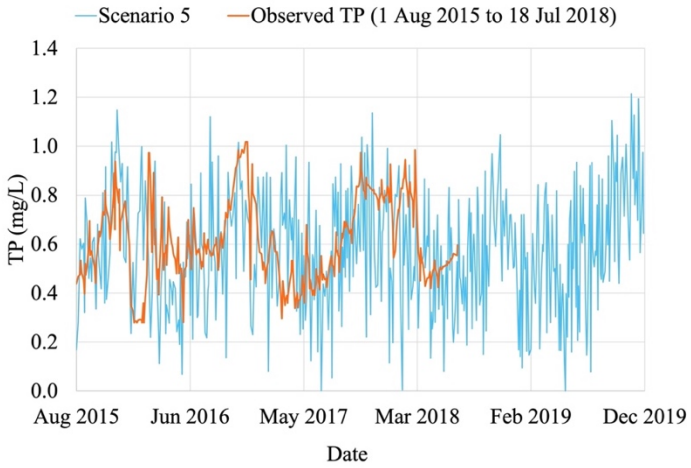


Figure 72: Plot of observed TP dataset and TP dataset generated in Scenario 5

Scenario 5 had a similar range to the observed TP dataset and followed the seasonal pattern in the observed TP dataset more closely than Scenario 3 did. To confirm this, the descriptive statistics were compared. The differences were calculated and are shown in Table 51.

Table 51: Descriptive statistics of observed TP dataset compared to that of Scenario 5

TP Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.17	0.61	0.57	2.45	0.33
Scenario 5	0.25	0.58	0.57	2.39	0.01
% difference	8.2	3.1	0.1	6.7	31.9

The average difference was 10% which was 1% higher than in Scenario 3 and much lower than in the other scenarios. It appears that Scenario 5 fitted the observed TP dataset better than Scenario 4 but not necessarily visually better than Scenario 3. Therefore, a statistical test was done to select a generated TP dataset.

A.8 Selecting a TP Dataset

A statistical test was done to determine which generated TP dataset represented the observed TP dataset the best. A two sided t-test was done, in MATLAB, which tested whether the observed and generated TP datasets came from distributions with similar means and variances (Curran,

2013). Although the most used statistical test for comparing datasets is the KS test, the KS test could not differentiate the scenarios and resulted in p-values of 0 for all scenarios (Nguyen et al., 2017). Thus, the t-test was used to get comparable results as it tests the datasets in a similar way to the KS test – instead of comparing distributions, it compared the means and variances of the distribution the datasets come from. Figure 73 presents the p-values calculated for each generated TP dataset.

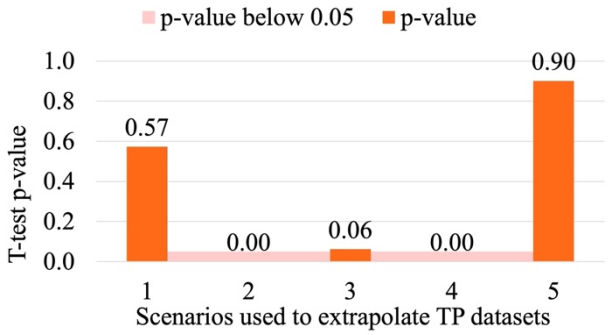


Figure 73: T-test p-values of the different TP datasets generated

The figure shows that Scenarios 2 and 4 were below 0.05 at zero, and thus did not have a PD with a similar mean and variance as the observed TP. This means that these scenarios did not represent the observed TP well which was implied when validating the datasets visually and comparing statistics. However, Scenarios 1, 3 and 5 fitted the observed TP well with p-values above 0.05. The best fitting simulation was Scenario 5 with a p-value of 0.9 indicating that the distribution in Scenario 5 had a mean and variance almost identical to that of the observed TP dataset, as the value was close to 1. Therefore, the TP dataset generated in Scenario 5 was used in model building.

APPENDIX B Details of Extrapolating TN Dataset

This appendix discusses in detail the methods used and results of the extrapolated TN dataset. The methods used were the same as those used to extrapolate the TP dataset with only some details and results differencing. Figure 40 in Chapter 6 outlines the process followed and sections presented in this appendix. The TN dataset was extrapolated after the missing data was imputed and the two nitrogen variables were added to form the TN dataset.

The first step in the process was time series analysis were the seasonal patterns and significant lags were identified to be later used in developing the five scenarios to extrapolate the TN dataset. The scenarios developed made use of three methods: Monte Carlo simulation, interpolating from a curve, and lagging. These methods were carried out and then applied to the scenarios in the ways detailed in subsequent sections. The best generated TN dataset, determined using a statistical test, was then used in model building.

B.1 Time Series Analysis of TN Dataset

Figure 74 shows the observed TN dataset after multiple KNN imputation from 12 January 1999 to 14 December 2011. From the time series plot, it is evident there was a seasonal pattern as shown by the hills that repeat yearly. These hills were of similar size with peaks occurring in the middle of each year. This suggests a seasonal pattern that repeats about every 6 months. However, some years seem to act like outliers such as 2000 and 2003 where the peaks, circled in black, were higher and lower, respectively, compared to in other years. In addition, from 2006 there appears to be an increasing trend as the peaks of the seasonal pattern increase yearly from 2006.

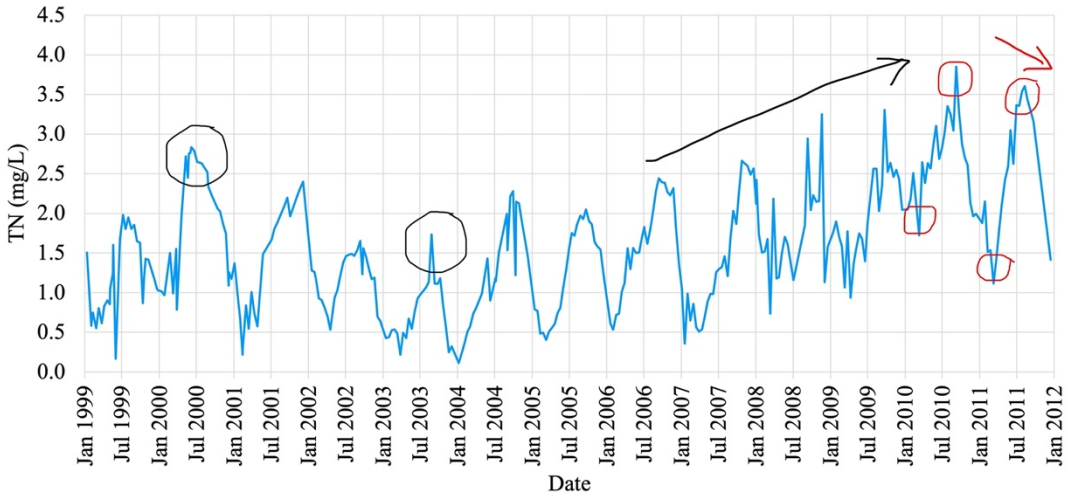


Figure 74: Time series plot of observed TN dataset with trends

The red circles show that January and July in 2011 decreased slightly compared to January and July in 2010. Thus, to confirm seasonality and the increasing trend, an ACF was plotted in MATLAB using the econometric app, as shown in Figure 75.

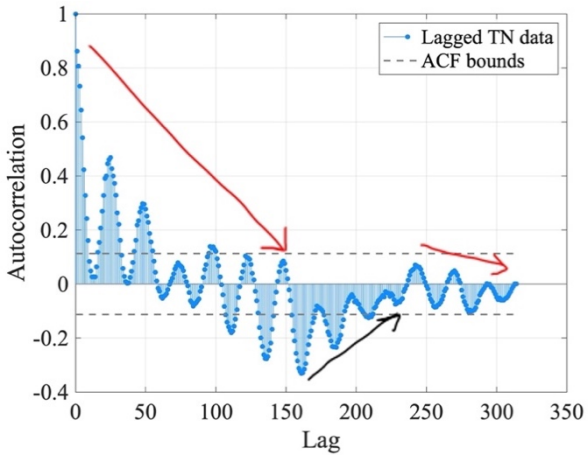


Figure 75: ACF plot of the observed TN dataset

The lags refer to the number of the data point after the dataset is shifted forward by one data point, ending at 314 as the TN dataset only had 315 data points. The ACF bounds as well as the autocorrelations were automatically calculated using MATLAB. The multiple parabolas or hills seen in the ACF confirms the up-and-down seasonal trend implicit in Figure 75. It is also clear that there was a decreasing trend that was followed by an increasing trend from about lag 150 to lag 250, but the troughs and peaks started decreasing again thereafter. This was not as clear in Figure 74. Therefore, the ACF confirmed that the increasing trend did not continue into 2012 and potentially into the years 2015 to 2019. Carrol and Curtis (2021) state that the influx of nutrients, TN and TP, in the Hartbeespoort dam from 2010 to 2016 were increasing at a rate higher than the outflux suggesting that there was a steady increase in TN as seen in the figures. However, the researchers also make it clear that the measures of TN and TP in the dam have been worse in the past, and that regardless of the increasing trend seen, the nutrient levels would not be worse than in the past, which was confirmed by the start of a decrease in 2011. Based on this, it was assumed that the TN remained the same for the period 1 August 2015 to 28 December 2019 as the observed TN dataset. Therefore, the seasonal patterns, seen as hills in the figure, was the focus of the time series analysis. Figure 76 shows the first 80 lags of the ACF plotted.

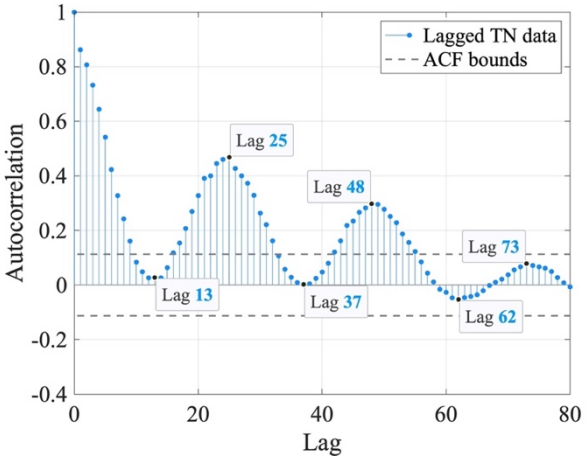


Figure 76: ACF plot of the observed TN dataset for the first 80 lags

The peaks and troughs occurred every 11, 12 or 13 points indicating a change in direction of the trend, as seen in Figure 76. Due to the fact that the TN dataset did not have data recorded daily;

every 12 points did not mean every 12 days. Figure 77 shows dates that relate to the data point numbers for the first 20 lags.

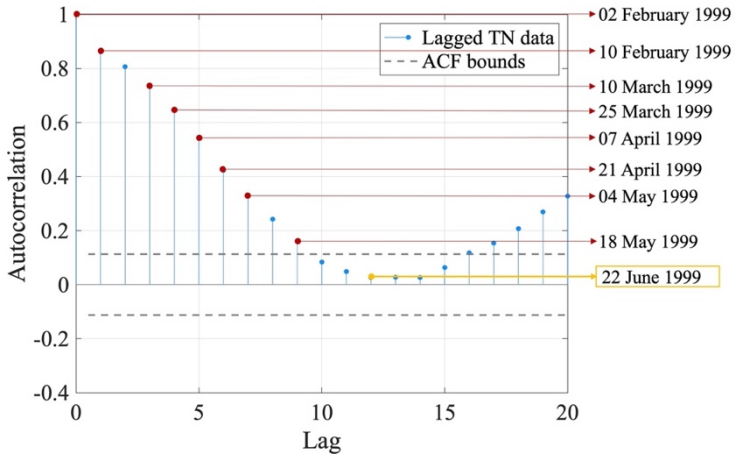


Figure 77: ACF plot of the observed TN dataset with dates in relation to the first 20 lags

As the figure shows, the 12th point was 22 June 1999. Almost every 12th or 13th data point that was a peak or trough related to a date that was about 6 months from the previous. This means that a seasonal pattern occurs about every 6 months as anticipated in the time series plot in Figure 74. Using this information, a TN dataset for 1 August 2015 to 28 December 2019 was simulated, ensuring the critical seasonal patterns were reflected in the simulation. Finally, a PACF was plotted to determine which lags were significant and should be used in simulating the datasets, as done in the extrapolation of the TP dataset.

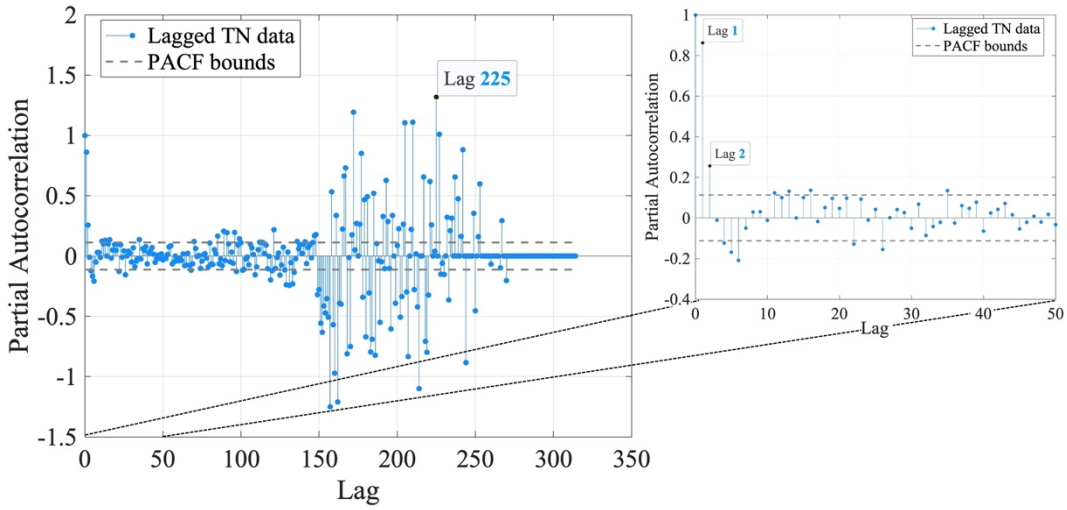


Figure 78: PACF plot of TN data

The PACF showed that lag 225 was the most significant as it had the highest partial autocorrelation. However, if only the first 50 lags are considered, lags 1 and 2 are very significant compared to the rest. Therefore, lags 1 and 2 were used in simulating the TN dataset. This is because if a lag of 225 data points were used, only 90 data points would be left over which is too little to use.

B.1.1 Summary of time series analysis on TN datasets

In summary, the analysis had similar results to that of the TP dataset: there was annual seasonal trends with peaks occurring in the middle of each year. The trends identified proved that TN data would remain similar in the required timeline to the observed TN dataset and thus the same assumption for extrapolating the TP dataset is made for the TN dataset. This assumption was that the annual seasonal curve remained the same for all years including the years 2015 to 2019, and thus could be used to generate a TN dataset.

B.2 Developing Five Scenarios for Extrapolating TN Dataset

The method followed to develop the scenarios was the same as that used to extrapolate the TP dataset. Figure 79 outlines the processes completed to develop the scenarios in Table 52.

Table 52: Summary of scenarios used to extrapolate the TN dataset

Scenario	Summary of method
1	Interpolated TN data were used to calculate the ranges each random value should fall into in the Monte Carlo simulation.
2	Monte Carlo simulation of the difference between the TN dataset and itself lagged by 1 data point (called TN-1) was added to the TN dataset generated in Scenario 1.
3	Monte Carlo simulation of the difference between the TN dataset and itself lagged by 2 data points (called TN-2) was added to the TN dataset generated in Scenario 1.
4	The TN-1 dataset, simulated in Scenario 2, was added to the TN data interpolated from the polynomial curve representing the annual seasonal pattern.
5	The TN-2 dataset, simulated in Scenario 2, was added to the TN data interpolated from the polynomial curve representing the annual seasonal pattern.

The scenarios are the same as those used to extrapolate the TP dataset with only the number of lags being different.

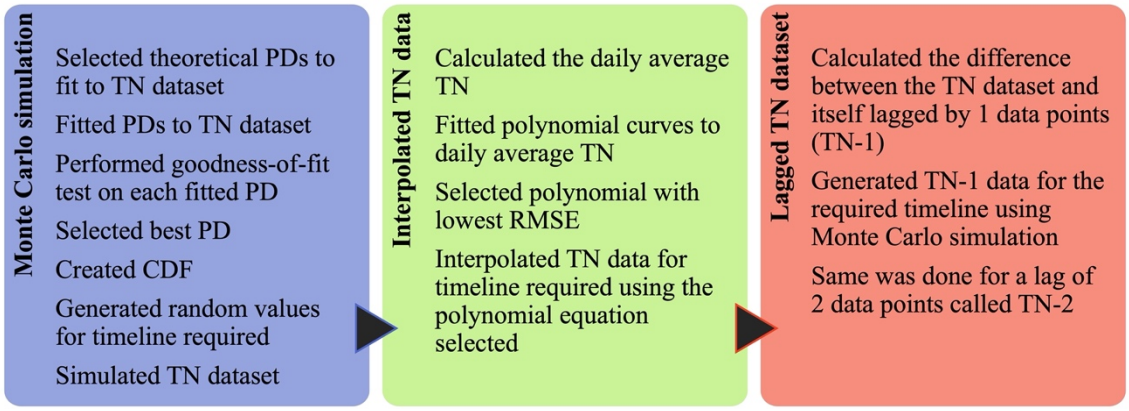


Figure 79: Development of scenarios used to extrapolate the TN dataset

As seen in the figure, three methods were used being the Monte Carlo simulation, interpolation of TN data from a curve, and lagging the TN dataset. This section discusses the results of each, using the same steps as used in Appendix A.

B.2.1 Monte Carlo simulation

Three PDs were fitted to the observed TN dataset after which a goodness-of-fit test was performed to determine the best fitting PD. Next, the PD was used to simulate the TN dataset using Monte Carlo simulation. The random numbers were generated on Excel and were completely random.

Fitting a PD to observed TN dataset

The descriptive statistics of the observed TN datasets, shown in Table 53, were used to suggest a few theoretical distributions to be fitted to the observed TN dataset.

Table 53: Descriptive statistics of the observed TN dataset

Standard Deviation	Mean	Median	Kurtosis	Skewness	Coefficient of Variance
0.78	1.59	1.54	2.56	0.38	0.49

Three theoretical PDs were selected based on the reasons shown in Table 54, which were the same as those selected for the TP dataset.

Table 54: Theoretical PDs selected for fitting to the observed TN dataset

Theoretical PD	Reason for Selection
Normal	skewness is between -0.5 and 0.5 kurtosis is close to 3
Lognormal	skewness > 0 coefficient of variance > 0.3
Weibull	skewness > 0

Next, the goodness-of-fit test was done to determine which of the above three was a better fit and would describe the observed TN dataset well.

Goodness-of-fit test

As done for the distributions fitted to the observed TP dataset, the loglikelihood, the AD test p-value, and the AIC were calculated. The results are shown in Table 55.

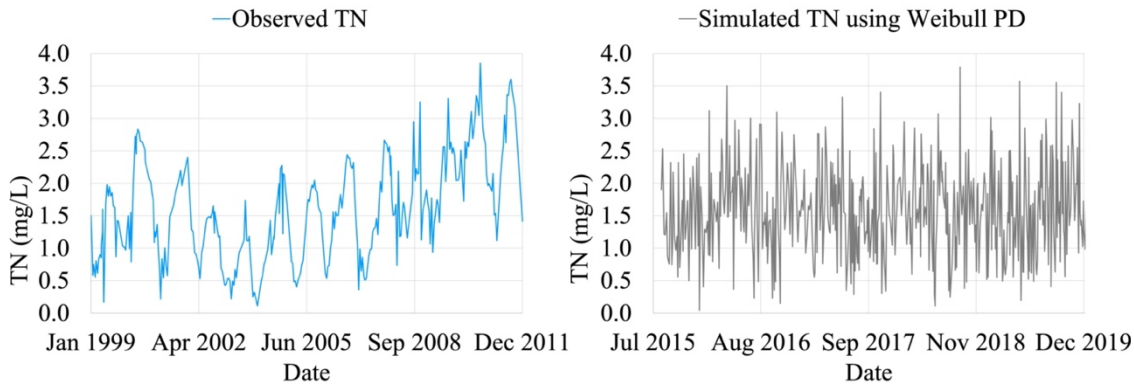
Table 55: Goodness-of-fit results for the PDs fitted to observed TN dataset

Theoretical PD	AD p-value	AIC
Weibull	0.401	**716**
Normal	0.002	740
Lognormal	0.001	772

The Weibull distribution was the best fitting PD as it was the only PD with a p-value greater than 0.05 and it had the lowest AIC value. The shape and scale parameters of the distribution were 1.7972 and 2.1621, respectively, which were then used to simulate the TP dataset using Monte Carlo simulation.

Problems with Monte Carlo simulation

The seasonal trends evident in the observed TN dataset are not present in the TN dataset simulated using Monte Carlo simulation due to the randomness the method depends on. In addition, the randomness changes for each iteration. For instance, the simulation generates 3.366 mg/L of TN for 1 August 2015, however when ran again, 0.025 mg/L of TN is generated for the same date. That difference would have influenced the WHC predictions due to the hyperbolic relationship WHC and TN have (Aoyama & Nishizaki, 1993). Figure 80 plots the simulated TN dataset.



(a) (b)

Figure 80: Scatter plot of observed TN dataset (a) and the simulated TN dataset (b)

Unlike the observed TP dataset, the observed TN dataset did not have any observed data points within the required timeline which could be used to validate the generated TN dataset. It is for this reason more than one scenario was developed and that the datasets were validated through visual inspection, comparing descriptive statistics and a statistical test. The statistical test was ultimately used to choose a generated TN dataset as explained in later sections.

B.2.2 Interpolated TN data

The annual seasonal TN trend was represented by a curve that was fitted to the average daily TN data, shown in Figure 81.

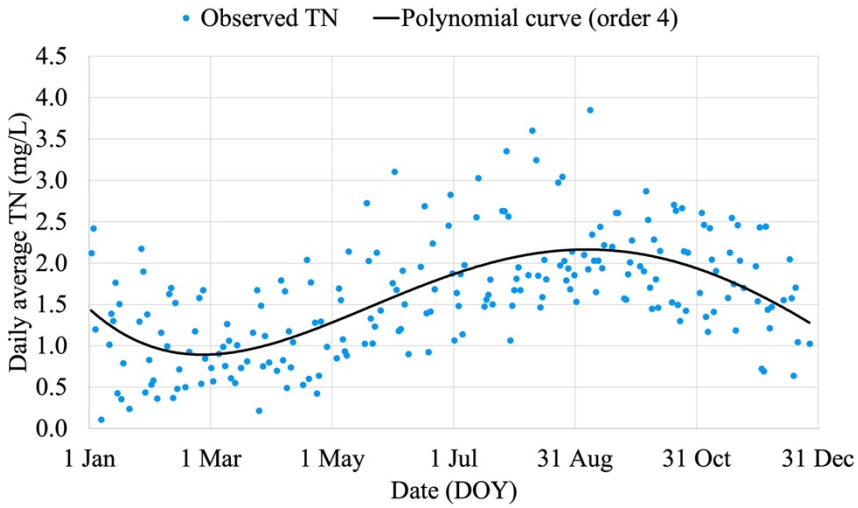


Figure 81: Scatter plot of the daily average TN data

DOY refers to the Day of Year that was calculated in MATLAB so a generic year could be plotted. After the curve was fitted, and the one with the lowest RMSE chosen, the TN data for the required timeline was interpolated. This section discusses how the TN data were interpolated.

Fitting polynomial curves to the observed TN dataset

Figure 82 displays the RMSE calculated for each of the three polynomials fitted to the daily average TN data.

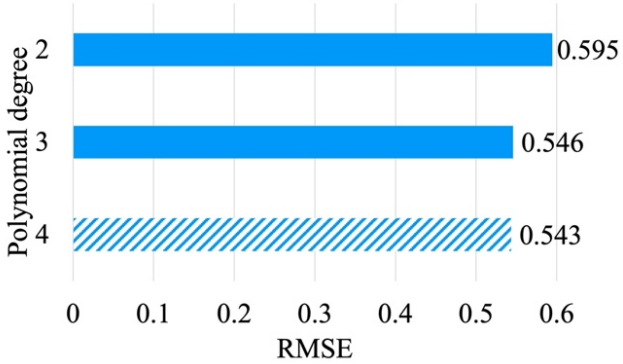


Figure 82: RMSE of fitted polynomial curves to the daily average TN data

The figure shows that the polynomial with 4 degrees had the lowest RMSE, as was the case for the daily average TP data. The details of the polynomial as shown in Table 56, which was then used to interpolate the TN data for the required timeline using the steps outline in Appendix B.2.

Table 56: Details of the polynomial fitted to the daily average TN data

Polynomial Equation	Details
$y = p_1 * x^4 + p_2 * x^3 + p_3 * x^2 + p_4 * x + p_5$ where: x is DOY, normalized by mean 180.8 and standard deviation 103.8 y is daily average TN, in mg/L	Polynomial with 4 degrees. Coefficients (with 95% confidence interval): $p_1 =$ 0.0920 (−0.0164 , 0.2005) $p_2 =$ −0.3053 (−0.3998 , −0.2108) $p_3 =$ −0.4377 (−0.7254 , −0.1500) $p_4 =$ 0.8699 (0.6881 ,1.052) $p_5 =$ 1.8520 (1.7130 , 1.991)

Problems with interpolating TN data from a curve

A scatter plot was made of the interpolated TN data as shown in Figure 83.

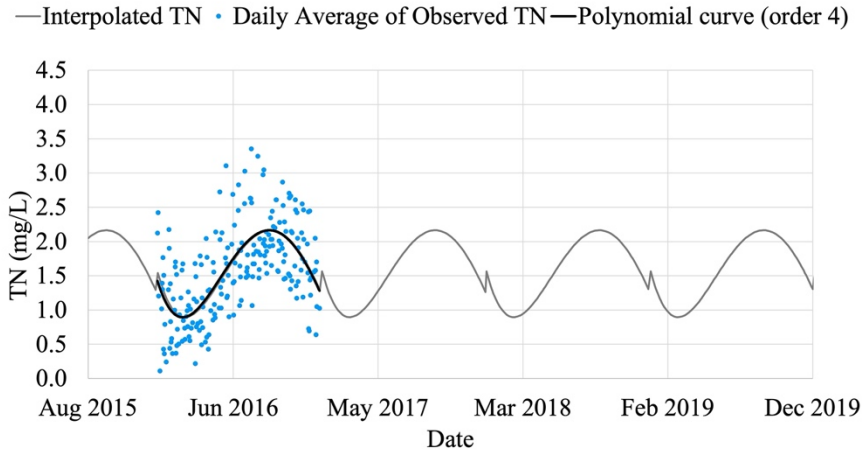


Figure 83: Scatter plot of the interpolated TN data

As seen in the figure, the interpolated TN data follows the polynomial curve that was fitted to the daily average TN data. Note that the daily average TN is the same for any year and was plotted for the year 2016 for ease of comparison, and that there were no outliers present in the daily average observed TN data. The interpolated TN repeats the curve every year. The figure also shows that the interpolated TN is very generalised as it does not capture the full range of the daily average TN. However, the curve is useful in reflecting the annual seasonal trend evident in the observed TN dataset.

B.2.3 Lagged TN dataset

For Scenarios 2 to 5 the datasets were lagged to remove some of the seasonality to improve simulations. This method is explained in more detail in Appendix A as the same method was used when lagging the TP dataset. The observed TN dataset was lagged by 1 then 2 data points as shown in the extract in Table 57.

Table 57: Extract of observed TN dataset and lagged datasets

			Lags/Deltas			
Data Point	Date	Observed TN	1	TN-1	2	TN-2
1	1999/01/12	1.5051				
2	1999/02/02	0.5827	1.505125	-0.922411		
3	1999/02/10	0.7481	0.582714	0.165411	1.5051	-0.757
4	36217	1	0.748125	-0.196625	0.5827	-0.031214
5	1999/03/10	0.8051	0.5515	0.253625	0.7481	0.057
6	1999/03/25	0.6129	0.805125	-0.19225	0.5515	0.061375
7	1999/04/07	0.8304	0.612875	0.2175	0.8051	0.02525
8	1999/04/21	0.9023	0.830375	0.071875	0.6129	0.289375
9	36284	1	1	0	0.8304	0.023625
10	1999/05/05	1.0472	0.8540	0.1932	0.90225	0.1450

The lags are in the columns labelled 1 and 2 in blue, and thus, the columns leave out the first 1 and 2 dates, respectively. To get the deltas later used to simulate the TN dataset, the lags were subtracted from the observed TN dataset. This is shown in the columns labelled TN-1 and TN-2 which are the calculated differences, deltas, for the first and then second lags.

Fitting distributions to TN-1 and TN-2

As done in Appendix A, the descriptive statistics of TN-1 and TN-2 were calculated for theoretical PDs to be chosen, which are shown in Table 58.

Table 58: Descriptive statistics of the TN-1 and TN-2 datasets

Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness	Coefficient of Variance
TN-1	0.41	0.00	0.01	7.42	-0.52	-1410.90
TN-2	0.47	0.01	0.04	5.70	0.00	59.59

The normal, t Location Scale and logistic distributions were fitted to both TN-1 and TN-2 for the reasons listed in Table 59. The goodness-of-fit results are shown in Table 60.

Table 59: Theoretical PD selected for fitting to deltas used in simulation scenarios

Dataset	Theoretical PD	Reason for Selection
TN-1	Normal	skewness almost between -0.5 and 0.5 mean almost equal to median
	t Location Scale	similar to normal
	Logistic	kurtosis > 3 so not perfect normal as has wider tails
TN-2	Normal	skewness = 0 mean almost equal to median
	t Location Scale	similar to normal
	Logistic	kurtosis > 3 so not perfect normal as has wider tails

Table 60: Goodness-of-fit results for distributions fitted to TN-1 and TN-2 datasets

Dataset	Theoretical PD	AD p-value	AIC
TN-1	t Location-Scale	0.88	**262**
	Logistic	0.12	283
	Normal	0.00	333
TN-2	t Location-Scale	0.59	**395**
	Logistic	0.43	397
	Normal	0.00	425

The deltas TN-1 and TN-2 had one and two distributions with p-values greater than 0.05, respectively, meaning that those distributions fitted the deltas well. Hence, the selected distributions were based on the AIC values calculated. The best fitting PD was thus the t Location-Scale distribution for both TN-1 and TN-2, with parameters listed in Table 61.

Table 61: Parameters of the t Location Scale distribution fitted to the TN-1 and TN-2 datasets

		Parameters		
Dataset	**PD**	**Location**	**Degrees of Freedom**	**Scale**
TN-1	t Location-Scale	0.0176	2.5324	0.2382
TN-2	t Location-Scale	0.0214	4.1569	0.3498

The distributions were used to calculate the CDFs used in simulating the differences for the required timeline.

B.3 Generating TN Dataset using Scenario 1

The method used is explained in detail in Appendix A as it was the same method. Monte Carlo simulation was done using random numbers generated within the range determined for each date, using the interpolated CDF values. Figure 84 shows the calculated CDF and confidence intervals.

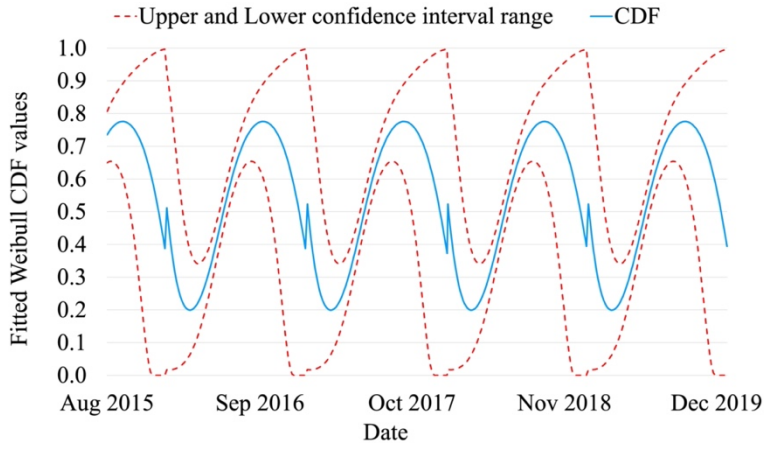
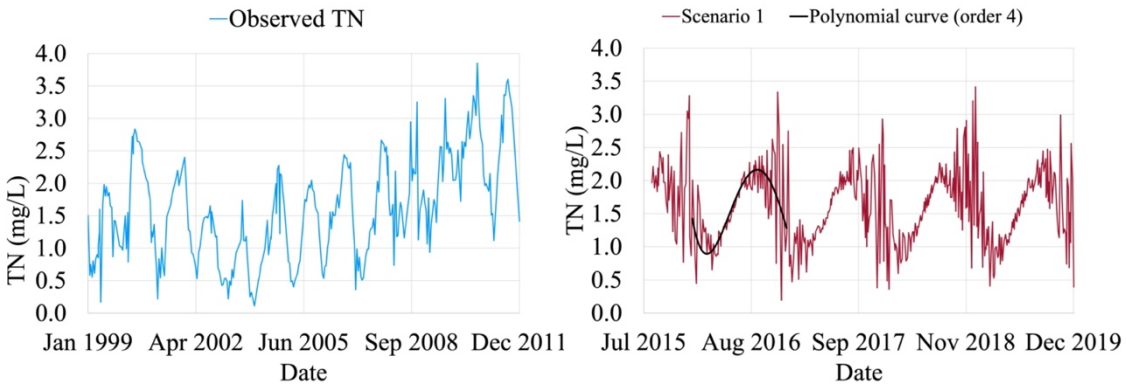


Figure 84: Plot of CDF with confidence interval ranges for the interpolated TN data

From the figure, the random number generated for 1 August 2015 for TN fell in the range of about 0.6 and 0.8 mg/L, as an example. Figure 85 shows the observed TN dataset alongside the TN dataset generated in Scenario 1.



(a) (b)

Figure 85: Plot of observed TN dataset (a) and TN dataset generated in Scenario 1 (b)

As can be seen in the figure, the generated TN dataset is less random than in Section B.2. It also has the same range of TN as the observed TN dataset. This shows that including the curve in the simulations improved results. Note that the increasing trend apparent in Figure 85(a) from 2006 to 2011 was proven to not continue into the years 2015 to 2019 and that the TN will remain the same as in 1998 to 2011 in general – refer to Section B.1. Thus, only the yearly seasonal trend is simulated. The descriptive statistics, shown in Table 62, were calculated and compared to validate that generated dataset.

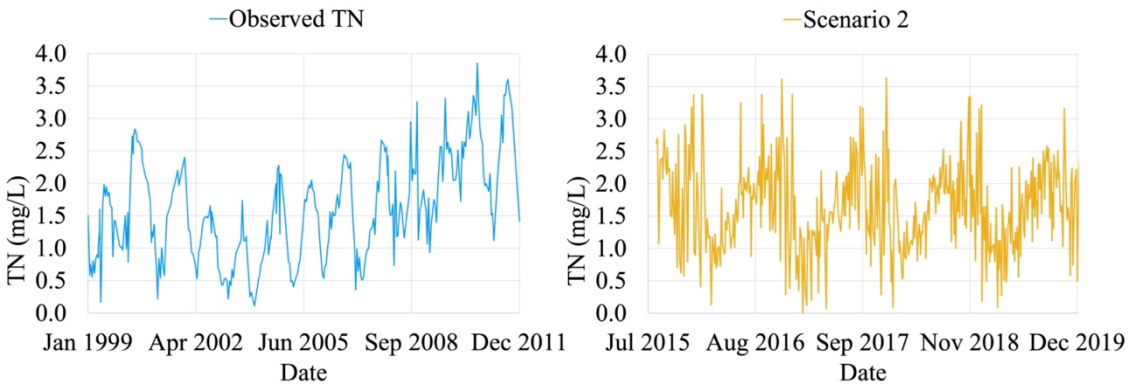
Table 62: Descriptive statistics of the TN dataset generated in Scenario 1

TN Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.78	1.59	1.54	2.56	0.38
Scenario 1	0.57	1.61	1.60	3.11	0.32
% difference	20.9	1.6	5.8	55.1	5.7

The statistics show the TN dataset generated in Scenario 1 were close to the observed TN statistics as all except kurtosis and standard deviation were below 6% different. The green cell indicates a difference less than 5%, which was the mean. The kurtosis had the highest difference of 55.1% meaning that the simulations had wider tails than the observed. Hence, the generated TN dataset had more extreme values than the observed TN dataset. Otherwise, the simulations were close to the observed in terms of statistics.

B.4 Generating TN Dataset using Scenario 2

The PACF suggested one lag is significant. Consequently, the difference between the TN dataset and itself after being lagged by one data point was calculated (TN-1). This was done by subtracting a data point from the data point that is one point before it. However, the exact same deltas could not be used as it is from a different time period. For this reason differences for the years 2015 to 2019 were simulated by means of Monte Carlo simulation. To do so, a PD was first chosen and then differences were simulated. These simulated differences were then added to the TN dataset generated in Scenario 1. Figure 86 shows the observed TN dataset and the TN dataset generated in Scenario 2.



(a) (b)

Figure 86: Plot of observed TN dataset (a) and TN dataset generated in Scenario 2 (b)

When comparing the the TN dataset generated in Scenario 2, shown in the figure, to Scenario 1 in Figure 85, it is clear adding the differences results in more variability to the dataset so the data is not as clustered around the mean. To further analyse the difference between the generated and observed TN datasets, the descriptive statistics were calculated and are shown in Table 63.

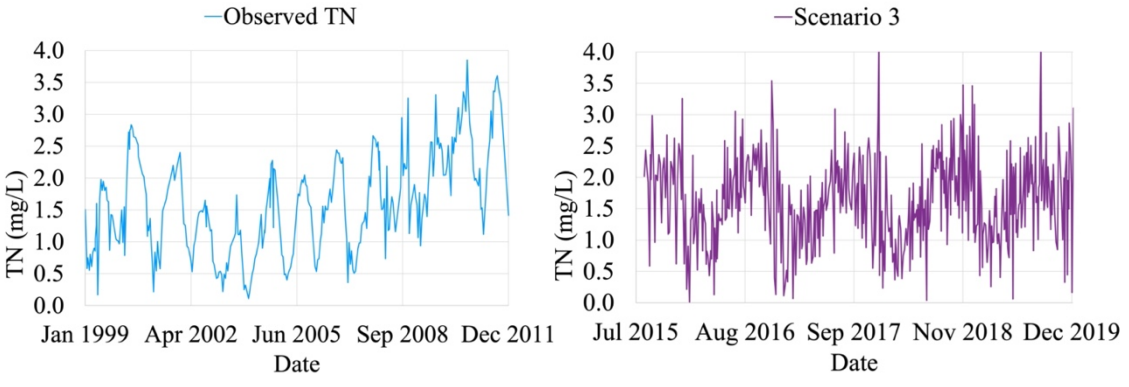
Table 63: Descriptive statistics of the TN dataset generated in Scenario 2

TN Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.78	1.59	1.54	2.56	0.38
Scenario 2	0.68	1.63	1.58	3.04	0.30
% difference	9.4	3.6	4.2	47.9	7.4

It is clear from the statistics that the generated TN dataset was similar in statistics to the observed TN dataset as most of the statistics were less than 10% different – an average difference of 14%. However, the kurtosis had a high difference of 47.9% meaning the generated TN dataset had heavier tails than the observed. Therefore, the generated TN dataset had more extreme values.

B.5 Generating TN Dataset using Scenario 3

Similarly to Scenario 2, the observed TN dataset was lagged but by 2 data points instead of 1. After simulating the differences, TN-2, the TN dataset was generated by adding those differences to the TN dataset generated in Scenario 1. Figure 87 plots the generated TN dataset next to the observed TN dataset.



(a) (b)

Figure 87: Plot of observed TN dataset (a) and TN dataset generated in Scenario 3 (b)

From visual inspection, the generated TN dataset seems to have a few outliers but otherwise appears to represent the observed TN dataset well. To examine the difference between the generated and observed TN datasets, the descriptive statistics were compared in Table 64. The statistics were calculated after replacing the outliers, which were identified using Equations (3) and (4).

Table 64: Descriptive statistics of the TN dataset generated in Scenario 3

TN Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.78	1.59	1.54	2.56	0.38
Scenario 3	0.71	1.63	1.62	2.97	0.17
% difference	7.2	3.7	7.8	41.3	20.2

The average difference was 16% with similar results to the TN dataset generated in Scenario 2. From the statistics, Scenario 2 appears to be closer to the observed TN dataset.

B.6 Generating TN Dataset using Scenario 4

As summarised in Table 52, Scenario 4 added the simulated TN-1 differences to the interpolated TN dataset to form the TN dataset. Figure 88 shows this dataset and compares it to the observed TN dataset.

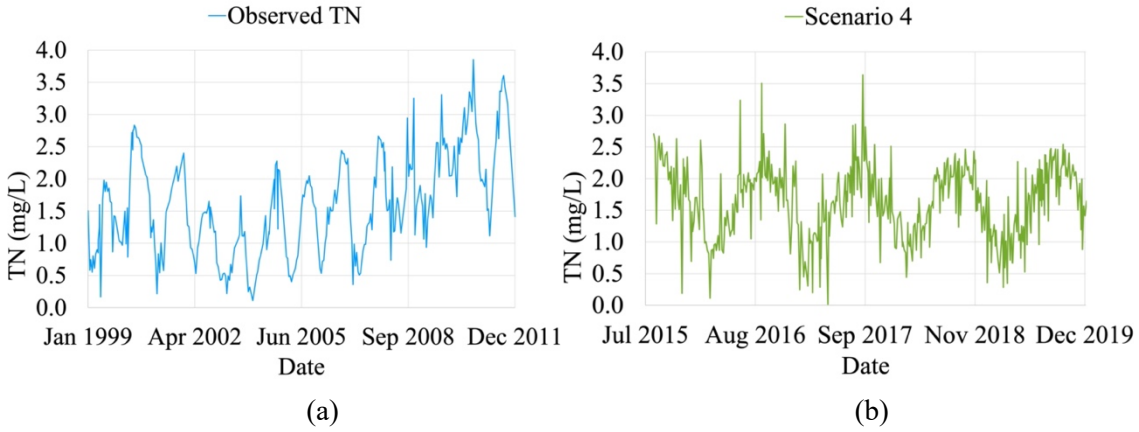


Figure 88: Plot of observed TN dataset (a) and TN dataset generated in Scenario 4 (b)

The figure suggests that the TN dataset generated in Scenario 4 had a few outliers, which were then replaced. The dataset represented the seasonal trends well. However, the data was very similar in each year. Table 65 shows the difference in descriptive statistics to validate the generated TN dataset.

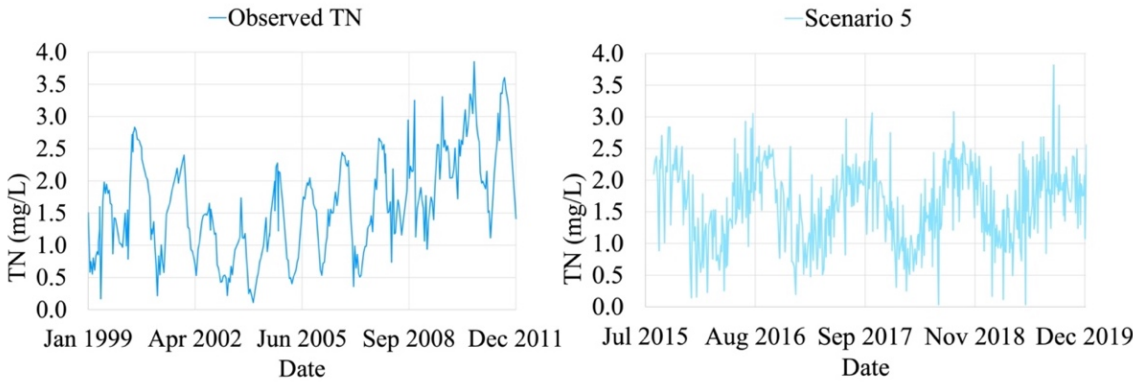
Table 65: Descriptive statistics of the TN dataset generated in Scenario 4

TN Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.78	1.59	1.54	2.56	0.38
Scenario 4	0.57	1.62	1.65	2.86	-0.08
% difference	21.2	2.6	10.3	30.5	45.4

As with the other generated TN datasets, Scenario 4 had a high difference in kurtosis and skewness however the other statistics were close to the observed TN dataset. On average the difference was 22% which was not as low as in Scenarios 2 and 3.

B.7 Generating TN Dataset using Scenario 5

Scenario 5 followed the same method as Scenario 4, but using TN-2. Figure 89 plots the TN dataset generated in Scenario 5.



(a) (b)

Figure 89: Plot of observed TN dataset (a) and TN dataset generated in Scenario 5 (b)

Similarly to in Scenario 4, the TN dataset generated in Scenario 5 had a few outliers that were replaced, and the seasonal patterns were evident. The dataset also had more variability in each year which better represented the observed TN dataset. The descriptive statistics, shown in Table 66, were close to the descriptive statistics in Scenario 5 and the same conclusions were made. However, the average difference was 16% which meant the generated TN dataset in Scenario 4 was slightly worse in terms of statistical difference compared to in Scenario 5. The difference in Scenario 3 was also 16% and so the two datasets were statistically similar.

Table 66: Descriptive statistics of the TN dataset generated in Scenario 5

TN Dataset	Standard Deviation	Mean	Median	Kurtosis	Skewness
Observed	0.78	1.59	1.54	2.56	0.38
Scenario 5	0.63	1.61	1.65	2.64	-0.05
% difference	15.0	2.1	10.4	8.2	42.8

The scenarios were compared using a statistical test to select the most appropriate TN dataset.

B.8 Selecting a TN Dataset

The KS test was done on each TN dataset generated. This tested whether the observed and generated TN datasets came from the same unknown PD. The KS test p-value was calculated in MATLAB and if it was equal to or above 0.05, then the datasets came from similar PDs and, thus, was a good fit (Ricci, 2005). Figure 90 shows the p-values for each generated TN dataset.

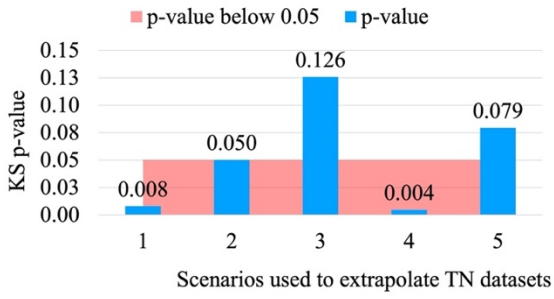


Figure 90: KS p-values of the different generated TN datasets

In the figure, Scenarios 2, 3 and 5, were above 0.05 and therefore had the same PD as the observed TN dataset, while Scenarios 1 and 4 did not. Scenario 3 resulted in the highest p-value of 0.126 and thus was the best generated TN dataset out of the other generated TN datasets. The TN dataset generated Scenario 3 was used in model building.

APPENDIX C Codes for Multiple KNN Imputation

```
%% A FUNCTION FOR MULTIPLE KNN IMPUTATION OF DATA

% Function first needs all variables to be defined for the loop to work. It then
% creates a while loop which loops until the number of repeats are completed.
% The loop first permeates randomly from 1 to the end of the dataset so that
% the dataset can be split into groups with random data points. There are g
% number of groups (determined by silhouette method) that are then created using
% the indices determined. Each group then gets sgl imputed using the single
% imputation function created. The mean is then taken (by row ie by date) and
% then the final means are placed into a table with the dates in a column for
% future use.

% % % n = total number of data points in dataset
% % % AllData = complete dataset with all NaN
% % % g = number of groups
% % % repeat = number of iterations

function MultiImpt = multimpt(AllData,n,g,repeat)

    MIdata = zeros(n,repeat);   % creating a var to put multiple imputation
results
    iteration = 1;

    % going to split the data (x data points) into groups (g)
    x = ceil(n/g);
    leftover = x*g - n;

    groups = zeros(x,g);    % creating a var to put indices

    data = zeros(n,1);      % creating var to put data after sglimpt
    date = zeros(n,1);      % creating var to put dates after sglimpt

    while  iteration < (repeat + 1)
        % permeates the data points randomly so that i can split it into g
        i = randperm(n)';

        %  splitting into g groups so now have indicies for the different g
        for m = 1:g
            f = x*(m-1) + 1;
            e = x*m;
            if e > n
                f = f - leftover;
                e = n;
            end
            groups(:,m) = i(f:e);   % determining groups

            datam = AllData(groups(:,m),:);   % creating the group
            MIm = sglimpt(datam);   % sglimpt the group

            date(f:e) = datenum(MIm.Date);
            data(f:e) = MIm.Data;    % saving the data and dates of each g
        end

        names = {'Date','Data'};
        T = sortrows(table(datetime(date,"ConvertFrom","datenum"),data,...
            'VariableNames',names),'Date','ascend');

        MIdata(:,iteration) = T.Data;

        iteration = iteration + 1;
    end

    % now can calculate the average of the iterations to get final dataset
    MultiImpt = table(AllData.Date,round(mean(MIdata,2),6),'VariableNames',...
              names);
end
```

```
%% FUNCTION FOR IMPUTING MISSING DATA

% Function first converts table data into a matrix with date in first row and
% data in second row (dates are converted to serial number), then calculates k
% and imputes the matrix using knnimpute, the imputed matrix is then converted
% to a table by changing the serial numbers into dates and having dates in
% column 1 then data in column 2.

% % % Z = Dataset in table format with date labelled as Date and data, such as
% % %            TN or WHC, labelled as Data.

function Impt = sglimpt(Z)
    % converting to matrix
    Dates = datenum(Z.Date)/10^5;
    DataMatrix = [Dates, Z.Data]';

    % choose k by using Rule of Thumb
    k = round(sqrt(numel(DataMatrix(:,1))));

    % imputing data by means of knn
    ImputedData = knnimpute(DataMatrix,k);

    % converting the serial numbers back into the dates
    Dates = datetime(Dates*10^5,"ConvertFrom","datenum");

    % putting all data back into table for feature selection later on
    Impt = table(Dates,ImputedData(2,:)','VariableNames',{'Date','Data'});
end
```

APPENDIX D Code for Final Model

```
% Returns a trained supervised regression model and its RMSE. This code recreates
% the model trained in Regression Learner app. Use the generated code to automate
% training the same model with new data, or to learn how to programmatically
% train models.

%  Input:
%      trainingData: A table containing the same predictor and response columns
%      as those imported into the app.

%  Output:
%      trainedModel: A struct containing the trained regression model. The
%      struct contains various fields with information about the trained model.

%      trainedModel.predictFcn: A function to make predictions on new data.

%      validationRMSE: A double representing the validation RMSE. In the app,
%       the Models pane displays the validation RMSE for each model.

% Use the code to train the model with new data. To retrain your model, call
% the function from the command line with your original data or new data as the
% input argument trainingData.

% For example, to retrain a regression model trained with the original data set
% T, enter:
%   [trainedModel, validationRMSE] = trainRegressionModel(T)

% To make predictions with the returned 'trainedModel' on new data T2, use
%   yfit = trainedModel.predictFcn(T2)

% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedModel.HowToPredict

% Auto-generated by MATLAB on 13-Jun-2024 17:25:25

function [trainedModel, validationRMSE] = EnsembleBoostFI(trainingData)

% Extract predictors and response
% This code processes the data into the right shape for training the model.
inputTable = trainingData;
predictorNames = {'MaxTemp', 'MinTemp', 'TP', 'TN', 'AvgTemp', 'DateNum', ,...
             'TNTP'};
predictors = inputTable(:, predictorNames);
response = inputTable.WH;
isCategoricalPredictor = [false, false, false, false, false, false, false];

% Data transformation: Select subset of the features
% This code selects the same subset of features as were used in the app.
includedPredictorNames = predictors.Properties.VariableNames([true true ...
             false false false true true]);
predictors = predictors(:,includedPredictorNames);
isCategoricalPredictor = isCategoricalPredictor([true true false false ...
             false true true]);

% Train a regression model
% This code specifies all the model options and trains the model.
template = templateTree(...
    'MinLeafSize', 9, ...
    'NumVariablesToSample', 'all');
regressionEnsemble = fitrensemble(...
    predictors, ...
    response, ...
    'Method', 'LSBoost', ...
    'NumLearningCycles', 500, ...
    'Learners', template, ...
    'LearnRate', 0.1026575025717173);
```

```
% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
featureSelectionFcn = @(x) x(:,includedPredictorNames);
ensemblePredictFcn = @(x) predict(regressionEnsemble, x);
trainedModel.predictFcn = @(x)
ensemblePredictFcn(featureSelectionFcn(predictorExtractionFcn(x)));

% Add additional fields to the result struct
trainedModel.RequiredVariables = {'MaxTemp', 'MinTemp', 'TP', 'TN',...
             'AvgTemp', 'DateNum', 'TNTP'};
trainedModel.RegressionEnsemble = regressionEnsemble;
trainedModel.About = 'This struct is a trained model exported from ...
             Regression Learner R2023b.';
trainedModel.HowToPredict = sprintf('To make predictions on a new table, T,...
    use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the name of the ...
    variable that is this struct, e.g. ''trainedModel''. \n \nThe table, T,...
    must contain the variables returned by: \n  c.RequiredVariables ...
    \nVariable formats (e.g. matrix/vector, datatype) must match the ...
    original training data. \nAdditional variables are ignored. \n \nFor ...
    more information, see <a href="matlab:helpview(fullfile(docroot,...
    ''stats'', ''stats.map''), ''appregression_exportmodeltoworkspace'')...
    ">How to predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'MaxTemp', 'MinTemp', 'TP', 'TN', 'AvgTemp',...
             'DateNum', 'TNTP'};
predictors = inputTable(:, predictorNames);
response = inputTable.WH;
isCategoricalPredictor = [false, false, false, false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedModel.RegressionEnsemble, 'KFold', 5);

% Compute validation predictions
validationPredictions = kfoldPredict(partitionedModel);

% Compute validation RMSE
validationRMSE = sqrt(kfoldLoss(partitionedModel, 'LossFun', 'mse'));

end
```