Data-driven sensitivity mitigation techniques for Genetic Algorithm - Long Short Term Memory water quality prediction model

Dhruti Dheda, 458389

UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

School of Electrical & Information Engineering
University of the Witwatersrand, Johannesburg, Gauteng

April 28, 2021

A dissertation presented for the degree of Masters of Electrical and Information Engineering

Declaration

I hereby declare that this dissertation titled "Data-driven sensitivity mitigation techniques for Genetic Algorithm - Long Short Term Memory water quality prediction model" is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

Dhruti Dheda

Signed this 28 day of April 2021

Abstract

A long short-term memory (LSTM) model developed for the prediction of water quality, based on the historical data of a particular water body, and as such a particular water quality dataset, will only be applicable to that dataset. Thus if a specific LSTM prediction model is applied to another dataset, then it is quite possible that the prediction model will fail to make an accurate prediction. These models tend to be case study specific. This research focuses on improving the tolerance (mitigating the discrepancies in model prediction capability that arise from differences in datasets) of LSTM prediction models. The two different LSTM models developed from two different water quality datasets, the Burnett and Baffle models, are optimised using the metaheuristic genetic algorithm (GA). The two hybrid GA-optimised LSTM base models, the GA-Burnett and GA-Baffle models, are fused together using a weight-based approach to form a final robust and tolerant predictive ensemble model. Both the models contribute equally to the average ensemble model. In the weighted ensemble model, the GA-Burnett model only has a 10% greater contribution than the GA-Baffle model.

Generally, the ensemble models outperform the GA-optimised hybrid LSTM models. The four models are tested on unseen and unrelated datasets and the performance of all the models are consistently similar to one another on each dataset. The consistency of performance exhibited by the different models on any particular dataset is evidence of the successful mitigation of the discrepancies of the individual LSTM models through the implementation of the linear weight

based fusion of two hybrid GA-optimised LSTM models. The models are not only applicable for the prediction of water quality, but also for domains outside of the water sector; thus asserting the relevance of the models, especially the weighted ensemble model in the wider field of LSTM and ensemble prediction.

This research involves the water quality of rivers. Water is a critical natural resource that is currently under threat, especially rivers. The models are able to successfully predict the quality of river water ahead of time, in terms of dissolved oxygen concentration. Water quality prediction aids in increasing the efficiency of water quality monitoring. Efficient water quality monitoring enables effective water management. Effective water management is necessary for the preservation of rivers.

Dedication

I dedicate this research to all the researchers and academics, who have worked tirelessly throughout the COVID-19 pandemic to achieve and complete research outcomes despite the ongoing strain and uncertainty of the current situation. Many researchers were directly impacted and tested COVID positive and had to work towards recovering both psychically and mentally, whist still completing their research. Other researchers, were indirectly impacted and lost loved ones to the pandemic, or had to sacrifice weeks and sometimes months of research time to take care of loved ones recovering from COVID, or had to shoulder more responsibilities at home to accommodate family members who were front-line workers, or had to find additional work to foot the bills if their household lost an income during pandemic layoffs, all whilst managing their psychological and physical well-being and completing their research. There is no researcher who was unaffected by the COVID pandemic and yet so many researchers across numerous disciplines persevered and achieved their research outcomes. As a researcher who has had to deal with these problems, I took heart from other researchers who had managed to overcome them and hence I dedicate this completed dissertation to all researchers and academics, who symbolise the persevering light of intellectual pursuit even in the darkest of times.

Acknowledgements

I would like to thank the DSI CSIR-Interbursary Support Programme for awarding me with the necessary funding to complete my research.

I would like to acknowledge Dr. Adnan M. Abu-Mahfouz from the CSIR Emerging Digital Technologies for 4IR (EDT4IR) Research Centre for his support in the completion of this study.

I would like to especially thank my supervisor, Prof. Ling Cheng for his unwavering guidance, support, patience and most importantly for the faith he had in me as I embarked on a tedious and insightful journey into a field of research which was completely new to me. Prof. Cheng's guidance not only helped me complete my research, but also made me a better researcher and a more focused thinker. One of my new favourite idioms, courtesy of Prof Cheng is "to toss a brick to attract jade".

I'm grateful to Mr. Craig Carlson and Dr. Stephen Levitt for the assistance given to me during the ethics waiver application process.

I would like to acknowledge Dr. Dheda and Mrs. Dheda for their time and support.

A special thank you is always in order for my favourite person and number one cheerleader, my sister, Shiksha Dheda. The voice of reason in confusing times.

I would like to thank Veeddeya Dheda for her unique perspective in all matters.

I would like to thank Rohan Chhipa for his advice in technical matters and for letting me use

his hardware to complete this study.

I would also like to thank Tanya Blaeser for listening to my research woes for an entire year and Aarti Panday for her patience.

Also a huge special thank you to Aviv for being my muse and for inspiring creativity in me.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
AR	Autoregression
ARIMA	Auto Regressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BP	Back Propagation
C	Celsius
CM	Centimeter
DE	Differential Evolution
DNN	Deep Neural Network
DO	Dissolved Oxygen
EC	Evolutionary Computation algorithms
GA	Genetic Algorithm
GPU	Graphics Processing Unit
IoT	Internet of Things
IQR	Interquartile Range
L	Liter
LSTM	Long Short-Term Memory

MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MG	Milligrams
MLP	Multilayer Perceptron
MLR	Multilinear Regression
mS	Millisiemens
MSE	Mean Squared Error
NAR	Nonlinear Autoregressive
NAV	Net Asset Value
NN	Neural Network
NTU	Nephelometric Turbidity Unit
PCA	Principle Component Analysis
PSO	Particle Swarm Optimisation
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RMSprop	Root Mean Square Propagation
RNN	Recurrent Neural Network
SVR	Support Vector Regression
USA	United States of America
WQI	Water Quality Index

List of Symbols

Vectors are defined by bold variable symbols and matrices are defined with capital variable symbols

α	Activation function of RNN
$\circ$	Hadamard product
$\cdot$	Multiplication
$\forall$	For all
$\hat{y}_i$	Predicted output value of the model's i^{th} observation
$\mathbf{b}_o$	Bias vector of output gate
$\mathbf{b}$	Deviation vector for calculation of the state of RNN hidden layer
$\mathbf{c}$	Deviation vector for calculation of expected RNN output
$\mathbf{h}_t$	Hidden state of LSTM cell
$\mathbf{x}(t)$	Input at time step t of RNN
$\overline{x}$	Mean of the data
σ_x	Standard deviation of the data
$\sqrt{x}$	Square root

$\sum$	Mathematical sum
$\tilde{\mathbf{c}}_t$	Intermediate cell state of LSTM cell
n	Number of samples
Q_1	Lower boundary (25^{th} percentile)
Q_3	Upper boundary (75^{th} percentile)
R^2	Coefficient of determination
$\tanh$	Hyperbolic tangent function
t	Time step
U	Weight matrix between the input layer and the hidden layer of RNN
V	Weight matrix between the hidden layer and the output layer of RNN
W_f	Weight of the forget gate
W_i	Weight of the input gate
w_i	Weight assigned to the i^{th} forecast
W	Weight matrix between the current hidden layer and the hidden layer at the next time step of RNN
x'	Normalised data value
x	Original data value
y_i	Desired output/true values at the model's i^{th} observation
$\mathbf{b}_c$	Bias vector of intermediate cell state

$\mathbf{b}_f$ Bias vector of forget gate

$\mathbf{b}_i$ Bias vector of input gate

$\mathbf{c}_t$ Current cell state of LSTM

$\mathbf{c}_{t-1}$ Memory of the previous LSTM cell state

$\mathbf{f}_t$ Forget gate of LSTM cell

$\mathbf{h}_{t-1}$ Output from previous hidden state of LSTM cell

$\mathbf{i}_t$ Input gate of LSTM cell

$\mathbf{o}_t$ Output gate of LSTM cell

$\mathbf{s}(t)$ State of the hidden layer at time step t of RNN

$\mathbf{s}(t-1)$ State of the hidden layer at the previous time step of RNN

$\mathbf{x}_t$ Input at current time step of LSTM cell

W_c Weight of the intermediate cell state

W_o Weight of the output gate

$\mathbf{y}(t)$ Expected output of the time step t of RNN

σ Logistic sigmoid

$^\circ C$ Degree Celsius

Chapter 1

Introduction

1.1 Introduction

Increased urbanisation, climate change and poor water infrastructure has placed a significant amount of pressure on existing water resources, such as rivers. Rivers are invaluable as inland water resources for industrial and recreational purposes, agricultural needs and human consumption. The stress placed on existing water resources, especially rivers necessitate the efficient management of these resources. To effectively manage these water resources (rivers), the quality of the water needs to be continuously monitored [7].

Water quality is commonly approximated through expensive and time-consuming laboratory analyses. This method includes water sample collection from the relevant river, the correct storage and transportation of samples to the laboratory, chemical laboratory tests and analysis (which require a fair amount of time and costly equipment) after which the water quality results can be obtained. Throughout this long process there is more than enough room for error and inefficiency [8]. Water management can be made more efficient, if along with data analysis, water quality can be predicted beforehand [9].

Thus, this study suggested optimised Long Short-Term Memory (LSTM) water quality

prediction models, which will be used as base models for the development of a robust and tolerant ensemble water quality prediction model.

Recurrent neural networks (RNNs) take the temporal aspects of data into account and hence are an appropriate choice for sequential data, such as time series water data. A major drawback of RNNs is vanishing and exploding gradients. LSTM can mitigate this drawback due its unique architecture. LSTM is the most appropriate network for sequential data in which temporal dependency is an implicit feature [3] such as time sequential water quality data for several years used to make predictions about water quality in the future.

Discrepancies can arise when water quality datasets are taken from different sites (rivers) and/or times. These discrepancies can make the development of an LSTM model for the prediction of the water quality of one river (using a water quality dataset specific to that river) only applicable to that river. This implies that if a specific LSTM water quality prediction model is applied to another river and thus another water quality dataset, the chances that the prediction model will fail to make an accurate prediction is quite high. Thus, these models are case study specific. However, this research aims to improve the tolerance (mitigate the discrepancies) of these LSTM prediction models through the optimisation of the LSTM network using a metaheuristic algorithm, namely the genetic algorithm (GA) and then the fusion of two GA-optimised LSTM models through a weight-based approach for a final tolerant LSTM ensemble model.

As this research focused on increasing the tolerance of the LSTM models, the final ensemble model should not only be applicable to the prediction of water quality, but should also be capable of prediction in domains outside of the water sector and thus was tested in other domains.

1.2 Motivation for the application of neural networks for water research

To improve water resource management and plan ahead, it is important to evaluate water quality parameters, in order to enhance the performance assessment of water resources [10]. Najah et al. [10] uses Artificial Neural Networks (ANNs) to predict water quality parameters at Johor River Basin, located in Johor state, Malaysia. The study concluded that the ANN prediction model was robust and reliable for both the simulation and prediction of the water quality parameters. In the study by Diamantopoulou et al. [11], by using existing water quality parameters as input variables, ANNs were able to predict the monthly values of three water quality parameters of the Strymon river at a station located in Sidirokastro Bridge near the Greek - Bulgarian borders. The ANNs were trained through the cascade correlation algorithm (a supervised algorithm in the multilayer feed forward ANNs) and Kalman's learning rule was used to modify the weights of the ANNs. The study concluded that the model was successful in predicting the water quality parameters and also had the capacity to find and fill in the missing values of the time series data of the other water quality parameters (which is a serious problem facing Greek water monitoring stations) [11].

Another study by Areerachakul et al. [12] aimed to make a model that can quickly and efficiently classify the water quality of 288 canals in Bangkok, Thailand. Previously, lab instruments and sensors had been used to monitor the water quality of the canals, which was costly and time consuming (as stated by this dissertation as well). This particular study employed the Multilayer perceptron (MLP) neural network with the Levenberg- Marquardt algorithm as it was considered to be more powerful than the conventional gradient descent techniques. The model achieved a high accuracy of 99.34% in classifying the water quality of the canals. Thus, the cost and time of water resource management could be minimized [12].

Three neural networks, namely the back propagation neural network, the modular neural network and the radial basis function network were used to model and predict the Water Quality Index (WQI) of the rivers, Pahang and Selangor in Malaysia in a study by Khuan et al. [13]. The performance of the ANNs, at an optimally tuned set of parameters, was satisfactory and was able to both simplify and decrease the computation time for the computation of the WQI, saving a significant amount of money and time. The modular neural network model was able to closely replicate the real WQI model, in terms of accuracy and fast learning time [13].

Another study by Khan et al. [9] developed a reliable water quality prediction model using a feed-forward neural network with Nonlinear Autoregressive (NAR) time series model; NRA-NN exploited the benefits of both NRA and ANN. Principal Component Analysis (PCA) had been used to determine relationship among different water quality parameters. In Zhou et al. [14], water quality data from the Yangtze River was used to predict the water quality through three ANN networks; a particle swarm optimization (PSO) based ANN was compared to a back propagation (BP) based ANN and GA based ANN. The PSO based ANN and the GA based ANN performed much better than BP based ANN; with only slight differences between the accuracy of the PSO and GA models. Another study of a river in Agra city by Abbaa et al. [15], developed and compared three models, namely multi-linear regression (MLR), ANN and adaptive neuro fuzzy inference system (ANFIS) to predict the dissolved oxygen concentration. The study concluded that both ANN and ANFIS performed considerably better than MLR, with the ANN model producing slightly better results than ANFIS.

Hence it can be surmised that ANNs have been successfully applied to water quality data for the prediction of the water quality of rivers and has outperformed other methods such as MLR and ANFIS in this regard. It can further be surmised that ANNs that have been

optimised by metaheuristic algorithms such as GA and PSO have performed better than ANNs optimised by other methods and it should be noted that PCA has been successfully used in water quality research for determining interactions between water quality parameters.

RNNs are a type of deep neural network (DNN) that accounts for the temporal aspects of data and thus is an appropriate choice for sequential data, such as time series water quality data. However, RNNs face a major drawback with regards to vanishing and exploding gradients. LSTM networks, a specific type of RNN, can mitigate this drawback due to its unique architecture. Its unique architecture allows LSTM to be well suited to sequential data in which temporal dependency is an implicit feature [3] and this is a promising advantageous option for making predictions about water quality in the future based on historical time sequential water quality data. Although the application of ANNs for the prediction of water quality has been common in recent years, only a handful of studies apply LSTM for water quality prediction, which is odd considering the properties of the LSTM architecture.

A study which focuses on the use of LSTM in an internet of things (IoT) environment for the analysis and prediction of the water quality by Liu et al. [16] of the Guazhou Water Source of the Yangtze River in Yangzhou, concluded that the predicted water quality values resultant from the LSTM network were very close to the actual water quality values and validated the use of LSTM for the prediction of water quality. Another study that involves the use of a deep LSTM network to predict the water quality in a smart water mariculture environment by Hu et al. [17], found the prediction accuracy of the LSTM network to be 98.57% and 98.97% for the prediction of pH and water temperature respectively. The study prioritised the LSTM approach over more traditional forecasting methods which were found to have lower accuracy, poor generalisation, and were more time consuming. The LSTM model also performed better than the RNN based prediction model, both in terms of higher prediction accuracy and lower time cost.

Research in a paper by Zhou et al. [18], involved the application of LSTM to two water quality datasets from Tai Lake and Victoria Bay for the forecasting of future water quality. The paper states that the LSTM approach was adopted due to the time sequence of the water quality information. The paper concludes that due to the LSTMs unique architecture, it can take full advantage of the time sequence water quality information and thus outperformed both the Back Propagation (BP) predictive water quality neural network and the Auto Regressive Integrated Moving Average (ARIMA) for water quality prediction. Similarly, another study by Wang et al. [19] used LSTM to predict the water quality of Taihu Lake using previous water quality data and compared its predictive ability to both the BP neural network and online sequential extreme machine learning and found the LSTM approach to not only be more accurate but to also be more generalised than the other approaches. In a different study by Prasad D et al. [20], deep learning models such as ANN, RNN and LSTM were used to estimate the WQI; metrics such as accuracy, precision and execution time were used for comparing the performance of the models. The study concluded that LSTM achieved the highest accuracy at the shortest execution time from all three models.

It can be deduced from the research explored in this study that due to its architecture, LSTM can take advantage of the structure of time sequential water quality data and outperform other methods such as ANNs, RNNs, BP, ARIMA and other traditional methods. LSTMs have also achieved high prediction accuracies in terms of water quality parameters at a low time expense and as such have validated the use of LSTM for water quality prediction.

The success of LSTMs depends heavily on the hyperparameters used to train the LSTM network as these hyperparameters determine numerous aspects of the behaviour of the algorithm. Manual and automatic hyperparameter optimisation are two hyperparameter optimisation methods. The manual hyperparameter optimisation approach involves manually changing and testing each hyperparameter and its effect on the success of the LSTM; this

method is tedious and relies heavily on the knowledge and experience of the researcher. Automatic hyperparameter optimisation approaches range from the modest Grid and Random search to more complicated model-based algorithms. The grid search approach explores all the possible combinations of hyperparameters until the global optima are found; this is a tedious and exhaustive approach. Random search algorithms are easy to implement and are based on direct search methods. The drawback is that these algorithms converge slowly and hence take a fair amount of time to find the global optima [21].

Evolutionary computation algorithms (EC), are a type of random search algorithm that converges faster than the others to find an optimal solution in a small amount of time. The GA, PSO and differential evolution (DE) are types of EC [21]. This research will explore the use of GA to optimise the hyperparameters of the LSTM.

GA is a stochastic and metaheuristic-search-based algorithm, inspired from biology by Darwinian evolution and they can find optimal hyperparameters by simulating a natural evolutionary process [3]. The concept of GA was first published by A. Fraser [22]; these concepts of GA were then further developed by J. H. Holland [23]. One of the earliest studies to use GA to optimise ANN is shown by Montana et al. [24]; researchers attempted to optimise the weights of the network, to overcome the shortcomings of back propagation, the study achieved great results at that time.

One study by Kim et al. [25] used GA for feature discretisation to reduce the dimensionality of the feature space and optimise the weights between layers. The research by Kim et al. [26] used a GA-based multiple classifier technique to predict the stock market index. Another study successfully uses GA to optimise the topology of a neural network by Mahajan et al. [27]; through the binary encoding of the connections between the neurons. Research by Islam et al. [28] sought to optimise the topology of an ANN using GA to optimise the number of neurons per layer, as well as the number of layers for an ANN system trained through back

propagation; both the model computational time and the mean absolute percentage error (MAPE), were distinctly reduced after optimisation with GA. A comparative study by Defilippo et al. [29], compared a neural network optimised by GA in terms of architecture and training parameters to four benchmarks, namely a linear method, two naïve time series forecasts and a neural network trained with grid search parameters for the forecasting of a load in Brazil. Consequently, the GA optimised network outperformed all four benchmarks with the lowest MAPE error.

GAs have also been specifically applied for the optimisation of LSTM. A hybrid GA-LSTM approach was taken by Santra et al. [30] for the forecasting of the electricity load and its performance was compared to that of a standard LSTM; GA-LSTM outperformed the standard LSTM, by significantly minimising the training data MAPE. Another study by Chung et al. [6] takes a similar approach and uses GA to optimise LSTM for stock market prediction, using data from the Korea Stock Price Index (KOSPI), in terms of the time window size and topology of the LSTM network. The hybrid approach was found to perform better than the benchmark model. Research by Gorgolis et al. [31] optimises the hyperparameters of a common LSTM language model using GA and concluded that the hybrid model improved on the performance of a commonly used default model.

Another paper focused on time series forecasting using LSTM optimised by multi heuristics algorithms in a study by Hendri et al. [32]. This paper compared the performance of two EC algorithms for LSTM optimisation, a GA optimised LSTM model was compared to a PSO optimised LSTM model for the forecasting of selected Indonesian mutual funds' Net Asset Value (NAV); the paper concluded that PSO achieves a lower Root Mean Squared Error (RMSE) than GA, however, the GA had a shorter computational execution time than the PSO. Similarly, another study used a multi-sequenced LSTM-RNN deep learning model and metaheuristic algorithms for electric load forecasting by Bouktif et al. [3]; applying both GA

and PSO for the optimisation of LSTM hyperparameters. The study concluded that both the GA-optimised and the PSO-optimised multi sequence LSTM models outperformed the other four benchmark models, namely standard multi-sequence LSTM, random forest, ANN, support vector regression (SVR) and extra trees regressor in terms of RMSE and mean absolute error (MAE) values. The GA-optimised multi-sequence LSTM outperforms the PSO-optimised model with lower RMSE and MAE values.

It can be inferred from the research reviewed that GA has been extensively used for the optimisation of various neural networks, reducing both error and computational time. GA has also been successfully applied for the optimisation of LSTMs in numerous fields, increasing the accuracy of the LSTM and thus validating the use of GA to optimise LSTM in this research.

1.3 Research Question

Can the LSTM based ensemble scheme proposed by this research, improve the tolerance (mitigate the discrepancies of the individual LSTM models) of the hybrid Genetic Algorithm optimised Long Short-Term Memory (GA-optimised LSTM) water quality prediction model, for different water datasets taken from different sites and/or times?

1.4 Aims and Objectives of Research

1.4.1 Aim of research

This research aimed to improve the tolerance (mitigation of discrepancies) of the optimised GA-optimised LSTM water quality prediction model for water quality data from different sites and/or times and for any other possible unknown circumstantial causes resultant from using different water quality datasets (other than site and time) that may lead to possible discrepancies for LSTM models. The scope of this research thus includes different water

quality datasets collected from different sites (water bodies) and/or at different times and other unknown circumstantial causes of discrepancy (this conjecture will be based on the observation of different LSTM models derived from similar water quality datasets).

1.4.2 Objectives of research

1. Model development and optimisation: The development of two water quality prediction LSTM models from two different time-sequential water quality datasets. This is followed by the optimisation of the LSTM models through the application of the genetic algorithm to increase the robustness and efficiency of the LSTM models.

2. Ensemble model creation: The development of a single ensemble water quality prediction model from the two previously developed hybrid GA-optimised LSTM water quality prediction models (now used as base models for the ensemble model), through the application of a weight-based technique.

3. Model tolerance and generalisation: The application of the final ensemble prediction model to datasets other than water quality and/or water-related datasets, such as climate, pollution, and temperature datasets, to assess the performance and thus tolerance of the final ensemble model.

1.5 Contribution of research

This research produced three main contributions. The first contribution was adaptation and optimisation. The adaptation of the LSTM network for water quality prediction from two different temporal-based water quality datasets. This was followed by the subsequent development of the two hybrid GA-optimised LSTM water quality prediction models, for the improvement of the efficiency and robustness of the original LSTM models. The second

contribution was fusion, or rather the combination of the two models. This was achieved through the application of a weight-based approach/technique for the combination of the two hybrid GA-optimised LSTM water quality prediction models, to develop a final more tolerant and robust ensemble model. The third and final contribution of this research was generalisation. Generalisation refers to the exploration of the possible use of the final LSTM-based ensemble prediction model in areas other than water quality, to assert the model's relevance and tolerance in the wider field of LSTM and ensemble prediction models.

1.6 Research layout

The dissertation begins with Chapter One, introducing the research ideation, underlying concepts, and contribution. Chapter Two then provides the necessary background and related techniques to some of the concepts introduced in Chapter One. Chapter Three involves the methods used to develop the two different LSTM models, then the process of the optimisation of both models through the genetic algorithm, and then the procedure using a weight based technique to fuse the optimised models together to create an ensemble model. Chapter Four focuses on the results generated from the different models on both the multivariate and univariate datasets, and critically analyses and discusses the results. Chapter Five provides suggestions for future work and concludes the study.

1.7 Summary of introduction

Recently, existing water resources, especially invaluable inland water resources such as rivers have been placed under pressure due to numerous factors. This necessitates the need for the efficient management of rivers. The efficient management of rivers can greatly benefit from the effective and continuous monitoring of river water quality. In general, conventional methods of

approximating water quality are expensive and time-consuming. The efficiency of water management can be improved if water quality data is analysed, and the water quality of a river can be predicted beforehand.

This study explored the use of LSTM for water quality prediction as LSTM is the most appropriate network for sequential data in which temporal dependency is an implicit feature i.e., time sequential water quality data. An LSTM water quality prediction model developed for a particular river will be specific to that river and might fail to make accurate predictions on other river water quality datasets. To improve the tolerance of LSTM prediction models, this study suggested the development of LSTM water quality prediction base models, optimised by the genetic algorithm, to be combined through a weight based scheme for the development of a final robust and tolerant water quality prediction ensemble model.

The main contributions of this study include the development and optimisation of LSTM models, GA-optimised LSTM model fusion/combination through a weight based scheme for the creation of a final tolerant and robust ensemble model, and lastly the application of the ensemble model to areas other than water quality to assert the model's relevance in the greater field of time-sequential prediction.

Chapter 2

Background and related techniques

2.1 Artificial neural network

Artificial intelligence has many sub-fields, one of which is machine learning. Machine learning algorithms analyse data and find patterns within the data. A specific set of algorithms used in machine learning is ANNs, which mimic the human brain or rather is based on the brain. The brain's ability to analyse, recognise from what has been analysed, learn from what has been recognised, memorise what has been learned, and then generalise patterns from what was learned and memorised is similar to the algorithmic modeling of neural networks [33, 34]. The biological neuron forms the template for the artificial neuron. Much like the biological neuron, the artificial neuron, processes the signals that it receives externally from other neurons and then transmits a signal to all the other neurons connected to it when excited [33].

These artificial neurons form a layered network to make an ANN. The most general neural network consists of three parts, an input layer, an output layer, and a middle layer (which can be one or more hidden layers). Numerical weights connect the neurons in each layer to one another [14]. The process of receiving and transmitting a signal is affected by the numerical weights and the activation function. The neurons receive an incoming signal and then calculate

a net input signal as a function of the numerical weights, which is then sent to the activation function [33,34]. The activation function determines to what level the artificial neuron is excited and how powerful the output signal will be [14].

2.2 Recurrent neural network

One of the subsets of machine learning is Deep Neural Networks (DNNs). A notable difference between DNNs and conventional machine learning is that DNNs can process data directly in its raw form. The approach of traditional machine learning would be to alter the raw data into a feature vector using a feature extractor after which a classifier (learning system) is then able to detect patterns from the feature vector. The entire process of the alteration of raw data can be disregarded when using a DNN, which can automatically learn from unstructured raw data [35].

The back propagation algorithm is responsible for enabling the learning and optimisation of neural networks. Through a recursive application of the chain rule, the back propagation method can calculate the gradient of each neuron in the network [36]. The neural network learns as the gradients of each neuron alter the internal weights of the network. A small gradient value will cause a small weight adjustment. The gradients of each neuron are calculated in terms of the behaviour of the gradients in the previous layer. Hence a big weight adjustment in the previous layer will cause the weight adjustment in the current layer to be even bigger than it was previously. The opposite is true for small gradient values [35].

In general, most neural networks do not model memory well and are incapable of capturing sequences as they are feed forward neural networks. The RNN is a DNN that can utilise sequential input data [35]. Time sequential water quality data, the focus of this study, could easily utilise RNNs for modeling as RNNs can account for the temporal nature of data when most ANNs are unable to [6].

An illustration of how the RNN functions can be found below in Figure 2.1, which shows a single RNN on the left and the same RNN rolled out in layers on the right. From Figure 2.1, it can be seen that the RNN functions much like a typical feed forward neural network, except it has a loop that can pass information pertaining to previous states forward. This looping mechanism enables the RNN to retain the information of previous elements in the sequence that it has processed, whilst working with sequential data. The retention of this information, referred to as the hidden state, allows the information to be passed forward from one step to the next. In this manner, the RNN can model memory [35].

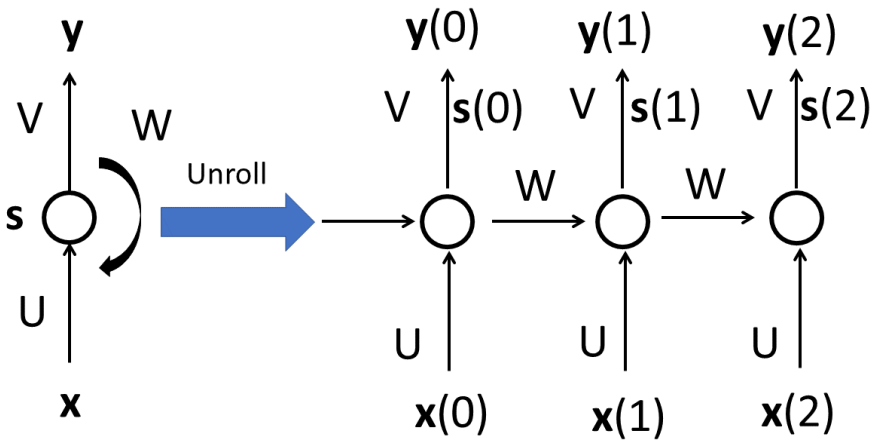

Figure 2.1: Single RNN (left) and the same unrolled RNN where each layer has the same weights, biases and activation functions (right) adapted from [1]

In accordance to Figure 2.1, sequence vectors $\mathbf{x} = [\mathbf{x(0)}, \mathbf{x(1)}, \mathbf{x(2)}]$ pass to the RNN one by one based on the set time step; sequence vectors $\mathbf{y} = [\mathbf{y(0)}, \mathbf{y(1)}, \mathbf{y(2)}]$ represent the expected output and can be describe by the following equations adapted from [1]:

$$\mathbf{s}(t) = \alpha\,(U \cdot \mathbf{x}(t) + \; W \cdot \mathbf{s}(t-1) + \; \mathbf{b}) \tag{2.1}$$

where α is the activation function, $\mathbf{x(t)}$ is the input at time step t and $\mathbf{b}$ is the deviation vector [1], U and W are the weight matrices; U is the weight between the input layer and the hidden layer, W is the weight between the current hidden layer and the hidden layer at the next time step, $\mathbf{s(t)}$ is state of the hidden layer at time step t, $\mathbf{s(t-1)}$ is the state of the hidden layer at the previous time step in (2.1) [19].

$$\mathbf{y(t)} = \alpha\,(V \cdot \mathbf{s(t)} + \mathbf{c}) \tag{2.2}$$

where $\mathbf{y(t)}$ is the expected output of the time step t, $\mathbf{c}$ is a deviation vector [1] and V is a weight matrix- weight between the hidden layer and the output layer in (2.2) [19].

As counter intuitive as it might seem at first, RNNs are trained using back propagation through time. If the RNN is rolled out through time as illustrated in Figure 2.1 and considered at discrete time steps, then each time step in the RNN can be thought of as a layer on its own [34, 35]. The rolled-out individual hidden layers (hidden state) each have the same weights, biases, and activation functions and hence it now seems possible to calculate the gradients of the neurons by applying back propagation through time for the forward computation of the neural network [35].

Simply put, RNNs contain feedback connections that permit the perseverance of past information (data from the beginning of the sequence) and is capable of both non-linear and time sequential predictions and is thus a good choice of network for time sequential water quality data [6].

A major drawback of RNNs is the network's inability to learn longer time dependencies i.e., RNNs are incapable of learning sequences when there are more than a few steps in length. Consequently, as the number of time steps increase, the information from the past disappears- information about the data at the very beginning of the time series begins to vanish, thus

skewing results, commonly referred to as the vanishing gradient and exploding gradients [6]. When training RNNs using back propagation, the gradients either shrink or grow at each time step as the gradients at each layer are mentioned calculated with respect to the gradients of the previous layer as earlier. Thus, the gradients will either vanish or explode over multiple time steps [37]. As the gradients become smaller, the internal weights of the network are barely adjusted, causing the earlier layers of the network to not learn anything. This is referred to as the vanishing gradient, which renders the RNN incapable of learning long-range dependencies across time steps. Consequently, RNNs can learn short term but not long-term dependencies [36].

The difficulties faced when training RNNs have been extensively discussed. Hochreiter and Schmidhuber were the first to successfully address the problem of vanishing gradients in 1997 [38]. Hochreiter and Schmidhuber proposed the architecture of the LSTM to address the complications created by the vanishing gradient. This has since become a popular means of mitigating the problem [35]. Exploding gradients, the phenomena of gradients increasing exponentially in size, is simpler to deal with through the use of a technique known as gradient clipping. Gradient clipping entails shrinking the gradient when norms exceed a particular threshold [34, 35].

2.3 Long short-term memory network

An advanced member of the RNN family is LSTM. LSTM can retain information for several thousands of time steps and can thus scale to much longer sequences than the average RNN, due to its unique architecture [6]. Thus LSTM is the most appropriate network for sequential data in which temporal dependency is an implicit feature; and when the retention of the knowledge of the earlier stages of the sequential data is necessary for the forecasting of future trends [3]; such as when time sequential water quality data over several years is used to make predictions

about water quality.

Mechanisms referred to as gates enable LSTMs to learn long term dependencies. Each LSTM cell has a cell state, considered to be the memory of the network that transfers relative information all the way down the sequence chain [34,39]. The various LSTM gates control which information is added to or removed from the cell state. The gates learn which information is important to retain and which information can be disregarded and thus only relevant information is used to make predictions. The forget, input and output gates are the three different LSTM gates, each with a sigmoid activation [39].

The architecture of the LSTM is illustrated in Figure 2.2 [2] for further clarity, where the internal structure of an LSTM recurrent network cell is shown in accordance with the LSTM equations expressed below.

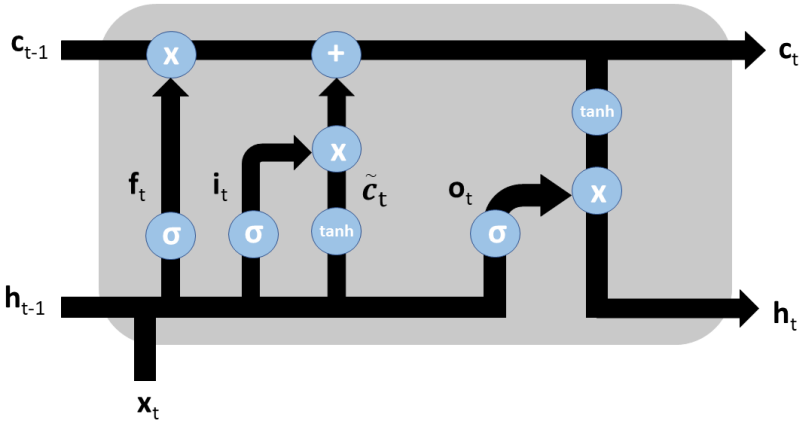

Figure 2.2: Internal structure of LSTM recurrent network cell adapted from [2].

The forget gate controls which information is removed and retained. The current input to the LSTM cell and the previous hidden state is individually multiplied by the weight of the forget gate and summed together and then passed through the sigmoid activation [39] as shown in (2.3). Notably, each gate has a different set of weights. The equations expressed below were adapted from [3]:

$$\mathbf{f}_t = \sigma(\ (\mathsf{W}_f \mathbf{x}_t + \ \mathsf{W}_f \mathbf{h}_{t-1}) + \ \mathbf{b}_f) \tag{2.3}$$

where $\mathbf{f}_t$ is the forget gate, W_f is the weight of the forget gate, $\mathbf{h}_{t-1}$ is the output from previous hidden state, $\mathbf{x}_t$ is the input at current time step and σ is the logistic sigmoid and $\mathbf{b}_f$ is the bias vector.

When the previous hidden state and current input is passed through the sigmoid activation, it determines which values will be updated and hence the input gate is able to update the cell state [39] in (2.4):

$$\mathbf{i}_t = \sigma(\ (\mathsf{W}_i \mathbf{x}_t + \ \mathsf{W}_i \mathbf{h}_{t-1}) + \ \mathbf{b}_i) \tag{2.4}$$

where $\mathbf{i}_t$ is the input gate, W_i is the weight of the input gate, $\mathbf{b}_i$ is the bias vector.

An intermediate cell state is calculated, where the output of the previous hidden state and the current input is passed through a hyperbolic tangent function to regulate the network [34, 39] as expressed in (2.5):

$$\tilde{\mathbf{c}}_t = \tanh(\ (\mathsf{W}_c \mathbf{x}_t + \ \mathsf{W}_c \mathbf{h}_{t-1}) + \ \mathbf{b}_c) \tag{2.5}$$

where $\tilde{\mathbf{c}}_t$ is the intermediate cell state, W_c is the weight of the intermediate cell state, $\mathbf{b}_c$ is the bias vector and $\tanh$ is the hyperbolic tangent function.

The product of the output of the forget gate and the previous cell state is used to calculate the current cell state by adding it (through point-wise addition) to the product of the output of the intermediate cell state and the output of the input gate, in this way the sigmoid output determines what should be retained from the hyperbolic tangent function ($\tanh$) output [39] as expressed in (2.6):

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t \tag{2.6}$$

where $\mathbf{c}_t$ is the current cell state, $\mathbf{c}_{t-1}$ is the memory of the previous state, $\circ$ is the Hadamard product, element-wise product.

Using the current cell state, the output gate then determines what the next hidden state should be. First, the previous hidden state and the current input is passed through sigmoid activation [39] as shown in (2.7):

$$\mathbf{o}_t = \sigma\,[\ (\mathbf{W}_o\mathbf{x}_t + \mathbf{W}_o\mathbf{h}_{t-1}) + \mathbf{b}_o\ ] \tag{2.7}$$

where $\mathbf{o}_t$ is the output gate, $\mathbf{W}_o$ is the weight of the output gate, $\mathbf{b}_o$ is the bias vector.

The hidden state is calculated by passing the current cell state through the hyperbolic tangent (tanh) activation and multiplying it by the output of the output gate in (2.8). This determines the information that the hidden state will carry to the next time step [39].

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh\mathbf{c}_t \tag{2.8}$$

where $\mathbf{h}_t$ is the hidden state.

The sigmoid functions for the three gates modulate the output between zero and one; if the output is zero, then the signal is blocked by the gate. The output is dependent on the input at the current time step t and the output from the previous hidden state. The weights and biases for each gate are learned by the model through minimising the differences between the LSTM outputs and the actual training samples [40].

RNNs can process sequence data to make predictions but are only capable of learning short term dependencies across time steps and not long-term dependencies. LSTMs are capable of

learning long-term dependencies across time steps through the use of the three gates and thus do not have the limitations of the RNN. In general, LSTMs are better utilised for the modeling of longer sequences with long term dependencies [39].

The "aged RNN" might suffer from short term memory, but they do have certain advantages over the newer LSTM. RNNs can train faster and use less computational energy as fewer tensor operations are required to be completed. Hence deciding which network, the LSTM or the RNN, would be better for use would depend on both the data and the purpose for which the data is being modeled. Recently, LSTMs have proven to be more effective than traditional RNNs [36].

2.4 Genetic algorithm

LSTMs also have a couple of drawbacks. LSTMs have several parameters, such as the number of layers, the number of time steps, the number of neurons per layer, to name a few, which are referred to as hyperparameters. Hyperparameters need to be appropriately selected for the network to run successfully. The determination of these hyperparameters has previously depended on the experience of the researcher and the repetition of experiments (trial-and-error) until the optimal hyperparameters were found. This can be rather tedious and even computationally costly if the grid search is used [31]. Despite the obvious importance of hyperparameters for the success of a network, research into and analysis of optimal hyperparameters for LSTM networks have made very little headway [32]. Without the use of optimal hyperparameters, the LSTM network may not produce the most accurate forecasting results. Thus, an optimal LSTM configuration is essential for the correct detection of water quality patterns and time series dynamics in the water quality domain [3]. An automatic method of determining the optimal hyperparameters from a large search space would be greatly advantageous.

Another drawback of the LSTM network is that although a relatively good prediction can be

made by the network, a precise explanation for the solution achieved by the network cannot be provided. A hybrid solution that proposes the integration of the genetic algorithm, a metaheuristic algorithm with the LSTM network has been suggested [32] to alleviate both problems.

Metaheuristic algorithms are known for solving complex optimization problems and for finding a relatively good solution despite time constraints and limited computational capacity. These algorithms are innately stochastic (probabilistic) and their behaviour mimics observed natural behaviour and each algorithm is termed accordingly. GAs are inspired from biology by Darwinian evolution and they can find optimal hyperparameters by simulating a natural evolutionary process. GA is a stochastic and metaheuristic-search-based algorithm recommended for its ability to lessen search complexity, its capacity to learn from any domain where applied and its capability to find optimal (or close to optimal) values for tunable LSTM hyperparameters [3] and hence GA was the primary choice for finding the hyperparameters for water quality forecasting.

The fundamental underlying mechanism of GA involves a population of individuals repeating a process until the system can no longer be improved [3]. Each of these individuals can be considered or viewed as a potential solution to a target problem (finding the optimal hyperparameter). Individuals (solutions) are usually expressed as binary strings [32]. These individuals are generated randomly and the individual that produces the better solution can reproduce [6]. Hence, with every generation, the worst individuals are removed and new individuals are added to the population [3].

The GA process has six distinct stages: initialization, fitness calculation, termination condition check, selection, crossover, and mutation. During initialisation and fitness calculation, individuals in the search space are arbitrarily selected and in accordance with a predetermined fitness function, the fitness of each individual is evaluated [6]. The survival

cycle of the individuals in the population is characteristic of natural evolution where individuals with a higher fitness function produce more offspring than those with a lower fitness function [3]. The definition of a fitness function is a crucial factor for optimisation in a metaheuristic algorithm. Only individuals with excellent performance (the optimal solutions) are preserved for the next generation [6]. Figure 2.3 illustrates the basic process of GA.

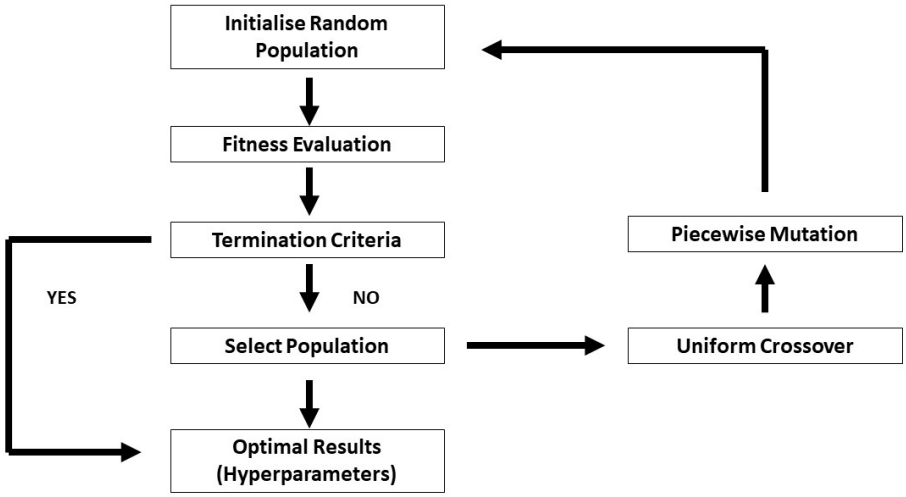

Figure 2.3: Basic process of the Genetic Algorithm (GA) adapted from [3].

As genetic diversity is required for the evolutionary process, genetic operators such as crossover, mutation and selection are used. The crossover and mutation operators allow for the exploration of new areas in the search space. The crossover operator achieves this by creating new individuals, by replacing some of the portions of the two parents' individuals. The mutation operator does this by changing (mutating) some portion of the individual strings [3]. The limitation of the crossover selector is that although it can create new individuals, it can only do so from the information of the previous parent individuals and thus it cannot generate any completely new information. This limitation is effectively overcome by the mutation operator which can generate completely new information by mutating portions of individual strings to generate completely new individuals [6].

The selection operator emphasises the stronger individuals with the reasoning that their offspring will have a higher fitness (fitness function) to survive the next generation; which effectively enables the exploitation of the information that accumulates from the GA search [3]. The standard GA procedure requires that the generated individual goes through the processes of selection, crossover and mutation, whilst calculating its fitness to the model and verifying the termination criteria; the GA procedure is only complete once the termination criteria have been satisfied [6].

To summarise, GAs explore iterative modifications of the population of individuals, and the value of individuals of the population is evaluated using a fitness function. This affords GAs the opportunity of finding optimal or near-optimal hyperparameters in a search space made of many peaks and valleys whilst traditional gradient-based methods can become stuck at the local optima [3].

By using different crossover and mutation operators, which generate a more diverse population, GA is capable of executing a global search and effectively exploring the entire search space. A drawback of the algorithm is accurately defining a suitable solution representation for a given problem. An inappropriate solution representation can lead to an inappropriate choice of mutation and crossover operators [3]. As such, research suggests that GA is the most appropriate choice for combinatorial optimization where a thorough search of all the possible solutions is required and would demand enormous computational power [3].

2.5 Hyperparameters for LSTM

The hyperparameters of the LSTM network need to be optimally determined before the training process. The success of LSTMs depends heavily on the hyperparameters used to train the LSTM network as these hyperparameters determine numerous aspects of the behaviour of the network. Manual and automatic hyperparameter optimisation are two hyperparameter

optimisation methods [3].

The manual hyperparameter optimisation approach involves manually changing and testing each hyperparameter and its effect on the success of the LSTM network. This method is tedious and relies heavily on the knowledge and experience of the researcher. Automatic hyperparameter optimisation approaches range from the modest grid and random search to more complicated model-based algorithms. The grid search approach explores all the possible combinations of hyperparameters until the global optima are found; this is a tedious and exhaustive approach. Random search algorithms are easier to implement and are based on direct search methods. The drawback is that these algorithms converge slowly and hence take a fair amount of time to find the global optima [21].

The values of the hyperparameters can have a big impact not only on the training process but also on model underfitting or overfitting and thus the final accuracy of the network. The number of hyperparameters that can be optimised can vary from a couple of parameters to several parameters, such as the number of layers, the number of units in the hidden layer, time window size, batch size and the type of activation [3].

Research suggests that given the temporal nature of the chosen data (time sequential water quality data), that the time window size and number of LSTM units in hidden layers are two of the most important hyperparameters to be optimised. Especially the time window size, as the appropriate window size should be able to contain the complete dataset background as the LSTM network uses past information during the learning process [32]. If the time window is too large, the model will overfit on the training data, alternatively, if the time window is too small, the model will neglect important information [6]. This research will emphasise the optimisation of these two hyperparameters.

2.6 Ensemble models through weight-based fusion

In the book entitled *Ensemble Methods: Foundations and Algorithms*, a line reads, "Making decisions based on the input of multiple people or experts has been a common practice in human civilization and serves as the foundation of a democratic society," [41]. Thus, it is this inherent human behaviour and approach to problem solving and making decisions, which forms the basis of ensemble learning; "ensemble methodology imitates our second nature to seek several opinions before making a crucial decision," [42]. Asking several different opinions from different sources to make a final decision can be compared to using several different models and combining their predictions to make a final prediction.

An ensemble model is created from a combination of a finite number of neural networks or any other sort of predictor that is trained for the same task, with the intention of improving the overall prediction. In general, the individual networks are trained independently, and their predictions are combined through the use of some mathematical rule [43].

Time series forecasting is particularly challenging in areas such as energy, finance, geology, water quality and information technology where data is dynamic and non-stationary. Forecasting instability is also further increased by the volatility of the available data in these areas. In recent times, there's an understanding that the above-mentioned challenges can be overcome through the use of ensemble forecasting, to enhance the forecasting accuracy instead of using a single individual model [44].

A weight-based approach refers to the use of a weighting scheme for the ranking of features [45] or in this case the ranking of models. The most popular model combination techniques are weighted linear combination techniques. The weights allocated to each individual model are either equal or determined through a mathematical rule. A popular weighted linear combination is the simple average and by extension, the weighted method. The combined forecasts for the

LSTM time series models are evaluated through a linear function of the individual forecasts from each model [46].

These linear combinations can be calculated as follows from [46]:

Let $\mathbf{Y}$ be the actual time series that will be forecasted using n different models, where

$$\mathbf{Y} = [y_1, y_2, \ldots, y_N]^T. \tag{2.9}$$

Let $\hat{\mathbf{Y}}^{(i)}$ be the forecast obtained from the i'th model ($i = 1, 2, \ldots, n$) where [46]:

$$\hat{\mathbf{Y}}^{(i)} = [\hat{y}_1^{(i)}, \hat{y}_2^{(i)}, \ldots, \hat{y}_N^{(i)}]^T. \tag{2.10}$$

A linear combination of these n forecasted series of the original time series can be shown as follows [46]:

$$\hat{\mathbf{Y}}^{(c)} = [\hat{y}_1^{(c)}, \hat{y}_2^{(c)}, \ldots, \hat{y}_N^{(c)}]^T \tag{2.11}$$

which was produced by [46]:

$$\hat{y}_k^{(c)} = f(\hat{y}_k^{(1)}, \hat{y}_k^{(2)}, \ldots, \hat{y}_k^{(n)}) \forall k = 1, 2, \ldots N \tag{2.12}$$

where f is a linear function of the individual forecasts [46]

$$\hat{y}_k^{(i)} (i = 1, 2, \ldots n; k = 1, 2, \ldots N) \tag{2.13}$$

which results in [46]:

$$\hat{y}_k^{(c)} = w_1\hat{y}_k^{(1)} + w_2\hat{y}_k^{(2)} + \ldots + w_n\hat{y}_k^{(n)} = \sum_{i=1}^{n} w_i\hat{y}_k^{(i)} \; \forall k = 1, 2, \ldots N \tag{2.14}$$

where w_i is the weight assigned to the ith forecast. It is assumed that the added weights amount to unity [46].

In the simple average approach, all the models are assigned equal weights [46] with

$$w_i = \frac{1}{n}(i = 1, 2, \ldots, N) \tag{2.15}$$

The simple average approach faults when it comes to skewed distributions and outliers i.e., extreme values; thus weighted combination schemes and techniques have been suggested to overcome this limitation [44].

Although LSTMs are rather popular when it comes to modeling temporal sequences and long-term dependencies i.e. time sequences, little research has been conducted in the applicability of LSTMs in the context of ensemble forecasting and so the best way in which to combine the forecasts of numerous individual LSTM models remains challenging and undecided [44].

In the weighted approach, the weights of the individual models are not equal, and instead greater weight is assigned to the more skilled model and a smaller weight is assigned to the less skilled model. The weight of each model indicates the percentage of trust in each model or rather the expected performance from each model. The underlying principle is that the weighted ensemble model, in which the skilled model is given more importance than the less skilled model, will perform better than an average ensemble model, where the skilled and not as skilled models are given equal importance. The weights of each model are positive small values and the sum of all the weights of each individual model in the ensemble must be equal to one [43].

The weight-based approach will aid in the fusion of the two hybrid GA-LSTM models.

Researchers, Jae Choi and Bumshik Lee believe that a considerable amount of diversity between individual LSTM models is necessary to ensure that the ensemble forecasting approach will be effective [44].

In this study, the diversity in the individual LSTM models exists in terms of the window size and the number of LSTM units in the hidden layers. Model A (later referred to as the Burnett model) will have a different time window size to Model B (later referred to as the Baffle model). Each model will have a different overall number of LSTM units in its hidden layers. The number of LSTM units in the first and second hidden layers will also be different from each other.

Using multiple sequence lengths i.e., different time window sizes for each model enables the development of individual LSTM models with a different number of LSTM units. It is suggested that this will have a complementary effect on the model learning the characteristics and internal dynamics of the time series data being used for prediction. Thus, this is of great advantage for efficient modeling of highly non-linear statistical dependencies. In this manner, it can be hypothesised that an ensemble model that is composed of individual LSTM models of different time window sizes and hence different LSTM units, then has the capability of handling the non-stationary and dynamic nature of real-world time series [44].

Sascha Krstanovic and Heiko Paulheim, suggest in [47], that configuring the individual LSTM models that make up the ensemble, through tuning of LSTM parameters, will increase the general quality of the individual LSTM models and thus of overall quality of the ensemble. Although, it must be ensured that a sufficient amount of diversity exists between the ensemble member models.

The individual LSTM models used in this study were optimised (tuned) using the genetic algorithm to increase the quality of the models used in the final ensemble.

2.7 Selected water bodies

Data were taken from two main water bodies for the development of the LSTM models and the ensemble models, namely the Burnett river and the Baffle river.

The Burnett river was named after the surveyor J.C. Burnett, in 1847, the first explorer of the river. The Burnett River can be found in southeast Queensland, Australia, where it rises on the western slope of the Burnett Range, east of the Eastern Highlands. The river flows the following 435 km course: from southwest to Eidsvold river and turns east at Mundubbera river and then northeast through Gayndah and Bundaberg rivers and eventually enters the Pacific Ocean at Burnett Heads. The three main tributaries to the river from the right are the Auburn river, the Boyne river, and Barambah Creek [48].

The source of the Burnett river is Mount Gaete and the river eventually flows into the Coral Sea of the South Pacific Ocean at Burnett Heads over the 435 km course mentioned above, during which it descends 485 m [4]. The river is part of the South East Queensland and Brigalow Belt bio-regions and the areas around the river are mainly dedicated to growing small crops and sugar cane [49]. A map showing the Burnett river can be found in Figure 2.4 [4]. The Burnett River is shown by a yellow line.

The Baffle Creek, sometimes referred to as the Baffle river as it is in this study, was named by politician and pastoralist, William Henry Walsh in the 1850s when he led an expedition up the river to track a group of raiders. He was unable to track down the group of raiders as their footprints disappeared in the dense bush found along the banks of the creek, leading him to name the creek, Baffle Creek [50].

Baffle creek is located in southeast Queensland, Australia. The source of the river is Arthurs Seat at an elevation of 282 m and the river flows through a 124 km course into the Coral Sea of the South Pacific Ocean, during which it makes a descent of 287 m [51]. The main tributaries to

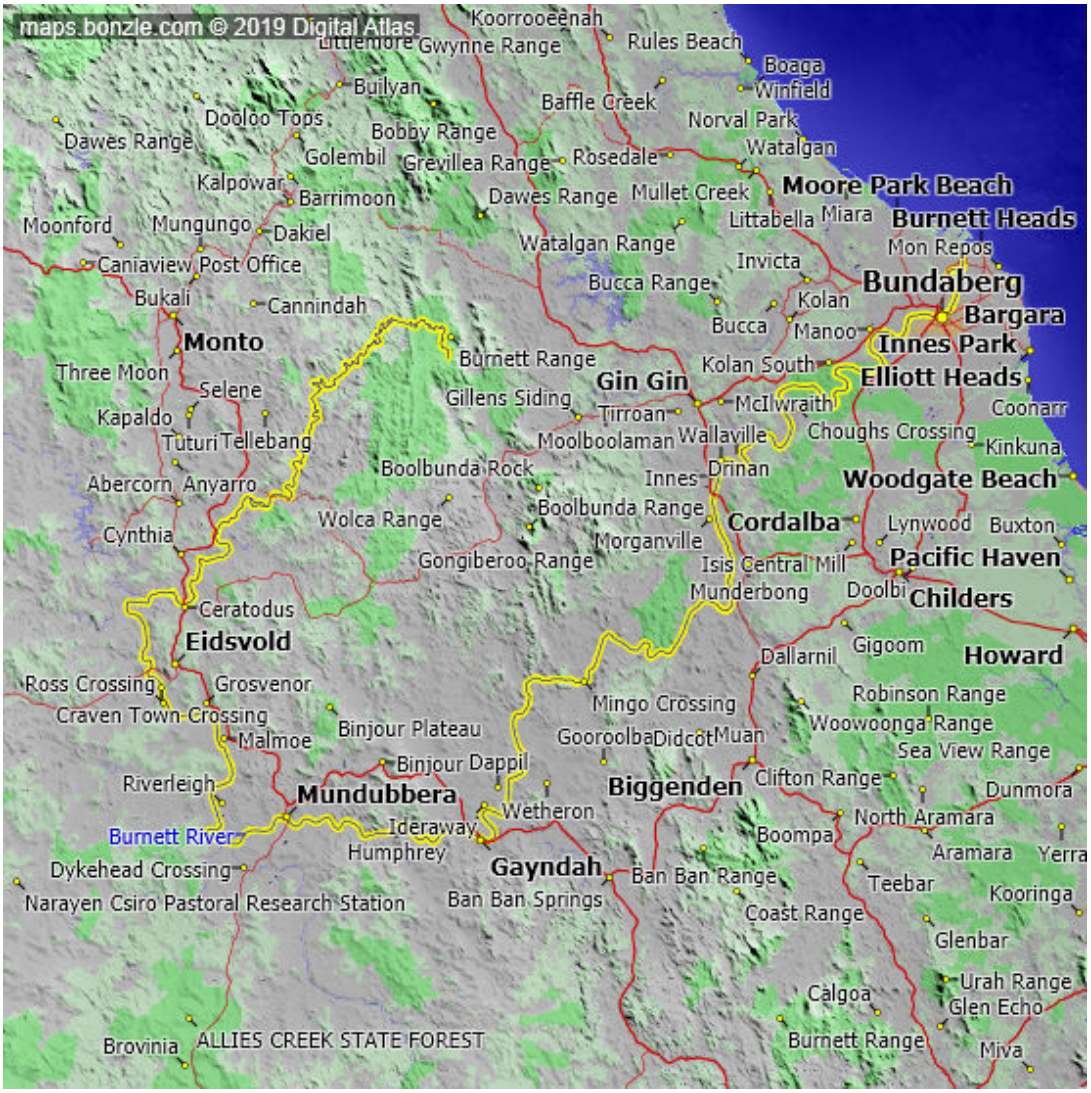

Figure 2.4: Map of the Burnett River [4]

Baffle creek from the right are Scrubby Creek, Granite Creek, Grevillea Creek, and the Three Mile Creek. The main tributaries from the left to Baffle Creek are Euleilah Creek and Island Creek [5]. A map showing the Baffle river can be found in Figure 2.5 [5]. The Baffle River is shown by a yellow line.

2.8 Water quality parameters

The most significant water quality parameter/feature to be predicted is dissolved oxygen. The dissolved oxygen concentration is reflective of the equilibrium between the oxygen producing

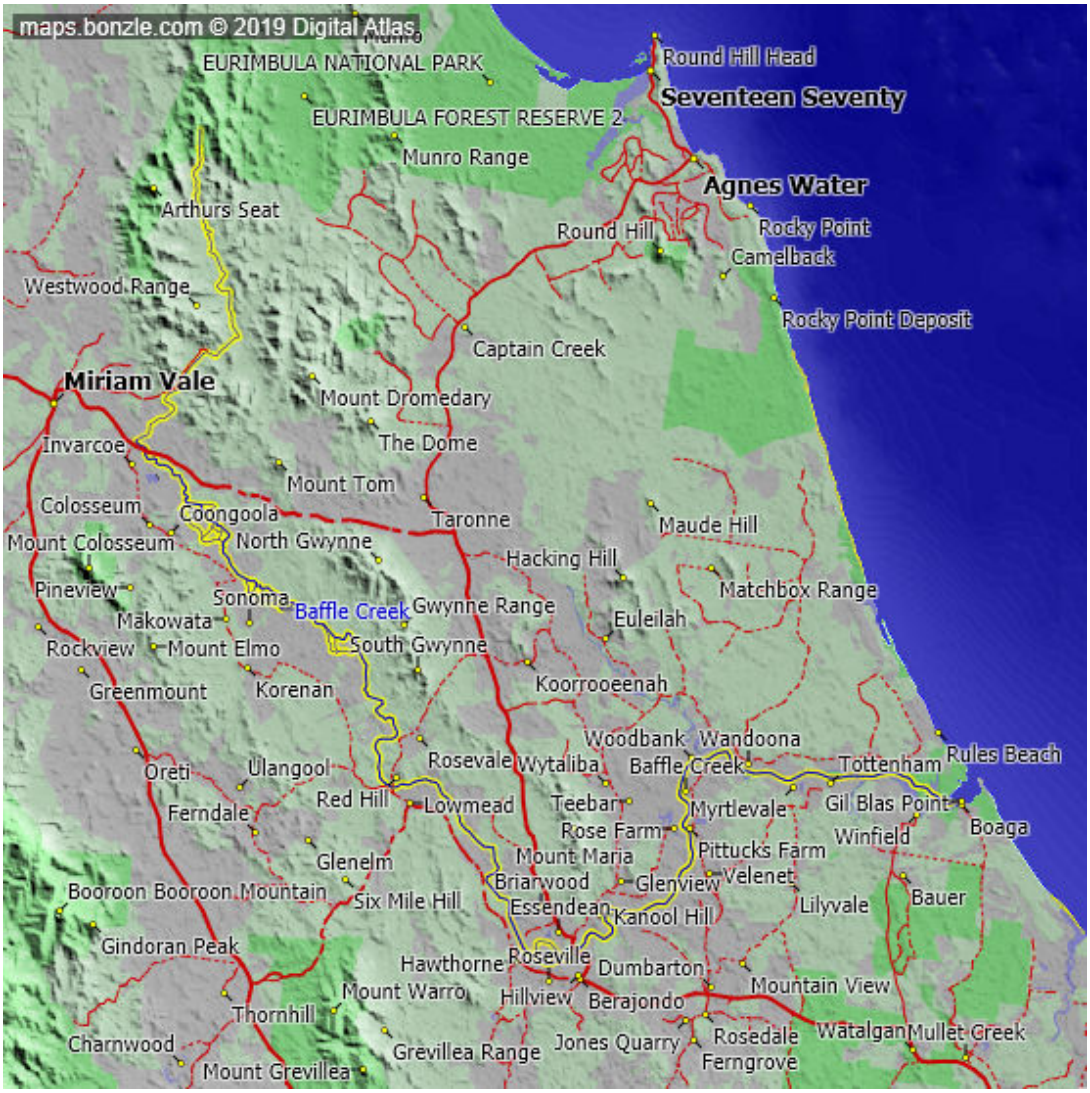

Figure 2.5: Map of the Baffle River [5]

and the oxygen consuming processes in a water body (river) [52]. Dissolved oxygen can also be referred to as the amount of free non-bonded, non-compound oxygen present in water [53]. Numerous factors affect dissolved oxygen levels either directly or indirectly, such as temperature, salinity, oxygen depletion, pH, turbidity and many others; as such the dissolved oxygen level of a water body is often used as the criterion of the health of the water body. Thus, it is of great significance to develop predictive dissolved oxygen models for rivers to ensure that water quality control measures can be optimised [52].

Dissolved oxygen levels also have a significant effect on organisms that exist in the rivers, such

as fish, invertebrates, plants and bacteria, which utilise dissolved oxygen in water just as land organisms utilise oxygen in the atmosphere. Different water dwelling organisms will require varying amounts of dissolved oxygen for numerous reasons and will acquire it through a number of different means. Dissolved oxygen levels fluctuate constantly and if the levels are too high or too low, it can adversely affect water quality and consequently harm aquatic life. Thus the monitoring of and the ability to predict dissolved oxygen levels ahead of time is of paramount importance [53].

Oxygen has limited solubility in water and thus dissolved oxygen concentrations, which are reflective of the equilibrium between oxygen producing and oxygen consuming processes, have hardly ever been recorded over 14 mg/L; the solubility of oxygen in water has been observed to range between 6 and 14 mg/L [54]. It has also been reported that healthy water bodies should have dissolved oxygen concentrations above 6.5-8 mg/L [55].

As the temperature of the water increases, the solubility of gases (including oxygen) will decrease. This implies that warmer rivers, lakes and streams will hold less dissolved oxygen than colder waters [53].

The Burnett and Baffle datasets included dissolved oxygen, water temperature, pH, electrical conductivity and turbidity as water quality parameters.

The temperature of the water is a physical property that describes how hot or cold the water is or more specifically it is the measurement of the average thermal energy of the water [56].

It is very difficult to define typical water temperatures as water temperature has an extreme range and can go from boiling to freezing point. The temperature of a river is dependent on four main factors such as the type and depth of the river, the environment surrounding the river and the season in which the water temperature measurements of the river were recorded [56].

There may not be any typical water temperature ranges that apply to all rivers, yet a specific

river can have a general temperature pattern that it follows annually and any observed unusual temperatures that deviate from the general temperature pattern must be viewed in context. Shallow and broad rivers will be warmer than deeper rivers. In general, rapid and greater temperature fluctuations are observed in rivers and streams than in lakes and oceans. Observed seasonal temperatures across water bodies in the USA show that on average in cold weather, water temperatures can be as low as between 1-4.5°C and as high as between 30-35°C [56].

The temperature of river water can be affected by numerous ambient conditions, such as solar radiation/sunlight, stream confluence, atmospheric heat transfer, turbidity, and man-made influences such as thermal pollution, runoff (water runoff from impervious surfaces), deforestation and impoundments (such as dams which do not directly affect the river water temperature but can alter the natural patterns of water temperature warming and cooling). Water temperature can change the physical and chemical properties of water and also influence several other water quality parameters, such as electrical conductivity in water, salinity, dissolved oxygen (as previously mentioned), other dissolved gas concentrations, compound toxicity, water density and pH [56].

The pH scale spans from 0 to 14 and is a measure of the activity of the hydrogen ion (H+) in water; pH is typically reported as the reciprocal of the logarithm of the hydrogen ion activity. For example, a water body with a pH of 7 has 10^{-7} moles per liter of hydrogen ions, likewise a pH of 9 would imply the water body has 10^{-9} moles per liter of hydrogen ions [57]. The pH of surface water systems i.e., rivers range from 6.5 to 8.5 and is slightly lower for ground water: 6 to 8.5 [57].

Conductivity measures the water body's ability to pass an electrical current. The presence of inorganic dissolved solids in water affects the conductivity of the water; inorganic solids with a negative charge like chloride and inorganic solids with a positive charge like sodium, increase

the conductivity of the water. Organic compounds such as oil and sugar, have low conductivity in water as they are poor conductors of electrical current [58]. In general, the conductivity of rivers ranges from 50 to 1500 µmhos/cm i.e. from 0.05 to 1.5 mS/cm. Inland freshwaters have also reported ranges from 150 to 500 µmhos/cm (0.15 to 0.5 mS/cm) and heavily polluted industrial waters with conductivity as high as 10 000 µmhos/cm (10 mS/cm) [58].

Turbidity is an optical characteristic of water and a measure of the relative clarity of river water. It is commonly referred to as how cloudy or murky the river water is i.e., it is caused by the solid particles that are dissolved or suspended in the water that scatter light, causing the water to have a cloudy or murky appearance [59]. Hence turbidity is a measurement of the amount of light that is scattered by the suspended particles when light is shined through a water sample [60]. Solid particles that are suspended or dissolved in water, causing turbidity include sediment (clay and silt), algae, fine inorganic and organic matter, plankton, and a variety of microscopic organisms [59].

River water with high turbidity levels will lessen the light penetration of the water and hence alter the ecological productivity, habitat quality and aesthetic value of the river. Change in habitat quality can cause harm to fish and aquatic life by degrading spawning beds, lessening food supplies and reducing gill function. A high level of suspended particulate matter also provides a place for other pollutants, such as bacteria and metals, to attach themselves, enabling the potential pollution of the river [60].

As suspended solid particles absorb heat, high levels of turbidity will cause an increase in water temperature. At higher temperatures, the concentration of dissolved oxygen in the river water decreases. Due to the greater number of suspended particles, the amount of sunlight penetrating the river is reduced, this inhibits photosynthesis by river plants, hindering the production of dissolved oxygen, further lowering the dissolved oxygen content of the river [61].

Typical turbidity levels can range from 1 to 1000 NTUs, with smaller values being indicative of lower levels of turbidity and thus a healthier river [61]; on average rivers usually have a range from 10 to 25 NTUs [59].

The observed typical values for each water quality parameter have been summarised in Table 2.1 below.

Table 2.1: Typical values for water quality parameters

Water Quality Parameter	Range
Dissolved oxygen	6 mg/L-14 mg/L [54]
pH	6.5-8.5 [57]
Temperature	1-4.5°C - 30-35°C [56]
Conductivity	0.05 mS/cm - 1.5 mS/cm [58]
Turbidity	1 NTU - 1000 NTUs [61]

2.9 Summary of background and related techniques

DNNs can process data in its raw form and thus differ from conventional machine learning techniques. RNNs are a DNN that is specific to sequential data and accounts for the temporal aspects of data when conventional ANNs cannot, making them an appropriate choice for water quality sequential data. A major drawback of RNNs is their inability to learn longer time dependencies due to the shrinking or exploding of gradients when RNNs are trained through back propagation. Thus, RNNs can learn short term but not long-term dependencies.

LSTM is an advanced member of the RNN family, which can scale to much longer sequences than the average RNN, due to its unique architecture. LSTMs are capable of learning long term dependencies using mechanisms called gates. There are three different gates. The forget, input and output gates regulate information through the LSTM cell. Thus, LSTM is the most

appropriate for sequential temporal data, where the knowledge of the earlier stages of the sequential data is necessary for the forecasting of future trends such as the time sequential water quality data that is used in this study for the prediction of future water quality.

LSTMs have several hyperparameters that need to be appropriately selected for the LSTM to produce the most accurate forecasting results. The optimal LSTM configuration (hyperparameters) is vital for the correct detection of water quality patterns and time series dynamics in the water quality domain. GA (stochastic and metaheuristic-search-based algorithm) can be used to determine the optimal hyperparameters from a large search space. The fundamental underlying mechanism involves a population of individuals (potential solutions to a target problem) repeating a process of six distinct stages: initialization, fitness calculation, termination condition check, selection, crossover, and mutation until the system can no longer be improved. The optimal individuals present at the end of the process represent the optimal hyperparameters for the LSTM.

Given the temporal nature of the chosen data and in accordance with research, the time window size and number of LSTM units in hidden layers are two of the most important hyperparameters to be optimised. This research will emphasise the optimisation of these two hyperparameters. An appropriate window size should be able to contain the complete dataset background. A time window that is too large, will lead to a model which is overfit on the training data and a time window that is too small, will neglect important information.

The most common model combination techniques for the development of ensemble models are weighted linear combination techniques. The weights allocated to each individual model are either equal or determined through a mathematical rule. The combined forecasts for the LSTM time series models are evaluated through a linear function of the individual forecasts from each model.

Water quality data was taken from two main rivers for the development of the LSTM models

and the ensemble model, namely the Burnett and Baffle rivers, situated in southeast Queensland, Australia.

Dissolved oxygen is the most significant water quality feature to be predicted as it is reflective of the equilibrium between the oxygen producing and the oxygen consuming processes in a river. It is also affected by numerous water quality parameters, such as temperature, salinity, turbidity etc. As such the dissolved oxygen level is often used as the criterion of the health of a river. Thus, it is of paramount importance to develop predictive dissolved oxygen models to ensure that water quality control measures can be optimised for rivers.

Chapter 3

A robust and tolerant water quality prediction LSTM based ensemble scheme

This study proposed an optimised LSTM prediction model that could improve the tolerance of LSTM models through the application of a weight-based ensemble technique to develop a robust and tolerant ensemble prediction model.

A general overview of the development of the LSTM based ensemble prediction model is shown in Figure 3.1 for an illustration of this research methodology. Various steps of the method will be described with respect to the diagram in Figure 3.1.

3.1 Water quality datasets and data preparation

Two water quality datasets were used to develop the two LSTM models. The Burnett river water quality dataset, which is also referred to as dataset A, seen in Table 3.1 and the Baffle river water quality dataset which was referred to as dataset B, seen in Table 3.2. The following

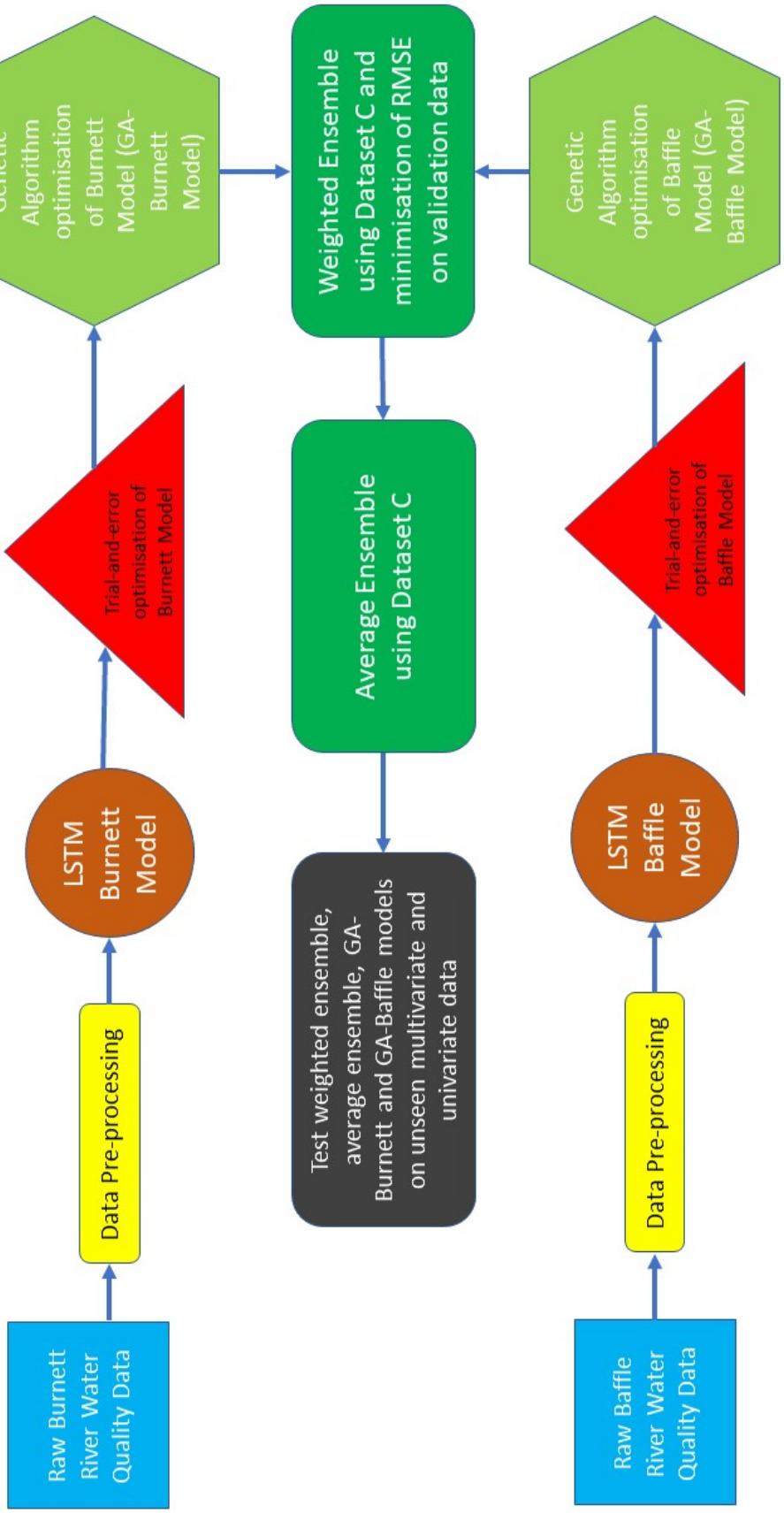

Figure 3.1: Schematic diagram of proposed methodology

tables contain the details of the water quality datasets.

Table 3.1: Burnett river

River	Burnett
Data	Dataset A
Time Period	Beginning of January 2017 to end of December 2019
Total Observations	48 observations per day, 17 520 observations per year, 52 560 observations for 3 years

Table 3.2: Baffle river 2019

River	Baffle
Data	Dataset B
Time Period	Beginning of December 2018 to end of November 2019
Total Observations	144 observations per day, 52 560 observations per year

Historical water quality data for the Burnett river and Baffle river was provided through a publicly available data source from the Ambient Estuarine Water Quality Monitoring Programme of the Queensland Government open data portal [62]. The data included water quality parameters such as water temperature (°C), pH, electrical conductivity (mS/cm), dissolved oxygen (mg/L) and turbidity (NTU).

Prior to using the data, the data was pre-processed and cleaned, this refers to the yellow block in Figure 3.1. The following steps were taken for data pre-processing:

1. The removal of any censored or inconsistent observations. This was done strategically as one could argue that certain inconsistent values should not be removed as they could represent a reality that might be necessary to record. Thus, only values that were highly problematic and could not possibly be realistic were removed. For example, it is highly

probable that a negative pH value is the result of an incorrect recording or faulty machinery as the pH scale only runs from pH of 0 to pH of 14. Note: Typical values for each of the water quality parameters were presented in Table 2.1.

2. Statistical analysis of the data was carried out. The minimum, maximum, mean, standard deviation, and the first, second and third quartile values of each water quality parameter were calculated to gain a greater understanding of the data. It was easier to remove outliers once the upper boundary Q_3 i.e. (75^{th} percentile) and lower boundary Q_1 (25^{th} percentile) of each parameter was found and the interquartile range (IQR) could be used to removed outliers. (3.1) shows how to calculate the IQR [63] and can be found below.

3. Duplicate observations were also removed to prevent repetition from distorting results.

4. The sequential data was then checked for any observations that were not in line with the regular time intervals and observations at inconsistent time intervals were removed. Missing values at regular time intervals were found through interpolation.

$$\mathrm{IQR} = \mathrm{Q}_3 - \mathrm{Q}_1 \tag{3.1}$$

After the data was processed, the relationship between the different water quality parameters was explored. The strength of the relationships between the water quality parameters would determine which parameters would be relevant as inputs for the development of the water quality predictive models.

A commonly used correlation method, Spearman's Correlation, is named after the founder, Charles Spearman. Correlation is the measure of association between two parameters and is often a value between -1 and 1. When parameters share a positive association, then as one parameter increases so does the other parameter. When they share a negative association,

then as one parameter increases, the other parameter decreases. With a neutral correlation, one which is close to the value of zero and indicates that there is no relationship between the parameters. When a strong correlation exists between two parameters, it is evidence of the strong relationship between two parameters and implies that they can be used to build a model [64] [34].

The Spearman's coefficients in terms of each parameter/variable in relation to dissolved oxygen (target feature) for both the Burnett and Baffle datasets are shown below in Table 3.3:

Table 3.3: Spearman's correlation coefficients

Parameters	Burnett Data	Baffle Data
Temperature	-0.437	-0.419
pH	0.361	-0.055
Dissolved oxygen	1.00	1.00
Electrical conductivity	-0.080	-0.075
Turbidity	-0.129	-0.100

After the correlation coefficients were found and the relationship between the parameters was established, it was evident which parameters could be used as inputs for the models. The values in Table 3.3 show that dissolved oxygen shares the strongest correlation with temperature, albeit a negative one, after the absolute correlation that it shares with itself. This strong negative correlation with dissolved oxygen is present for both the Burnett and Baffle rivers. This strong negative correlation implies that as the water temperature increases, the level of dissolved oxygen will decrease. This is in accordance with the research discussed in Chapter 2: Background and related techniques, 2.8: Water Quality Parameters. The correlation between dissolved oxygen and electrical conductivity and turbidity is small enough to be considered insignificant for both the Burnett and Baffle rivers. The relationship between

pH and dissolved oxygen is rather difficult to gauge from these values. pH shares a moderate positive correlation with dissolved oxygen in the Burnett dataset and a weak negative correlation in the Baffle dataset.

Whilst the behaviour of all the other parameters seems to be consistent across both datasets, pH is the parameter that behaves differently in the two datasets. This inconsistent behaviour indicates that correlation does not always equate to causation. If the moderate positive correlation between dissolved oxygen and pH that exists in the Burnett dataset was not compared to the significantly different correlation that exists between the same water quality parameters in the Baffle dataset, it would have been assumed that a causal relationship exists between the two water quality parameters, which is not the case.

Causation and correlation should not be confused, which can often be the case with time series data [65]. A causal relationship exists between two parameters when three requirements are met: there is an association between the parameters, there is an appropriate time order and the other parameters have been eliminated [66]. Usually, the elimination of other parameters as stated in the third criterion can only occur under experimental conditions. However, that is not the only way to establish causality in accordance with the third criterion. Another method is to simply check if the causal stimulus works i.e., to see if the manipulation of one parameter causes a change in the other parameter and there will be no need to eliminate the other parameters under experimental conditions. If the behaviour of one parameter affects the other parameter sufficiently, then a causal relationship can be established [67].

The behaviour of the two water quality parameters, dissolved oxygen and pH, is different in the Burnett and Baffle dataset and hence any assumed causation can be disregarded. Also, at the time of this study, there was no documented causal relationship between pH and dissolved oxygen and so this study saw it fit to overlook the positive correlation present in the Burnett river.

Hence dissolved oxygen and temperature were used as the input parameters for the models for the prediction of dissolved oxygen. The other water quality parameters were dropped from the dataset and a new dataset was created for the Burnett river, which only contained dissolved oxygen and water temperature. The same process was applied to the Baffle dataset.

3.2 Development and optimisation of LSTM Burnett model and LSTM Baffle model

3.2.1 Development of a multivariate multi-step stacked LSTM model

Two water quality predictive LSTM models were developed, one for the Burnett river and the other for the Baffle river, this refers to the brown circle in Figure 3.1. Both models were developed in Keras, using Tensorflow backend. Keras is a publicly available, free open source Python library for developing and evaluating deep learning models, which was started as a project by Francois Chollet in 2015. The use of Tensorflow (deep learning mathematical library) backend (i.e., used to perform computation) allowed for models to be defined through the use of efficient numerical libraries with a clean user friendly interface. The models were developed in Google Colab (Google Colaboratory). Google Colab is a hosted Jupyter notebook service that allows users to write and execute python code through the browser, whilst providing free access to computing resources, such as GPUs [68].

The Burnett model, also referred to as Model A was developed as a multivariate multi-step stacked LSTM model. The two parameters, dissolved oxygen and water temperature were used as features to predict the target parameter, dissolved oxygen. Models in Keras are defined as a sequence of layers. The input layer has two features, for the two input parameters. The model was developed to have two hidden layers, the first hidden layer with a larger number of LSTM units than the second hidden layer. The output layer is a dense layer

that connects the whole model and outputs dissolved oxygen values 24 time steps ahead for a multi-step prediction. The optimiser used was Root Mean Square Propagation (RMSprop) and the activation function was Rectified Linear Unit (ReLU).

The Baffle model, also referred to as Model B was developed to have a very similar architecture to the Burnett model. The Baffle model has two features in the input layer for the two parameters, dissolved oxygen and temperature. This model also has two hidden layers, with the number of LSTM units in the first layer being greater than the number of units in the second layer; and a dense output layer that predicts dissolved oxygen values 24 time steps ahead. Again, RMSprop was used as an optimiser and ReLU as the activation function.

The only difference between the two models is that the overall number of LSTM units in the Baffle model was greater than the number of LSTM units in the Burnett model. This was largely due to the nature of the different datasets.

Both the Baffle dataset and the Burnett dataset have the same number of total observations, 52 560 observations. However, the Burnett dataset, has this number of observations spread out over three years and the Baffle dataset has the same number of observations spread out over just one single year. As the data in the Burnett dataset is spread out over three years, it is easier for the LSTM network to pick up variations in the data and learn trends over a three-year period, hence an LSTM model with an overall smaller architecture was used for the Burnett model. The Baffle data is more dense and thus has lots of close clusters of points that are similar and have fewer variations between points, hence learning a trend for the purpose of prediction could be more difficult. Thus, an overall bigger LSTM architecture was used to accommodate the Baffle dataset.

The activation function employed in both models was the Rectified Linear Unit (ReLU) activation function. The ReLU can be regarded as a piece-wise linear function or a hinge function, which implies that the function can behave differently based on the input value; for

example, the function can be linear for some of the input but nonlinear for the rest depending on whether the input is positive or negative. When values are positive i.e., greater than zero, the ReLU function is linear, thus retaining the properties of linearity when training neural networks with back propagation. For values that are negative i.e. less than zero, the ReLU function adopts the behaviour of a nonlinear function and outputs a zero [34, 39].

Of late, ReLU has become a common, trusted default activation function for developing most types of neural networks. The advantages of the ReLU function were highlighted by Xavier Glorot et.al [69] in the paper, *Deep Sparse Rectifier Neural Networks*. The advantages of the ReLU activation include representational sparsity, linear behaviour and computational simplicity. As the ReLU function does not require the computation of an exponential function, such as other popular activation functions like the sigmoid and hyperbolic tangent functions and thus it is easier to implement ReLU at a lower computational cost [34, 69].

Whenever an activation function is capable of outputting true zero values from negative inputs, it is referred to as representational sparsity. The ReLU activation function, can produce a true zero value from negative inputs, thus allowing hidden layers in the neural network to be activated and hence to contain at least one true zero value. This gives the ReLU activation function an advantage over the sigmoid and hyperbolic tangent functions which approximates a value close to zero but not a true zero value [39].

Usually, neural networks that are linear or almost linear are easier to optimise than other types of neural networks. In general, the ReLU activation function appears and behaves like a linear activation function [39], causing the gradients to remain proportional to the node activations and thus neural networks trained with the ReLU activation function do not have to deal with the issue of vanishing gradients [34, 69].

The optimiser of choice for both models was the Root Mean Square Propagation (RMSprop) optimiser. RMSprop is notorious for being an unpublished optimiser which was suggested in a

lecture by Geoff Hinton. RMSprop was developed to address the problem of drastically diminishing learning rates that were observed when using the Adagrad optimiser and thus RMSprop is an adaptive learning rate method. [70].

Before the pre-processed data was fed to the model, the data was split into training data, validation data and testing data, in a 50%, 20%, 30% percent ratio respectively. The training dataset was the biggest dataset, to ensure that the LSTM network had enough samples to learn from.

As the water quality parameters or features had different ranges and scales, the features were normalised before that data was fed to the models. Although it is possible for a model to converge without the normalisation of data, it is sufficiently more difficult to train such a model and the model is then heavily dependent on the input unit choice. The mean and standard deviation of the data were the statistics used to normalise the data in both the training and validation datasets, in accordance to (3.2) [47]:

$$x' = \frac{x - \overline{x}}{\sigma_x} \tag{3.2}$$

where x is the original value of each entry in the dataset, x' is the normalised value of each entry in the dataset, $\overline{x}$ is the mean of the data and σ_x is the standard deviation of the data.

The test dataset was not normalised. In this way, the not normalised test data assesses the generality of the model, along with the pre-processing of the data, which is an advantage when developing a predictive algorithm. Often when the entire dataset (including the test dataset) is normalised then the testing process will simply validate this model and not be of much assistance in model generality [47]. It should be noted that the predictions from the models were scaled back to the original scale, by reversing the normalisation in (3.2), before any of the performance metrics were calculated and evaluated.

Thus, the Burnett model and Baffle model were then trained with their normalised training datasets, respectively for various epochs. An epoch is one learning iteration, which consists of two phases: a feedforward pass and back propagation. The feedforward pass calculates the output value of the neural network, during each training pattern. Back propagation occurs when the error value is propagated to the input layer from the output layer. By this process, the weights of the neural network are then adjusted as a function of the back propagated error signal [33].

3.2.2 Trial-and-error optimisation of a LSTM Model

Through trial-and-error, the two multivariate multi-step stacked LSTM models were initially optimised in terms of time window size and the number of LSTM units in the two hidden layers, this is illustrated by the orange triangles in Figure 3.1.

For the Burnett model, arbitrary values for the window size and the number of units in the two hidden layers were chosen. Thus, three arbitrary values were chosen, one for the window size, one for the number of LSTM units in the first hidden layer and another for the number of LSTM units in the second hidden layer. The number of units in the first hidden layer was chosen as being greater than the number of units in the second hidden layer. The model was trained and the RMSE value was calculated to decipher the performance of the model. The smaller the RMSE, the better the model is at predicting dissolved oxygen values from historical dissolved oxygen and temperature data.

In a second run, all three of the values were arbitrarily lowered and the model was trained, a prediction was made and the RMSE was calculated to evaluate the model. Then all three of the values were arbitrarily increased and the process was repeated. The RMSE value was noted. Arbitrary combinations of the values of the number of LSTM units in the hidden layers were chosen and tested keeping the window size constant. The model was trained at these

values and each time the RMSE was calculated to evaluate the model after a prediction was made. The number of units in the hidden layers was kept constant, whilst the window size was changed. Again, the model was trained at these different values for the window size and the RMSE was calculated after a prediction was made.

Patterns in the process were observed and directed the choice of arbitrary values for each run. If smaller values for the number of units in the hidden layers were observed to achieve a lower RMSE, then consistently lower values were chosen, until the RMSE values started increasing at lower values for the number of LSTM units in the hidden layers.

The same process was repeated for finding the window size and then for finding the combination of the window size and number of units in the hidden layers that gave the lowest RMSE value.

For the Burnett model, the first hidden layer was optimised to have 32 LSTM units and the second hidden layer to have 16 LSTM units. A large window size of 100 time steps was found to be optimal.

The Baffle model was also optimised through trial-and-error. The first hidden layer was optimised to have 64 LSTM units and the second hidden layer to have 32 units. A window size of 150 time steps was found to be optimal. As was expected, the Baffle LSTM model had a bigger network architecture than the Burnett LSTM model, with almost double the amount of LSTM units and a bigger window size.

The 'optimal' values found for the Burnett and Baffle LSTM models, in terms of window size and the number of units in the two hidden layers (given above), through the trial-and-error process, will be used to initialise the optimisation of both these models by the genetic algorithm (GA).

3.3 Hybrid genetic algorithm optimised LSTM Burnett and Baffle models

A hybrid GA-optimised LSTM water quality prediction model for the Burnett water quality dataset, GA-Burnett LSTM model and the Baffle water quality dataset, GA-Baffle LSTM model was developed in accordance to Figure 3.2 [6] to find the optimal window size and the optimal number of units in the first and second hidden layers for each model. This part of the method is illustrated by the light green hexagon as step five in Figure 3.1.

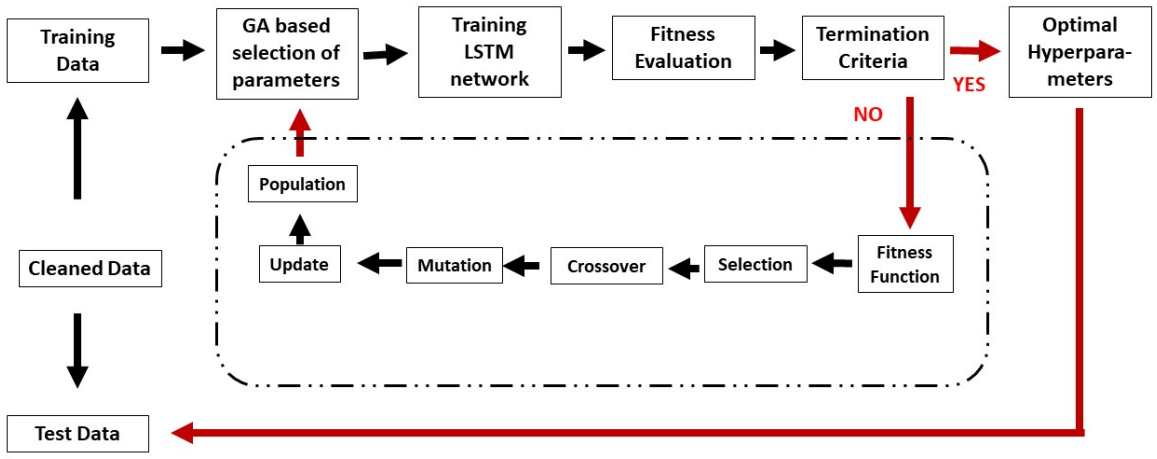

Figure 3.2: Illustration of the optimisation of LSTM by GA adapted from [6]

In the first stage (detailed in the previous section), the basic form of the Burnett LSTM model and the Baffle LSTM model was designed and 'optimised' through trial-and-error. The values obtained for each model in terms of window size and the number of units in the two hidden layers are used as the initial values during the GA optimisation process (referred to below).

In the second stage, the Burnett and Baffle LSTM models were trained and evaluated using

Keras and Tensorflow and the optimisation occurred through the application of the genetic algorithm from the python package called "Distributed Evolutionary Algorithms in Python" also referred to as DEAP [71].

The optimisation of the Burnett model and the Baffle model was done separately.

During the optimisation process, the number of units in the hidden layers and the size of the time windows were used to evaluate the fitness of the genetic algorithm. The values of these variables were initialised at the values that were found through the trial-and-error process, prior to the exploration of the search space by the genetic operators. As the genetic operators explore the search space, the population becomes composed of possible solutions i.e., possible optimal values for the window size and the number of units. Binary bits encode the chromosomes which represent the number of the LSTM units and the window size. The selection and recombination operators, look for the best solution within the population, which is made up of all the possible solutions. Each solution is evaluated in accordance with the predefined fitness function and the chromosomes with the best performance are chosen for reproduction [6].

Defining the fitness function is thus of great significance. This study chose to use the RMSE value to evaluate the fitness of each chromosome. The chromosome which provided the combination of the window size and number of units which resulted in the lowest RMSE value was regarded as the optimal or near-optimal solution. As the termination criteria were satisfied by the optimal solution, the solution can now be used in the LSTM prediction model to make a prediction. In the case of the optimal solution not being found and the termination criteria not being satisfied, the cycle of selection, crossover and mutation continues until the optimal solution is found [6].

To obtain a solution, particular genetic parameters, namely population size, number of generations, crossover and mutation rate were defined. The parameters used for the genetic

algorithm are shown in Table 3.4, the values were chosen with reference to similar studies [6] of LSTM optimisation by GA:

Table 3.4: Genetic algorithm parameters

Population Size	70
Number of Generations	10
Crossover Rate	0.7
Mutation Rate	0.15

These values were defined using the DEAP package. This research used a population size of 70, a mutation rate of 0.15, a crossover rate of 0.7 and the number of generations was set at 10 [6]. DEAP was used to define other parameters pertaining to the type of crossover, mutation and selection used, such as ordered crossover, shuffle mutation and roulette wheel selection. These parameters were chosen as they gave the best possible results from the other options that were available. To fulfil the aim of this optimisation, the RMSE value has to be minimised, this is achieved by defining the Fitness Maximum using DEAP as -1.0 and not 1.0, which would maximise accuracy [71].

As stated previously, the solution was illustrated through binary representation. This resulted in a binary solution of the length 14. The first 6 digits were for the window size, the next four digits were for the number of LSTM units in the first hidden layer and the last four digits were for the number of LSTM units in the second hidden layer.

The pseudo-code of the GA-Burnett and GA-Baffle models is shown in Algorithm 1.

Algorithm 1 GA-optimised LSTM model

Split the data into training (30%), validation (50%) and test (20%) data;
The LSTM is evaluated on the validation data.
Initialise the population size (70), the number of generations (10) and the length of the chromosome at 14 (binary style);
Set RMSE as the the fitness function;
if window size == 0
or number of units A == 0 ▷ number of units in first hidden layer
or number of units B == 0 ▷ number of units in second hidden layer **then**
    Probability of 0.15 for mutation of new chromosomes;
    Probability of 0.7 for crossover of chromosomes;
    Evaluation of the freshly generated chromosome through use of the fitness function;
    return RMSE of 100 ▷ Stopping condition as minimisation of RMSE is the aim
End If
Choose the best individual chromosome which represents the optimal time window, the optimal number of units in the first hidden layer and the optimal number of units in the second layer;
Apply the optimal window size and optimal number of units in the two hidden layers in the LSTM and make a prediction on the unseen test data;

3.3.1 The optimised LSTM hyperparameters

The genetic algorithm was used to optimise two hyperparameters, the window size and the number of LSTM units in the two hidden layers of each model.

For the Burnett model, the optimal window size was found to be 57 time steps; the optimal units in the first hidden layer was found to be 10 units and 8 units in the second hidden layer. For the Baffle model, the optimal window size was found to be 63 time steps; with 12 units in the first hidden layer and 10 units in the second hidden layer. As expected, the overall architecture of the Baffle model is bigger than that of the Burnett model. Table 3.5 shows the difference in architecture and window size of the original LSTM models and their GA-optimised versions.

Table 3.5: Hyperparameter values for the LSTM models

	Burnett Model	GA-Burnett Model	Baffle Model	GA-Baffle Model
Window Size	100	57	150	63
Units 1st layer	32	10	64	12
Units 2nd layer	16	8	32	10

3.4 Ensemble LSTM-based model through a weight-based technique

The weight based ensemble approach was be used to incorporate the results from the two models by allocating a weight to the prediction of each model. The weight was indicative of each individual model's contribution to the new ensemble model. The ensemble model was created using dataset C. Dataset C is a Baffle river water quality dataset from 2015. This is different to the Baffle river dataset that was used to develop the Baffle model, which was taken from in 2019. The details of dataset C (Baffle 2015) are shown below.

Table 3.6: Baffle river 2015

River	Baffle
Data	Dataset C
Time Period	Beginning of January 2015 to end of December 2015
Total Observations	48 observations per day, total of 17 520 observations for the year

The process of developing the ensemble models was inspired by [72] and is detailed below and refers to the dark green rectangles in Figure 3.1.

Data preparation and validation dataset

A new water quality dataset, one which was not used to develop the two GA-optimised LSTM models i.e., not the Burnett and/or the Baffle 2019 dataset was used to develop the ensemble models. The Baffle 2015 dataset (dataset C), a river water quality dataset for the year 2015, taken from the beginning of January 2015 to the end of December 2015 was used to develop the ensemble models. The Baffle 2015 dataset was divided differently from the two datasets that were used to develop the two LSTM models. The training dataset was 30% of the data, the validation dataset was 50% of the data and the testing dataset was 20% of the data. The validation dataset was the largest dataset now.

The optimal weight for each model in the ensemble model was not explicitly calculated, instead the holdout validation dataset, unseen by the individual models during the training process, was used to estimate the weight of each model in the ensemble model. The validation dataset was used instead of the training set to prevent the model over-fitting. Hence the need for a large representative validation dataset.

Model predictions and comparison

The two individual LSTM models, GA-Burnett model and GA-Baffle model were defined according to their found hyperparameters and were then trained using the training dataset. After the individual GA-optimised LSTM models were trained, predictions were made by each model using the validation dataset as the test dataset. The predictions made by each model was evaluated through comparison to the actual values (the actual dissolved oxygen values) by calculating the RMSE value. The lower the RMSE value, the better the overall performance of the model. This is necessary as the performance of the individual models can now be compared to each other and that of the ensemble models.

Weight coefficients and weight grid search

An exhaustive, yet simple grid search was employed to find the optimal weight values i.e., the weight coefficients of the prediction of each model, which represented the size of the contribution that each model made to the final ensemble prediction. A coarse grid of weights ranging from 0.0 to 1.0 with increasing increments of 0.1 was defined.

The weight values represented in the grid were multiplied by predictions made by the GA-Burnett and GA-Baffle models until the best combination of weights was found. The best possible weight combination was found through a function that minimizes the RMSE value i.e., finds the lowest RMSE through the combination of optimal weight values from the grid search. A necessary restriction of the grid search is that the sum of the two possible weight values must be equal to one. The validation dataset, which was unseen by the individual models, which were trained using the training dataset, was used to simultaneously perform the grid search of possible weights for the optimal weight combination of the two models, whilst each individual model made a prediction, with the minimisation of the RMSE value as the final goal.

Once the optimal weight combination was found, the final weighted ensemble model was evaluated on the test dataset.

Weighted ensemble model and average ensemble model

For a model with continuous valued output, the prediction of the average ensemble model can be calculated as the average of the individual member predictions. With regards to the method mentioned above, the weight values have to be equal to 0.5 for each model as there are only two models i.e 1/2 is 0.5, to find the RMSE value of the average ensemble model.

A limitation of this method is that the individual models contribute equally to ensemble model prediction. It could be possible that one model may be more skilled than the other model and

hence should have a greater contribution to the final ensemble model- this is possible through the weighted ensemble model. An average ensemble model was developed to be used as a point of comparison to the weighted ensemble model. It is expected that a well-configured weighted ensemble model will outperform an average ensemble model.

The pseudo-code for the ensemble model is shown in Algorithm 2.

Algorithm 2 Ensemble Model

1: Split the data into training (30%), validation (50%) and test (20%) data; ▷ validation data must be the biggest dataset
2: The individual model weight combinations are evaluated on the validation data.
3: Fit the GA-Burnett model on training data;
4: Fit the GA-Baffle model on training data;
5: Make a prediction with each model using the validation data.
6: Create a grid of weights from 0.0 to 1.0 with increments of 0.1.
7: Define a function which multiplies a weight from the grid with the prediction from each model and then sums the product of the associated model weight and prediction for each model to make a final prediction:
8: $\mathsf{pred}_{\mathsf{ensemble}} = \mathsf{w}_a * \mathsf{pred}_a + \mathsf{w}_b * \mathsf{pred}_b$ ▷ a, b = GA-Burnett, GA-Baffle
9: Define a function to evaluate the ensemble prediction by calculating the RMSE score with the ensemble prediction and the true values of the validation data;
10: **while** $\mathsf{w}_a + \mathsf{w}_b = 1$
11: and $\mathsf{w}_a, \mathsf{w}_b > 0$ **do**
12: minimise RMSE score for the weighted ensemble
13: End while
14: The best weight for each model will form the weight combination that gives the lowest RMSE on the validation data
15: The best weight combination for the weighted ensemble can be used to make a prediction on the unseen test data
16: For the average ensemble, w_a and w_b are equal i.e. 0.5
17: The average ensemble is also used to make a prediction on the unseen test data

An illustration of the development of the weighted and average ensemble model from the two GA-optimised LSTM models, GA-Burnett model and GA-Baffle model is shown in Figure 3.3.

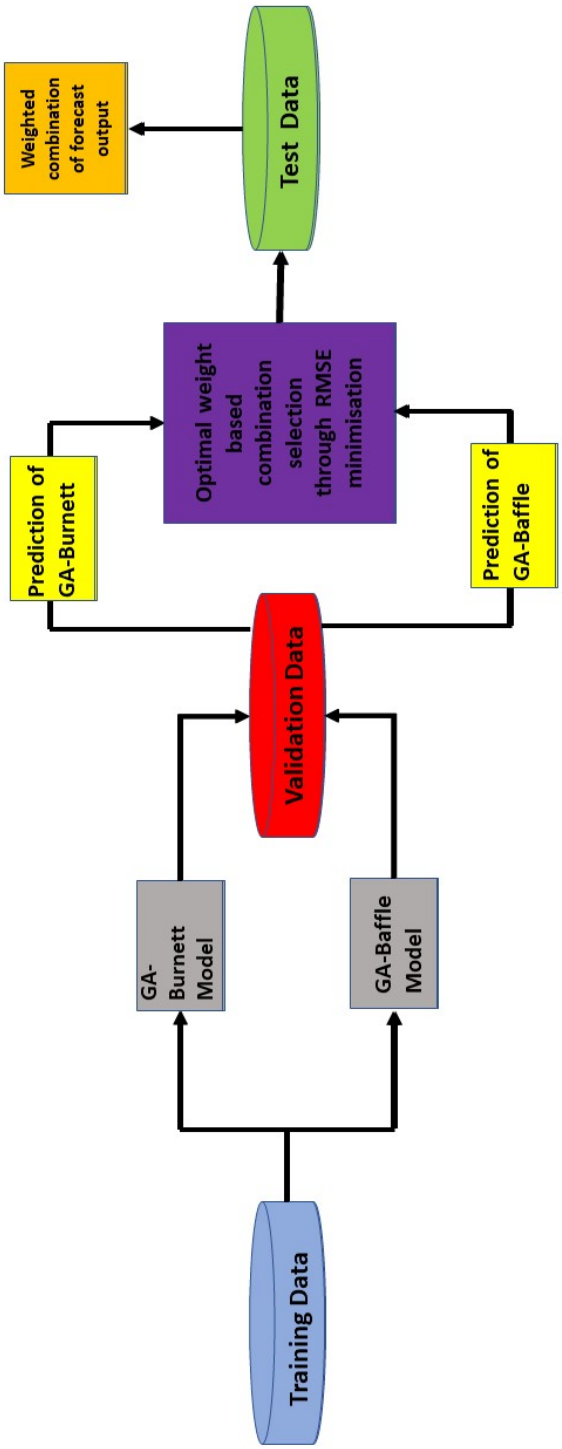

Figure 3.3: Diagram of the development of the ensemble model

3.5 Performance metrics for models

The ensemble model was assessed by computing the Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Error Percentage (MAPE) and the Root Mean Squared Error (RMSE). These are common performance metrics used for continuous output ensemble models. The MSE is a measure of the average squared difference between the predicted values and the actual values [73]. The smaller the MSE, the more accurate the model is at predicting future values based on historical data.

The MSE was calculated in accordance to (3.3) [6]:

$$\mathsf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.3}$$

Where n is the number of samples, y_i is the desired output and $\hat{y}_i$ is the predicted output value of the model's i^{th} observation.

The MAE (3.4), MAPE (3.5) and RMSE (3.6) were evaluated in accordance to the following equations [6]:

$$\mathsf{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{3.4}$$

The MAE is the average of the absolute difference between the predicted values and the actual values [74]. The smaller the MAE value, the closer the predicted values are to the actual values, the better the performance of the model.

$$\mathsf{MAPE} = \frac{\sum_{i=1}^{n}|(y_i - \hat{y}_i)/y_i|}{n} \times 100 \tag{3.5}$$

The MAPE is the average of the absolute percentage of the difference between the estimated values and the actual values. During computation, the sum of the percentage errors can be

found without regarding the sign of errors. The MAPE value provides a measure of the error in terms of percentage, which is easy to understand. Simply put, the smaller the MAPE, the better the forecast [75].

$$\mathsf{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.6}$$

The RMSE value refers to the quadratic average of the differences between the predicted values and the actual values, put in another way, it is the square root of the average of the squared differences between the predicted and actual values. RMSE is commonly used to measure the accuracy of the predictive capacity of a model on a particular dataset and not between datasets as it is scale dependent [76]. A lower RMSE value is better than a higher one and a RMSE value of zero (which is hardly ever achieved) would be indicative of a perfect fit between model and data. The RMSE is sensitive to outliers, as outliers will have the biggest impact on the difference between the predicted and actual values, which will disproportionately affect the RMSE value [77].

R^2 score refers to the co-efficient of determination and provides a measure of how differences in one variable can be explained by a difference in another variable i.e. the variation in y (dependent variable) explained by x (independent variable). [78] It is indicative of how well the model fits to the data and thus a measure of how well the unseen data is likely to be predicted by the model [74]. The R^2 score can be calculated using (3.7) [74], where $\overline{y}$ is the average of the desired outcomes i.e. actual values:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \overline{y})} \tag{3.7}$$

The Median Absolute Error gives a measure of the error by finding the median of all the absolute differences between the predicted values and the desired outcomes (actual values). It

can be found according to (3.8) [74]:

$$\mathsf{MedAE} = \mathsf{Median}(|y_1 - \hat{y}_1|, \ldots, |y_n - \hat{y}_n|) \tag{3.8}$$

The maximum error calculates the maximum residual error by capturing the worst error between the predicted value and the desired outcome. The maximum error would be zero for a perfectly fitted model i.e. showing the extent of error that the model had when fitted and can be found with (3.9) [74]:

$$\mathsf{MaxError} = \mathsf{max}(|y_i - \hat{y}_i|) \tag{3.9}$$

The explained variance score is a measure of how well the model accounts for the dispersion (variation) of a particular dataset. The closer the score is to one, the better the performance of the model, lower values are indicative of poorer performance. The explained variance is calculated through (3.10) [74], where Var is variance, the square of the standard deviation:

$$\mathsf{explainedvariance} = 1 - \frac{\mathsf{Var}\{y - \hat{y}\}}{\mathsf{Var}\{y\}} \tag{3.10}$$

3.6 Assessment of models on unseen data

The four models, the GA-Burnett, GA-Baffle, average ensemble and weighted ensemble models were all tested on unseen data, three multivariate datasets and one univariate dataset. As previously mentioned, the third objective of this study was model tolerance and generalisation. Generalisation refers to the exploration of the possible use of the models in areas other than water quality, with the purpose of asserting the four models, especially the final ensemble model's relevance and tolerance in the wider field of LSTM and ensemble prediction models.

This was achieved through the application of the models to datasets other than water quality and/or water related datasets, such as climate, pollution, and temperature datasets, to assess the performance and thus tolerance of all the models, especially the final ensemble model.

3.6.1 Details of unseen datasets

The models were tested on various unseen multivariate datasets, such as a dataset to predict wind power generation, another to predict air temperature and one to predict the level of pollution, as well as one univariate dataset to predict minimum temperature. All the datasets used in this study are publicly available online.

Multivariate datasets

1) Wind Power Forecasting Dataset

The wind power forecasting dataset [79] was found on the Kaggle: Machine Learning and Data Science Community as part of their Global Energy Forecasting Competition 2012 on Wind Forecasting. The data is publicly available and visible on their site as part of the competition that was held 8 years ago. The dataset includes 7 columns, excluding the date column, labelled "wp1" to "wp7" for the wind power measurements of seven different wind farms. The wind power dataset was normalised by the data provider and thus the original values are unknown, and the units and scale of the original values are also unknown.

2) Air Temperature Dataset

Referred to as the air temperature dataset [80] in this study, the Jena Climate Data set was found on Kaggle. The air temperature dataset includes various meteorological parameters along with air temperature, such as air pressure, air density etc., which was used to predict the air temperature ahead of time.

3) Pollution Dataset

The Beijing PM2.5 Data dataset [81] referred to as the pollution dataset in this study was also found on Kaggle. The pollution dataset includes parameters such as temperature, pressure, dew point, wind speed and direction etc., along with the pollution level which was used to predict the future level of pollution. The measure of pollution in the dataset is referred to as PM2.5, which is the atmospheric particulate matter (PM) that has a diameter less than 2.5 micrometers [82].

Univariate dataset

The daily minimum temperatures in Melbourne dataset [83] was found and viewed on Kaggle, whilst the original dataset was hosted by the Data Market Qlik Sense Data Sources. The dataset contained only one parameter, the minimum daily temperature in Melbourne, which was used to predict the future minimum temperatures in Melbourne.

Summary of datasets

A summary of all the datasets used in this study is presented in Table 3.7 below. The summary contains the name of the dataset, the number of samples in the dataset, the parameter that was chosen to be predicted and the mean and standard deviation of the chosen parameter.

Table 3.7: Summary of datasets

Dataset	Chosen Parameter	Samples	Mean	Std.Dev
Burnett	DO(mg/L)	52 560	6.58	0.88
Baffle 2019	DO(mg/L)	52 560	6.77	0.96
Baffle 2015	DO(mg/L)	17 520	6.79	0.97
Air Temperature	Temp(°C)	420 551	9.45	8.42
Pollution	PM2.5(ug/m^3)	43 800	94.01	92.25
Minimum Temp	Temp(°C)	3650	11.18	4.07
*Wind Power	Wind Power(normalised)	18 757	-	-

*As the wind power dataset was already normalised and the original values were unknown, mean and standard deviation could not be calculated.

3.6.2 Comparison of models and classical time series forecasting methods

Comparison is one of the best performance measures. Thus, four classical time series forecasting methods were applied to the univariate time series dataset used in this study with the purpose of comparing their predictive performance capability to that of the final weighted ensemble LSTM model. These four classical models are namely, Autoregression (AR), Moving Average (MA), Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA).

3.7 Summary of the development of a robust and tolerant water quality prediction based ensemble scheme

Data preparation

Prior to model development, the data was pre-processed and cleaned through the removal of any censored or inconsistent observations, the removal of outliers through the use of the interquartile range (IQR), duplicate values were removed to prevent repetition from distorting results, observations at inconsistent time intervals were removed and missing values at regular time intervals were found through interpolation.

After the data was processed, the relationship between the different water quality parameters was explored using the Spearman's Correlation Coefficient, to determine which water quality parameters could be used as inputs for the models.

Dissolved oxygen shares a strong negative correlation with temperature, implying that as the water temperature increases, the dissolved oxygen level decreases. This correlation is present for both the Burnett and Baffle rivers. The correlation between dissolved oxygen and the other water quality parameters were insignificant. Only dissolved oxygen and water temperature were used as the input parameters for the models for the prediction of dissolved oxygen.

Development of a multivariate multi-step stacked LSTM model

Two water quality predictive LSTM models were developed, for the Burnett and Baffle (2019) rivers. Both models were developed in Keras, using Tensorflow backend, in Google Colab (Google Colaboratory).

The Burnett and Baffle models were developed as multivariate multi-step stacked LSTM models. The two parameters, dissolved oxygen and water temperature were used as features to predict the target parameter, dissolved oxygen. The input layer has two features. The model was

developed to have two hidden layers, the first hidden layer with a larger number of LSTM units than the second hidden layer. The output layer is a dense layer that connects the whole model and outputs dissolved oxygen values 24 time steps ahead for a multi-step prediction. The optimiser, Root Mean Square Propagation (RMSprop) and the activation function, Rectified Linear Unit (ReLU) was used for both models.

The only difference between the two models is that overall, the number of LSTM units in the Baffle model was greater. As it is easier for the LSTM network to pick up variations in the data and learn trends over a three-year period, an LSTM model with an overall smaller architecture was used for the Burnett model (data spread over three years) than the Baffle model (data spread over a single year).

Trial-and-error optimisation of a LSTM Model

Through trial-and-error, the two multivariate multi-step stacked LSTM models were optimised in terms of time window size and number of LSTM units in the two hidden layers.

Three arbitrary values were chosen, one for the window size, one for the number of LSTM units in the first hidden layer and another for the number of LSTM units in the second hidden layer. The number of units in the first hidden layer was chosen as being greater than the number of units in the second hidden layer. The model was trained and the RMSE value was calculated to evaluate the model. The smaller the RMSE, the better the model is at predicting dissolved oxygen. This process was repeated until the smallest RMSE value for each model was calculated.

Hybrid genetic algorithm optimised LSTM Burnett model and LSTM Baffle model

The window size and the number of units in the two hidden layers, obtained in the trial-and-error process was used as the initial values during the GA optimisation process. GA optimisation was done using Keras and Tensorflow and the python package called

"Distributed Evolutionary Algorithms in Python" (DEAP). The optimisation of the models was done separately.

The number of units in the hidden layers and the size of the time windows are used to evaluate the fitness function. When the genetic operators explore the search space, the population becomes composed of possible solutions. Selection and recombination operators, look for the best solution within the population. Each solution is evaluated in accordance with the predefined fitness function and the chromosomes (solution) with the best performance are chosen for reproduction. Values were defined using the DEAP package: population size (70), mutation rate (0.15), crossover rate (0.7), number of generations (10).

This study chose to use the RMSE value to evaluate the fitness of each chromosome. The solution with the lowest RMSE value was regarded as the optimal or near-optimal solution.

Ensemble LSTM model through weight-based technique

A new water quality dataset, Baffle 2015 dataset (which was not used to develop the two LSTM models) was used to develop the ensemble models. The dataset was split differently: 30% (training), 50% (validation) and 20% (testing). The validation dataset was the largest now. The optimal weight for each model in the ensemble was not explicitly calculated, instead the holdout validation set, unseen by the individual models during the training process, was used to estimate the weight of each model in the ensemble. The validation dataset was used instead of the training set to prevent model over-fitting.

Predictions were made by models using the validation dataset as the test dataset and then evaluated through the RMSE value. The lower the RMSE value, the better the overall performance of the model. An exhaustive, yet simple grid search was employed to find the optimal weight values/ coefficients of each model. The optimal weight combination of the two models was found through the minimisation of the RMSE value.

Chapter 4

Results and analysis

4.1 The prediction performance of the trial-and-error optimised LSTM Burnett and Baffle models

The two graphs below show the ability of the Burnett model in Figure 4.1 and the Baffle model in Figure 4.2 to predict dissolved oxygen from historical water temperature and dissolved oxygen data. These models were optimised through trial-and-error. Through this method, the Burnett model was found to have a window size of 100 time steps, 32 LSTM units in the first hidden layer and 16 LSTM units in the second hidden layer. The Baffle model was found to have a window size of 150 time steps, 64 LSTM units in the first hidden layer and 32 LSTM units in the second hidden layer. The historical dissolved oxygen data is represented by a blue line, the predicted dissolved oxygen values (that were predicted using the historical data shown in the graph) for the 24 time steps ahead are represented with red dots, whilst the actual or true dissolved oxygen values are shown with green dots for comparison.

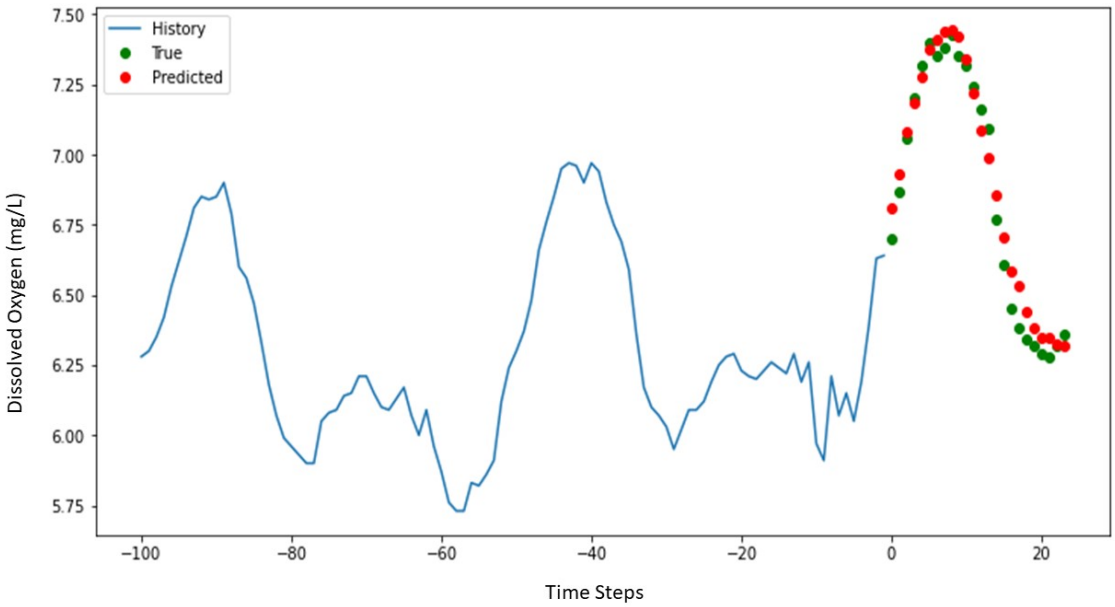

Figure 4.1: Performance graph of the Burnett Model for the prediction of dissolved oxygen (Model A)

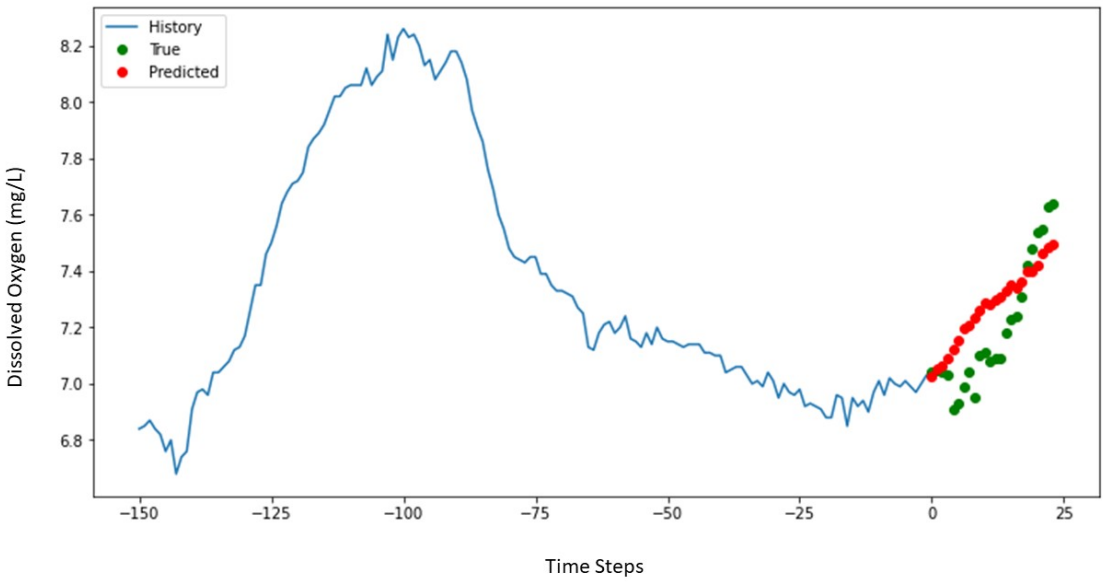

Figure 4.2: Performance graph of the Baffle Model for the prediction of dissolved oxygen (Model B)

4.1.1 The comparison of the performance of Burnett and Baffle LSTM models

It is obvious from the two graphs, that the graph in Figure 4.1, which shows the Burnett model, has a greater ability to predict dissolved oxygen than the Baffle model in Figure 4.2. The green and red dots are fairly well aligned in Figure 4.1 when compared to the green and red dots in Figure 4.2, which are hardly aligned and only intersect at one point.

This notion is further supported by the graph in Figure 4.3, which shows a detailed comparison of the predictive ability of the Burnett and Baffle models in terms of various performance criteria.

Figure 4.3 compares the performance of the Burnett and Baffle models in terms of the performance criteria commonly used to assess LSTM performance and regression capability; namely the RMSE (Root Mean Square Error), MSE (Mean Sqaured Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Performance Error), Max Error (Maximum Error), Median AE (Median Absolute Error), EV Score (Explained Variance Score) and R^2 score (co-efficient of determination).

The Burnett model has lower RMSE, MSE and MAE values when compared to the Baffle model. The RMSE value (0.99) for the Baffle model is almost four times greater than that of the Burnett model (0.28), the same is true for the MAE value (0.65 for the Baffle model and 0.18 for the Burnett model), with the MSE value of 0.98 for the Baffle model, being 12 times greater than that of the Burnett model with 0.08. The MAPE value for the Burnett model is 2.7% and 9.4% for the Baffle model, approximately three times bigger. All these values point to a smaller difference between the predicted values and the actual values by the Burnett model in comparison to the Baffle model. The values for the performance metrics can be found in Table 4.1 below and is also illustrated in Figure 4.3.

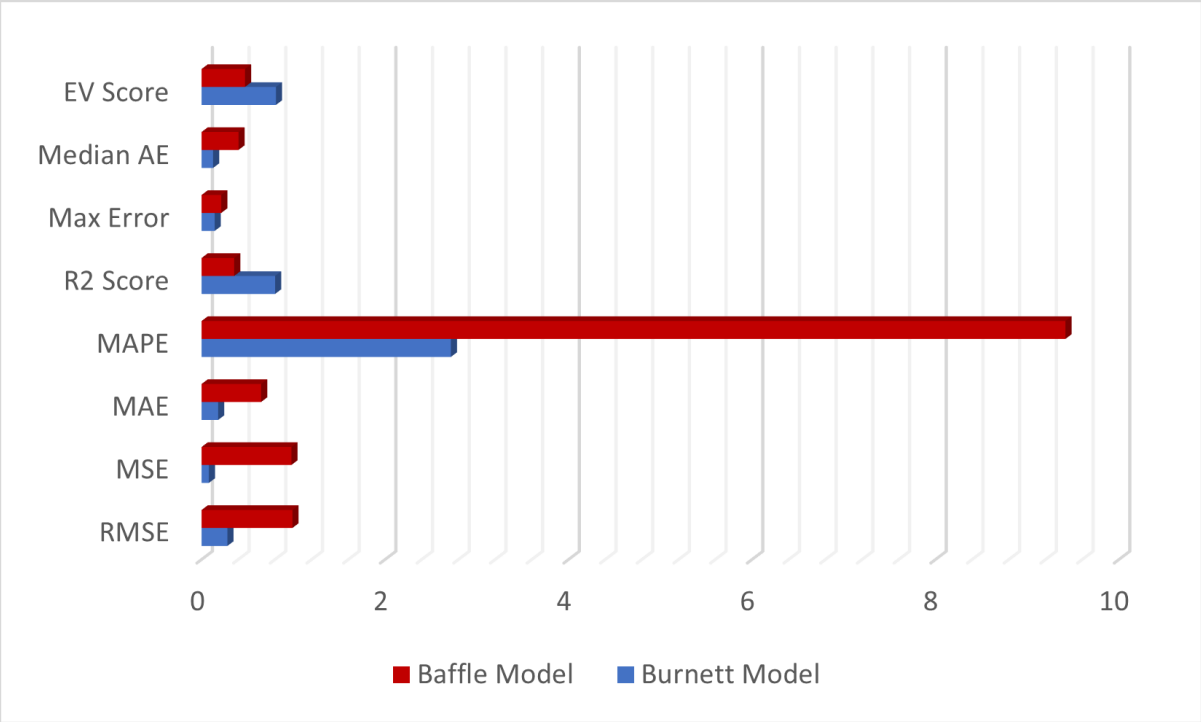

Figure 4.3: Performance comparison the Burnett Model and the Baffle Model

Table 4.1: Performance metrics of Burnett and Baffle LSTM models

Performance Metric	Burnett	Baffle
RMSE	0.28	0.99
MAE	0.18	0.65
MSE	0.08	0.98
MAPE	2.7%	9.4%
R^2 Score	0.80	0.36
EV Score	0.81	0.47
Max Error	0.14	0.21
Median AE	0.13	0.40

The R^2 score is a measure of how well the model is able to fit the data, by evaluating how close the predicted values are to the actual values. The closer the value is to one, the better

the fit of the model. The Burnett model achieves a R^2 score value of 0.80, whilst the Baffle model achieves an R^2 score value of 0.36 (almost half of the Burnett model), proving that the Burnett model has a greater predictive capability. The Burnett model has an explained variance score of 0.81 and the Baffle model achieves a score of 0.47, following the same trend as the R^2 score. This indicates that the Burnett model is more capable than the Baffle model in terms of accounting for variation in data.

The worst error obtained from the Burnett model is 0.14 and 0.21 for the Baffle model as shown by the maximum error, emphasising the better performance of the Burnett model. There is a median absolute error of 0.13 for the Burnett model and 0.40 for the Baffle model. This implies that the maximum error is close to the median absolute error for the Burnett model, whilst the median absolute error for the Baffle model is greater than the maximum error, further proving the superiority of the performance of the Burnett model. These performance metrics are recorded in Table 4.1 and be viewed graphically in Figure 4.3.

From the graph in Figure 4.1, it can be seen that despite having a smaller window size (100 time steps), than the graph in Figure 4.2 with a 150 time steps per window; the Burnett model is fed data over a greater time period i.e. three years (as is evident by the three peaks in the graph (a repeated trend) in Figure 4.1) and the Baffle model is fed denser spaced data over a single year (evident by the single peak in the graph in Figure 4.2). This greater time range allows for the greater diversity of data and also for the repetition of the same trend three times, enabling the Burnett model to be trained better than the Baffle model and to thus have greater predictive capacity.

4.2 The ability of the hybrid GA-optimised LSTM Burnett and Baffle models to predict water quality

The graphs below show the ability of the Burnett model and the Baffle model to predict dissolved oxygen after both models were optimised in terms of window size and number of LSTM units in the two hidden layers using the genetic algorithm. Figure 4.4 shows the performance graph of the GA-Burnett LSTM model and Figure 4.6 shows the performance of the GA-Baffle LSTM model (graph will appear further on in the study). Again, the blue line represents the historical dissolved oxygen values, the green dots represent the true or actual values, and the red dots represent the predicted dissolved oxygen values.

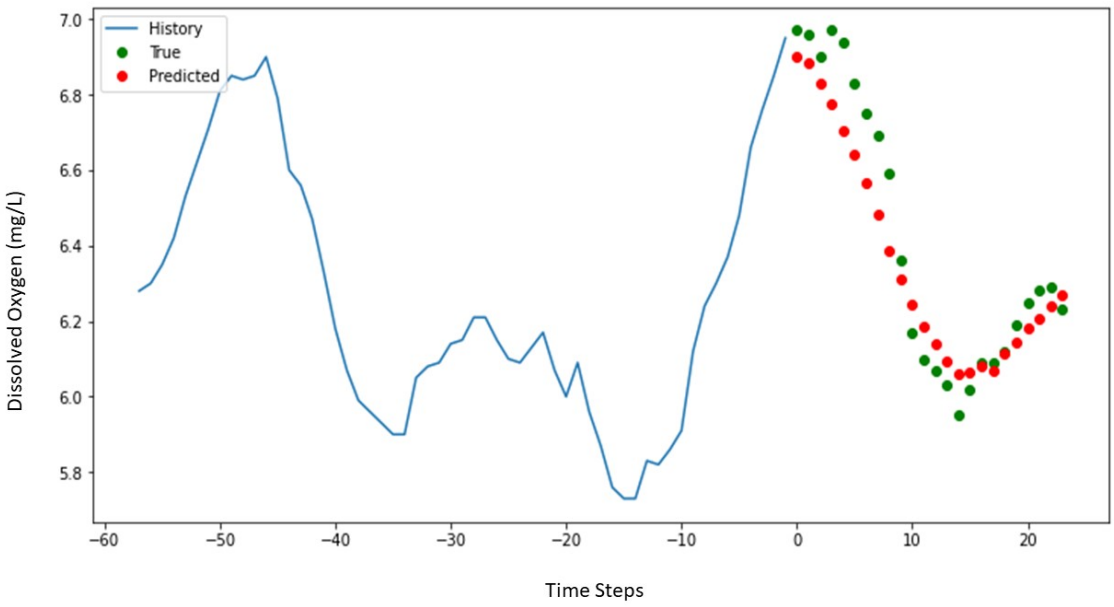

Figure 4.4: Performance graph of GA-optimised Burnett Model

4.2.1 The comparison of the performance of GA-Burnett and Burnett LSTM models

The graph in Figure 4.4 shows the performance of the GA-Burnett model. This graph shows how the GA-Burnett model predicted dissolved oxygen values 24 time steps ahead with a window size of 57 time steps of historical data. The window size for the GA-Burnett model is now almost half of what it was for the Burnett model. The number of parameters to be trained for the GA-Burnett model has also decreased as the units in the first and second hidden layers have decreased from 32 to 10 units and 16 to 8 units, respectively. The graph in Figure 4.4 is rather similar to the graph in Figure 4.1, with the red and green points aligning at numerous instances.

This similarity in performance is mirrored by the performance criteria shown in Figure 4.5 which compares the performance of the Burnett LSTM model to the GA-Burnett LSTM model, in terms of RMSE, MSE, MAE, MAPE, maximum error, R^2 Score, explained variance score and median absolute error. Figure 4.5 shows that the performance of the Burnett model has improved after optimisation by the genetic algorithm i.e., the GA-Burnett model. However, the improvement in performance is only slight as the difference between the performance measures for the Burnett model and the GA-Burnett model is relatively small. This can be clearly seen in Table 4.2 where there is little difference between the values achieved by the two models (GA-Burnett and Burnett models) on the same dataset. The GA-Burnett model has an RMSE, MAE, MSE and MAPE values of 0.25, 0.17, 0.06 and 2.5%, respectively. These values are just slightly lower than that of the Burnett model.

Despite the GA-Burnett model having a bigger maximum error (0.22) than the Burnett model (0.14), the median absolute error of the GA-Burnett model (0.12) is almost the same as the Burnett model (0.13), indicating that the maximum error of the GA-Burnett model is most probably calculated from an outlier prediction.

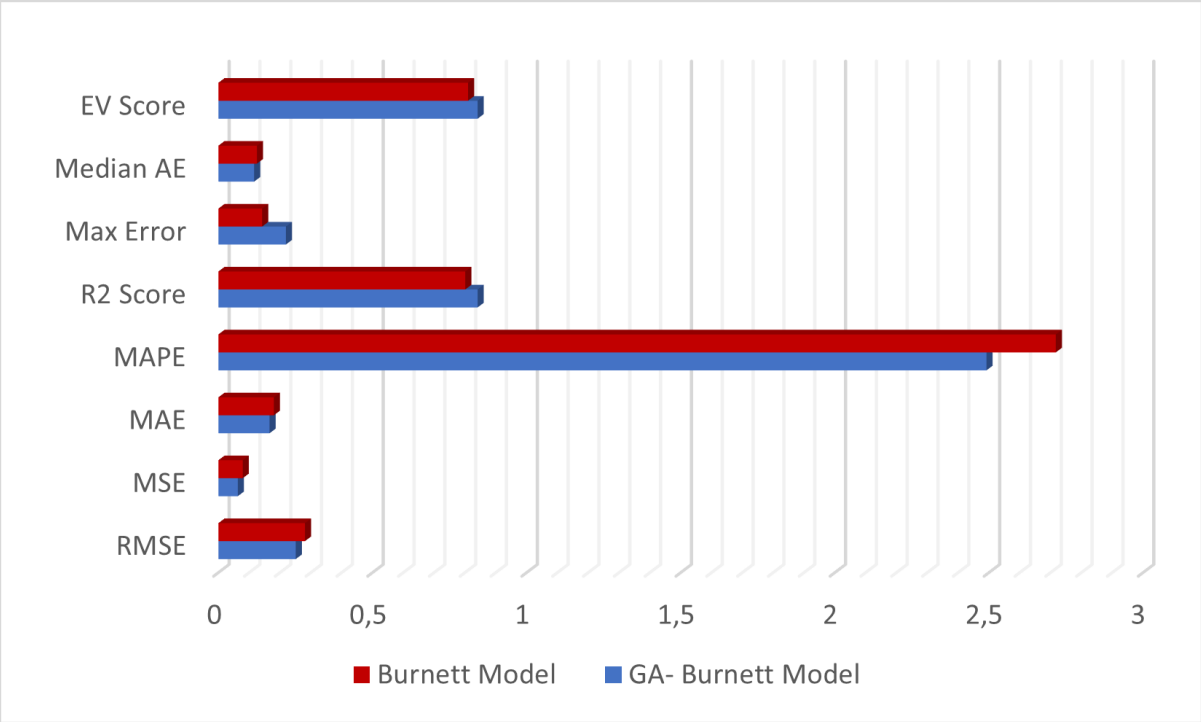

Figure 4.5: Comparison of the performance of the Burnett LSTM model to the Burnett GA-optimised LSTM model

Table 4.2: Performance metrics of GA-Burnett and Burnett LSTM models

Performance Metric	GA-Burnett	Burnett
RMSE	0.25	0.28
MAE	0.17	0.18
MSE	0.06	0.08
MAPE	2.5%	2.7%
R^2 Score	0.84	0.80
EV Score	0.84	0.81
Max Error	0.22	0.14
Median AE	0.12	0.13

The R^2 score of 0.84 and explained variance score of 0.84 are only slightly greater than that

of the Burnett model. Despite the obvious optimisation of the Burnett model by the genetic algorithm and the changes resultant from this optimisation, the change in predictive ability is almost negligible. It is probable that the Burnett model had already been optimised to a great extent, as shown by the good values obtained for the performance metrics in Figure 4.3 and Table 4.1 and the close alignment of the green and red points in Figure 4.1, and any further optimisation would not make much difference to the model's predictive ability.

4.2.2 The comparison of the performance of the GA-Baffle and Baffle LSTM models

The graph in Figure 4.6 shows the performance of the Baffle model after optimisation by the genetic algorithm, the GA-Baffle model's performance. The GA-Baffle model performance has significantly improved after optimisation as can be seen by the closer alignment of the red and green points in the graph. The green and red points only intersect once in the performance graph for the Baffle model shown in Figure 4.2.

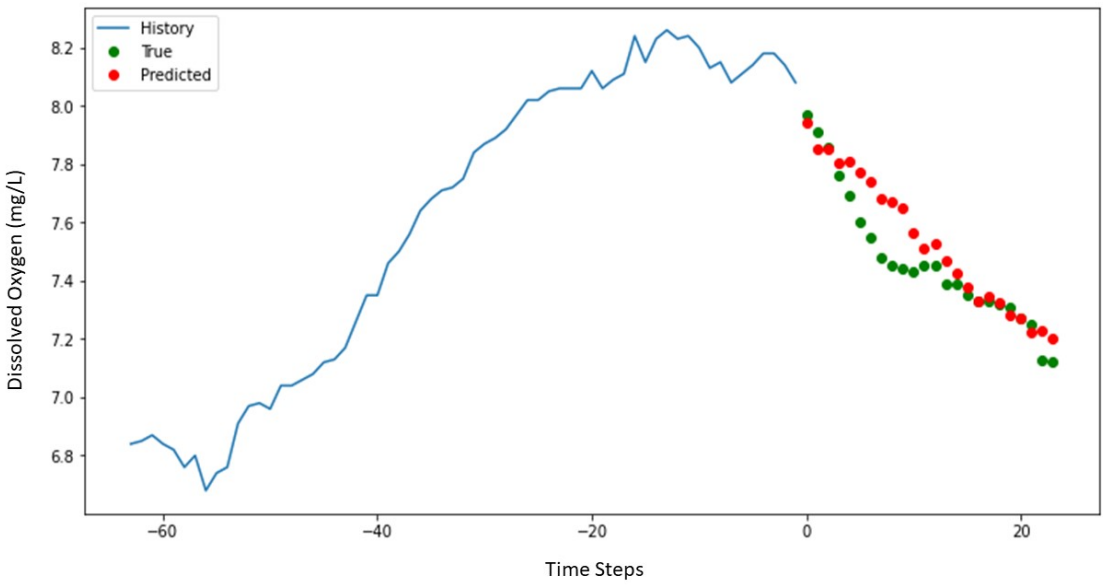

Figure 4.6: Performance graph of GA-optimised Baffle Model

This improved performance is also illustrated by the difference in performance metrics shown in Figure 4.7 which compares the performance of the LSTM Baffle model to the GA-optimised LSTM Baffle model, using the same performance criteria that were used to compare the two Burnett models.

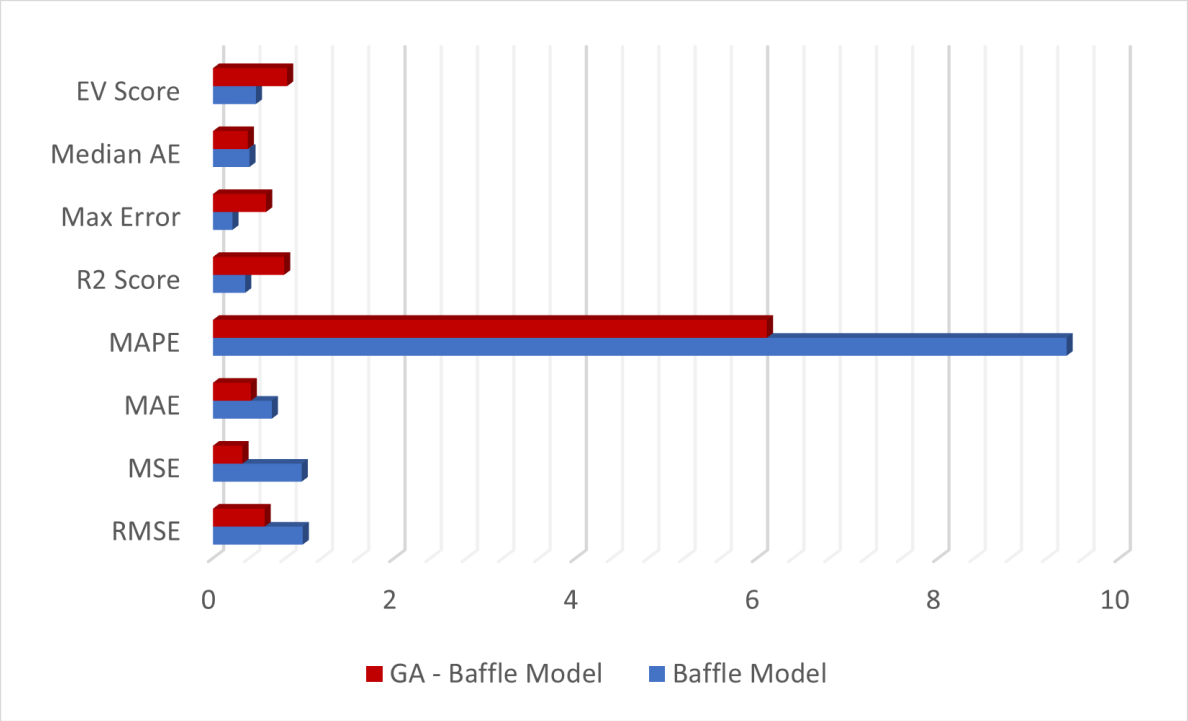

Figure 4.7: Comparison of the performance of the Baffle LSTM model to the Baffle GA-optimised LSTM model

The GA-Baffle model had obtained RMSE, MAE, MSE and MAPE values of 0.57, 0.42, 0.33 and 6.11%, respectively. These values can be seen in Table 4.3 and are illustrated in Figure 4.7. This improvement is the most evident in the increase of the R^2 score from 0.36 to 0.79 and the explained variance score from 0.47 to 0.82. The R^2 score is more than twice as large as it used to be and is now closer to one, showing that the GA-Baffle model is able to fit the data better than the Baffle model and thus makes predictions closer to the actual values and hence has greater predictive capability than the Baffle model.

The explained variance score has also doubled, indicating that the GA-Baffle model is more

capable than the Baffle model with regards to accounting for the variation in the data. The median absolute errors of the GA-Baffle model (0.39) and Baffle model (0.40) are very similar, but the maximum errors are quite different with the GA-Baffle model (0.59) and the Baffle model (0.21). The maximum error of the Baffle model is half the size of the absolute median error, showing the inferior performance of the Baffle model compared to the GA-Baffle model.

Table 4.3: Performance metrics of GA-Baffle and Baffle LSTM models

Performance Metric	GA-Baffle	Baffle
RMSE	0.57	0.99
MAE	0.42	0.65
MSE	0.33	0.98
MAPE	6.11%	9.4%
R^2 Score	0.79	0.36
EV Score	0.82	0.47
Max Error	0.59	0.21
Median AE	0.39	0.40

From Figure 4.7 it can be surmised that the GA-Baffle model performs significantly better than the Baffle model. This significant improvement was not present between the Burnett and GA-Burnett models. After optimisation, the window size for the Baffle model changed from 150 time steps to 63 time steps, the new window size is less than half the size it was. The units in the first and second layers were reduced from 64 to 12 and from 32 to 10 units, respectively. The architecture of the GA-Baffle model is much smaller than that of the Baffle model. The difference in the architecture of these two models is greater than the difference in architecture observed by the Burnett model after GA-optimisation. This greater difference in architecture is responsible for the more notable improvement in the performance of the Baffle model after

GA-optimisation.

Figure 4.8 shows a detailed comparison of the performance of the two GA-optimised LSTM models in terms of numerous performance criteria. From Figure 4.8 it can be seen that the GA-Burnett model performs better than the GA-Baffle model. Although, the GA-Baffle model

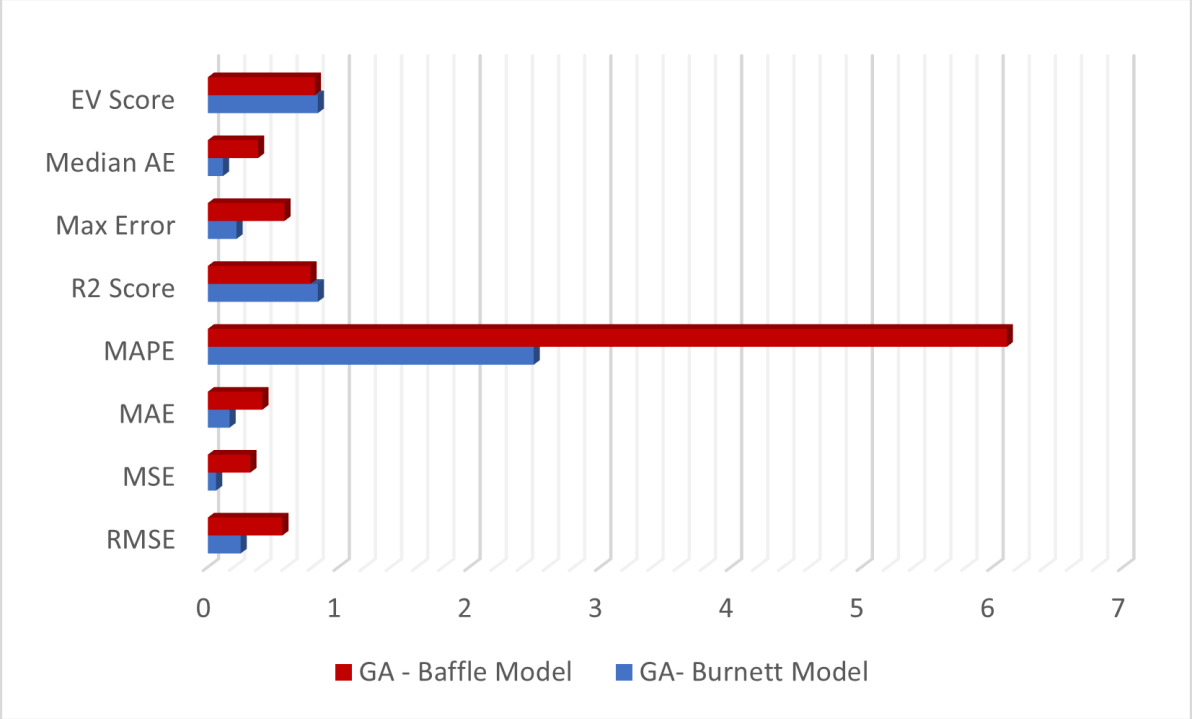

Figure 4.8: Comparison of GA-optimised Burnett Model and GA-optimised Baffle Model

showed improved performance in comparison to the Baffle model; the GA-Burnett model still outperforms the GA-Baffle model. The performance of the Burnett model was better than that of the Baffle model from the onset, primarily due to the data that was used to train the model. Through the application of the genetic algorithm, the already well performing Burnett model was only slightly optimised and the Baffle model was further optimised and yet the GA-Burnett model still outperforms the GA-Baffle model.

4.2.3 The comparison of the performance of the GA-optimised models to the LSTM models

Figure 4.9 shows various performance measures which compare the predictive capability of the two GA-optimised LSTM models to the original LSTM models.

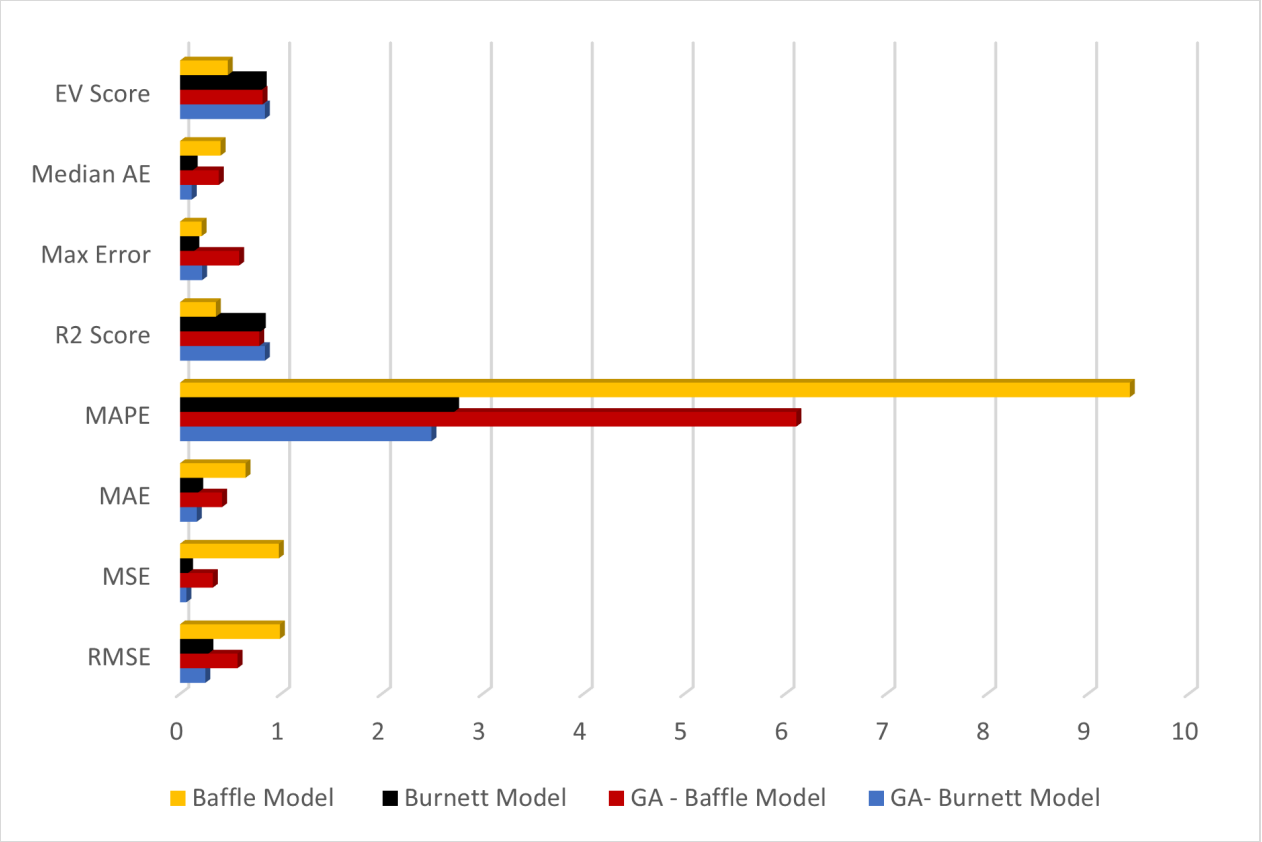

Figure 4.9: Comparison of the performance of the LSTM models to the GA-optimised models

The graph in Figure 4.9 visually summarises the performance metrics of all four models, the Burnett and GA-Burnett models as well as the Baffle and GA-Baffle models alongside one another for the purpose of comparison. The Baffle model has the largest and the worst RMSE, MSE, MAE and MAPE values from all four of the models, whilst the GA-Burnett model has the lowest values for these performance metrics and hence the best values. These values are only slightly better than that of the Burnett model and thus this difference is negligible. Three of the models: the GA-Burnett, Burnett and GA-Baffle models all have similar R^2

scores, around 0.8 and similar explained variance scores, also all around 0.8 (as can be seen in Table 4.2 and Table 4.3. The Baffle model has a much lower R^2 score of 0.36 and an explained variance score of 0.47 (as shown in Table 4.1), almost half of what the values are for the other models. Thus, the Baffle model only fits the data half as well as the other models and accounts for the variation in the data only half as well as the other models.

Interestingly, both the GA-optimised models, GA-Burnett and GA-Baffle models have greater maximum errors than their original respective counterparts, the Burnett and Baffle models. The median absolute errors of the GA-optimised models are smaller than their maximum errors, indicating that the GA-optimised models, despite producing better and mostly consistent predictions, are prone to a couple of outlier predictions. Overall, the GA-Burnett model shows the best predictive capability, and the Baffle model shows the worst performance from all four models.

In general, both GA-optimised models show improved predictive performance compared to their respective original counterparts, this improved performance is representative of the improved robustness of the original LSTM models.

4.3 The predictive capability of the ensemble models

Figure 4.10 shows the detailed predictive performance capacity, in terms of various performance metrics, of the average ensemble model and the weighted ensemble model.

In the average ensemble model, the GA-Burnett model and the GA-Baffle model contribute equally to the ensemble model. In the weighted ensemble model, the GA-Burnett model contributes 60% and the GA-Baffle model contributes 40% to the final ensemble model. The weight contribution in the weighted ensemble model was the optimal weight combination found by this study.

The weight combination in the weighted ensemble model of 60% and 40% is quite similar to that of the average ensemble of 50% and 50%. There is only a 10% difference. This is resultant from the similarities between the two base GA-LSTM models. Both models are multivariate stacked LSTM multi-step models, with 2 input features, 2 hidden layers (the first hidden layer always has a greater number of units than the second hidden layer in both models) and a single dense layer which consolidates the entire model and outputs 24 predicted values. Both models have the same optimiser and activation function.

The only difference between the two models is the number of units in the hidden layers which leads to the GA-Burnett model with 1344 trainable parameters and the GA-Baffle model with 1904 trainable parameters and a small difference of 560 parameters and 6 minutes of computation time. Trainable parameters and computation time are discussed in more detail in section 4.6 and Table 4.10.

The numerous similarities between the GA-optimised LSTM models is the cause of the (almost) equal strength of both models and thus even after an exhaustive search for the optimal weight combination, the found weight combination (60% and 40%), could not deviate much from the equal strength combination (50%-50%).

4.3.1 The comparison of the performance of the average ensemble model and weighted ensemble model

The comparison between the average ensemble model and the weighted ensemble model, in terms of performance metrics, is illustrated in Figure 4.10 and values can be viewed in Table 4.4. The difference in the performance of the two ensemble models is negligible. The average ensemble model achieved RMSE, MSE, MAE and MAPE values of 0.188, 0.034, 0.129 and 2.093%, respectively whilst the weighted ensemble model achieved similar RMSE, MSE, MAE and MAPE values of 0.186, 0.035, 0.129 and 2.083%, respectively. The ensemble models

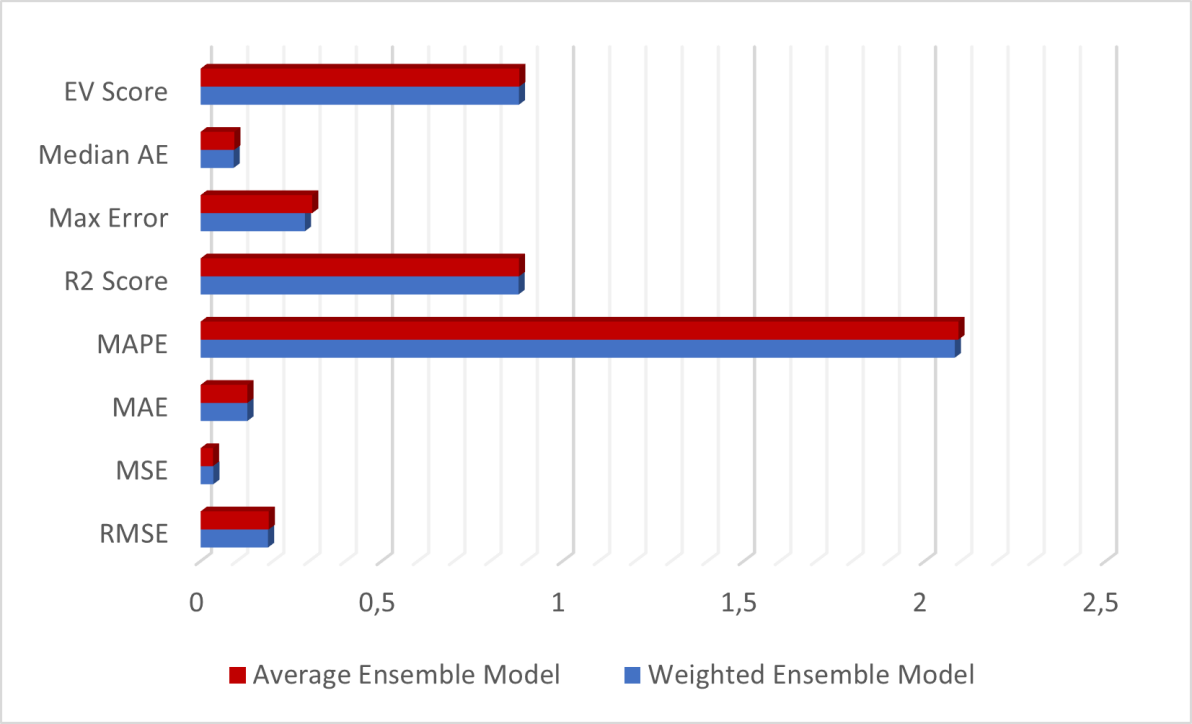

Figure 4.10: Comparison of the performance of the average ensemble and the weighted ensemble

have almost the same R^2 score, with 0.879 for the average ensemble and 0.878 for the weighted ensemble. The same is observed for the explained variance score, with 0.88 for the average ensemble and 0.879 for the weighted ensemble.

The maximum error of the average ensemble (0.31) and the weighted ensemble (0.29) are very similar and both models have the same median absolute error of 0.09. The huge difference between the maximum error and the median absolute error observed for both ensemble models reaffirms the trend of random outlier predictions that were also observed in the performance of the GA-optimised LSTM models, where there was a notable difference between the maximum and median errors. The GA-optimised LSTM models were used as base models to develop both ensemble models. It is evident that the behaviour of the GA-optimised LSTM base models has been transferred to ensemble models.

Table 4.4: Performance metrics of Average and Weighted ensemble models

Performance Metric	Average Ensemble	Weighted Ensemble
RMSE	0.188	0.186
MSE	0.034	0.035
MAE	0.129	0.129
MAPE	2.093%	2.083%
R^2 Score	0.879	0.878
EV Score	0.88	0.879
Max Error	0.31	0.29
Median AE	0.09	0.09

In accordance to Table 4.4 and Figure 4.10 the overall difference between the performance in the average ensemble model and the weighted ensemble model is negligible, as the weighted ensemble model only slightly outperforms the average ensemble model.

This closeness in performance is due to the weight of each model in the ensemble. As the GA-Burnett model only has 10% greater power in the weighted ensemble than in the average ensemble, the difference in performance and predictive capability of the two ensemble models can only be slight.

4.3.2 The predictive capability of the ensemble models compared to the individual GA-optimised LSTM models

The illustration of the performance of the individual LSTM models, GA-Burnett model and GA-Baffle model as well as the ensemble models, average ensemble model and the weighted ensemble model on the same Baffle 2015 dataset (dataset C) is shown in Figure 4.11. A clear conclusion from the graph in Figure 4.11 is that all the models obtained similar values for the performance

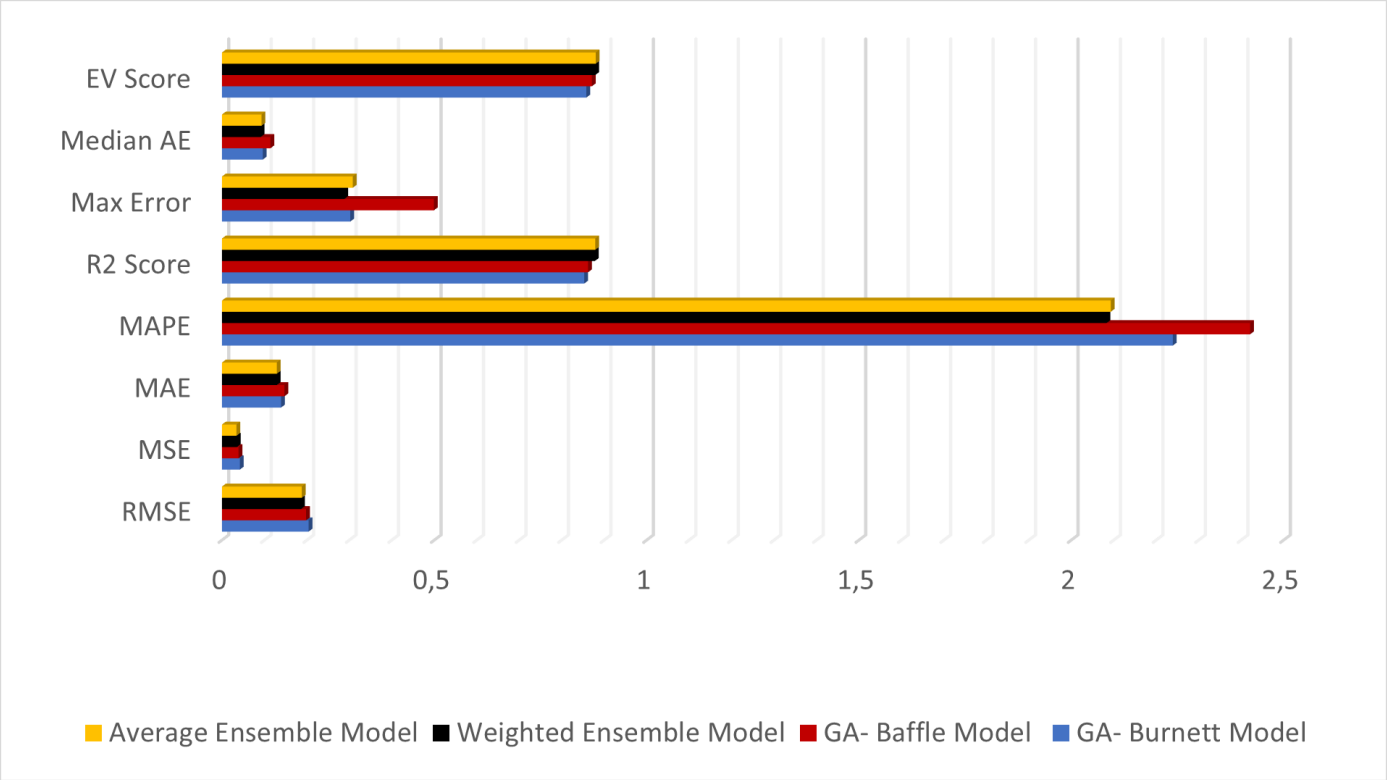

Figure 4.11: Comparison of the performance of the ensemble models to the individual GA-optimised models

metrics. The weighted ensemble model performs the best, with the average ensemble model performing almost just as well. Again, the difference in performance is negligible.

It should be noted that the performance of the GA-Baffle model is rather similar to that of the GA-Burnett model, with the GA-Baffle model achieving better results than the GA-Burnett model in some of the performance metrics. This closeness in performance is unexpected, considering the previous behavioural performance of the models, which entailed the Burnett model always outperforming the Baffle model.

A reason for this could be the trend or lack thereof exhibited by the Baffle 2015 dataset. The water quality data in the Baffle 2015 dataset (dataset C), is spread over a single year, with the yearly river water quality trend only shown once, just as in the Baffle 2019 dataset, the trend is also only shown over a single year. The data in Baffle 2019 dataset was the dataset used to train and optimise the GA-Baffle model. Thus, the similarity of only having a single year of water quality data and only observing a trend once with no further exposure to greater trends over more years could be the cause of the improved performance of the GA-Baffle model on this particular dataset.

Overall, the weighted ensemble performs the best from all four models. However, the difference in the performance of each model when compared to that of the other three models is only slight and can be considered insignificant. This demonstrates consistency in performance and predictive capability. This consistency is evidence of model robustness and can also be linked to improved model tolerance.

4.4 The performance of GA-optimised LSTM models and the ensemble models on unseen multivariate data

The models were tested on various unseen multivariate datasets, such as a multivariate dataset to predict wind power, another to predict air temperature and one to predict the level of pollution.

4.4.1 Wind power generation

The four models were tested on the wind power forecasting dataset, where the future values of wind power generation were to be predicted from the historical data of wind power generation from several wind farms. Figure 4.12 shows the performance of all the models on an unseen multivariate dataset for the prediction of wind power.

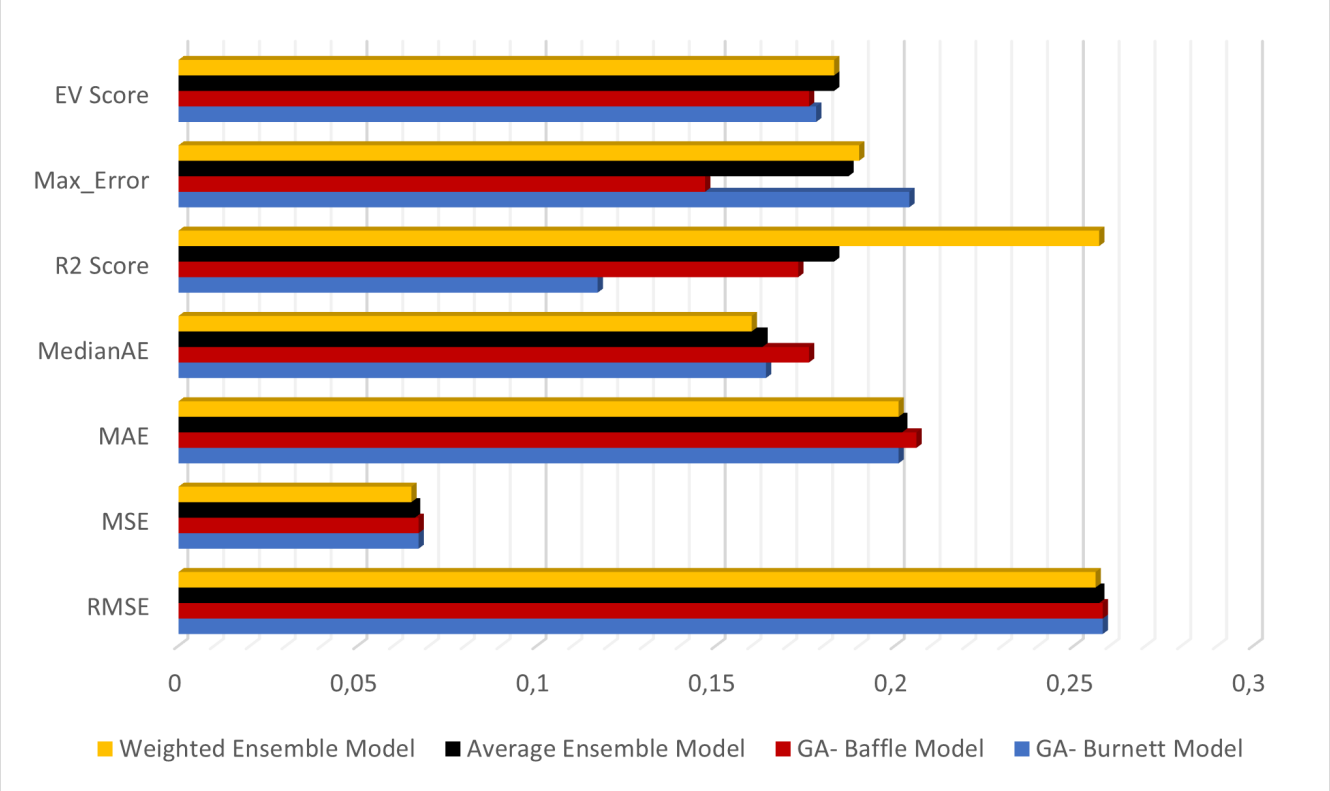

Figure 4.12: Comparison of the performance of the individual GA-optimised LSTM and ensemble models for the prediction of wind power

Figure 4.12 illustrates the consistency of the performance of all four models, the values of which can be seen in Table 4.5. The weighted ensemble model performs the best by a small margin. The GA-Burnett, GA-Baffle, average ensemble and weighted ensemble have similar RMSE values: 0.258, 0.258, 0.257 and 0.256 respectively; MSE values: 0.067, 0.067, 0.066 and 0.065 respectively; MAE values: 0.201, 0.206, 0.202 and 0.201 respectively. In general, the ensemble models outperform the individual GA-optimised LSTM models, with the weighted ensemble model performing the best. All these values are almost the same, with negligible differences.

Table 4.5: Performance metrics of GA-optimised LSTM models and the ensemble models for the prediction of wind power generation

Performance Metric	GA-Burnett	GA-Baffle	Average Ensemble	Weighted Ensemble
RMSE	0.258	0.258	0.257	0.256
MSE	0.067	0.067	0.066	0.065
MAE	0.201	0.206	0.202	0.201
R^2 Score	0.117	0.173	0.183	0.257
EV Score	0.178	0.176	0.183	0.183
Max Error	0.204	0.147	0.187	0.190

In terms of the R^2 score, the GA-Burnett, GA-Baffle, average ensemble and weighted ensemble achieved 0.117, 0.173, 0.183 and 0.257 respectively and explained variance scores of 0.178, 0.176, 0.183 and 0.183. The R^2 and explained variance scores are not only low, they are also very far from the numeric value of one and thus show that despite all the models achieving consistently low RMSE, MAE and very low MSE values, none of the four models performed well in terms of their ability to predict wind power generation.

The maximum error achieved by these models, the GA-Burnett, GA-Baffle, average ensemble

and weighted ensemble models are 0.204, 0.147, 0.187 and 0.190 respectively. The maximum error values, just like the RMSE, MSE and MAE values are relatively low and consistent for all four models. However, the wind power generation values used to train the models are given in positive decimals that are all less than one (as the dataset was already normalised by the data provider), thus in comparison, these seemingly low values are quite large. Thus, all four models show consistent predictive capability on this dataset, but none of the models has a good predictive capability on this dataset.

A possible reason could be the structure of the dataset. The wind power generation dataset has more than one parameter, but the parameters refer to the same feature i.e., the dataset has different columns but all the columns are wind power values- each column refers to the wind power generation from a different wind power station. In practice, these values are each treated as different parameters as they come from different systems (wind farms), whilst in essence, they are the same feature- wind power values.

This data structure differs from the data structure of the water quality datasets that were used to develop the four models. These datasets had several different parameters, each representative of a different feature of the same system, such as water temperature, dissolved oxygen, pH, electrical conductivity etc. These models clearly struggle to make predictions on datasets that do not have multiple features from the same system.

Another reason for this might be data preparation. All four models were originally trained on data that was normalised, using the mean and standard deviation of the dataset, to negate the effect of the different label parameters/features having different numeric scales. As all the values in the wind power generation dataset were at the same scale, due to prior normalisation done on the dataset by the dataset provider, no further normalisation was required.

However, the manner in which the data was normalised is not known and the normalisation carried out by the data provider might be different to the normalisation that was carried out

by this study on the other datasets. This possible difference in data preparation leads to weaker performance on the dataset. In conclusion, the models exhibit great consistency but low tolerance on the wind power generation multivariate dataset.

4.4.2 Air temperature

Figure 4.13 illustrates the performance of all the models on an unseen multivariate dataset for the prediction of air temperature.

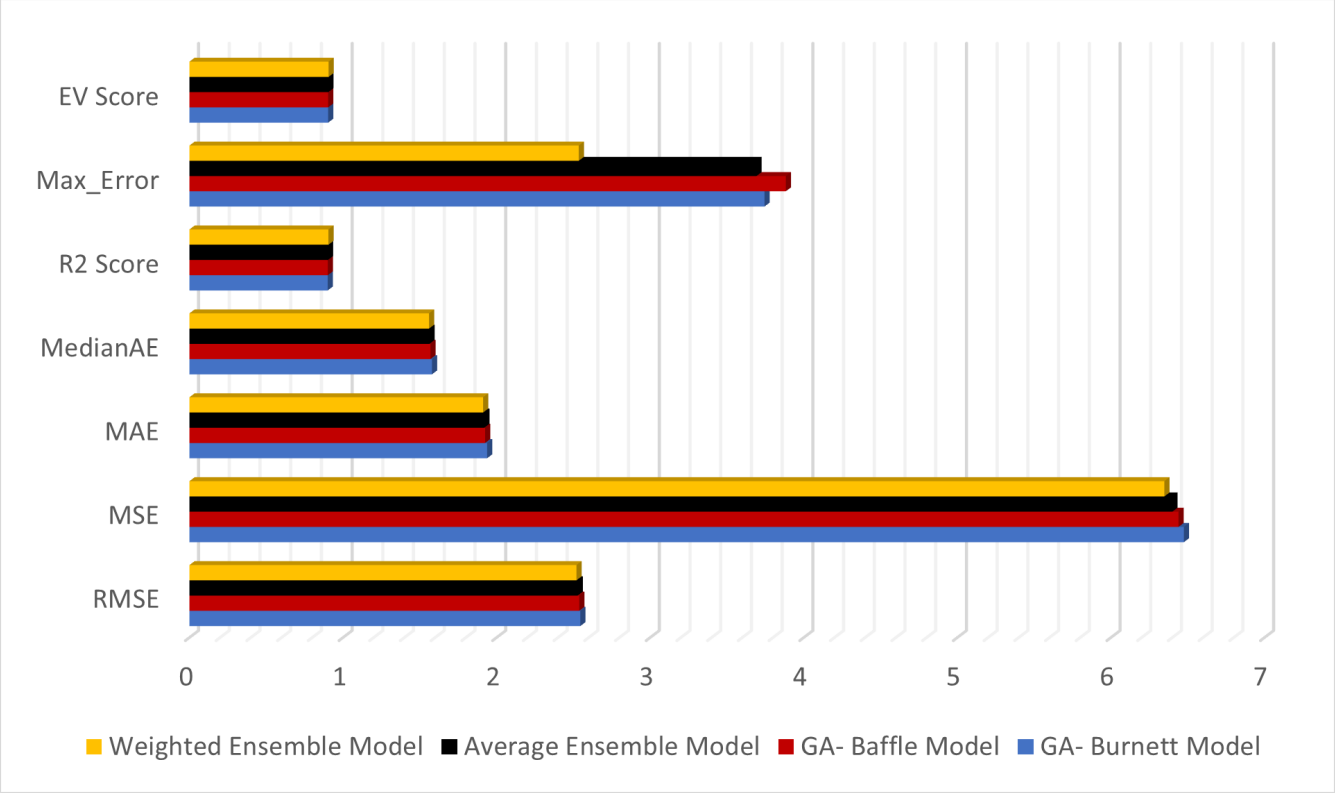

Figure 4.13: Comparison of the performance of the individual GA-optimised LSTM and ensemble models for the prediction of air temperature

Once again, the similarity in the performance of each model points towards the consistency in the predictive capability of each model. As with the wind power generation forecast, the weighted ensemble model performs the best, but the performance is comparable to that of the other models, with differences in model performance being negligible. The values of the performance metrics are shown in Table 4.6. The GA-Burnett, GA-Baffle, average ensemble

and weighted ensemble models obtained RMSE values of 2.545, 2.538, 2.528 and 2.520 respectively; MSE values: 6.475, 6.44, 6.40 and 6.348 respectively and MAE values: 1.938, 1.925, 1.920 and 1.912. These similar numeric values are reflective of the consistency amongst model performance.

Table 4.6: Performance metrics of GA-optimised LSTM models and the ensemble models for the prediction of air temperature

Performance Metric	GA-Burnett	GA-Baffle	Average Ensemble	Weighted Ensemble
RMSE	2.545	2.538	2.528	2.520
MSE	6.475	6.440	6.400	6.348
MAE	1.938	1.925	1.920	1.912
R^2 Score	0.900	0.901	0.902	0.905
EV Score	0.903	0.903	0.904	0.906

The R^2 score of the GA-Burnett, GA-Baffle, average ensemble and weighted ensemble models are 0.900, 0.901, 0.902 and 0.905 respectively, with explained variance scores of 0.903, 0.903, 0.904 and 0.906 respectively. These numeric values are almost identical. These numeric values are also quite high and very close to one, showing the good predictive capability of each model on this dataset as well as a good capacity to account for the variation in the dataset. This contrasts greatly with the poor performance of the models on the wind power generation dataset.

In terms of data structure, the dataset contained various related meteorological parameters along with air temperature, such as air density and wind speed etc. These parameters were all recorded as part of the same system. In a similar way, the four models were developed on water quality datasets that had several related parameters such as dissolved oxygen, water temperature, pH, electrical conductivity which were all recorded as part of the same system.

The similarity in the data structure of the water quality datasets that were used to develop the models and the air temperature dataset, is a possible reason for the good performance of the four models on this particular dataset.

In terms of the data preparation for the training of models on the air temperature dataset, all the values were normalised prior to training using the mean and standard deviation of the data. This method of data preparation mirrors the data preparation for the water quality datasets used in the development of the models.

Similarity, in data structure and data preparation of the air temperature dataset to the water quality datasets used to develop the models, could be the cause of not only the consistent predictive capability but also the good predictive capacity of all four models.

Figure 4.13 shows that same trend as Figure 4.12, where the ensemble models outperform the individual LSTM models; with the weighted ensemble performing the best. However, the difference in the performance of all four models can be considered unimpressive, and as such great similarity and consistency can be observed between the performance of all four models.

The air temperature dataset has 420 551 samples (the biggest dataset used by this study) and the wind power dataset only has 18 757 (one of the smaller datasets used by the study), as shown in Table 3.7. This could indicate that models might be better suited to bigger datasets that have multiple features from the same system. Interestingly, the GA-Burnett model performed the worst on this dataset, this could be due to the GA-Burnett model having the smallest architecture from all the models and thus not having the capacity to deal with such a big dataset. In summary, the models exhibit great consistency and high tolerance on the air temperature multivariate dataset.

4.4.3 Pollution level

Figure 4.14 shows the illustration of the performance of the four models for the prediction of the level of air pollution. The trend of model performance consistency, which was present with the previous two datasets is also present here. As with the two previous datasets, the weighted ensemble model performs the best from all the four models, but by an insignificant margin. The values of the performance metrics are similar and thus consistent and can be viewed in Table 4.7. The GA-Burnett, GA-Baffle, average ensemble and weighted ensemble models

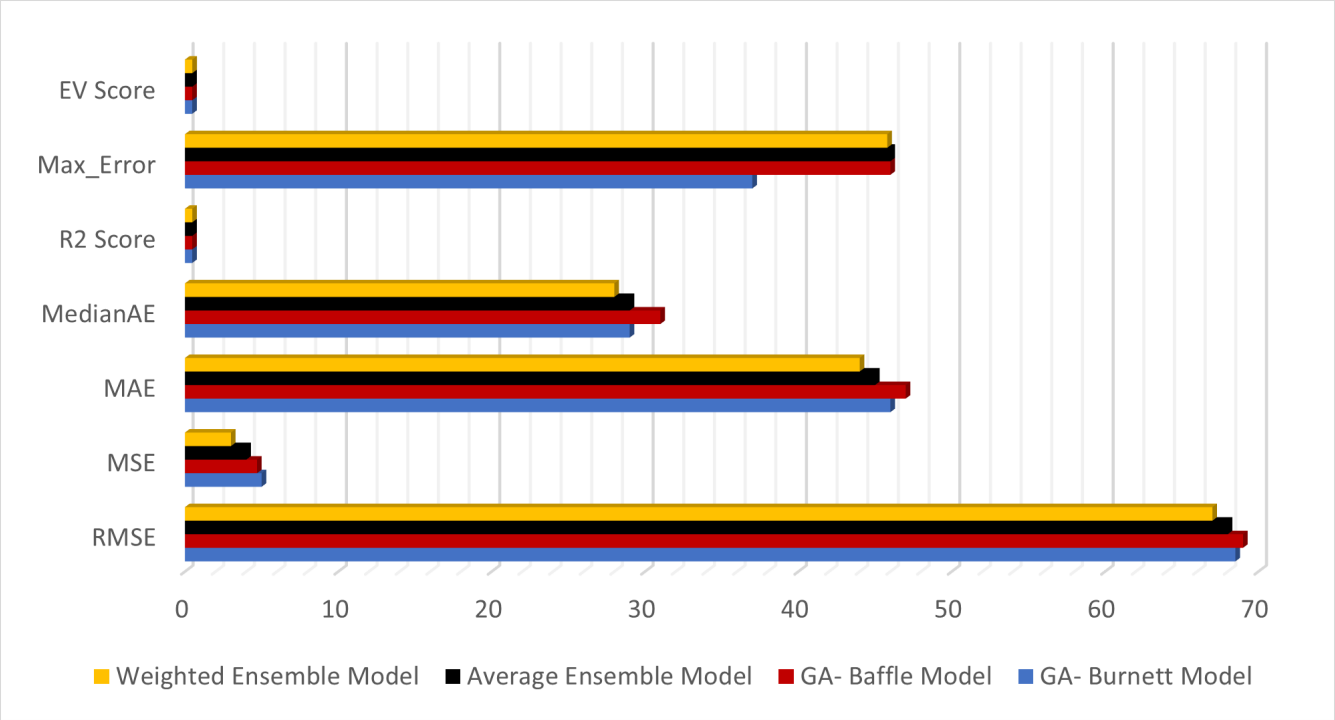

Figure 4.14: Comparison of the performance of the individual GA-optimised LSTM and ensemble models for the prediction of the overall level of pollution

achieved consistent values across performance metrics. The models achieved the following RMSE values 68.5, 69, 68 and 67 respectively; MSE values: 5, 4.7, 4 and 3 respectively and MAE values: 46, 47, 45 and 44 respectively.

The models achieved consistent R^2 and explained variance scores. The GA-Burnett, GA-Baffle, average ensemble and weighted ensemble achieved R^2 score values of 0.472, 0.472,

0.474 and 0.476 respectively; with explained variance scores of 0.473, 0.473, 0.477 and 0.478 respectively. These values are exceedingly similar. They are also evidence of the moderate performance of the models on the air pollution dataset, as the R^2 score and explained variance score values are all close to the 0.5 numeric value and are halfway from one. Hence the performance of the models can be considered neither good nor bad, but average. The models are moderately capable of making predictions and have an average ability to account for the variation in the pollution dataset.

Table 4.7: Performance metrics of GA-optimised LSTM models and the ensemble models for the prediction of pollution level

Performance Metric	GA-Burnett	GA-Baffle	Average Ensemble	Weighted Ensemble
RMSE	68.5	69.0	68.0	67.0
MSE	5.0	4.7	4.0	3.0
MAE	46.0	47.0	45.0	44.0
R^2 Score	0.472	0.472	0.474	0.476
EV Score	0.473	0.473	0.477	0.478

Figure 4.14 and Table 4.7 follow the trend of the ensemble models exhibiting an overall better performance than the individual GA-optimised LSTM models; with the weighted ensemble performing the best. Again, as with the other datasets, the difference in the performance of all four models is relatively low, thus emphasising the consistency in the performance of all four models.

As shown in Table 3.7, the pollution dataset has 43 800 samples, which makes it smaller than the air temperature dataset but bigger than the wind power dataset. The models seem to perform moderately or badly on smaller datasets. In summary, the models exhibit great consistency and moderate tolerance on the pollution dataset.

4.5 The performance of the models and the classical time series forecasting methods on unseen univariate data

4.5.1 The applicability of the four models on the univariate dataset

The four models (weighted ensemble model, average ensemble model as well as the GA-Burnett model and GA-Baffle model) were tested on an unseen univariate dataset to predict the minimum temperature. The performance is illustrated in terms of RMSE values in Figure 4.15. Although the values are close (as can be seen in Table 4.8 and illustrated in

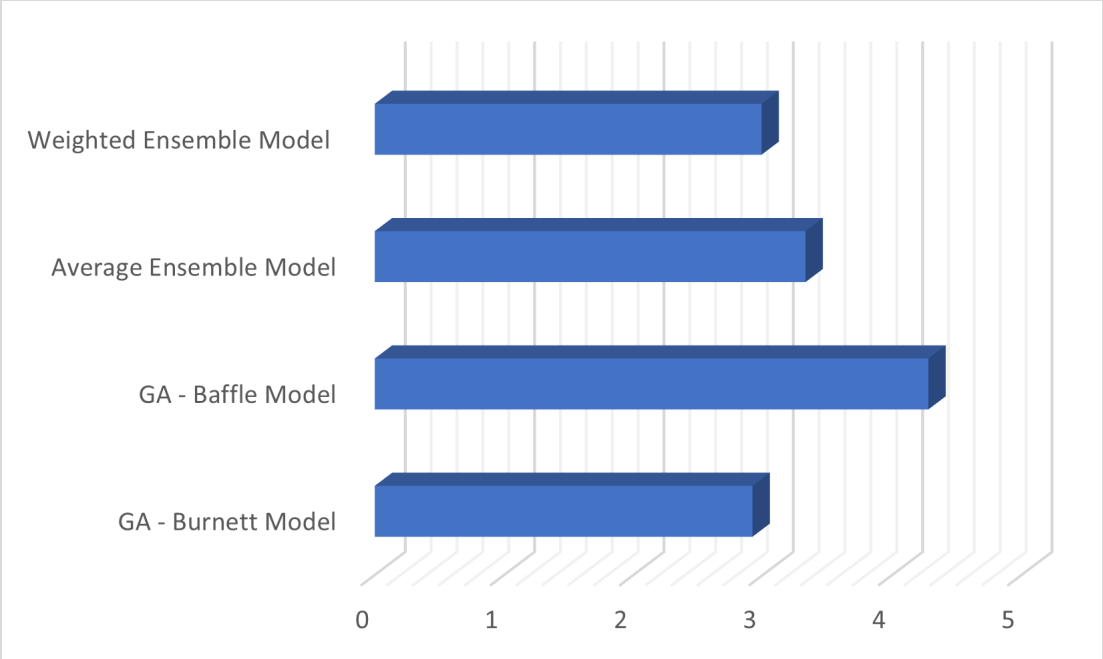

Figure 4.15: Comparison of the performance of the individual GA-LSTM and ensemble models for the prediction of the minimum temperature

Figure 4.15), it is evident GA-Baffle model performed the worst with an RMSE of 4.28, followed by the average ensemble model with an RMSE of 3.33; the performance of the GA-Burnett model and the weighted ensemble model was almost the same with RMSE values of 2.92 and 2.99 respectively. The GA-Burnett model performed slightly better than the weighted ensemble model. Although the difference in the performance of all four models is not

great, it is still more significant than the negligible difference in performance observed between models on the multivariate datasets.

Also, the performance of the GA-Burnett model and GA-Baffle model on the multivariate datasets had achieved very similar results, the first noticeable difference in their performance occurs with the univariate dataset.

Table 4.8: Performance metrics of GA-optimised LSTM models and the ensemble models for the prediction of minimum temperature

Model	RMSE
GA-Burnett	2.92
GA-Baffle	4.28
Average Ensemble	3.33
Weighted Ensemble	2.99

The univariate dataset only takes one feature into consideration when training the model and when making predictions. All four models were developed on multivariate water quality datasets, where two features- dissolved oxygen and water temperature were used to train the models. As such the architecture of the networks catered for multi-feature datasets.

The GA-Baffle model has a bigger architecture than the GA-Burnett model and thus performed poorly in comparison to the GA-Burnett model on the univariate dataset, as its network architecture might have been too large for the univariate dataset. This point is also evident because the GA-Burnett model, the model with the smallest network architecture from all four models performed the best on the univariate dataset. The average ensemble model which is 50% GA-Burnett and 50% GA-Baffle, performed worse than the weighted ensemble which is 60% GA-Burnett and 40% GA-Baffle. This shows that the ensemble model which had more of the model with the smaller architecture performed better.

The minimum temperature dataset only has 3650 samples (the smallest dataset in the study), which is another reason why the model with the smallest architecture performs the best on this dataset. In general, all four of the models do not perform well on the dataset, further indicating that the models do not perform well on smaller datasets.

Overall, it can be concluded that the GA-Burnett model performs the best with the lowest RMSE value, and the GA-Baffle model fairs the worst with the highest RMSE value.

4.5.2 The performance comparison of the models and classical time series forecasting methods

The performance of the four models on the univariate dataset was compared to the performance of the four classical time series forecasting methods namely, AR, MA, ARMA and ARIMA on the same univariate dataset. Figure 4.16 shows the performance of all the models as well as the classical forecasting methods on an unseen univariate dataset for the prediction of the minimum temperature.

It can be seen from Figure 4.16 and Table 4.9, that the classical time series forecasting methods perform better than the models on the univariate dataset, with the ARIMA method performing the best with the lowest RMSE of 2.316 and the ARMA method achieving almost the same value at 2.320. The AR and MA methods achieved RMSE values of 2.389 and 2.98 respectively; with the MA method performing the worst from all four methods. The performance of these four methods is comparable and so the performance difference is largely insignificant.

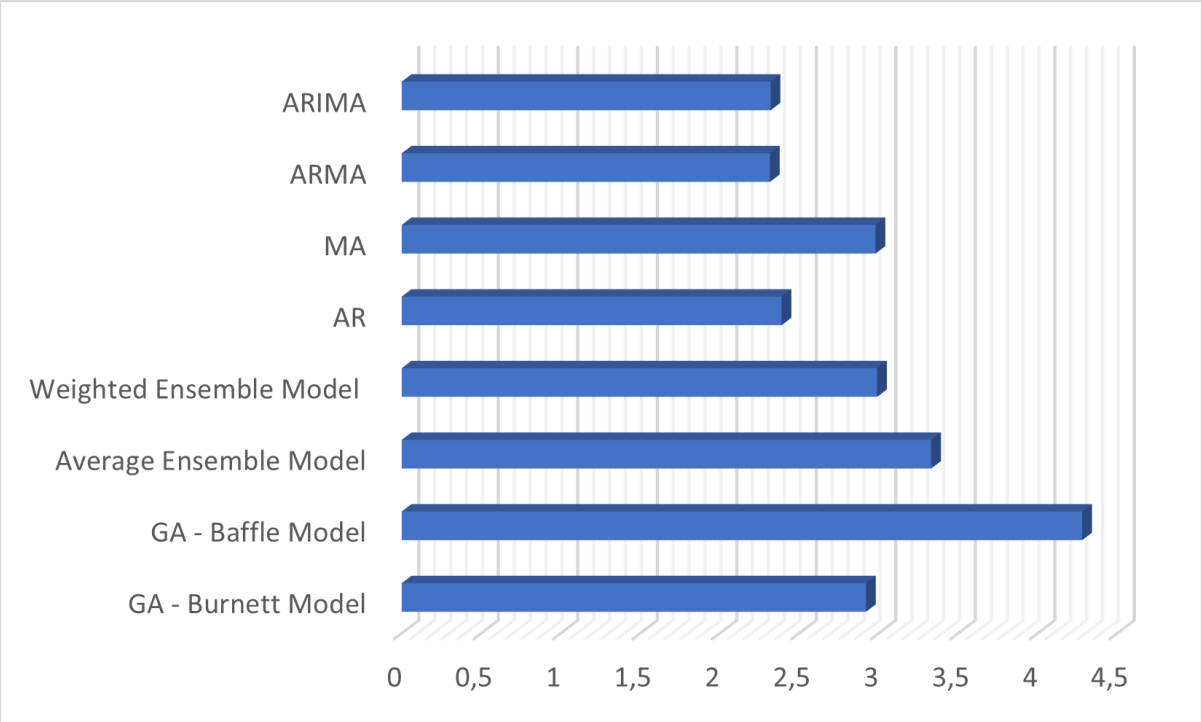

Figure 4.16: Comparison of the performance of the models and the classical forecasting methods for the prediction of minimum temperature

Table 4.9: Performance metrics of the classical time series forecasting methods and the GA-optimised LSTM models and the ensemble models for the prediction of minimum temperature

Method/Model	RMSE
AR	2.389
MA	2.980
ARMA	2.320
ARIMA	2.316
GA-Burnett	2.920
GA-Baffle	4.280
Average Ensemble	3.330
Weighted Ensemble	2.990

From Figure 4.16 and Table 4.9 it can be seen that many of the classical methods

outperform the models on the univariate dataset, with the ARIMA method achieving the lowest RMSE value. The GA-Baffle model performs the worst, with the highest RMSE value. The performance of the worst performing classical time forecasting method, MA is at the same level as the performance of the best performing model, GA-Burnett. This indicates that classical time series forecasting methods might still be better suited and more appropriate for univariate datasets when compared to the LSTM and ensemble models.

It also implies that from the four models, the GA-Burnett model might most closely mimic the behaviour of the classical time series forecasting methods on univariate datasets, this is largely due to the GA-Burnett model's small architecture. In conclusion, the models show a lower tolerance on univariate datasets than on multivariate datasets on average, with the models with smaller architectures having a higher tolerance on univariate datasets than models with bigger architectures.

4.6 Computation time and trainable parameters

Table 4.10 shows the number of trainable parameters for each model, based on the number of units in the two hidden layers of each model and the computation time taken to train each model. Trainable parameters refer to the number of trainable elements in a network-basically, the parameters that are changed during gradient computation by the optimiser after the application of back propagation [84].

Table 4.10: Total number of trainable parameters and the computation time for each model

Model	No. Trainable Parameters	Computation Time
Burnett	8024	23min 41s
GA-Burnett	1344	13min 6s
Baffle	30360	1h 10min 31s
GA-Baffle	1904	19min 29s
Ensemble	3248	25min 41s

As seen in Table 4.10 after GA-optimisation the trainable parameters for the Burnett model decreased by 6680 parameters and the computation time for training the model was reduced by 10 minutes. The number of trainable parameters was reduced by 28 456 parameters for the Baffle model and the computation time was reduced by 50 minutes, thus emphasising the impact of the optimisation by the genetic algorithm on the original Baffle model.

The total number of trainable parameters for the ensemble model (composed of the weight based fusion of the GA-Burnett and GA-Baffle models) is 3248, which is the sum of the number of trainable parameters for the GA-Burnett model and the GA-Baffle model. The computation time for training the ensemble model is 25 minutes and 41 seconds, which is less than the sum of the computation time that was taken to train the GA-Burnett and GA-Baffle models separately, which is 32 minutes and 35 seconds. The computation time taken to train the ensemble model is much less than the time taken to train the Baffle model and is comparable to the time taken to train the Burnett model.

The GA-optimisation of the LSTM models had a bigger impact on model robustness and model computation time than any of the other processes in the development of the ensemble models. The genetic algorithm optimisation decreased the number of trainable parameters as

well as the computation time for both the original Baffle and Burnett models. This concurs with literature, especially by Krstanovic and Paulheim [47], where it was suggested that by configuring the individual LSTM base models of the ensemble, through the tuning of the LSTM hyperparameters, the quality of the individual base models would increase and thus enhance the overall quality of the resultant ensemble model.

The weight based combination of the two models did not have an impact on the number of parameters and only had a small effect on the computation time. However, it did have an impact on model performance and model tolerance. On many of the datasets, the weighted ensemble model had the best performance or close to the best performance as is evident from Table 4.11.

Table 4.11 below shows the best and the worst performing model on each dataset in terms of RMSE values and the difference between the best performing model and worst performing model to show the extent of the performance difference.

In four out of the five datasets, the weighted ensemble model performed better than the other models, to differing extents. At times, the difference between the model performance on the datasets was negligible and at times it was more significant. The worst performing models on each dataset, at times were either the GA-Burnett or GA-Baffle models, even if it was by a small margin.

Thus the weight based combination of the GA-optimised models for the creation of the final ensemble model, does improve the performance of the individual GA-optimised models. This is evident even if it is only by a small margin in certain cases and it does so without substantially increasing computation time. This improvement in performance after the weight based model combination (ensemble model), is evidence of increased model robustness and tolerance.

Table 4.11: The best performing model on the datasets

Dataset	Best Model	Best RMSE	Worst Model	Worst RMSE	Performance Difference
Baffle 2015	Weighted ensemble	0.186	GA-Burnett	0.204	0.018
Wind power	Weighted ensemble	0.256	GA-Burnett GA- Baffle	0.258	0.002
Air temperature	Weighted ensemble	2.520	GA-Burnett	2.545	0.025
Pollution	Weighted ensemble	67.000	GA-Baffle	69.000	2.000
*Minimum temperature	GA-Burnett	2.920	GA-Baffle	4.280	1.360

*It should be noted that although the GA-Burnett model performs better than the weighted average model, the difference is insignificant. The GA-Burnett model achieves a RMSE of 2.92, whilst the weighted ensemble achieves a RMSE value of 2.99 on the minimum temperature dataset.

In summary, the GA-optimisation does have a huge impact on decreasing the number of trainable parameters and hence on decreasing the computation time. The weight based combination of the GA-optimised LSTM models for the formation of the ensemble model, did increase the model performance, with the ensemble model performing better than the individual GA-LSTM models, without increasing the computation time.

4.7 Descriptive statics and correlations of the water quality datasets

Table 4.12: Descriptive statistics of the datasets used to develop the models

Parameter	Mean	Std Dev	Min	25%	50%	75%	Max
			Burnett Dataset				
Temp(°C)	24.45	3.58	11.76	21.07	24.76	27.58	32.29
DO(mg/L)	6.58	0.88	6.00	6.04	6.54	7.04	13.90
pH	7.88	0.53	6.00	7.74	7.84	7.95	8.41
			Baffle 2019 Dataset				
Temp(°C)	25.10	3.94	16.82	21.24	25.98	28.50	32.63
DO(mg/L)	6.77	0.96	4.01	6.06	6.72	7.37	9.00
pH	7.37	1.08	4.53	7.65	7.76	7.85	8.21
			Baffle 2015 Dataset				
Temp(°C)	24.80	3.80	15.72	21.52	25.24	28.17	32.81
DO(mg/L)	6.79	0.97	3.93	6.09	6.83	7.53	9.18
pH	7.90	0.37	6.23	7.72	8.01	8.14	8.48

$*StdDev$:$StandardDeviation$,Min:$Minimum$,Max:$Maximum$,25%,50%,75%:25th,50th,75$thpercentile$

Table 4.12 shows the statics of the three water quality datasets used to develop the models, Burnett dataset, Baffle 2015 and 2019 datasets.

A summary of the Spearman's coefficients of the water quality parameters in relation to dissolved oxygen (target variable) for the water quality datasets is shown below in Table 4.13.

Table 4.13: Spearman's coefficients water quality datasets

Parameters	Burnett Data	Baffle Data 2019	Baffle Data 2015
Temperature	-0.437	-0.419	-0.752
pH	0.361	-0.055	0.731
Dissolved oxygen	1.00	1.00	1.00

4.7.1 Water quality analysis

Table 4.13 shows the Spearman's coefficients for the most significant water quality parameters in relation to dissolved oxygen for the three water quality datasets. All three water quality datasets, show a significant negative correlation between dissolved oxygen and water temperature. This implies that as the temperature of water increases, the dissolved oxygen concentration will decrease, which concurs with the research literature. The negative correlation is of a similar value for the Burnett dataset and the Baffle dataset 2019 (-0.437 and -0.419 respectively). This negative correlation is much stronger in the Baffle dataset 2015 at -0.752.

The relationship between pH and dissolved oxygen is not as clear for the correlation coefficients. Both the Burnett and Baffle 2015 datasets show a positive correlation between pH and dissolved oxygen. However, the correlation in the Baffle 2015 dataset (0.731) is much stronger than that in the Burnett dataset (0.361). The Baffle 2019 dataset shows a completely different correlation, instead of a moderate or high positive correlation, it shows a negative weak correlation between pH and dissolved oxygen.

Correlation does not necessarily translate to causation and as previously stated at the time of this study, there was no documented correlation between dissolved oxygen and pH. Also, the negligible correlation shown in Baffle 2019 does not support the existence of a causal

relationship between pH and dissolved oxygen and thus this study did not include pH in any of the models to predict dissolved oxygen.

The difference in the value of the pH- dissolved oxygen correlation between Baffle 2019 and the other datasets cannot be overlooked. The Baffle 2015 dataset has 17 520 observations spread over a single year (48 observations per day). This spread of data is similar to that of the Burnett dataset with a total of 17 520 observations per year- the difference being that the Burnett dataset has this same density of data over three years, with a total of 52 560 observations. This could be a possible reason for the higher positive correlation in the Baffle 2015 dataset than the Burnett dataset. Perhaps if the trend of pH and dissolved oxygen is observed for over three years, the correlation becomes weaker.

The data in Baffle 2019 dataset has a total of 52 560 observations over a single year i.e., 144 observations per day and is thus the most dense dataset from all three water quality datasets. It is much more dense than 48 observations per day of the Baffle 2015 dataset. Densely spaced datasets imply closer clusters of similar observations with little variation between them. This indicates that with more detailed observations per day, it is possible that pH and dissolved oxygen may not have any correlation at all. As such, it is also possible that the strong correlation shown in the Baffle 2015 dataset might not exist, if there were more observations per day.

Another possibility is that the pH may have been recorded incorrectly for the Baffle 2019 dataset. In Table 4.12 the minimum and maximum value of pH for the Baffle 2019 dataset is 4.53 and 8.21 respectively. The minimum value of pH is lower than the typically observed pH values which range from 6.5-8.5 and is indicative of highly acidic waters. This supports the notion of incorrect recording of pH values perhaps due to equipment error or possibly an external event which caused the pH values of the river to be so low- yet as none of the values of the other water quality parameters (dissolved oxygen and temperature) seem to be out of

range, the latter notion seems rather unlikely.

Despite the discrepancy in pH values, all three datasets have an average pH of around 7, which falls well within the range of typical pH values and is indicative of neutral river waters.

From Table 4.12 the mean temperature of the water quality datasets is similar with 24.5°C for Burnett, 25.1°C for Baffle 2019 and 24.8°C for Baffle 2015. The mean temperatures fall well within the range of typically observed water temperatures shown in Table 2.1. The mean temperatures fall on the upper end of the typical temperature range, thus indicating that both rivers are found in warm areas. This is true as both rivers are found in southeast Queensland, Australia- a fairly warm area.

The maximum temperature for all the datasets is around 32-33°C and the minimum temperatures for Baffle 2019 and 2015 are 16.8°C and 15.7°C respectively, with the minimum temperature for Burnett at 11.8°C. This could imply that the Baffle river is overall warmer than the Burnett river. Another possibility is that the Burnett dataset is over three years and not just a single year like the Baffle datasets and hence would have a bigger temperature range as it spans over a longer period of time.

The dissolved oxygen values for the Burnett data fall within the range of typical observed dissolved oxygen values i.e., 6 - 14 mg/L. The minimum values of Baffle 2019 and Baffle 2015 (4.0 mg/L and 3.9 mg/L respectively) are out of range. This could be due to the higher temperatures observed in the Baffle river which makes dissolved oxygen concentration lower- the minimum dissolved oxygen values are observed at minimum temperature values and the minimum temperature values for the Baffle datasets are higher than that on the Burnett dataset. The average dissolved oxygen values for all datasets are around 6.5-7 mg/L, which falls within the typical range and also within the range for healthy river waters (6.5-8 mg/L).

4.8 Summary of Main Findings

LSTM models and GA-optimised LSTM models

Overall, the LSTM Burnett model performed better than the LSTM Baffle model. The improvement in performance by the Burnett model after optimisation through the genetic algorithm (GA-Burnett model) is only slight, whilst the improvement in the performance of the Baffle model after optimisation (GA-Baffle model) is very significant. The GA-Burnett model performs better than the GA-Baffle model; in the same way that the Burnett model achieved a better performance than the Baffle model.

It can be surmised that the superior performance of the Burnett model is largely due to the structure of the dataset used to develop the model. The dataset used to create the initial Burnett model is less dense and more spread out, over a period of three years. Whilst the Baffle model was developed on data that is very closely spaced (dense) with the same number of data points as the Burnett dataset, but densely spaced over a period of a single year. The Burnett model benefits from witnessing a trend that occurs over a period of three years and the Baffle model is disadvantaged as it only witnesses the trend in data once.

The GA-optimised models outperform their original LSTM versions, thus emphasising the impact of optimisation on increased model predictive capability and robustness. Generally, from the four models (Burnett, Baffle, GA-Burnett and GA-Baffle models), the GA-Burnett model shows the best predictive capability, and the Baffle model shows the worst performance.

Average and weighted ensemble models

In the average ensemble model, both the GA-Burnett model and the GA-Baffle model contributed equally to the final ensemble model. For the weighted ensemble model, the GA-Burnett model contributes 60% and the GA-Baffle model contributes 40% to the final

ensemble model. The difference in the performance of the average ensemble model and the weighted ensemble model is negligible, with the weighted ensemble model only slightly outperforming the average ensemble model. The similarity in performance is largely due to the weight portions being similar between the weighted ensemble and the average ensemble i.e., the GA-Burnett model only has a 10% increase in power in the weighted ensemble model than it does in the average ensemble model.

Both the ensemble models outperform the individual GA-optimised LSTM models. Thus, highlighting the improved model performance that is resultant from the weighted combination of individual base models. Despite slight differences in model performance, the general performance of all four models (weighted ensemble, average ensemble, GA-Burnett and GA-Baffle models) is quite similar and hence consistent. Overall, the GA-Baffle model performs the worst of all four models, whilst the weighted ensemble performs the best.

Performance on unseen multivariate datasets

A general trend of model performance for all four of the models was observed on the multivariate datasets. The ensemble models outperform the individual GA-optimised LSTM models. The weighted ensemble performs the best. However, the difference in the performance of all four models can be considered unimpressive, and as such great similarity and consistency can be observed between the performance of all four models.

Performance on unseen univariate datasets

The four models were also tested on an unseen univariate dataset to predict the minimum temperature. Here, the GA-Burnett model performs the best with the lowest RMSE value, and the GA-Baffle model fairs the worst with the highest RMSE value.

When the performance of these four models was compared to that of four classical time series forecasting methods (AR, MA, ARMA and ARIMA), it was observed that the methods perform better than the models on the unseen univariate dataset. The ARIMA method achieved the lowest RMSE value from the methods and the models. This illustrates that classical methods might still be the best suited for univariate datasets, when compared to LSTM models, even LSTM-based ensemble models.

Generalisation

The GA-optimised LSTM and ensemble models show very little variation in performance when used on unseen datasets, this shows consistency. The models were more capable of making better predictions on certain datasets than on others. This implies that the performance of all four models would be consistently good on certain datasets and consistently bad on other datasets.

The models seem to perform better on the unseen multivariate datasets, than on the unseen univariate datasets, where classical time series forecasting methods still perform better. The models also perform better on bigger datasets, with a greater number of samples, than on smaller datasets with fewer samples. The structure of the unseen datasets well as the data preparation method, also influence the predictive capability of the models on the datasets. The ensemble models manage to achieve better results than the individual GA-optimised models, even by a small margin, without substantially increasing the computation time.

Chapter 5

Conclusion and future work

5.1 General Conclusion

This research aimed to answer the question of whether the proposed LSTM based ensemble scheme could improve the tolerance (mitigate the discrepancies of the individual LSTM models) of the hybrid genetic algorithm optimised long short-term memory (GA-optimised LSTM) water quality prediction model, for different water quality datasets taken from different sites and/or different times.

In order to achieve this aim, this study has made three main research contributions. The first being the development of two LSTM models for two different time sequential water quality datasets taken from two different sites and from two different time periods, each with a differently spaced/grouped data structure. Both the LSTM models, the Burnett and Baffle models, aimed to predict dissolved oxygen concentrations ahead of time, using historical dissolved oxygen concentrations and related water temperature values.

This was followed by the optimisation of both models through the use of the genetic algorithm, to increase the efficiency and robustness of each model. This resulted in the development of the two hybrid GA-optimised LSTM models, the GA-Burnett and GA-Baffle

models. The GA-Burnett and GA-Baffle models both outperform their original counterparts. This is evidence of the improved performance capacity and robustness of the original models through GA-optimisation.

The second contribution focused on the development of a final, more tolerant and efficient ensemble model from the weight based combination of the two individual hybrid GA-optimised LSTM models. This was achieved through the application of a linear weight based technique that combined the GA-Burnett and the GA-Baffle models, to create an ensemble model.

Two ensemble models, with different weight contributions of each of the individual GA-optimised LSTM models, were created. Both the models, the weighted ensemble and the average ensemble models, outperform the individual GA-optimised LSTM models, even if it is sometimes only by a slight margin, proving that the ensemble model is more robust and has a greater predictive capability than the individual LSTM base models.

The third and final contribution of this study involved testing model tolerance and generalisation. Thus, the use of the four models (GA-Burnett, GA-Baffle, weighted ensemble and average ensemble models) on unseen multivariate and univariate datasets that were unrelated to water quality, was explored and assessed. The performance of the four models was consistently similar to one another on each dataset. They were all good on a certain dataset or all bad, but the performance difference between the models was negligible.

This consistency of the performance exhibited by all four different models, which were developed from water quality datasets taken from different rivers and different time periods, on any particular dataset is evidence of the mitigation of the discrepancies of the individual LSTM models and hence the achievement of greater tolerance of these individual models through the employment of the GA-optimised LSTM based ensemble scheme. Thus, asserting the relevance and tolerance of the developed models, especially the weighted ensemble model in the wider field of LSTM and ensemble prediction models.

5.2 Specific conclusions

Through the course of the research, certain observations were made about the performance of each model and the possible reasons for model behaviour. These observations and the suggested causes of each observation, form a big part of this study. The most important findings are presented below.

5.2.1 Water quality prediction

Water is a critical natural resource that is currently under threat, especially rivers. The models were able to successfully predict the quality of water ahead of time, in terms of dissolved oxygen concentration, for both the Burnett and Baffle rivers. The low RMSE, MSE, MAE and MAPE values, along with the high R^2 and explained variance scores achieved by the models for the prediction of dissolved oxygen is proof of the effective and efficient prediction of water quality by the models. Water quality prediction aids in increasing the efficiency of water quality monitoring. Efficient water quality monitoring enables effective water management and effective water management is necessary for the preservation of rivers.

5.2.2 Predictive LSTM models and data structure

Generally, the LSTM model developed from the Burnett dataset performed better than the LSTM model developed from the Baffle dataset. This is mainly due to the difference in the structure of the datasets. The Burnett dataset has 52 560 observations spread over 3 years and the Baffle dataset is more dense with 52 560 observations spread over a single year. The Burnett dataset enables the Burnett model to observe a trend over three years with greater variation in data, whilst the Baffle dataset only allows the trend to be observed once by the Baffle model with a single year of densely spread data with little variation.

5.2.3 Optimisation of LSTM models

Both the GA-optimised LSTM models perform better than their original LSTM model versions, showing an enhancement in model performance through hyperparameter tuning.

The improvement in performance by the Burnett model after optimisation through the genetic algorithm (GA-Burnett model) is only slight; whilst the improvement in the performance from the Baffle model to the GA-Baffle model is greater. The Burnett model was already well optimised through the trial-and-error method, unlike the Baffle model and thus the improvement in the performance of the Burnett model after optimisation through the genetic algorithm is only slight.

This implies that the trial-and-error method can successfully optimise models that are based on datasets that are spread out over a greater time period and can thus exhibit repeated trends (such as the Burnett dataset) that are easier to learn from. When datasets are more densely spaced with little variation in observations over a shorter period of time and thus cannot show a repeated trend (such as the Baffle 2019 dataset), then a more powerful optimisation technique might be required, such as the metaheuristic genetic algorithm.

From the four models (the Burnett, Baffle, GA-Burnett and GA-Baffle models), the GA-Burnett model shows the best predictive capability, and the Baffle model shows the worst predictive ability. The Baffle model had the least fine-tuned (not optimal) hyperparameters and hence performed the worst.

The optimisation of the individual LSTM Burnett and Baffle models through the genetic algorithm had the biggest impact on decreasing computation time and trainable parameters and thus increasing model robustness than any other process in the ensemble model development. The greatest impact was the reduction of 28 456 trainable parameters and 50 minutes of computation time of the Baffle model to form the GA-Baffle model. This optimisation reduced the overall computation time for the process on the final ensemble

model. This proves that the enhancement of the individual base models, through hyperparameter optimisation can enhance the overall quality of the ensemble by reducing computation time and the number of trainable parameters. This highlights the use of base model optimisation as a crucial and almost necessary aspect of the development of an ensemble model.

Before the Burnett and Baffle models were optimised through the genetic algorithm, the performance of the models was not similar. After the models were optimised, the performance of the models became comparable.

5.2.4 Average and weighted ensemble models

For the average ensemble model, both the GA-Burnett model and the GA-Baffle model contributed equally to the ensemble model. For the weighted ensemble model, the GA-Burnett model only had a 10% greater contribution than the GA-Baffle model. The difference in the performance of the average ensemble model and the weighted ensemble model is negligible, with the weighted ensemble model only slightly outperforming the average ensemble model. This is largely due to the weight portions being similar between the weighted ensemble (60% and 40%) and the average ensemble (50% and 50%).

The optimal weight portions are so similar to the equal weight portions as the two base GA-optimised LSTM models are very similar in structure. The only difference is the number of LSTM units in the two hidden layers. The similarity in the model architecture of the GA-Burnett and GA-Baffle models led to similar model performance. The combination of two similar models, of similar performance capacity, can only lead to the development of a final ensemble model, in which the two models have an equal or near to equal contribution.

The weighted ensemble model (the more powerful ensemble model), has a greater contribution from the GA-Burnett model as the GA-Burnett model has a greater performance capability

than the GA-Baffle model.

The combination of the two GA-optimised models did improve the model performance of the final ensemble model, even if it was only by a small margin in certain cases and without significantly increasing the computation time. The performance of all four models (weighted ensemble, average ensemble, GA-Burnett and GA-Baffle models) cannot be significantly different from one another as the two LSTM base models are similar to each other and the resultant weighted and average ensemble models, have the similar base LSTM models combined in almost the same weight portions.

5.2.5 Model performance on unseen datasets

Model performance on unseen multivariate datasets

In general, the performance of all four models (weighted ensemble, average ensemble, GA-Burnett and GA-Baffle models) on the unseen multivariate datasets is quite similar. On the multivariate datasets, the ensemble models outperform the individual GA-optimised LSTM models; with the weighted ensemble performing the best. The difference in the performance of all four models can be considered unimpressive and as such great similarity and consistency can be observed between the performance of all four models.

The similar and consistent performance of all the models on unseen and unrelated datasets, by models that were developed from different water quality datasets taken from different rivers and different time periods, indicates that this study has managed to mitigate the discrepancies of the individual LSTM models, to a large extent.

The models performed poorly on the wind power dataset, largely due to differences in data structure, data preparation and possibly dataset size, to the datasets that were used to train and develop the models. This dataset has the lowest number of samples from all the unseen multivariate datasets. The models exhibit great consistency, but low tolerance on the unseen

wind power dataset.

The models exhibited great predictive capability on the air temperature dataset, this is largely due to the similarities that are present between this dataset and the datasets used to make develop the models, in terms of data structure, data preparation and possibly dataset size. This dataset has the highest number of samples from the unseen multivariate datasets, indicating that models might be better suited to bigger datasets. The models exhibit great consistency and high tolerance on the unseen air temperature dataset.

The models performed averagely on the pollution dataset, achieving results that were neither good nor bad. The models exhibit great consistency and moderate tolerance on this dataset. The models tend to perform moderately or poorly on smaller datasets.

The performance of the models and hence the predictive capability of all four models are rather similar and consistent, with the weighted ensemble model only marginally outperforming the other models. This indicates that on certain datasets, all the models will have consistent and good predictive capabilities and that they will have similar moderate to poor predictive capacities on other datasets. The models tend to perform better on bigger datasets than on smaller datasets. They also tend to perform better on datasets that have multiple features which are related to one another in a single system and in which the data can be prepared i.e., normalised in the same manner in which the data was normalised in the development of each of these models. In conclusion, these models perform well on datasets that are similar in structure, preparation and size to the datasets which were used to develop each of these models.

Model performance and classical time series forecasting methods on unseen univariate dataset

The four models were also tested on an unseen univariate dataset. The GA-Burnett model performs the best and the GA-Baffle model fairs the worst. The difference in model performance

is more significant on the univariate dataset than on the multivariate datasets. The discrepancies in individual model performance become more obvious on univariate datasets.

In general, the classical forecasting methods perform better than the models on the unseen univariate dataset. This illustrates that classical methods might still be the best suited for making predictions on univariate datasets, when compared to LSTM models, even LSTM based ensemble models. Also, smaller LSTM networks might be better suited to univariate datasets than bigger LSTM networks.

Generally, the models tend to exhibit great consistency in model performance on the multivariate datasets than the univariate datasets. This could be due to all the models having been developed on multivariate datasets.

5.3 Future work

5.3.1 The use of more than two LSTM base models

Various improvements can be made to this study. One such improvement would be the use of more than two LSTM base models to develop the ensemble model- such as a study into the optimum number of LSTM base models that can be used to make the most robust and tolerant ensemble model for water prediction and possibly other areas such as energy, finance, geology etc.

5.3.2 The use of more diverse water quality datasets

The three water quality datasets used to develop the models were from the Burnett and Baffle rivers. Although, the datasets were taken from different sites and different times. They still shared certain similarities. Both rivers are situated in southeast Queensland, Australia and both rivers eventually flow into the Coral Sea of the South Pacific Ocean. Future work should

consider developing models from water quality datasets that are not situated in a similar region- this might add more diversity to the datasets and perhaps increase the tolerance of the final ensemble model.

5.3.3 The use of more diverse LSTM base models

The two LSTM models used in this study were diverse in terms of having different time window sizes and a different number of LSTM units in the two hidden layers (with the number of units in the first hidden layer always being greater than the number of units in the second hidden layer). However, more diversity might be required to create a more tolerant final model.
Both the GA-optimised LSTM base models used to develop the ensemble models were similar in architecture, the only difference between the two models was the number of LSTM units in the two hidden layers. A future study could explore using different LSTM models. Perhaps, LSTM models that each have a different number of hidden layers, a different number of input parameters, different activation functions and optimisers. Further diversity in LSTM base models could result in a more tolerant and robust model.

5.3.4 The use of other optimisation algorithms

Further work can explore the optimisation of the LSTM network by different metaheuristic algorithms, such as particle swarm optimisation etc., to gauge if the use of different optimisation algorithms could affect the overall robustness and tolerance of the final ensemble model.

5.3.5 Greater emphasis on water temperature for predictive model development

This study focused on the prediction of dissolved oxygen levels as this water quality parameter is often used to gauge the overall health of a river. The research in this study has shown that

water temperature has a great influence over most of the other water quality parameters and hence future work should focus on developing a water temperature prediction model along with a dissolved oxygen prediction model.

5.3.6 Further exploration of the relationship between dissolved oxygen and pH

This study has mentioned the correlation between dissolved oxygen and pH and how this correlation might not necessarily translate into causation. Future research could be done into a possible link between pH and dissolved oxygen and whether they share a causal relationship and if it can be used in the development of water quality prediction models.

Bibliography

[1] Q. Ye, X. Yang, C. Chen, and J. Wang, "River water quality parameters prediction method based on lstm-rnn model," in *2019 Chinese Control And Decision Conference (CCDC), Nanchang, China*, pp. 3024–3028, 2019.

[2] M. Aslam, J. Lee, H. Kim, S. Lee, and S. Hong, "Deep learning models for long-term solar radiation forecasting considering microgrid installation: A comparative study," *Energies*, vol. 13, no. 147, pp. 1–15, 2020.

[3] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Multi-sequence lstm-rnn deep learning and metaheuristics for electric load forecasting," *Energies*, vol. 13, pp. 391– 412, 2020.

[4] Bonzle Digital Atlas of Australia, "Map of burnett river, qld." [Online] `http://www.bonzle.com/c/a?a=p&p=210993&cmd=sp`, [Accessed: 2 January 2020].

[5] Bonzle Digital Atlas of Australia, "Map of baffle creek, qld." [Online] `http://www.bonzle.com/c/a?a=p&p=210011&cmd=sp`, [Accessed: 20 January 2020].

[6] H. Chung and K. shik Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustainability, MDPI, Open Access Journal*, vol. 10, no. 10, pp. 1–18, 2018.

[7] H. Razmkhah, A. Abrishamchi, and A. Torkian, "Evaluation of spatial and temporal variation in water quality by pattern recognition techniques: A case study on jajrood river (tehran, iran)," *Journal of Environmental Management*, no. 91, pp. 852–860, 2010.

[8] U. Ahmed, R. Mumtaz, H. Anwar, A. A. Shah, R. Irfan, and J. García-Nieto, "Efficient water quality prediction using supervised machine learning," *Water*, no. 11, pp. 1–14, 2019.

[9] Y. Khan and C. S. See, "Predicting and analyzing water quality using machine learning: A comprehensive model," in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA*, no. 11, pp. 1–6, IEEE, 2016.

[10] A. Najah, A. Elshafie, O. A. Karim, and O. Jaffar, "Prediction of johor river water quality parameters using artificial neural networks," *European Journal of Scientific Research*, vol. 28, no. 3, pp. 422–435, 2009.

[11] M. J. Diamantopoulou, D. M.Papamichail, and V. Z. Antonopoulos, "The use of a neural network technique for the prediction of water quality parameters," *Operational Research. An International Journal*, vol. 5, no. 1, pp. 115–125, 2005.

[12] S. Areerachakul and S. Sanguansintukul, "Water quality classification using neural networks: Case study of canals in bangkok, thailand," in *2009 International Conference for Internet Technology and Secured Transactions, (ICITST), London*, pp. 1–5, IEEE, 2009.

[13] L. Y. Khuan, N. Hamzah, and R. Jailan, "Prediction of water quality index (wqi) based on artificial neural network (ann)," in *Student Conference on Research and Development Proceedings, Shah Alam, Malaysla*, pp. 157–161, IEEE, 2002.

[14] C. Zhou, L. Gao, H. Gao, and C. Peng, "Pattern classification and prediction of water quality by neural network with particle swarm optimization," in *2006 6th World Congress on Intelligent Control and Automation, Dalian*, pp. 2864–2868, IEEE, 2006.

[15] S. Abbaa, S. J. Hadia, and J. Abdullahia, "River water modelling prediction using multi-linear regression, artificial neural network, and adaptive neuro-fuzzy inference system techniques," in *9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW, Budapest, Hungary*, pp. 75–82, Elsevier, 22-23 August 2017.

[16] P. Liu, J. Wang, A. K. Sangaiah, Y. Xie, and X. Yin, "Analysis and prediction of water quality using lstm deep neural networks in iot environment," *Sustainability*, vol. 11, pp. 2058– 2072, 2019.

[17] Z. Hu, Y. Zhang, Y. Zhao, M. Xie, J. Zhong, Z. Tu, and J. Liu, "A water quality prediction method based on the deep lstm network considering correlation in smart mariculture," *Sensors*, vol. 19, pp. 1420 – 1440, 2019.

[18] J. Zhou, Y. Wang, F. Xiao, Y. Wang, and L. Sun, "Water quality prediction method based on igra and lstm," *Water*, vol. 10, pp. 1148–1159, 2018.

[19] Y. Wang, J. Zhou, K. Chen, Y. Wang, and L. Liu, "Water quality prediction method based on lstm neural network," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Nanjing*, pp. 1–5, IEEE, 2017.

[20] V. V. Prasad D, L. Y Venkataramana, P. S. Kumar, Prasannamedha G, Soumya K, and Poornema A.J, "Water quality analysis in a lake using deep learning methodology: prediction and validation," *International Journal of Environmental Analytical Chemistry*, vol. 3, pp. 1–10, 2020.

[21] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, "Long short term memory hyperparameter optimization for a neural network based emotion recognition framework," *IEEE Access*, vol. 6, pp. 49325 – 49337, September 28, 2018.

[22] A. Fraser, "Simulation of genetic systems by automatic digital computers introduction," *Australian Journal of Biological Sciences*, vol. 10, p. 484–491, 1957.

[23] J. H. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[24] D. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," in *International Joint Conference on Artificial Intelligence (IJCAI), Detroit*, vol. 1, pp. 762–767, 20-26 August 1989.

[25] K. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert System Application*, vol. 19, p. 125–132, 2000.

[26] M. Kim, S. Min, and I. Han, "An evolutionary approach to the combination of multiple classifiers to predict a stock price index," *Expert System Application*, vol. 31, p. 241–247, 2006.

[27] R. Mahajan and G. Kaur, "Neural networks using genetic algorithms," *International Journal of Computer Applications*, vol. 14, no. 77, pp. 6–11, 2013.

[28] B. ul Islam, Z. Baharudin, M. Q. Raza, and P. Nallagownden, "Optimization of neural network architecture using genetic algorithm for load forecasting," in *Proceedings of the 2014 5th IEEE International Conference Intelligent and Advanced Systems (ICIAS), Kuala Lumpur, Malaysia*, p. 1–6, IEEE, 3–5 June 2014.

[29] S. B. Defilippo, G. G. Neto, and H. S. Hippert, "Short-term load forecasting by artificial neural networks specified by genetic algorithms–a simulation study over a brazilian dataset," in *Proceedings of the XIII Simposio Argentino de Investigación Operativa (SIO)—JAIIO 44, Rosario, Santa Fe, Argentina*, p. 1–6, 31 August–1 September 2015.

[30] A. S. Santra and J.-L. Lin, "Integrating long short-term memory and genetic algorithm for short-term load forecasting," *Energies*, vol. 12, no. 11, p. 1–6, 2019.

[31] N. Gorgolis, I. Hatzilygeroudis, Z. Istenes, and L. Gyenne, "Hyperparameter optimization of lstm network models through genetic algorithm," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), PATRAS, Greece*, pp. 1–4, IEEE, 15-17 July 2019.

[32] Hendri, R. N. Sari, and A. Wibowo, "Timeseries forecasting using long short-term memory optimized by multi heuristics algorithm," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 11492–11500, November 2019.

[33] A. P. Engelbrecht, *Computational Intelligence: An Introduction, 2nd Edition.* John Wiley & Sons Ltd, 2007.

[34] D. Dheda and L. Cheng, "A multivariate water quality parameter prediction model using recurrent neural network," 2020.

[35] Y. LeCun, Y. Bengio, and G. Hinton, "Review: Deep learning," *Nature*, vol. 521, pp. 436–444, 280 May 2015.

[36] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *32nd International Conference on Machine Learning, Lille, France*, pp. 2342–2350, 7-9 July 2015.

[37] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena, Special Issue on Machine Learning and Dynamical Systems*, vol. 404, pp. 1–43, February 2020.

[38] S. Hochreiter and J. Schmidhuber, "Long short term memory," *Neural Computation*, vol. 9, no. 8, p. 1735–1780, 1997.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning series)*. Cambridge, 2016.

[40] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, pp. 1636 – 1656, 2018.

[41] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series*. Chapman and Hall/CRC, 1 ed., 6 June 2012.

[42] L. Rokach, *Pattern Classification Using Ensemble Methods (Series in Machine Perception and Artificial Intelligence)*. World Scientific Publishing Company, 2 ed., 30 November 2009.

[43] P. Sollich and A. Krogh, "Learning with ensembles: How over-fitting can be useful," in *Advances in Neural Information Processing Systems*, pp. 190 – 196, 27-30 November 1995.

[44] J. Y. Choi and B. Lee, "Combining lstm network ensemble via adaptive weighting for improved time series forecasting," *Hindawi, Mathematical Problems in Engineering*, vol. 2018, pp. 1–8, 5 August 2018.

[45] L. Wolf and A. Shashua, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *Journal of Machine Learning Research*, vol. 6, p. 1855–1887, 2005.

[46] R. Adhikari and R. K. Agrawal, "Combining multiple time series models through a robust weighted mechanism," in *2012 1st International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India*, pp. 455–460, 15-17 March 2012.

[47] S. Krstanovic and H. Paulheim, "Ensembles of recurrent neural networks for robust time series forecasting," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 1–14, November 2017.

[48] A. Tikkanen, "Burnett river, encyclopdia britannica." [Online] https://www.data.qld.gov.au/dataset/ambient-estuarine-water-qualitymonitoring-data-near-real-time-sites-2012-to-present-day, [Accessed: 20 January 2020].

[49] Department of Environment and Science, Queensland, "Upper burnett river drainage sub-basin — facts and maps." [Online] https://wetlandinfo.des.qld.gov.au/wetlands/facts-maps/sub-basin-upper-burnett-river/, [Accessed: 4 January 2020].

[50] Business and Industry, Queensland Government, "Baffle creek - watercourse." [Online] `https://www.resources.qld.gov.au/qld/environment/land/place-names/search#/search=Baffle_Creek&types=0&place=Baffle_Creek1219`, [Accessed:6 January 2020].

[51] Department of Environment and Science, Queensland, "Baffle drainage basin." Online: https://wetlandinfo.des.qld.gov.au/wetlands/facts-maps/basin-baffle/, [Accessed: 2 February 2020].

[52] E. Olyaie, H. Z. Abyaneh, and A. D. Mehr, "A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in delaware river," *Geoscience Frontiers*, vol. 8, pp. 517–527, 2017.

[53] Fondriest Environmental, Inc., "Dissolved oxygen. fundamentals of environmental measurements.." [Online] https://www.fondriest.com/environmental-measurements/parameters/water-quality/dissolved-oxygen/, [Accessed:2 November 2020].

[54] D. W. Connell and G. J. Miller, *Chemistry and Ecotoxicology of Pollution.* John Wiley & Sons, Inc., 3 ed., 20 March 1984.

[55] Environment and Natural Resources, "Dissolved oxygen (do)." [Online] `https://www.enr.gov.nt.ca/sites/enr/files/dissolved_oxygen.pdf`, [Accessed:2 February 2020].

[56] Fondriest Environmental, Inc., "Water temperature. fundamentals of environmental measurements." [Online] https://www.fondriest.com/environmental-measurements/parameters/water-quality/water-temperature/, [Accessed: 2 February 2020].

[57] B. Oram, "Water research center: The ph of water." [Online] https://water-research.net/index.php/ph, [Accessed:13 March 2020].

[58] United States Environmental Protection Agency, "Water: Monitoring & assessment: Conductivity." [Online] https://archive.epa.gov/water/archive/web/html/vms59.html, [Accessed: 13 January 2020].

[59] Minnesota Pollution Control Agency, "Turbidity: Description, impact on water quality, sources, measures - a general overview." [Online] `https://www.pca.state.mn.us/sites/default/files/wq-iw3-21.pdf`, [Accessed:10 January 2020].

[60] United States Geological Survey (USGS), "Water science school: Turbidity and water." [Online] https://www.usgs.gov/special-topic/water-science-school/science/turbidity-and-water, [Accessed:10 March 2020].

[61] United States Environmental Protection Agency, "Water: Monitoring & assessment: Turbidity." [Online] https://archive.epa.gov/water/archive/web/html/vms55.html, [Accessed:13 January 2020].

[62] The State of Queensland, "Ambient estuarine water quality monitoring data (includes near real-time sites)- 2012 to present day." [Online] https://www.data.qld.gov.au/dataset/ambient-estuarine-water-qualitymonitoring-data-near-real-time-sites-2012-to-present-day, [Accessed: 2 January 2020].

[63] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3 ed., 22 June 2011.

[64] J. Brownlee, *Statistical Methods for Machine Learning*. Cambridge, 2015.

[65] C. Lesmeister, *Mastering Machine Learning with R to deliver insights for complex projects*. Packt Publishing Ltd, 2 ed., 28 October 2015.

[66] A. Agresti and B. Finlay, *Statistical Methods for the Social Sciences*. Prentice Hall, 2 ed., 1997.

[67] J. Cowls and R. Schroeder, "Causation, correlation, and big data in social science research," *P & I Policy & Internet*, vol. 7, pp. 447–472, 30 August 2015.

[68] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, and I. Goodfellow, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *Preliminary White Paper*, pp. 1–19, 9 November 2015.

[69] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *14th International Conference on Artificial Intelligence and Statistics*, p. 315–323, 2011.

[70] S. Ruder, *An overview of gradient descent optimization algorithms*. Aylien Ltd, 2017.

[71] F.-A. Fortin, F.-M. De Rainville, M. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, July 2012.

[72] J. Brownlee, *Better Deep Learning Train Faster, Reduce Overfitting, and Make Better Predictions*. Cambridge, 2018.

[73] H. Pishro-Nik, "Introduction to probability. statistics and random processes." [Online] https://www.probabilitycourse.com/chapter9, [Accessed:2 January 2021].

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

[75] P. Swamidass, "Mean absolute percentage error (mape)," *Encyclopedia of Production and Manufacturing Management*, p. 30, 2000.

[76] R. J.Hyndman and A. B.Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, pp. 679–688, October–December 2006.

[77] R. G. Pontius, O. Thontteh, and H. Chen, "Components of information for multiple resolution comparison between maps that share a real variable," *Environmental and Ecological Statistics*, vol. 15, p. 111–142, 2008.

[78] S. Kotz, N. L. Johnson, and C. B. Read, *Encyclopedia of Statistical Sciences*, vol. 9. Wiley-Interscience, 1 ed., 12 May 1988.

[79] Kaggle, "Global energy forecasting competition 2012 - wind forecasting." [Online] https://www.kaggle.com/c/GEF2012-wind-forecasting/data, [Accessed:8 October 2020].

[80] Stytch and Kaggle, "Jena climate 2009-2016." [Online] https://www.kaggle.com/stytch16/jena-climate-2009-2016, [Accessed: 1 December 2020].

[81] D. Havera and Kaggle, "Beijing pm25 data." [Online] https://www.kaggle.com/djhavera/beijing-pm25-data-data-set, [Accessed:10 December 2020].

[82] M. Siddhartha and Kaggle, "Beijing multi-site air-quality data set." [Online] https://www.kaggle.com/sid321axn/beijing-multisite-airquality-data-set/metadata, [Accessed:05 June 2020].

[83] P. Brabban and Kaggle, "Daily minimum temperatures in melbourne." [Online] https://www.kaggle.com/paulbrabban/daily-minimum-temperatures-in-melbourne, [Accessed: 5 December 2020].

[84] Tensorflow, "Models and layers." [Online] `https://www.tensorflow.org/js/guide/models_and_layers#model_summary`, [Accessed: 5 November 2019].