Predicting Reservoir Water Levels using Deep Learning

Ahmed Nuzhi Meyen

168247G

MSc in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

May 2018

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: …… Date: …….

Name: A. Nuzhi Meyen

The supervisor/s should certify the thesis/dissertation with the following declaration.

I certify that the declaration above by the candidate is true to the best of my knowledge and that this report is acceptable for evaluation for the MSc Thesis.

Signature of the supervisor: ……. Date: ……..

Name: Dr. A. Shehan Perera

Acknowledgements

My sincere appreciation goes to my family for the continuous support and motivation given to make this thesis a success. I also express my heartfelt gratitude to Dr. Amal Shehan Perera, my supervisor, for the supervision and advice given throughout to make this research a success. I also thank my parents, sister and brother-in-law for their heartfelt support. I am also thankful to Mr. Sanjaya Ratnayake for providing the required data. Last but not least I also thank my friends who supported me in this whole effort.

ABSTRACT

Currently the authorities in the field of water resource management for irrigation and hydro power electricity in Sri Lanka make use of basic forecasting methodology in order to make decisions with respect to water resource management. The results of this research would be useful to the relevant authorities as it would provide them an indication of the excepted water levels allowing them to make vital decisions regarding the competing needs such as water resource management for irrigation as opposed to water resource management for hydro power electricity generation during the monsoonal as well as inter-monsoonal periods with the use of the latest predictive framework with respect to artificial intelligence technology.

Most of the current research in this area use models such as multi-linear regression, support vector machines and artificial neural networks such as adaptive neuro-fuzzy inference systems to provide predictions for hydrological models. The models are developed for varying levels of granularity with respect to time such as daily and weekly depending on the need to forecast water levels for reservoirs.

This research will focus on the novel deep learning techniques of LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) recurrent neural networks as opposed to the conventional machine learning approaches. Historical daily water levels will be used as inputs along with meteorological variables at other nearby reservoirs to do forecast future values. These methods will be benchmarked against traditional baseline machine learning techniques to validate how much of a predictive gain can be obtained by use of the deep learning techniques. Furthermore, this research will evaluate the suitability of the aforementioned techniques to make predictions regarding the water level input by the usage of various metrics such as mean square error as the cost function which can be used to validate the output of the above models.

Contents

List of Figures

List of Abbreviations

Abbreviation	Description
ARMA	Auto Regressive Moving Average
MLR	Multi Linear Regression
ANN	Artificial Neural Network
AR	Auto Regressive
MLP	Multi-Layer Perceptron
MA	Moving Average
MSE	Mean Square Error
MAE	Mean Absolute Error
RBF	Radial Basis Function
ANFIS	Adaptive Neuro-Fuzzy Inference System
SVM	Support Vector Machine
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory

Chapter 1

1. Introduction

1.1 Kotmale Reservoir

The Kotmale hydro power project in Sri Lanka was aimed at utilizing the country's major river, the Mahaweli Ganga, for power generation and irrigation. The Kotmale project includes a 90 meters high rock filled dam across the Kotmale Oya, which is a tributary of the Mahweli Ganga¸ a six kilometers long reservoir holding 175 million cubic meters of water and an underground power station to which water is fed through a seven kilometers long tunnel [1]. The Kotmale hydropower project was an integral part of the large scale Accelerated Mahaweli Programme, which was in turn based on the Mahaweli development programme. The Mahaweli programme was a multi-purpose river development programme which included energy production, storage reservoirs, canals and irrigation projects as well as newsettlements. To reduce the completion time from implementation, in 1977, the Sri Lankan Government decided to concentrate on a few major reservoir projects under the Accelerated Mahaweli programme. With regards to the Kotmale project a full feasibility study was conducted from 1973 to 1976 by the Water Power Development Consultancy Services of India, and this provided the basis for the project [1].The initial plan was to construct a reservoir 2 km wide 10 km long with a storage capacity of 174 cubic million meters. The water impounded by the reservoir is conducted through a 7-km long headrace tunnel, down a sharply inclined high-pressure shaft and on to an underground tunnel, the first of its kind in Sri Lanka. [1]. It has a total installed capacity of 201 MW courtesy of three Francis turbines. After power generation, the water is discharged through a 645 meters long tailrace tunnel to the outfall at the Mahaweli Ganga at the Atabaga Oya confluence. In addition to the generation of power, the main purpose of the project, the regulated water will also increase the supply for water for irrigation purposes through the Polgolla tunnel and also improve power output at the Ukuwela and Bowatenne power stations (Figure 1.1). [1]

The Kotmale Oya is one of the seven main tributaries of the Mahaweli Ganga. The headwaters of the Mahaweli Ganga rise in the core of the Central Highlands at Nuwara Eliya with the river originating from in the Hatton plateau which is situated southwest to the Central Highlands with elevations around 1400 meters. The river

flows north from this plateau until it curves around the city of Kandy in the Kandyan plateau at an elevation of 500 meters. From this plateau, it flows east towards the dry zone lowlands, crossing the wet zone boundary at an elevation of 100 meters. Afterwards the river turns north again and reaches the sea near Trincomalee about 180 km north-north east of its origin. (Figure 1.3). Approximately 4 km upstream from the confluence of the Mahaweli Ganga and the Kotmale Oya, the latter has been impounded forming the Kotmale reservoir which has a catchment of about 554 square km upstream from the dam site (Figure 1.1) [1].

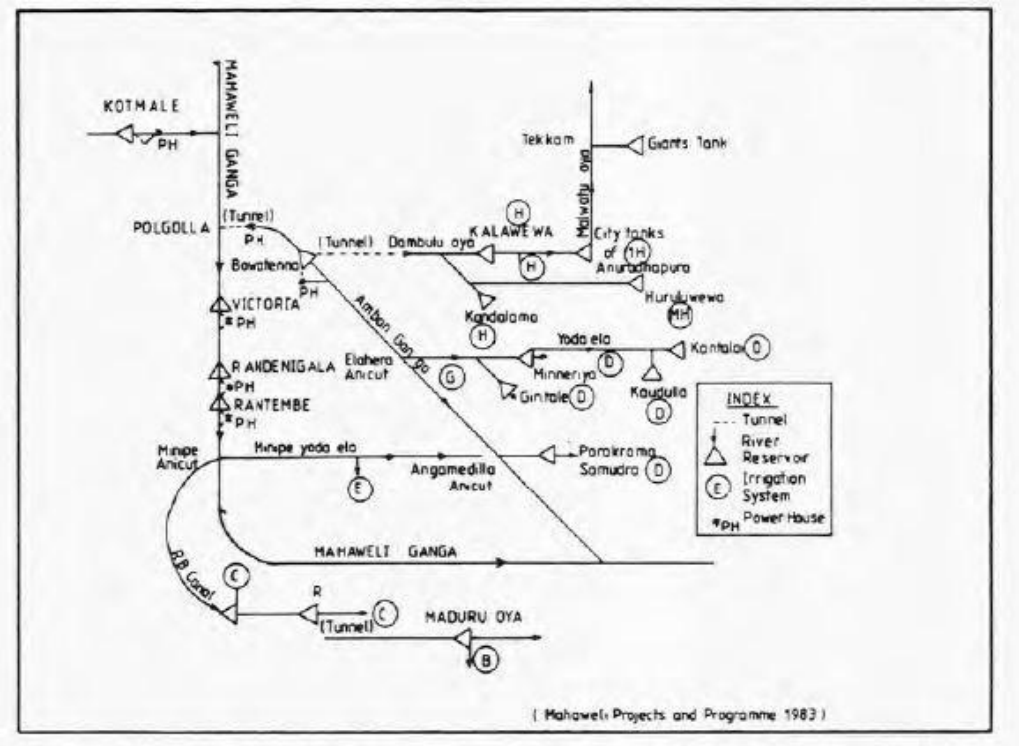

Figure 0.1: Water Distribution Diagram of the Accelerated Mahaweli Program

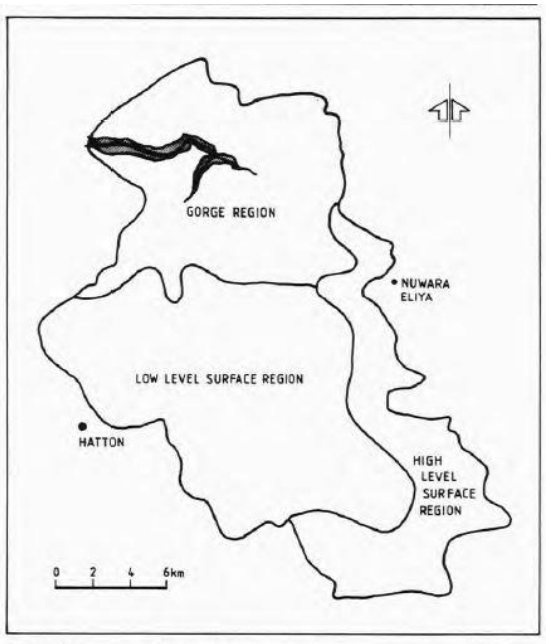

Figure 1.2: The three physiographic regions of the catchment above the Kotmale Reservoir

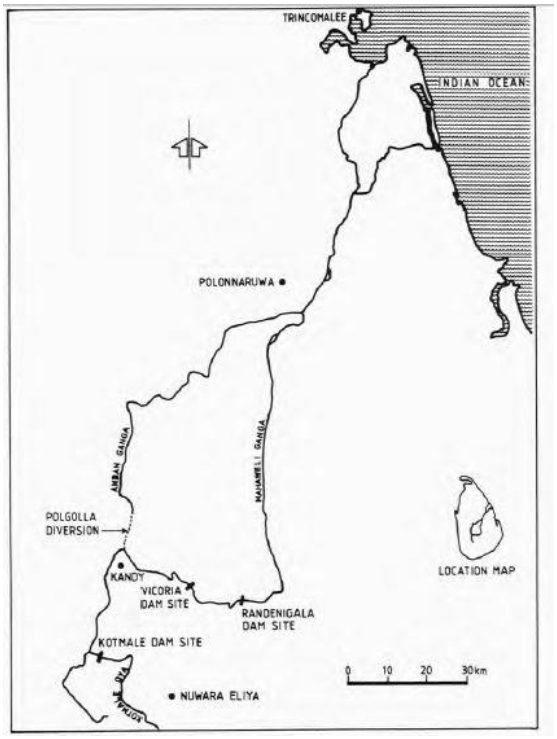

Figure 1.3: Mahaweli Ganga, Kotmale Oya and the location of some major reservoirs

When the catchment area of the Kotmale Reservoir is considered, it includes:

- A high-level surface, which covers about 20 percent of the area. This surface lies at around an altitude of about 2,200 -2,300 meters, consisting of low undulating hills covered by dense montane forests.
- A low-level surface at an altitude of about 1,400 meters. This surface of low undulating hills is used almost exclusively for tea cultivation and covers about 40 percent of the catchment area.
- A gorge region of steep slopes in an area of high relief where the three main tributaries, the Kotmale, the Pannu and Pundula Oya, flow at the bottom of the gorge region. Land use is complex in this region, with paddy on the mudslide aprons in the valley floors, home gardens and occasionally patches of semi-natural forest on the boulder tongues and steeper slopes and tea cultivations on the higher areas. The gorge region covers about 40 percent of the catchment area. [1].

When considering climatic conditions, it can be seen that the lowest part of the catchment lies at about 700 meters while highest point, the Totupolakanda Peak rises to 2,380 meters. The large difference in altitude are naturally reflected in the climatic parameters and a large variation of, for instance, temperature and rainfall may occur even for localities geographically closer to each other.

The Kotmale catchment experiences a monsoonal distribution of rainfall during the Southwest monsoon or Yala, from May to September. The next wettest months are usually the intermonsoon periods, October, November and March and April. The Northeast monsoon, or Maha, during December and January produces the least rainfall and the month of February is usually the driest of the year. However, large seasonal variations may occur from year to year [1]. Furthermore, the large seasonal variations in rainfall are strongly reflected in the inflow to the Kotmale reservoir since there are no lakes or swamps that retard the water flow whilst a large part of the catchment area is covered by tea plantations with low- water retaining capacity [1].

The flooding of the narrow valley has resulted in an elongated shape of the reservoir, such that the main part of the reservoir is oriented in an east-west direction where at the inflow of the Kotmale river the direction is North to South. The length of the East-West part is about 8.5 km and the Norht-South part about 3 km at full supply level. The width, which is about 1 km near the dam site, slowly decreases towards the inflow at Pannu (Puna) and Kotmale rivers. At full supply level, the surface area is about 6.15 sq km. The greatest depth, about 75 meters, at full supply level is found at the dam site [1].

The relationships between water levels and storage volumes is shown in Figure 1.4 whilst the relationships between water levels and surface area are given in Figure 1.5.

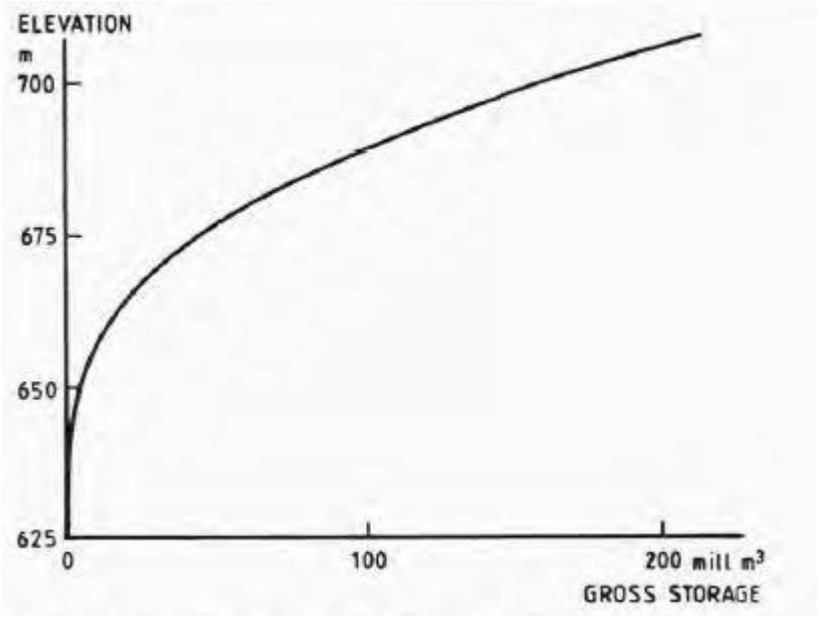

Figure 1.4: Storage of the Kotmale Reservoir at various water levels

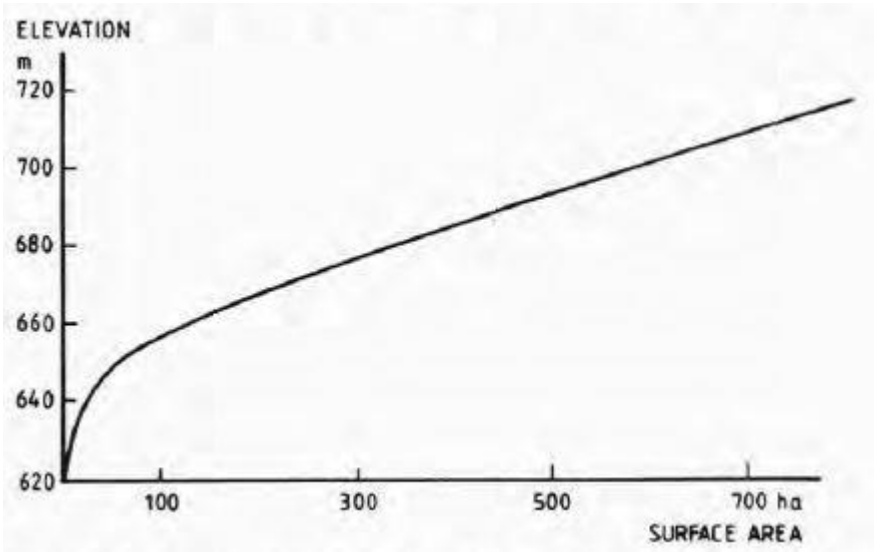

Figure 1.5: Surface area (ha) of the Kotmale Reservoir at various water levels

1.2 Problem

Reservoir water level prediction is important since reservoir input level fluctuations play an important role in the planning, designing and operating of the reservoir. Furthermore, reservoir water input level forecasting across time is an important issue in water resources planning. It should be noted that water resources planning includes the need for water resources to be utilized for a variety of competing needs from irrigation and farming to hydro-electric power generation. The variation in reservoir water levels can be attributed due to the complex outcomes of many environmental factors such as precipitations, direct and indirect runoffs from neighbor catchments, evaporation from free water body, air and water temperature, and interactions between the reservoir and low lying aquifers [1].The main research problem intended to be studied by this research is the prediction of reservoir water level for the Kotmale Reservoir utilizing some of the hydrological parameters mentioned above as input variables.

In order to develop a viable solution for the identified research problem it is required to integrate knowledge from domain experts in the field of hydrological studies along with areas such as machine learning/deep learning to develop a model for forecasting the reservoir level input.

1.3 Objectives

The main objective of this research is to explore the use of existing researches carried out for predicting reservoir input level in the field of hydrological sciences and to develop a forecasting tool using advanced machine learning/deep learning techniques, benchmarked against a baseline model which can then be used by the relevant authorities to predict the water input level for a rolling window of time steps of their choice depending on the quantity of data being collected.

The second major objective of this research is to validate the forecasting tool to be developed by using a cost function/ metric which is common both to the baseline model and forecasting model which is to be developed.

1.4 General Objectives

- To develop a forecasting tool based on deep learning techniques which could be utilized by relevant authorities to within a reasonable degree of accuracy.

1.5 Specific Objectives

- To select the best deep learning algorithm which would be the most accurate given the selected feature set.
- To benchmark the performance of the model relative to baseline models.
- To engineer features which could prove to be most appropriate to the model to be developed.

1.6 Prior Work

When considering the prior research carried out in the field of predicting water input levels of reservoirs it can be seen that for decades a wide variety of methods have been proposed including statistical black box models to physical (conceptual) models. However due to the nonlinear, uncertainty and time varying characteristics of the measured hydrological parameters used in prediction it is difficult to point to any single model as being superior being compared to another. [2].

In the recent past with the acceptance of artificial neural networks as a viable and potentially useful tool for predicting as well as modelling highly complex nonlinear systems has seen its adoption for the predicting of water input levels in reservoirs as well [2].

However, it can be seen that with respect to deep learning methods that relatively less amount of work is available in the public domain when considering the specific problem of predicting the water levels to reservoirs, which this study proposes to address.

Chapter 2

2. Literature Review

2.1 Introduction

Considering the previous work carried out by researchers in this domain it can be seen that as mentioned earlier a variety of methods including time series models such as Auto-Regressive Moving Average (ARMA) models, Multi-Linear Regression (MLR) models to Artificial Neural Networks (ANN) have gained increasing popularity [3]. Furthermore, it can be seen that in the use of these varying methodologies that there is a tradeoff between the parsimony of the models developed as opposed to the accuracy of the prediction (i.e. the tradeoff between simple models as opposed to black box models such as ANN's.) [3].

The selection of input parameters for the development of a particular model is also important and can affect the outcome of the prediction being made [4]. Furthermore, with respect to ANN's, the division of the available data in to training and validation sets along with data preprocessing also need to be handled carefully due to the "black box" nature of ANN's.

In this chapter, previous methodologies used in for predictions in the hydrological sciences along with the advantages and disadvantages of the proposed methodologies are discussed in detail.

2.2 Multi-Linear Regression (MLR) model

In Unes et.al [3], the writers use a fourfold methodology approach which includes multi-linear regression (MLR) for modelling daily reservoir levels in the Millers Ferry Dam on the Alabama River in the USA. MLR was introduced as a statistical model to map the linear relationship between a dependent variable and two or more independent variables. The MLR method is generally based on least squares (i.e. minimizing the squares of differences between the actual and forecasted values on the training set.) Although reservoir density flow problems are highly nonlinear the MLR model was developed to be used as a benchmark to compare the other models developed.

In general, the MLR model can be specified as follows. If there are m independent variables and one dependent variable the multi-linear regression model can be specified as,

$$y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_m x_m + \varepsilon$$

where y is the dependent variable, a_i is the coefficient of the i^{th} variable and ε refers to the residual error term.

It could be seen that in the studies carried out with respect to predicting the hydrological parameters of reservoirs that MLR was used as a baseline method against which models such as time series and artificial neural networks were benchmarked, as MLR is a simple model [3]. Furthermore, it could be seen that when compared with more complex models such as time series and artificial neural networks when MLR was evaluated using metrics such as Mean Square Error (MSE), Mean Absolute Error (MAE) and correlation coefficient (R) that the while the MLR models outperformed the time series models the artificial neural networks outperformed the MLR models. [3].

2.3 Auto Regressive (AR) model

The Auto Regressive (AR) model is a simple univariate time series model, where the time series output is regressed on its previous values to develop a model which can then be used to predict future values. An AR (p) model can be specified as follows,

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + c + \varepsilon_t$$

where c is a constant, φ_i represents the weights or coefficients for the auto regressive terms indexed by time and ε_t represents a purely random process or white noise. The coefficients can be estimated using Yule-Walker equations.

This model was one of the approaches used by Unes et al. to forecast daily reservoir levels [3]. It could be seen with respect to predicting reservoir water level inflow that the AR models developed were marginally less accurate with respect to metrics such as MSE, MAE and R coefficient when compared with methods such as MLR and ANNs [3].

2.4 Auto Regressive Moving Average (ARMA) model

The ARMA (Auto Regressive Moving Average) model is a univariate time series model, used to model a stationary stochastic process. consisting of two components which are namely the Auto Regressive (AR) and Moving Average (MA) components. The ARMA model is used to model time series due to its relative parsimony with respect to interpreting the model. It was used in the work of Unes et al. as one of the methods to model daily reservoir levels [3]. The ARMA $(p.q)$ model can be parameterized in terms of its coefficients as follows,

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

Where y_i is, the predicted value indexed by time, ε_i represents the white noise due to the moving average terms and φ_i , θ_i represent the coefficients to be estimated for the autoregressive and moving average components respectively. The Box – Jenkins methodology, provides a systematic means of determining the optimal number of terms to be used.

When considering the approach of using ARMA models to predict reservoir water inflow for the Millers Dam it could be seen that when considering criteria such as MSE, MAE and R coefficient that while the ARMA model was marginally better than AR models that it was outranked by the performance of the MLR and ANNs models developed [3].

2.5 Artificial Neural Networks (ANNs)

2.5.1 Introduction

Artificial Neural Networks (ANNs) can be considered as an umbrella term used to describe computational models which are loosely based on the way biological brains solve problems by the use of connected neurons. The idea of ANNs for modelling non-linear systems was first mooted by McCulloch and Pitts [5]. With respect to modelling hydrological reservoirs it can be seen that ANNs of different architectures have been used and will be examined in detail in the course of this literature review. [2], [3], [4], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]

2.5.2 Multi-Layer Perceptron (MLP)

At its simplest form the mathematical model of a neuron as described by McCulloch and Pitts [5] consists of the following components.

- A set of weighted inputs (w_i),
- An adder that sums the input signals,
- An activation function (a threshold function) that decides whether the neuron fires('spikes') for the current inputs.

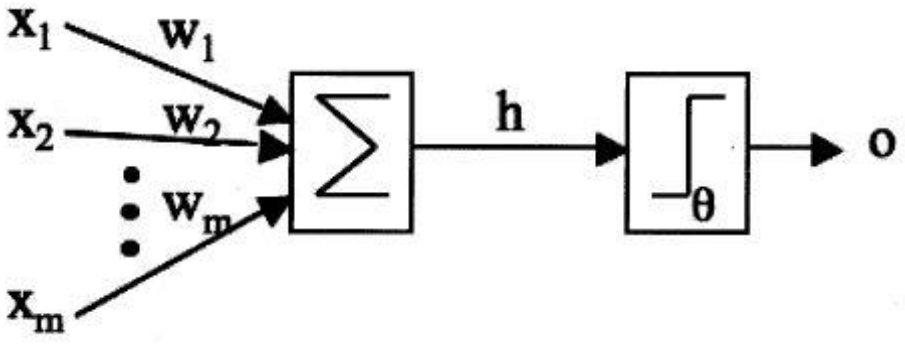

Figure 2.1- McCulloch and Pitt's mathematical model of a neuron. The inputs x_i are multiplied by the weights w_i, and the neurons sum their values. If their sum is greater than the threshold θ then the neuron fires, otherwise it does not.

The Perceptron can be considered as a collection of McCulloch and Pitts neurons combined together with a set of inputs and some weights to fasten the inputs to the neurons.

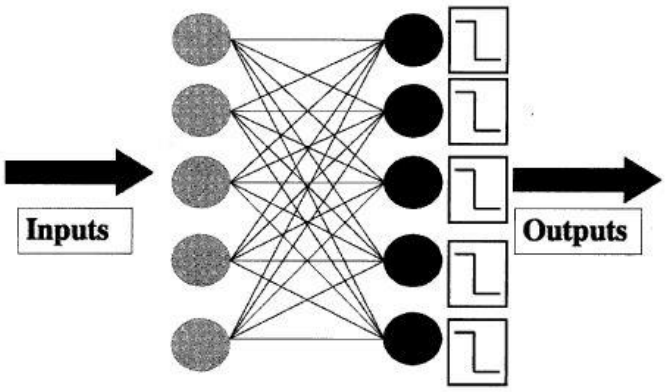

Figure 2.2:The Perceptron network, consisting of a set of input nodes (left) connected to McCulloch and Pitts neurons using weighted connections.

Multi-Layer Perceptrons (MLPs) are a network of Perceptrons with hidden layers which were able to solve the two-dimensional XOR problem, which could not be solved by a model such as the single layer Perceptron. [18].

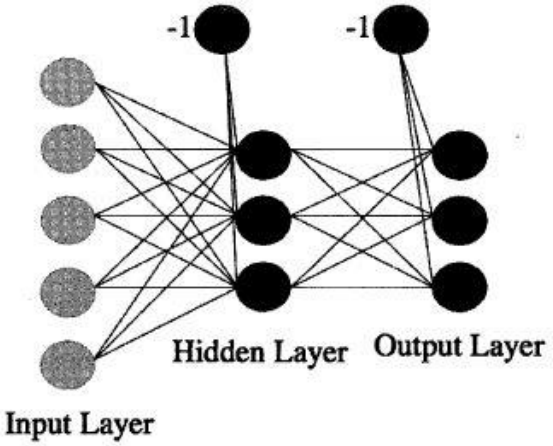

Figure 2.3:The Multi-Layer Perceptron (MLP) Network, consisting of multiple layers of connected neurons

MLPs are also considered to belong to the category of feed forward networks as it is an example of an artificial neural network where connections between the units do not form a cycle. MLPs utilize a suitable algorithm which minimizes a loss function with

respect to the predicted and actual outputs, to determine the values of the weights in the different layers. With respect to the use of MLPs in water resource management it can be seen that various algorithms such as the Levenberg-Marquardt backpropagation algorithm [3] and Output Weight Optimization – Hidden Weight Optimization (OWO-HWO) algorithm [15], [19] were used for determining the weights corresponding to neurons in the network.

When considering the application of MLPs with respect to the field of water resources management it can be seen that they have been used for diverse areas such as short and midterm reservoir inflow forecasting [17] to reservoir water level forecasting [15]. When compared with time series and MLR models it could be seen that MLPs tend to outperform these models, with respect to forecasting input water level in reservoirs, when metrics such as MAE, MSE and R coefficient are taken in to consideration. [3]

2.5.3 Radial Basis Function (RBF) networks

Radial Basis Function (RBF) networks are similar to MLPs, in that they are a form of feedforward networks, with the main differentiator being the type of activation functions used in the networks. While MLPs utilize sigmoid type functions for activation functions, RBF networks differ in that they use Radial Basis Functions. RBF networks have many uses across diverse domains and has been used for classification, system control, function approximation as well as time series forecasting.

When considering the development of RBF networks to develop hydrological forecasting models, the learning algorithm of the RBF network is considered to be critical. The different types of learning algorithms used to determine network configuration and parameters have great influence on the performance of the derived RBF-based models. For instance, if the number of hidden neurons used in the RBF network is too large, it could lead to potential overfitting, whilst on the other hand, if the number of hidden neurons is too small it may fail to map input patterns onto output patterns.

The usage of RBF networks with respect to the modelling of hydrological parameters can be seen in the work of Lin and Wu [8], where they developed a RBF-based model with a two-step learning algorithm to yield the hourly forecast of inflow to a Taiwanese reservoir during typhoons. The RBF network was combined with a two-step learning algorithms (unsupervised and supervised) in order to obtain the optimal configuration of the RBF network. Firstly, in the unsupervised learning step, using a data analysis technique a number of candidates for centers of the hidden neurons of the RBF network were obtained followed by the supervised learning step, where the optimal set of centers were selected from thee candidates and were then used as the centers of the hidden neurons of the RBF network.

Furthermore, in the work of Lin and Wu [8], a RBF based model was evaluated along with a Self-Organizing Radial Basis Function (SORBF) network as well as Back-Propagation Network (BPN). When the RBF model was compared with the other models it could be seen that when evaluated with metrics such as Mean Coefficient of Efficiency (MCE), Mean Coefficient of Correlation (MCC) and Mean Root Mean Square (MRMSE), that the RBF had a higher degree of accuracy.

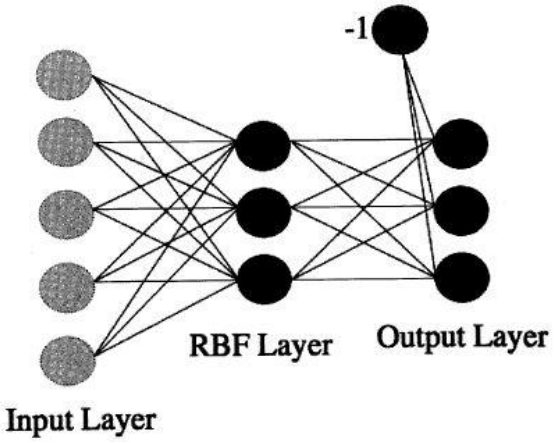

Figure 2.4: The Radial Basis Function network consists of input nodes connected by weight to a set of RBF neurons, which fire proportionally to the distance between the input and the neuron in weight space.

2.5.4 Adaptive Neuro-Fuzzy Inference System (ANFIS)

The Adaptive Neuro-Fuzzy Inference System can be considered as a multilayer feedforward network utilizing neural network learning algorithm and fuzzy reasoning to map an input space to an output space [2]. In the recent past, fuzzy logic has been highly recommended for modeling reservoir operation due to the inherent imprecision and vagueness which characterize problems related to reservoir operations. By augmenting neural networks with a fuzzy system ANFIS has been useful in modeling a variety of processes such power systems dynamic load, motor fault detection and diagnosis, forecasting system for the demand of teacher human resources and real-time reservoir operation.

ANFIS has the advantage of allowing fuzzy rules to be extracted from numerical data or expert knowledge and thereby adaptively construct a rule base. In addition to this, it can also handle the complicated conversion of human intelligence to fuzzy systems. However, a major drawback of the ANFIS model is the time taken for training the structure as well as the time taken for determining the parameters which take a significant time interval.

A Fuzzy Inference System usually consists of four steps. They can be described as follows:

- Step 1: Fuzzification-

At this stage the crisp inputs are transformed into fuzzy sets and the degree to which these inputs belong to each of the appropriate fuzzy sets are determined. The membership function determines how fuzzy the inputs are.

- Step 2: Fuzzy rule base-

The fuzzy rule base is where the system stores the relevant knowledge and information regarding the proposed problem.

- Step 3: Fuzzy Inference Engine

The fuzzy inference engine can be considered to be the brain-like component of the entire system. Based on the fuzzy rule base defined in Step 2, this component can simulate human inference, thinking and decision-making abilities to solve problems.

- Step 4: Defuzzification

The final step of the fuzzy inference system is defuzzification, which consists of transforming the fuzzy outcome into a non-fuzzy output.

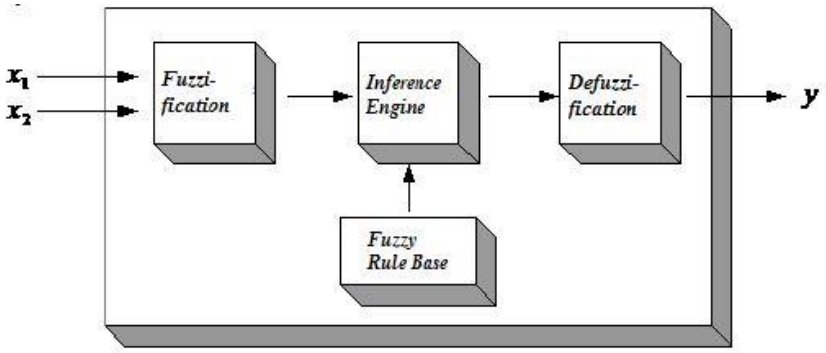

Figure 2.5: Fuzzy Inference System

A fuzzy rule, typically known as a rule of Sugeno's model can be written as:

$$If\ x\ is\ A_1\ and\ y\ is\ \ B_1\ Then\ z = f(x, y)$$

where x and y are inputs and $z = f(x, y)$ is a crisp function in the consequent. When z is a first order polynomial then,

$$z = f(x, y) = px + qy + r$$

If there are two input variables, the rules will be as follows:

$$If\ x\ is\ A_1\ and\ y\ is\ \ B_1\ Then\ f_1 = p_1x + q_1y + r_1$$

$$If\ x\ is\ A_2\ and\ y\ is\ \ B_2\ Then\ f_2 = p_2x + q_2y + r_2$$

As a general case, the ANFIS model consists of a neural network with five layers. Given below is a diagram detailing the ANFIS architecture.

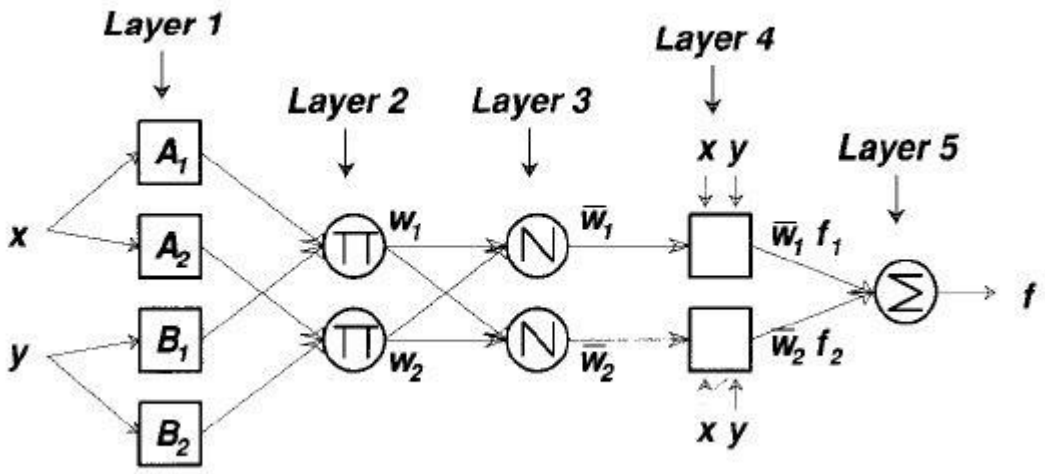

Figure 2.6: Architecture of Adaptive Neruor-Fuzzy Inference System

The following description outlines in details the calculations carried out at each layer of the ANFIS model:

- Layer 1:

Layer 1 consists of the input and fuzzification layer as mentioned previously.

- Layer 2:

This is the rule layer, where a rule neuron receives inputs from the corresponding fuzzification neurons and calculates the firing strength of the rule it represents. The symbol π in the previous figure represents the operator product, where the output of neuron i in Layer 2 is obtained by,

$$O_{2,i} = w_i = \mu_{Ai}(x).\mu_{Bi}(y) \ , i = 1,2$$

- Layer 3:

Layer 3 denotes the normalization layer. The capital letter N on the neurons in the previous figure denotes normalization. Each neuron in this layer calculates the normalized firing strength of a given rule. Thereby the output of neuron i in Layer 3 is determined as,

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2} \quad , i = 1,2$$

- Layer 4:

Layer 4 is the defuzzification layer where a defuzzification neuron calculates the weighted consequent value of a given rule as,

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r) \quad , i = 1,2$$

- Layer 5:

Layer 5 is designed to calculate the summation of outputs of all the defuzzification neurons in the previous layer.

$$O_{5,i} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad , i = 1,2$$

It could be seen from prior work [11], that when comparing the ANFIS model to the MLP and RBF based ANN on the same feature set, that the ANFIS model outperforms the other two models with respect to evaluation metrics such as the Correlation Coefficient, Mean Absolute Prediction Error (MAPE) and Relative Root Mean Square Error (RRMSE).

2.6 Support Vector Machines (SVMs)

Support Vector Machines were developed by Vapnik in 1995 for the supervised learning tasks of classification and regression. [20] Based on statistical learning theory it has been proven that SVMs tend to have certain advantages over backpropagation based ANNs. For instance, SVMs tend to have better generalization capabilities as opposed to Backpropagation Networks (BPNs) [21] . In addition to this the architecture and weights of the SVMs are guaranteed to be unique and globally optimal which tends to make SVM models more robust when compared to BPNs. Finally, SVMs can be trained more rapidly and this feature can prove to be very useful when constructing efficient forecasting models.

With regard to the use of SVMs in hydrological forecasting it can be seen that studies have been carried out to evaluate both SVMs and BPNs with respect to forecasting hourly reservoir inflow forecasting during typhoon- warning periods. [21]. The findings of this study indicated that SVMs are better suited long lead time forecasting when compared with BPN based models due to its better generalization ability. In addition to this SVMs also proved to be more robust and efficient with regards to the predictions made.

2.7 Deep Learning and Applications

Deep learning at its core can be considered to be a sub class of machine learning that has come to prominence in the last few years with the growth of cores available on GPUs for computing. Machine Learning in turn can be defined as a sub field of Artificial Intelligence which is concerned with developing algorithms which can aid to make data driven predictions or decisions. In the field of deep learning currently there are several tools used by the industry to power applications such as self-driving cars to automatically generating captions for videos and images.

Especially with respect to big data solutions it can be seen that deep learning is playing a pivotal role since it can harvest valuable insight from complex systems [22].Currently deep learning has become one of the most active research areas particularly in the machine learning community since its advent in 2006. Even though

the origins of deep learning can be traced back to the 1940s it could be seen that traditional training strategies for multi-layer neural networks either could not guarantee convergence or resulted in a locally optimal solution. It was due to this that the multi-layer neural networks did not receive widespread application even though it was realized that these networks could achieve better performance with respect to representation and feature learning. With Hinton's proposal of a two-stage strategy for pre-training and fine tuning, in 2006, for deep learning in an effective manner the first breakthrough in deep learning was achieved. Furthermore, the increase in computing power and size of datasets also contributed to the popularity of deep learning. With the advent of the era of big data a large number of sample could be collected to train the parameters of deep learning models. With the usage of GPU- based frameworks, the training time for deep learning models such as large-scale deep belief networks with more than 100 million free parameters and millions of training samples can range from several weeks to about one day.

Considering the past few years it can be seen that deep leaning has made significant progress especially with respect feature learning. When compare to conventional shallow machine learning techniques such as Naïve Bayes and Support Vector Machines, deep learning models can take advantage of large number of samples to extract high level features as well learn hierarchical representations by leveraging the low-level input in a more efficient manner.

Deep learning algorithms are constituted of Artificial Neural Networks with hidden layers, which in turn were inspired by biological neural networks [23]. Deep neural networks can be used to model complex non-linear relationships in both supervised (where historical data is used to make prediction about future outcomes) and unsupervised (clustering and finding new patterns and anomalies in data) settings [24]. Compared to traditional machine learning algorithms, deep learning models can provide significant improvement in areas such speech recognition and language translation as evidenced by the significant improvement in Google Translate after switching from Phrase Based Machine Translation (PBMT) to Neural Machine

Translation (NMT) [25] . The different type of Deep Learning models includes Deep Autoencoders, Restricted Boltzmann Machines (RBMs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks and Long Short-Term Memory (LSTM) models depending on whether the application domain is for supervised or unsupervised learning [26].

At the time of writing with respect to applications with regard to forecasting water levels in reservoirs there do not appear to be any literature which cite the use of deep learning techniques and as such the application of deep learning models in this research reflects the novel nature of the approach used, which is based on the success of this approach in other fields.

Chapter 3

3. Methodology

.

Currently even though data regarding the water input level at the Kotmale Reservoir is being collected it can be seen that no active analysis of the data is being carried out. This research plans to remedy this issue by using novel deep learning techniques to give the decision makers better insight with regards to managing competing needs for water resources.

It is evident there are several conventional methods existing for water level prediction currently and it is proposed to use a suitable parsimonious model as a baseline against which the deep learning techniques to be used in this research can be benchmarked against. However, improving upon the accuracy of an existing well-established technique could be a challenging research goal.

This section is concerned about the proposed methodology and approach to implement deep learning.

3.1 Proposed Methodology

In order to implement the proposed deep learning-based technique on the collected data, the approach as depicted in Figure 3.11 is intended to be used. It can be seen that this is a high-level architecture, which broadly outlines the process where the model building will be carried out reiteratively until a suitable degree of accuracy is obtained. It should be noted that the metric for defining the accuracy of a model should be common to all the models to be compared.

The various stages in the high-level architecture will be explained in detail in Chapter 4 –Solution Architecture and Implementation.

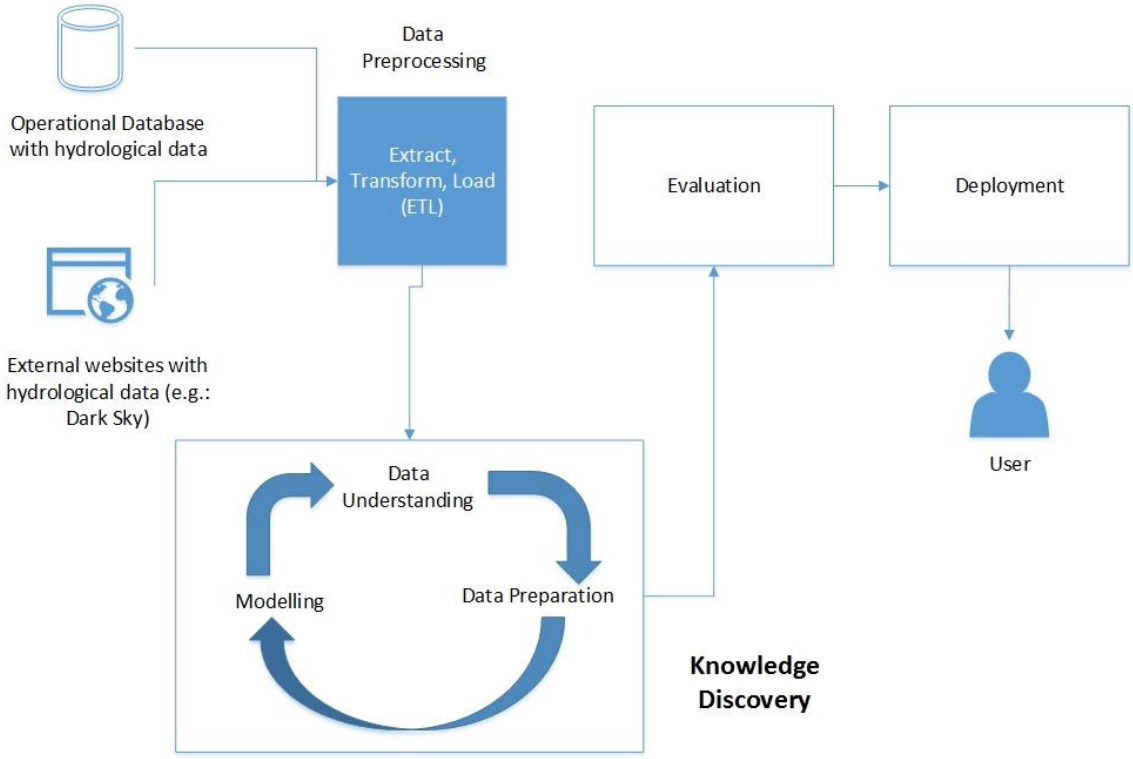

Figure 3.1: Architecture of proposed solution

3.2 Deep Learning

The generally accepted key difference between deep learning and traditional machine learning is that traditional machine learning techniques exploited at most one or two layers of non-linear transformations. [26] While the shallow architecture based traditional machine learning techniques have been found to be effective in solving well constrained or simple problems, when dealing with more complicated real-world applications involving natural signals such as natural sound and language, natural image and visual scenes and human speech , difficulties tend to arise due to their limited representational and modelling capabilities. However, based on human information processing mechanisms such as audition and vision, the need of deep

architectures for extracting complicated structure as well as the building of internal representation from sensory inputs can be seen.

Based on the literature review carried out it was found that the number of studies carried out in the field of reservoir level forecasting using deep learning is virtually nonexistent and this research hopes to address this issue by modelling the water level of the Kotmale Dam by utilizing a deep learning model detailed in the next sub-section to improve upon existing prediction models.

3.2.1 LSTM (Long Short-Term Memory) Recurrent Networks

Recurrent networks are neural networks which take as their input not just the current input example they see, but also what they perceived in the earlier time steps. The decision recurrent nets reach at time step *t-1* affects the decision that they will reach later at time step *t.* Therefore, recurrent networks have two sources of input which are the present as well as the recent past. Recurrent networks are also said to have memory since compared to Feedforward networks it can be seen that, Recurrent networks retain information regarding the temporal structure of the data. This sequential information is preserved in the recurrent network's hidden state, spanning many time steps as it cascades forward to affect the processing of each subsequent example.

The purpose of Recurrent networks is to accurately classify or predict sequential input. Backpropagation and gradient descent is utilized in order to achieve this. Recurrent networks can be visualized as shown in Figure 3.12.

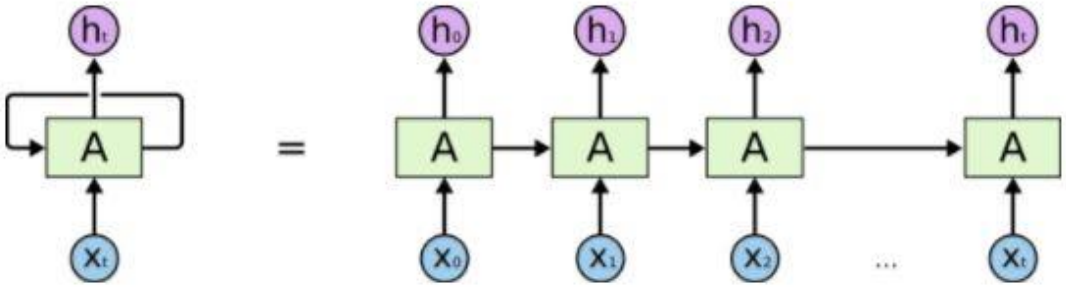

Figure 3.2: An unrolled recurrent neural network

In Feedforward networks, Backpropagation moves backward from the final error through the outputs, weights and inputs of each layer, assigning those weights responsibility for a portion of the error by the calculation of their partial derivatives. These derivatives are subsequently used by the gradient descent algorithm to adjust the weights to decrease error specified by a loss function. Recurrent networks rely on an extension of backpropagation called Backpropagation Through Time (BPTT).

Given below in Figure 3.13 illustrates the repeating module in a standard Recurrent Neural Network (RNN) which usually consists of a single tanh layer.

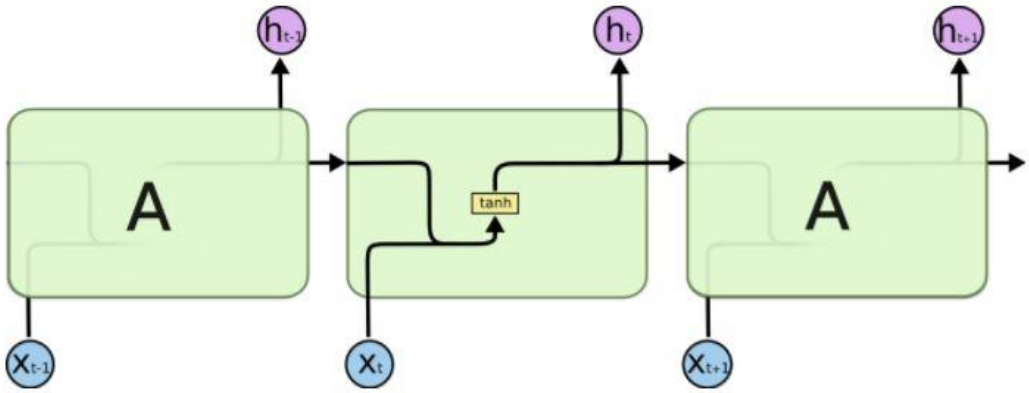

Figure 3.3 :The repeating module in a standard RNN

The vanishing gradient problem is a major obstacle that emerged with respect to recurrent net performance. Long Short-Term Memory units, or LSTMs, were proposed as a solution to the vanishing gradient problem.

LSTMs help preserve the error that can be backpropagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps, thereby opening a channel to link causes and effects remotely.

LSTMs contain information outside the normal flow of the recurrent network in a gated cell. Information can be stored in, written to, or read from a cell. The cell makes decision about what to store, and when to allow reads, writes and erasures, via gates that open and close. These gates act on the signals they receive, and similar to the

neural network's nodes they block or pass on information based on its strength and import, which they filter with their own sets of weights. These weights, similar to the weights that regulate input and hidden states, are adjusted via the recurrent networks learning process.

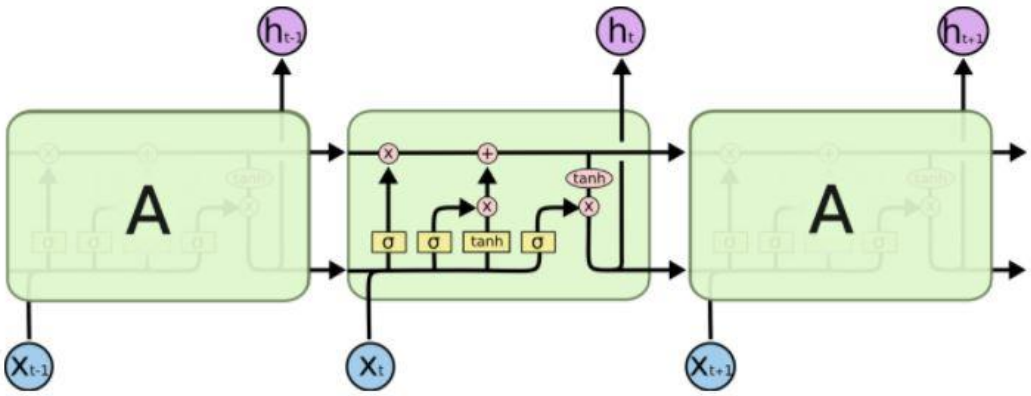

Figure 3.4:The repeating module in an LSTM consisting of four interacting layers

In addition to traditional LSTMs there exists a variant known as a Gated Recurrent Unit (GRU) which is basically an LSTM without an output gate, and therefore fully writes the contents from its memory cell to the larger net at each time step.

3.2.2 Proposed LSTM Model for Water Level Prediction

This research proposes to implement a spatio-temporal LSTM model which can predict the water level, in feet, at the Kotmale Reservoir. This is proposed to be done in the form of namely a LSTM model which will forecast hourly average water level for a half a day ahead time window. This model will use the following variables from the 10 most adjacent hydroelectric dams.

- Cloud Cover – The percentage of sky occluded by the sky, between 0 and 1, inclusive.
- Dew Point – The dew point in degrees Fahrenheit.
- Humidity – The relative humidity between 0 and 1 inclusive.

- Pressure – The sea-level air pressure in millibars.
- Temperature – The air temperature in degrees Fahrenheit.
- Visibility – The average visibility in miles, capped at 10 miles.
- Wind Bearing – The direction that the wind is coming from in degrees, with true north at 0^0 and progressing clockwise. (If Wind Speed is zero, then this value will not be defined.)
- Wind Speed – The wind speed in miles per hour.

The hydroelectric dams for which data points under the above variable are collected for the given period are as follows in Table 3.1,

Table 3.1: Latitude and Longitude of Reservoirs

Dam	**Latitude (Decimal)**	**Longitude (Decimal)**
Broadlands	6.980556	80.4525
Laxapana	6.918889	80.489444
Norton	6.913889	80.521667
Upper Kotmale	6.946667	80.658056
Castlereigh	6.873333	80.566389
Canyon	6.871667	80.526111
Maskeliya	6.843611	80.548889
Nilambe	7.188333	80.631111
Polgolla	7.321667	80.645
Victoria	7.241389	80.784722

Since the spatial element of the model is captured by incorporating the meteorological data across the various hydroelectric stations the temporal aspect is bought in to the model by stacking the last 1 day values for the different meteorological parameters across the different hydroelectric dams.

This can be mathematically expressed as follows,

$$\widehat{Y_t} = LSTM(Y_{t-1}, X_{i_{j_k}})$$

Where,

$$\widehat{Y_t} = Predicted\ Water\ Level\ \ from\ LSTM\ Model\ at\ time\ t\ for\ Kotmale\ Dam$$

$$Y_{t-1} = Actual\ Water\ Level\ at\ time\ t-1\ for\ Kotmale\ Dam$$

$$LSTM = LSTM\ Network\ which\ takes\ Y_{t-1}, X_{i_{j_k}}\ as\ inputs\ to\ output\ \widehat{Y_t}$$

$$X_{i_{j_k}} = Observed\ variable\ value\ at\ time\ i = t-1, \&\ j = 1,2, \dots 10\ \&\ k = 1,2, \dots 6$$

The subscript *i* denotes the time index while *j* denotes the various dams and *k* denotes the meteorological parameters measured for each of those dams. The developed LSTM models will be compared with linear regression models and XGBoost (Extreme Gradient Boost) models to verify the accuracy based on the Root Mean Square Error (RMSE) as the evaluation criteria.

3.3 Baseline Models

As mentioned earlier linear regression models and XGBoost will be used as the baseline models against which to compare the developed LSTM models. The following sections give an outline of how the features will be engineered to be incorporated in these two models.

3.3.1 Baseline Regression Model for Water Level Prediction

The regression model while being able to capture the spatial variation in the data would not be ideally suited to capture the temporal variation unless feature engineering is done on the data. In addition to the variables above, temporal variables such as the day, month and day of the week will be used as input for the baseline regression models to capture the temporal variation.

3.3.2 Baseline SVM (Support Vector Machine) Model for Water Level Prediction

As mentioned in Chapter 2, under Literature Review it was seen that Support Vector Machines (SVMs) represent the state-of-the art with regards to predicting water levels for reservoirs.

As in the case of the regression models, temporal variable will be used as input to incorporate the temporal variation across the data.

3.3.3 Baseline XGBoost (XGBoost) Model for Water Level Prediction

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data.

As in the case of the regression models in addition to the variables used in the LSTM models, temporal variables will be used as input to capture the temporal variation across the data.

Chapter 4

4. Solution Architecture & Implementation

4.1 Overview

As illustrated in Figure 3.11, the following steps were followed during implementation of the solution architecture the details of which will be discussed in detail in this chapter.

- Extract, Transform, Load (ETL) / Data Preprocessing
- Knowledge Discovery
    - Data Understanding
    - Data Preparation
    - Modelling

The evaluation of the developed models will be discussed in detail in Chapter 5 – Model Evaluation (Data and Analysis).

4.2 Extract, Transform, Load (ETL) Process / Data Preprocessing

Under the ETL process data from two separate sources were integrated. The first source was the data for the historical water levels at the Kotmale Reservoir which was available for a period of 546 days (from 30th July 2015 to 25th January 2017). The second was historical meteorological data which was obtained from the Dark Sky API, the details of which were outlined in Section 3.2.2. The Dark Sky API provides was chosen because it provides reliable information upon the supplying of the longitude and latitude co-ordinates of the location, provides a 7 day-ahead forecast and since it is trusted by enterprises such as Microsoft and Yelp.

The entire ETL process was carried out using Python with the urllib2 and json modules being used for accessing the APIs while the pandas and csv module were used for reshaping and outputting the data in flat file format.

Water level data at the Kotmale Reservoir was aggregated at 10 second intervals. The water level was aggregated at daily levels. The effective period of 546 days was arrived at by removing the consecutive days with missing values.

In addition to this, there were instances where values were not returned by the Dark Sky API for certain meteorological parameters. The case of missing values was handled by representing the missing values by the numerical value of -999, The handling of missing values in the model is discussed in detail in the following section under data preparation,

4.3 Knowledge Discovery

Knowledge Discovery consists of the process of understanding the data, preparing the data and developing a model on the prepared data. It consists of the following stages:

- Data Understanding
- Data Preparation
- Modelling

4.3.1 Data Understanding

Data Understanding involves generating descriptive statistics of the data and obtaining insights based on them to infer properties regarding the data. These inferences can be then used to aid decisions in designing and fine tuning the models developed for analysis.

This section is discussed in detail in Chapter 5 under Data and Analysis.

4.3.2 Data Preparation

For the baseline models, the data was standardized by subtracting the average value for a predictor variable from an individual value and dividing it by the standard deviation of the values for that variable

In the case of the proposed LSTM Recurrent Neural Network the values for a predictor variable normalized by using a Min-Max Transform to ensure that the transformed values lie between a minimum value of 0 and a maximum value of 1.

For the LSTM model, the temporal aspect was incorporated by using time lagged observations of the spatial meteorological variables across the different reservoirs.

4.3.3 Modelling

For the baseline models, apart from multiple linear regression the hyper parameters for both the XGBoost and SVM models were determined using grid search.

Similarly, for the LSTM model the hyperparameters of the model were also determined using grid search.

It should be noted that for the model being both for the baseline and LSTM model Cloud Cover and Pressure were excluded were excluded from the analysis since they had a considerable number of missing values. It was seen that none of the other predictor variables had missing values.

Root-Mean Square Error (RMSE) was used as the evaluation metric for assessing the performance of the models. The formula for RMSE is as given in the following page,

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}$$

$$where\ \hat{y}_t = Predicted\ value\ of\ response\ variable\ at\ time\ t$$

$$y_t = Actual\ value\ of\ response\ variable\ at\ time\ t$$

$$n = number\ of\ observations$$

Chapter 5

5. Data & Analysis

5.1 Descriptive Analysis

Descriptive Analysis was carried out by graphing the relationship of the various meteorological variables across the various reservoirs across time.

5.1.1 Descriptive Analysis for Modelling

The graphs obtained under this analysis are attached in the Appendix and include in detail the relationships.

5.1.2 Evaluation and Analysis of Baseline Models

5.1.2.1 Linear Regression Model

The model coefficients for the Linear Regression Model created on the dataset were as given in Table 5.1:

Table 5.1 - Coefficients for Linear Regression Model

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.82E+02	5.01E+01	5.627	5.21e-08
Broadlands_dewPoint	2.65E+02	4.41E+02	0.602	0.548079
Broadlands_humidity	-4.00E+02	1.03E+03	-0.388	0.698434
Broadlands_temp	1.35E+02	3.66E+02	0.369	0.712204
Broadlands_visibility	7.52E+02	7.04E+02	1.069	0.286267
Broadlands_windbearing	2.13E-01	2.48E-01	0.861	0.390056
Broadlands_windspeed	-2.74E+02	2.92E+02	-0.941	0.347879
Canyon_dewPoint	-1.08E+02	9.00E+02	-0.12	0.904755
Canyon_humidity	-1.78E+03	1.41E+03	-1.264	0.207549
Canyon_temp	-5.07E+02	9.57E+02	-0.53	0.596631
Canyon_visibility	2.15E+03	1.43E+03	1.506	0.133468
Canyon_windbearing	6.36E-01	5.02E-01	1.267	0.206526
Canyon_windspeed	-3.25E+02	9.86E+02	-0.33	0.741704
Castlereigh_dewPoint	-3.36E+02	5.95E+02	-0.565	0.572473
Castlereigh_humidity	6.33E+02	1.63E+03	0.388	0.698016

Castlereigh_temp	9.23E+02	5.84E+02	1.58	0.115489
Castlereigh_visibility	-5.40E+02	9.18E+02	-0.588	0.556871
Castlereigh_windbearing	1.64E-01	3.36E-01	0.488	0.626246
Castlereigh_windspeed	1.79E+02	5.96E+02	0.3	0.764085
Laxapana_dewPoint	-5.38E+02	9.09E+02	-0.592	0.554392
Laxapana_humidity	2.01E+03	1.49E+03	1.347	0.179131
Laxapana_temp	6.76E+01	8.63E+02	0.078	0.937587
Laxapana_visibility	9.19E+02	1.37E+03	0.673	0.501892
Laxapana_windbearing	-9.45E-02	3.70E-01	-0.256	0.798442
Laxapana_windspeed	9.44E+02	9.17E+02	1.03	0.304228
Maskeliya_dewPoint	6.27E+02	6.87E+02	0.912	0.362631
Maskeliya_humidity	-5.41E+02	1.63E+03	-0.331	0.740789
Maskeliya_temp	3.45E+02	7.32E+02	0.471	0.638169
Maskeliya_visibility	-6.25E+02	7.57E+02	-0.826	0.409639
Maskeliya_windbearing	-3.52E-01	4.84E-01	-0.726	0.468301
Maskeliya_windspeed	-8.41E+01	7.15E+02	-0.118	0.90639
Nilambe_dewPoint	1.61E+02	1.72E+02	0.935	0.350636
Nilambe_humidity	-1.07E+03	1.50E+03	-0.714	0.476169
Nilambe_temp	6.39E+02	1.75E+02	3.649	0.000325
Nilambe_visibility	-1.17E+03	2.95E+02	-3.954	0.000102
Nilambe_windbearing	3.01E-01	2.29E-01	1.316	0.189385
Nilambe_windspeed	-3.10E+01	6.81E+01	-0.456	0.648951
Norton_dewPoint	-2.26E+01	9.67E+02	-0.023	0.9814
Norton_humidity	2.60E+02	1.26E+03	0.206	0.83662
Norton_temp	-9.60E+02	1.13E+03	-0.852	0.395204
Norton_visibility	-2.87E+03	1.39E+03	-2.063	0.040174
Norton_windbearing	-5.63E-01	4.23E-01	-1.33	0.184754
Norton_windspeed	-4.65E+02	8.13E+02	-0.571	0.568459
Polgolla_dewPoint	-3.35E+01	2.62E+01	-1.276	0.20314
Polgolla_humidity	3.97E+01	7.10E+02	0.056	0.955462
Polgolla_temp	-4.93E+01	2.66E+01	-1.855	0.064856
Polgolla_visibility	1.60E+02	3.97E+01	4.026	7.67e-05
Polgolla_windbearing	-5.04E-02	1.08E-01	-0.465	0.642527
Polgolla_windspeed	-3.99E+00	8.40E+00	-0.475	0.63509
UpperKotmale_dewPoint	-1.50E+00	8.19E+01	-0.018	0.985417
UpperKotmale_humidity	8.32E+02	9.46E+02	0.879	0.380204
UpperKotmale_temp	-7.81E+01	8.59E+01	-0.909	0.364126
UpperKotmale_visibility	4.62E+02	1.42E+02	3.247	0.001339
UpperKotmale_windbearin	8.393e-02	1.93E-01	0.434	0.664759
UpperKotmale_windspeed	1.43E+01	4.68E+01	0.307	0.759309
Victoria_dewPoint	-1.41E+01	1.15E+02	-0.123	0.902456
Victoria_humidity	1.80E+01	1.15E+03	0.016	0.987505

Victoria_temp	-5.18E+02	1.18E+02	-4.383	1.77e-05
Victoria_visibility	7.55E+02	2.08E+02	3.633	0.000344
Victoria_windbearing	-2.77E-01	2.10E-01	-1.319	0.188451
Victoria_windspeed	4.55E+01	5.31E+01	0.857	0.392527

As there was a significant outlier on the test data which was also a leverage point it was removed for the analysis of the RMSE metric. The Linear Regression Model had a RMSE of 45.36 on the training data and a RMSE of 11.39 on the test data.

In addition to the above metrics, the predicted versus actual values were graphed along with the same values in time order as well as the standardized residuals versus the predicted values to perform model diagnostic checks. It was observed that because of the nonlinear relationship between the predictors and the dependent value that the standardized values do not appear to be completely random. At the same time, when plotting the predicted versus actual values in time order it could be seen that while there was difference in values the movement of the time series appeared to be in roughly the same pattern.

These patterns can be observed in Figure 5.1 – Figure 5.5 given below.

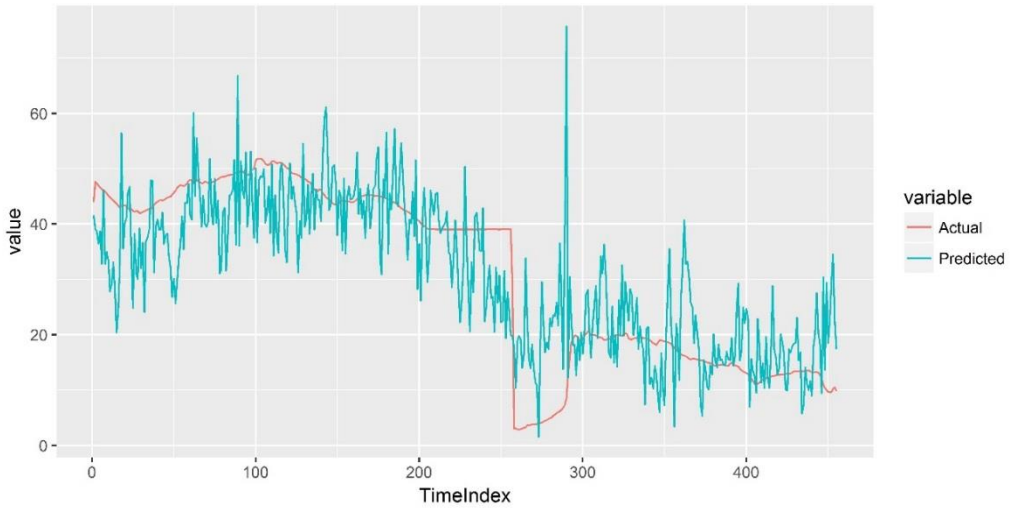

Figure 5.1 -Predicted Vs Actual Values Over Time using Linear Regression

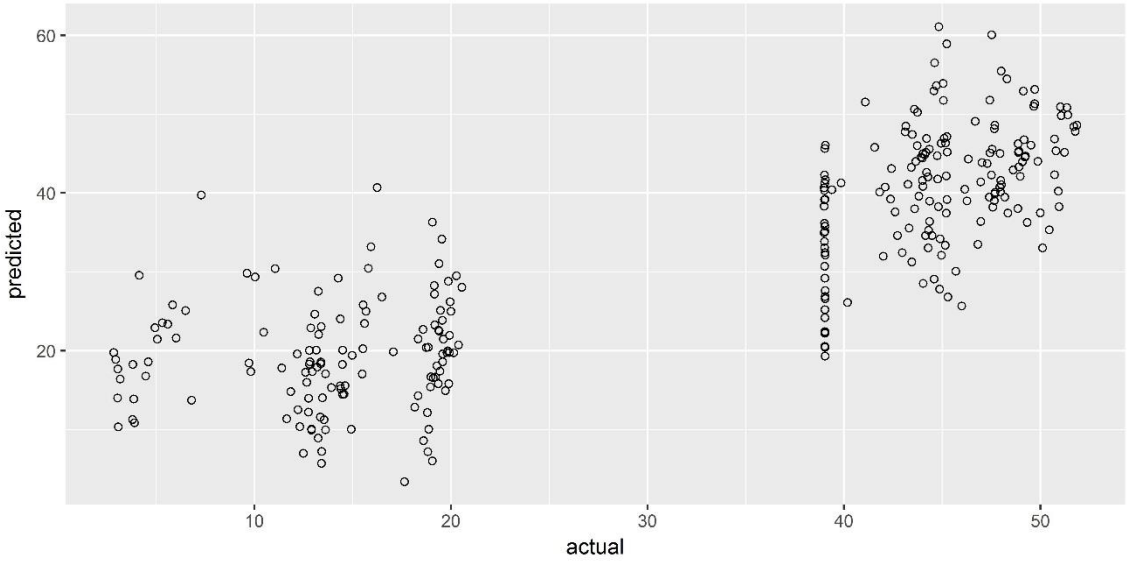

Figure 5.2- Predicted Vs Actual Values Scatterplot for Training Data using Linear Regression

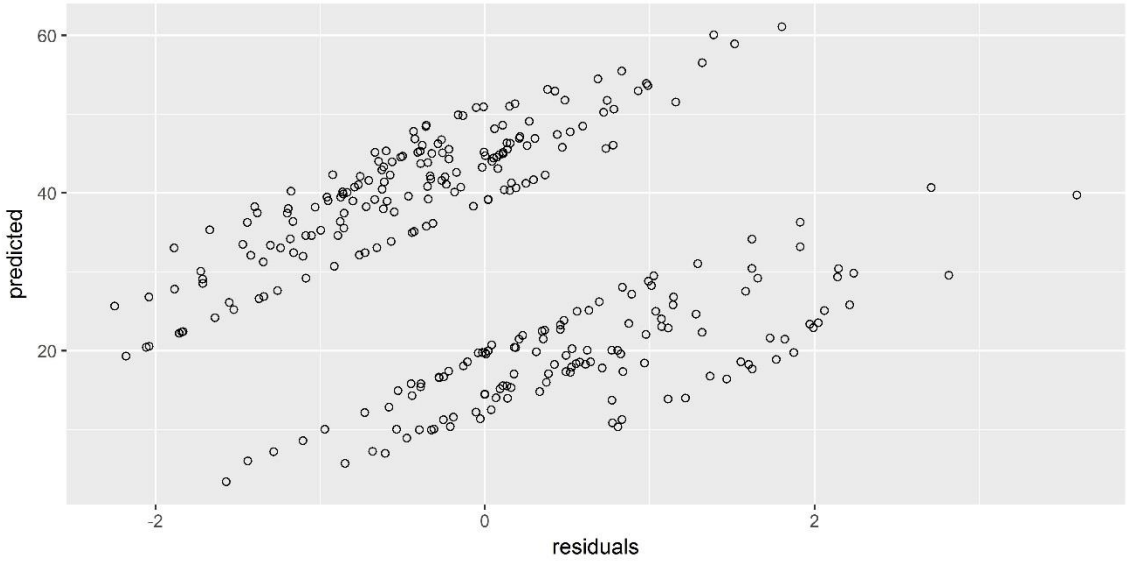

Figure 5.3-- Predicted Vs Standardized Residual Values Scatterplot for Training Data using Linear Regression

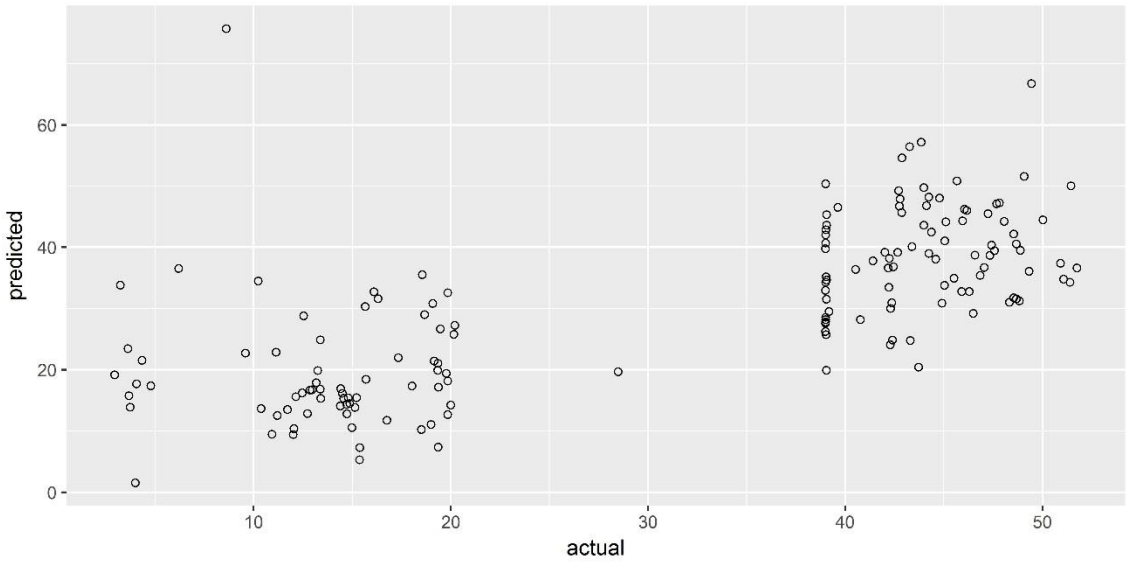

Figure 5.4 -Predicted Vs Actual Values Scatterplot for Test Data using Linear Regression

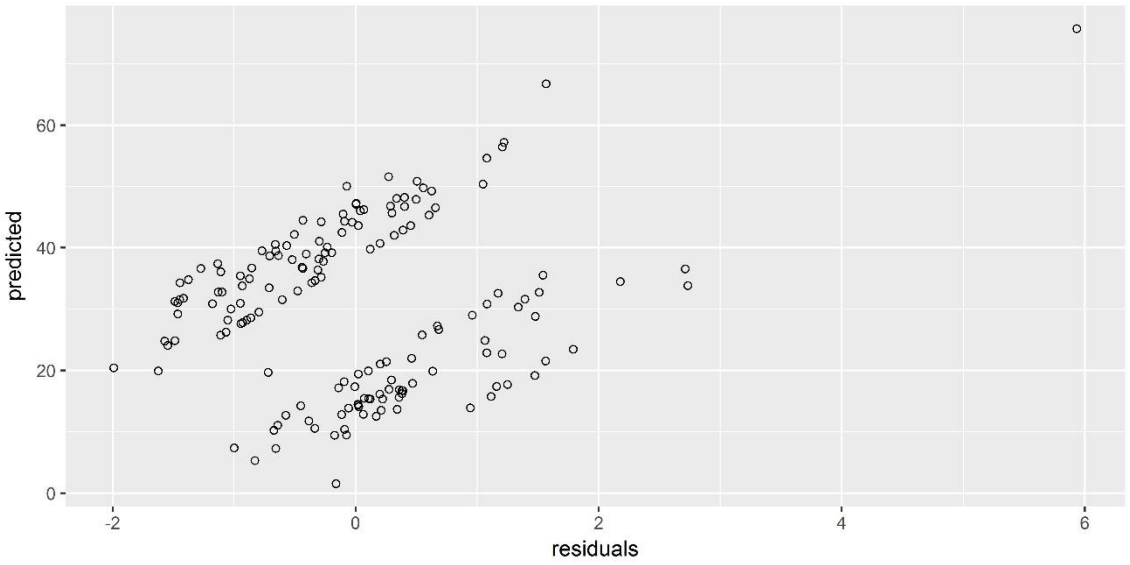

Figure 5.5 - Predicted Vs Standardized Residual Values Scatterplot for Test Data using Linear Regression

5.1.2.2 Support Vector Machines

Since the prediction of the water level is a regression problem epsilon support vector regression and nu support vector regression were used with three types of kernels, namely linear, polynomial and sigmoid kernels. This resulted in 6 models being developed under the baseline category for Support Vector Machines (SVMs). The models were chosen after optimizing the hyperparameters of each model by using grid search.

The Root Mean Square Errors (RMSEs) for the developed models on the training and test sets are given as follows in Table 5.2:

Table 5.2 - SVR Model with Train and Test RMSEs

SVR Model	RMSE (Train)	RMSE (Test)
Epsilon SVR – Linear Kernel	12.48	12.85
Epsilon SVR – Polynomial Kernel	14.99	15.24
Epsilon SVR – Sigmoid Kernel	51.73	48.00
Nu SVR – Linear Kernel	12.18	12.28
Nu SVR – Polynomial Kernel	14.70	14.39
Nu SVR – Sigmoid Kernel	45.36	40.25

It can be that for both Epsilon and Nu SVRs when the Sigmoid kernel is used that the RMSE obtained for the train and test sets is significantly larger compared to the other SVRs. This could be due to an anomaly arising due to the type of kernel used.

The predicted versus actual values in time order was plotted for all the SVR models. Furthermore, the predicted versus actual values for both the training and test sets were plotted for all the SVR models were plotted and are given in the Appendix.

5.1.2.3 XGBoost

The optimal XGBoost model had a RMSE of 0.00045 on the training data and a RMSE of 14.75 on the test data. The model which minimizes the RMSE was chosen after optimizing the hyperparameters of the models by using grid search.

In addition to the above metrics, the predicted versus actual values were graphed along with the same values in time order. The scatterplot of predicted versus actual values was also graphed and is given below.

The optimal value for the hyper parameters of the XGBoost model were found to be as follows:

- Maximum Depth = 8
- Eta = 1.0
- Number of Rounds = 25

The variation of predicted versus actual values across time and for the training and test data can be seen Figure 5.6, Figure 5.7 and Figure 5.8 respectively.

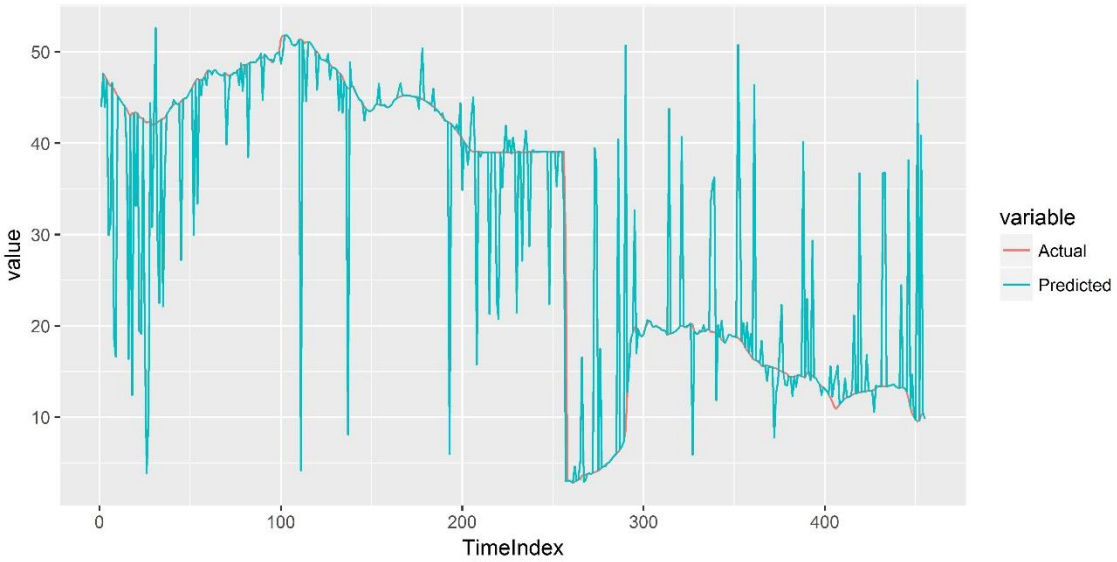

Figure 5.6 - Predicted Vs Actual Values Over Time using XGBoost

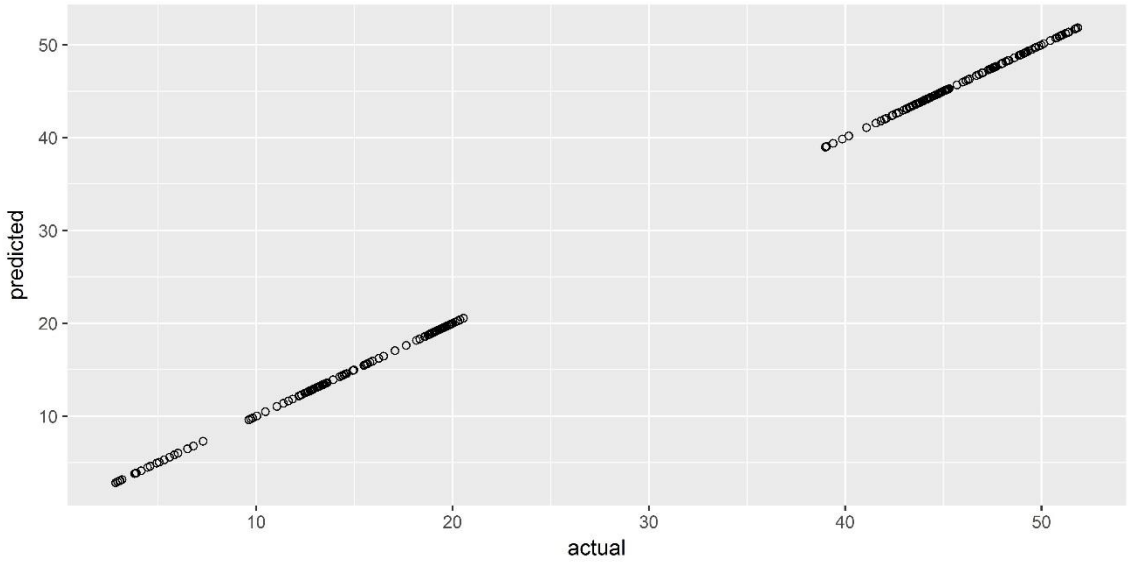

Figure 5.7 -Predicted Vs Actual Values Scatterplot for Training Data using XGBoost

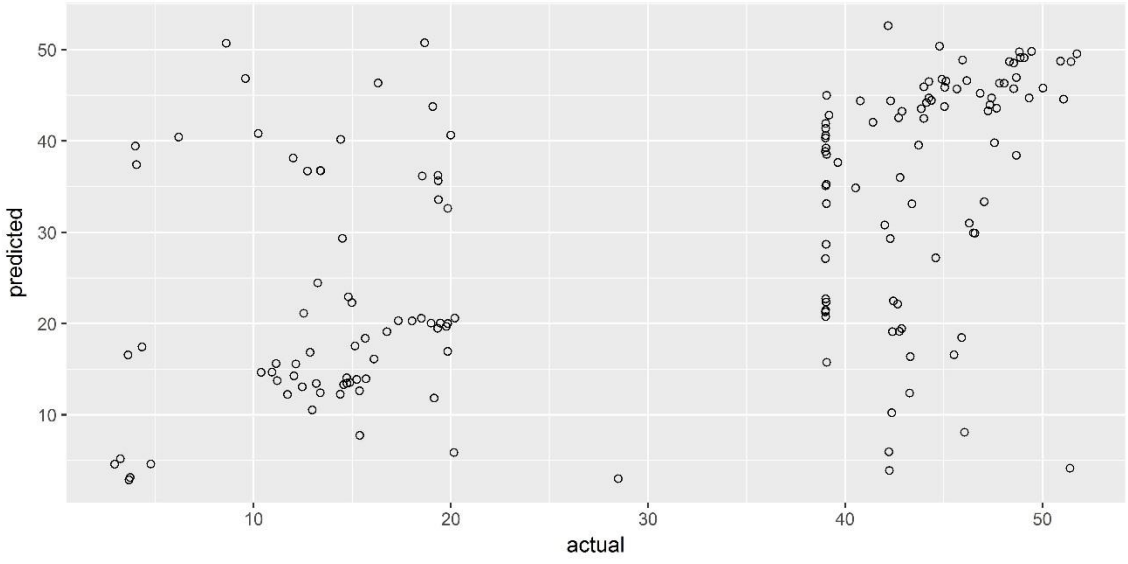

Figure 5.8:Predicted Vs Actual Values Scatterplot for Test Data using XGBoost

5.1.3 Evaluation and Analysis of LSTM Model

The LSTM models developed used both spatial and temporal variables. When developing the models, the validation error of the model oscillated which was probably due to the time ordering of the training and test data sets. The hyperparameters of the model were selected using grid search

Different models were fitted and the RMSE of the models were assessed. Among the different type of Recurrent Neural Network models used both LSTM and GRU neural networks were fitted with the number of neurons in the first layer being modified after initial model tuning. Given below are graphs which illustrate how the error varies in the train and test set over different epochs for the different type of neural networks employed.

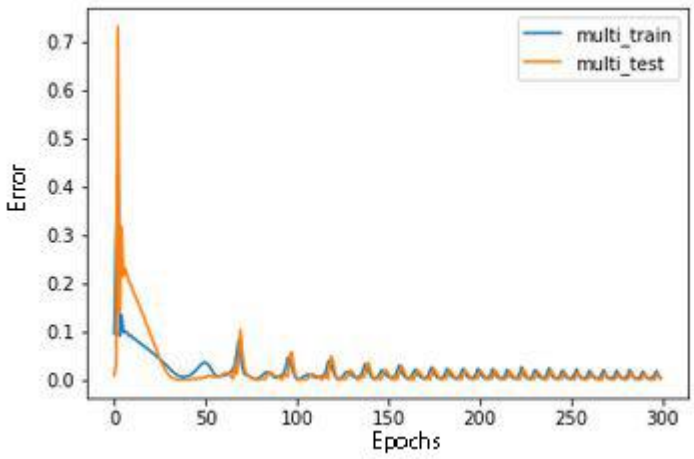

Figure 5.9- Variation of error in the train and test set for GRU RNN with 100 neurons in first layer

As seen in Figure 5.9, it can be seen that the test set error oscillates and then decreases an converges to a lower value as the epochs increases. This could be attributed to the time ordering inherent in the data which is fed in to the GRU RNN.

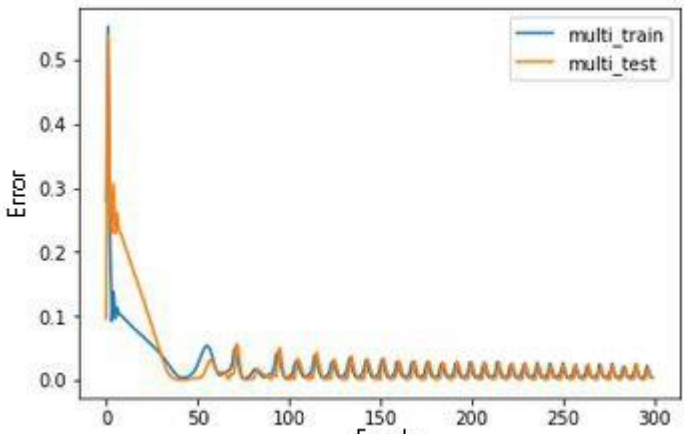

Figure 5.10- Variation of error in the train and test set for GRU RNN with 150 neurons in first layer

Compared to Figure 5.9 it can be seen that in Figure 5.10 that the test set error oscillates in a similar manner across epochs and converges to a lower value.

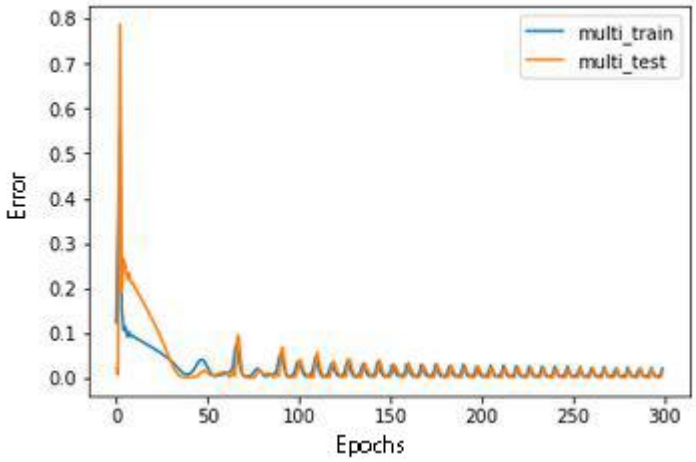

Figure 5.11- Variation of error in the train and test set for GRU RNN with 200 neurons in first layer

It can be seen from Figure 5.11 that it is once again similar to Figure 5.10 and that the test set error oscillates over epochs and converges to a lower value.

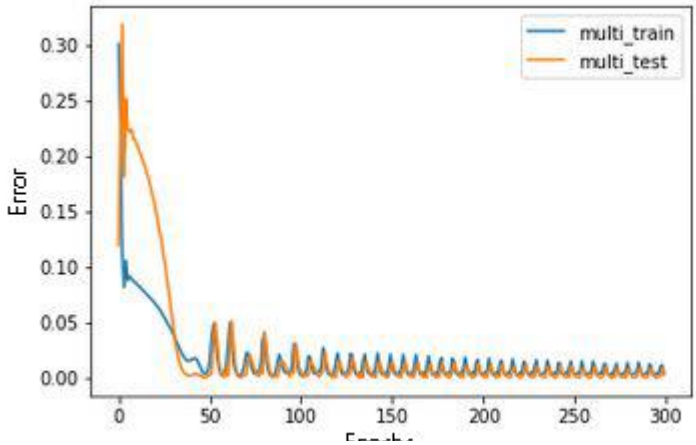

Figure 5.12- Variation of error in the train and test set for LSTM RNN with 100 neurons in first layer

In Figure 5.12, it can be seen that the test error oscillates across epochs and converges towards a smaller value.

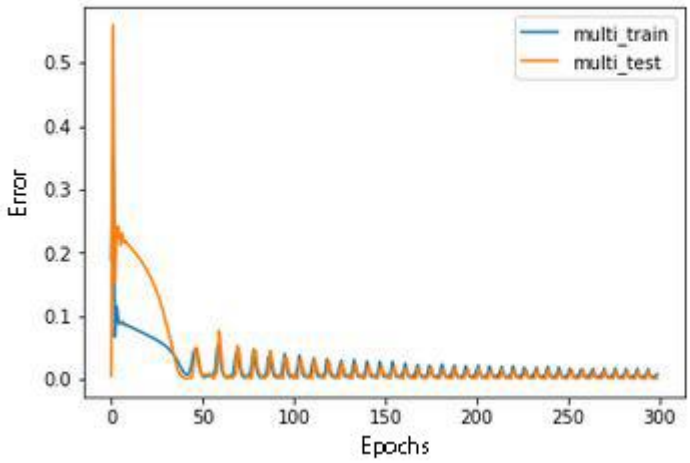

Figure 5.13- Variation of error in the train and test set for LSTM RNN with 150 neurons in first layer

It can be seen from Figure 5.13 that similar to Figure 5.14 that the test error and train error decrease and oscillate together converging towards a smaller value as the epochs increase.

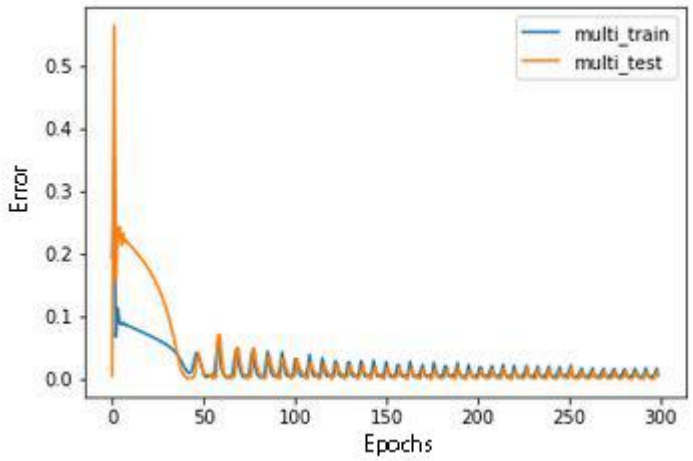

Figure 5.14- Variation of error in the train and test set for LSTM RNN with 200 neurons in first layer

From Figure 5.14 it can be seen that the test error oscillates across epochs and converges towards a smaller value as the number of epochs increase.

The model which minimizes the RMSE was chosen after optimizing the hyperparameters. It was the model which had 150 neurons in the LSTM layer. The optimal LSTM model had a RMSE of 3.24 on the training set and a RMSE of 2.14 on the test set. This is indicative that the model has a low validation error when generalizing to unseen data. In addition, the predicted versus actual values were graphed along with the same values in time order. The scatterplot of predicted versus actual values was also graphed. These can be seen in Figure 5.15, Figure 5.16 and Figure 5.17 respectively.

.

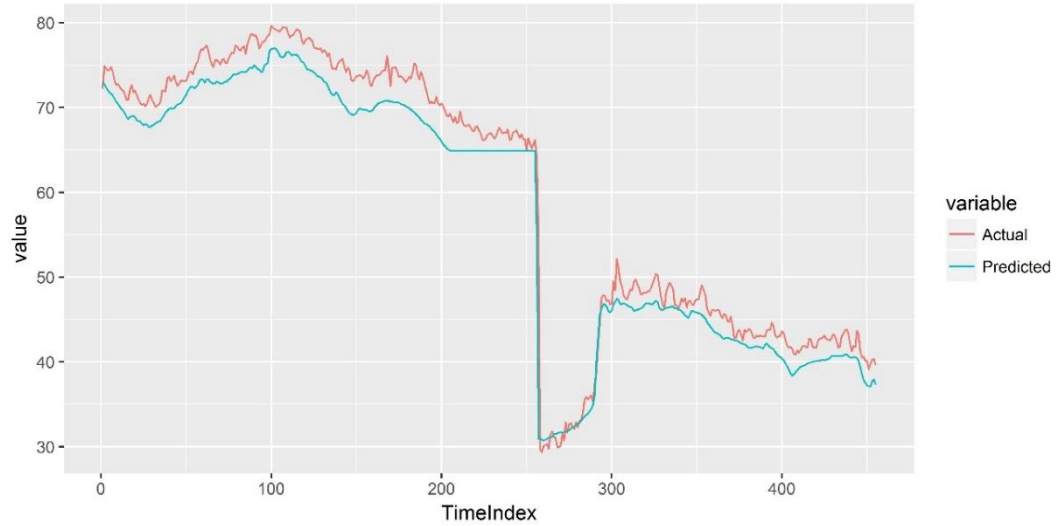

Figure 5.15 - Predicted Vs Actual Values Over Time using Optimal LSTM Model

Figure 5.15- Predicted Vs Actual Values Over Time using Optimal LSTM model

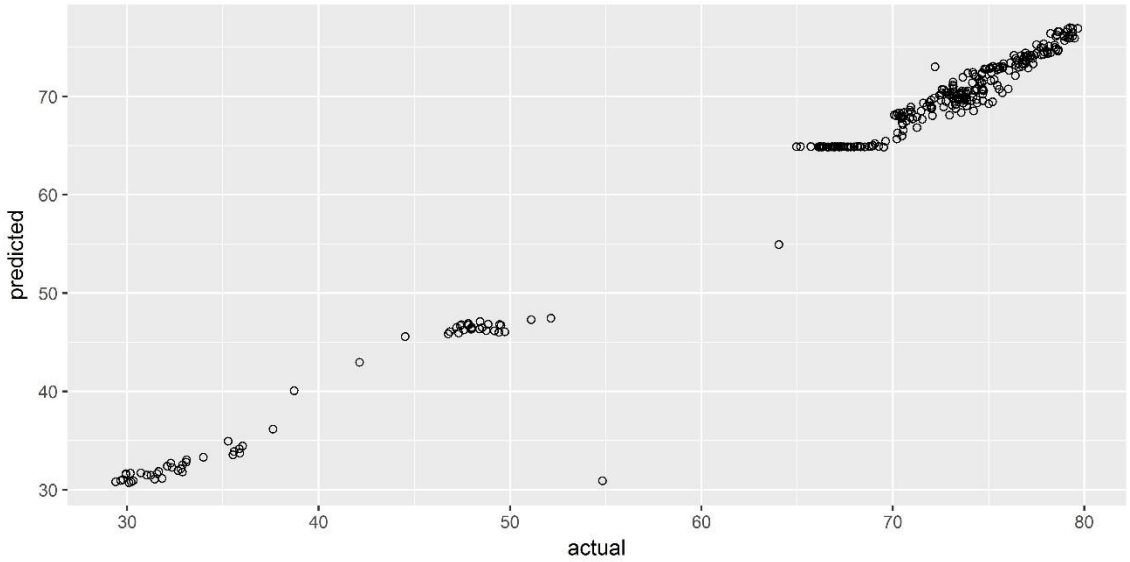

Figure 5.16- Predicted Vs Actual Values Scatterplot for Training Data using Optimal LSTM Model

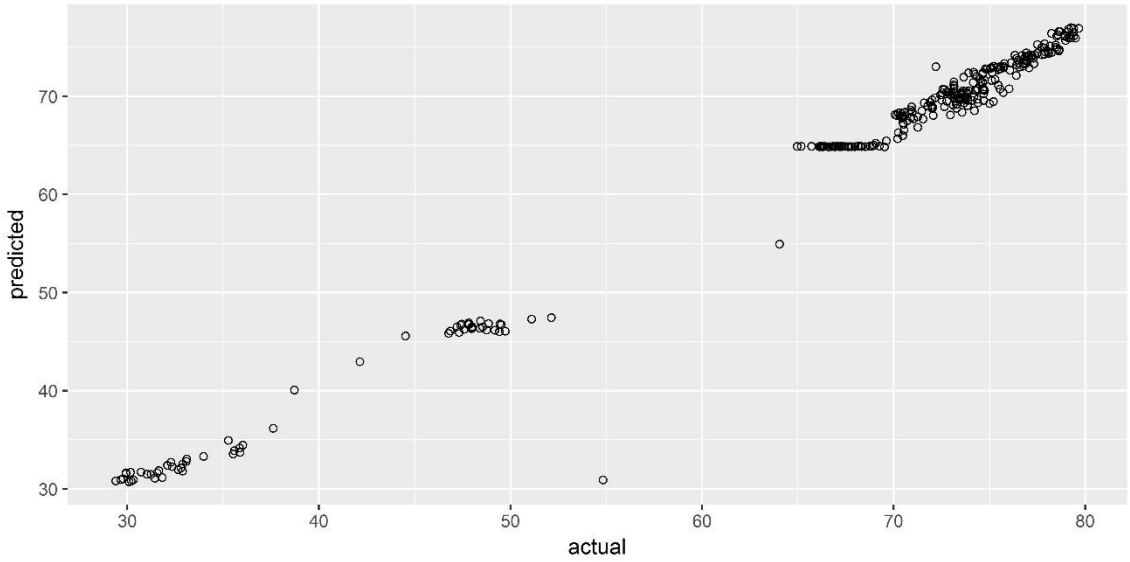

Figure 5.17- Predicted Vs Actual Values Scatterplot for Test Data using Optimal LSTM Model

Chapter 6

6. General Discussion & Conclusion

6.1 General Discussion on the Study

For the purpose of this study to see whether deep learning could be utilized to improve on traditional machine learning and predictive analytics models, 8 baseline models were used to benchmark versus the performance of the LSTM Recurrent Neural Network developed. It was interesting to note that possibly due to the limited amount of data available in the dataset that while the LSTM model achieved the lowest RMSE value on the training set as well as the lowest RMSE value for the test set except for the XGBoost model when the baseline models which implies that it was the best among all models as the other models at predicting out of sample values. This was also evinced by the descriptive analysis on the predicted versus actual values of the model.

It was interesting to note that for the XGBoost model that there was a significant difference between the training and test RMSE values and in addition from the predicted versus actual plot it could be seen that the predictions obtained from this model tended to fluctuate significantly as well. When considering the Linear Regression model it could be seen that the test set RMSE was significantly lower compared to the training set RMSE. In addition, it could be noted that while SVM regression models with Linear and Polynomial Kernels had similar values with respect to RMSE on the training and test set, the SVM model with Sigmoid Kernel that the training set and test set RMSEs were significantly higher. This could be attributed due to the parameterization of the Sigmoid Kernel resulting in these values.

When considering the diagnostic plots of the models it could be seen that the variation in the predicted values of the LSTM model with the actual values it can be seen that most of them lie along the line oriented at 45 degrees in the first quadrant to both the horizontal and vertical axes. When considering the predicted values for XGBoost and Regression models it could be seen that they tended to fluctuate rapidly between values, while for the SVM Models a similar pattern could be observed.

It was also interesting to note that for the different LSTM and GRU Recurrent Neural Networks developed that the train and test set error tended to oscillate across epochs. This could be attributed to the potential time ordering of data. In addition, it could be

seen that the error converged to a lower value as the number of epochs over which the model was trained increased. The optimal RNN which minimized the RMSE was found to be the LSTM RNN which used 150 neurons in its first layer.

6.2 Conclusions

The findings of this study are summarized below:

- Out of the models developed the LSTM and GRU models appeared to be the most stable with respect to time and both had the least RMSE on the test set and the model which optimized the RMSE on both the training and test sets was obtained from a LSTM model.

- While the XGBoost model performed relatively well the values predicted by the optimal XGBoost model tended to fluctuate rapidly results in spikes and troughs for certain predictions.

- Out of the SVM models developed while the SVM models with Linear and Polynomial Kernels had near similar performance with respect to the RMSE metric the SVM models with the Sigmoid Kernel had anomalously high values with respect to the RMSE metric. This could be attributed to the specific manner in which the Sigmoid Kernel is parameterized.

- The Linear Regression model had the highest RMSE for the training set when compared to the other baseline models.

6.3 Further Work

The following are some areas which could be researched into, based on the knowledge gathered from this study:

- Extend this research to the case where the number of observations collected from the Kotmale Reservoir was over an increased time span.

- Bring in more variables which could be used to derive more features during the feature engineering process.

References

[1] D. J. "The Kotmale Environment: A study of the environmental impact of the Kotmale Hydropower Project in Sri Lanka," Swedish International Development Authority, 1989.

[2] F.-J. Chang and Y.-T. Chang, "Adaptive neuro-fuzzy inference system for prediction of water level in reservoir," *Advance in Water Resources,* vol. 29, pp. 1-10, 2006.

[3] F. Unes, M. Demirci and O. Kisi, "Predicition of Millers Ferry Dam Reservoir level in USA using Artificial Neural Networks," *Periodica Polytechnica,Civil Engineering.,* vol. 59, no. 3, 2015.

[4] G. C. Dandy and H. R. Maier, "Neural Networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications," *Environemental Modelling and Software,* vol. 15, pp. 101-124, 2000.

[5] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics,* vol. 5, 1943.

[6] T.-H. Chang, A.-P. Wang and H.-Y. Liao, "Adaptive Neuro-fuzzy Inference System on Downstream Water Level Forecasting," in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, 2008.

[7] L. Hluchy, P. Krammer, O. Habala and M. Seleng, "Advanced Data Integration and Data Mining for Environmental Scenarios," in *12th International Symposium on Symbolic and Numerical Computing for Scientific Computing*, 2010.

[8] G.-F. Lin and M.-C. Wu, "An RBF network with a two-step learning algorithm for developing a reservoir inflow forecasting model," *Journal of Hydrology,* pp. 439-450, 2011.

[9] S. K. Jain, A. Das and D. K. Srivastava, "Application of ANN for Reservoir Inflow Prediction and Operation," *Journal of Water Resources Planning and Management,* vol. 125, no. 5, 1999.

[10] H. Aksoy and A. Dahamsheh, "Artificial neural network models for forecasting monthly precipitation in Jordan," *Stochastic Environmental Research and Risk Assessment ,* vol. 23, no. 7, pp. 917-931, 2009.

[11] E. Khadangi, H. R. Madvar and M. M. Ebadzedeh, "Comparison of ANFIS and RBF models in daily stream flow forecasting," in *2nd International Conference on Computer, Control and Communication.*, 2009.

[12] P. Coulibaly, F. Anctil and B. Bobee, "Daily reservoir inflow forecasting using artificial neural networks," *Journal of Hydrology,* no. 230, pp. 244-257, 2000.

[13] Y.-M. Chiang, L.-C. Chang, M.-J. Tsai, Y.-F. Wang and F.-J. Chang, "Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and ungauged sites.," *Hydrology and Earth System Sciences,* pp. 1309-1319, 2010.

[14] D.-H. Bae, D. M. Jeong and K. Gwangseob, "Monthly dam inflow forecasts using weather forecasting information and neuro-fuzzy technique," *Hydrological Sciences Journal,* vol. 52, pp. 99-113, 2007.

[15] S. Ondimu and H. Murase, "Reservoir Level Forecasting using Neural Networks: Lake Naivasha," *Biosystems Engineering,* vol. 96, no. 1, pp. 135-138, 2007.

[16] M. Firat and M. Gungor, "River flow estimation using adaptive neuro fuzzy inference system," *Mathematics and Computers in Simulation,* vol. 75, pp. 87-96, 2007.

[17] T. Stokelj, D. Paravan and R. Golob, "Short and Mid Term Hydro Power Plant Reservoir Inflow Forecasting," in *Proceedings of International Conference on Power System Technology, 2000.* , 2000.

[18] S. Marsland, Machine Learning: An Algorithmic Perspective, CRC Press, 2009.

[19] H.-H. Cheng, M. T. Manry and H. Chandrasekaran, "A neural network training algorithm utilizing multiple sets of linear equations," *Neurocomputing,* vol. 25, pp. 55-72, 1999.

[20] Vapnik, The nature of statistical learning theory, New York: Springer, 1995.

[21] G.-F. Lin, G.-R. Chen, P.-Y. Huang and Y.-C. Chou, "Support vector machine-based models for hourly reservoir inflow forecasting during typhoon-warning periods," *Journal of Hydrology,* vol. 372, pp. 17-29, 2009.

[22] Q. Zhang, L. T. Yang, Z. Chen and P. Li, "A survey on deep learning for big data," *Information Fusion,* vol. 42, pp. 146-157, 2018.

[23] K. Waehner, "Open Source Deep Learning Frameworks and Visual Analytics," [Online]. Available: http://www.datasciencecentral.com/profiles/blogs/open-source-deep-learning-frameworks-and-visual-analytics.. [Accessed 16 7 2017].

[24] J. O., "Models, Frameworks Accelerate Deep Learning for Enterprise Big Data," [Online]. Available: https://goparallel.sourceforge.net/models-frameworks-accelerate-deep-learning-enterprise-big-data/. [Accessed 16 07 2017].

[25] Observer, "Google Is Using Artificial Intelligence to Make a Huge Change to Its Translate Tool," [Online]. Available: http://observer.com/2017/03/google-translate-neural-update/. [Accessed 15 03 2018].

[26] L. Deng and D. Yu, Deep Learning: Methods and Applications, 2014.

Appendix

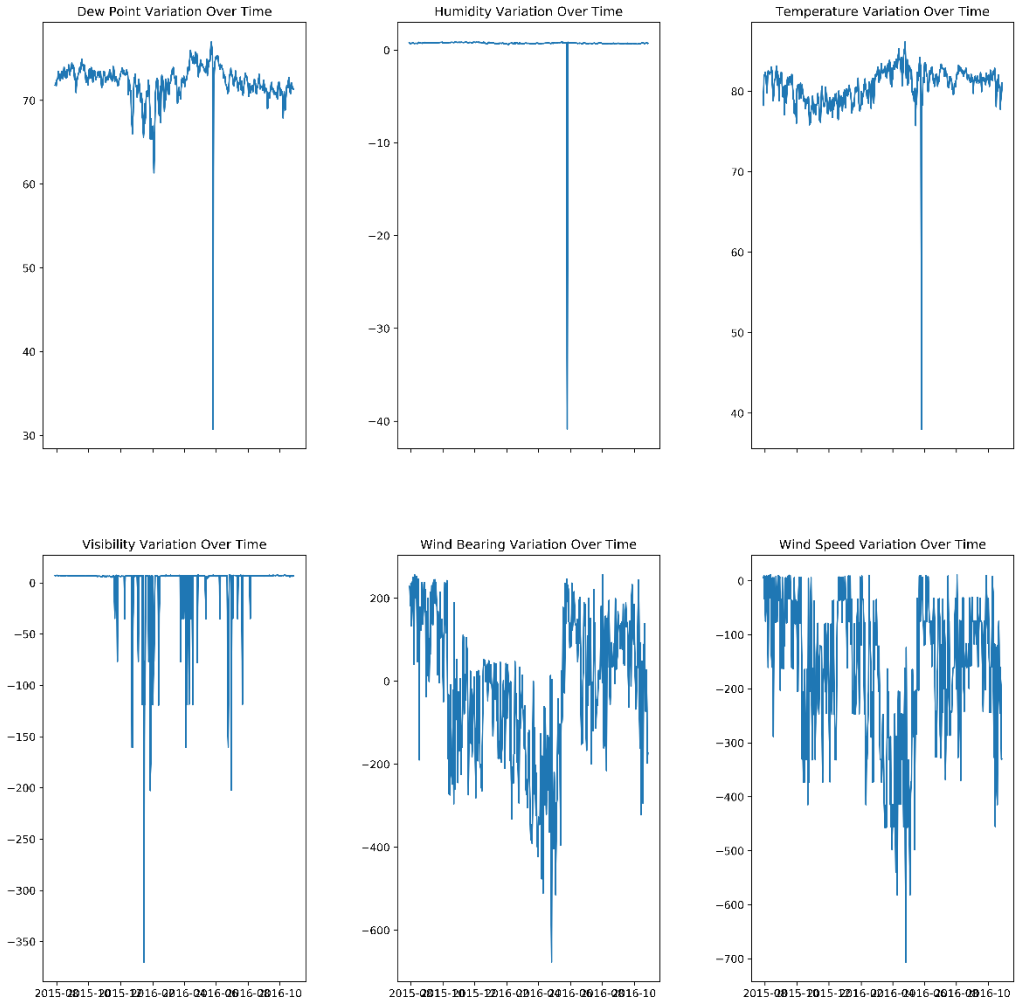

Figure A.1- Variation of Meteorological Variables with time (Broadlands Reservoir)

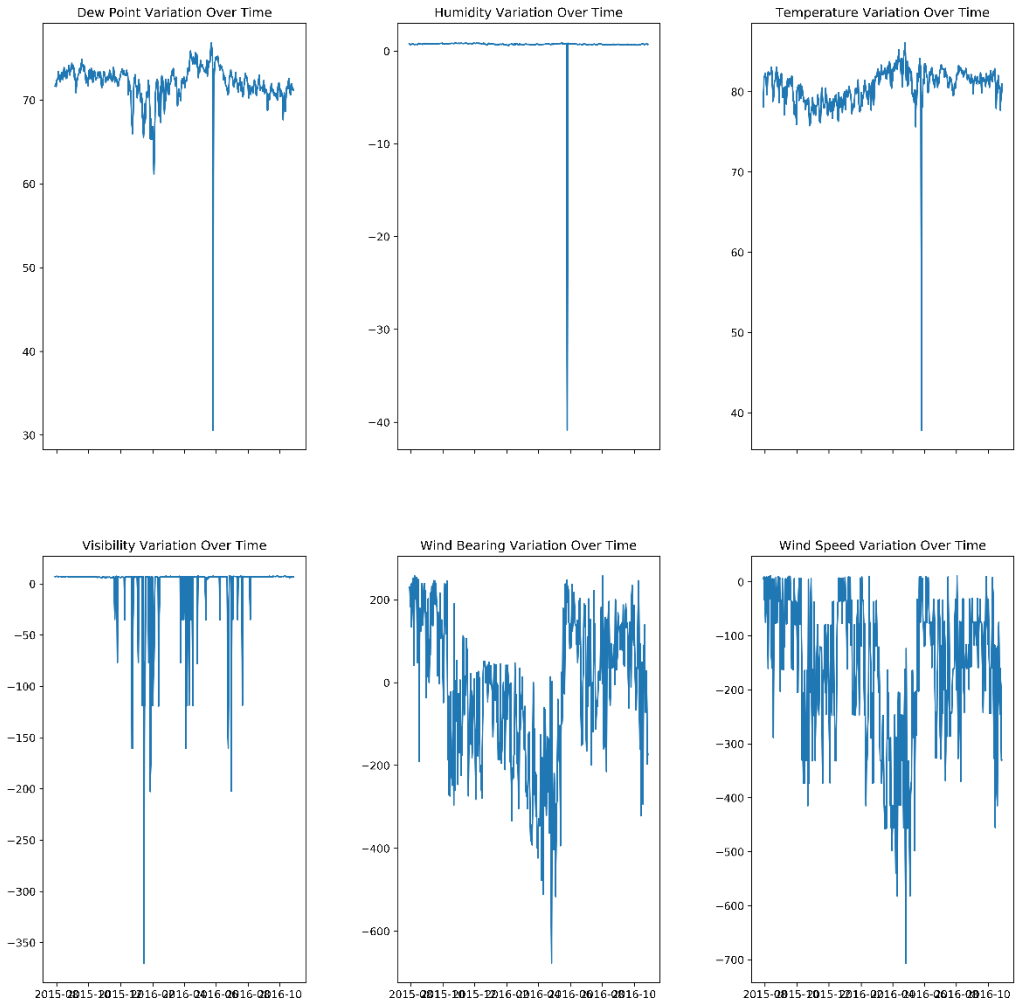

Figure A.2- Variation of Meteorological Variables with time (Laxapana Reservoir)

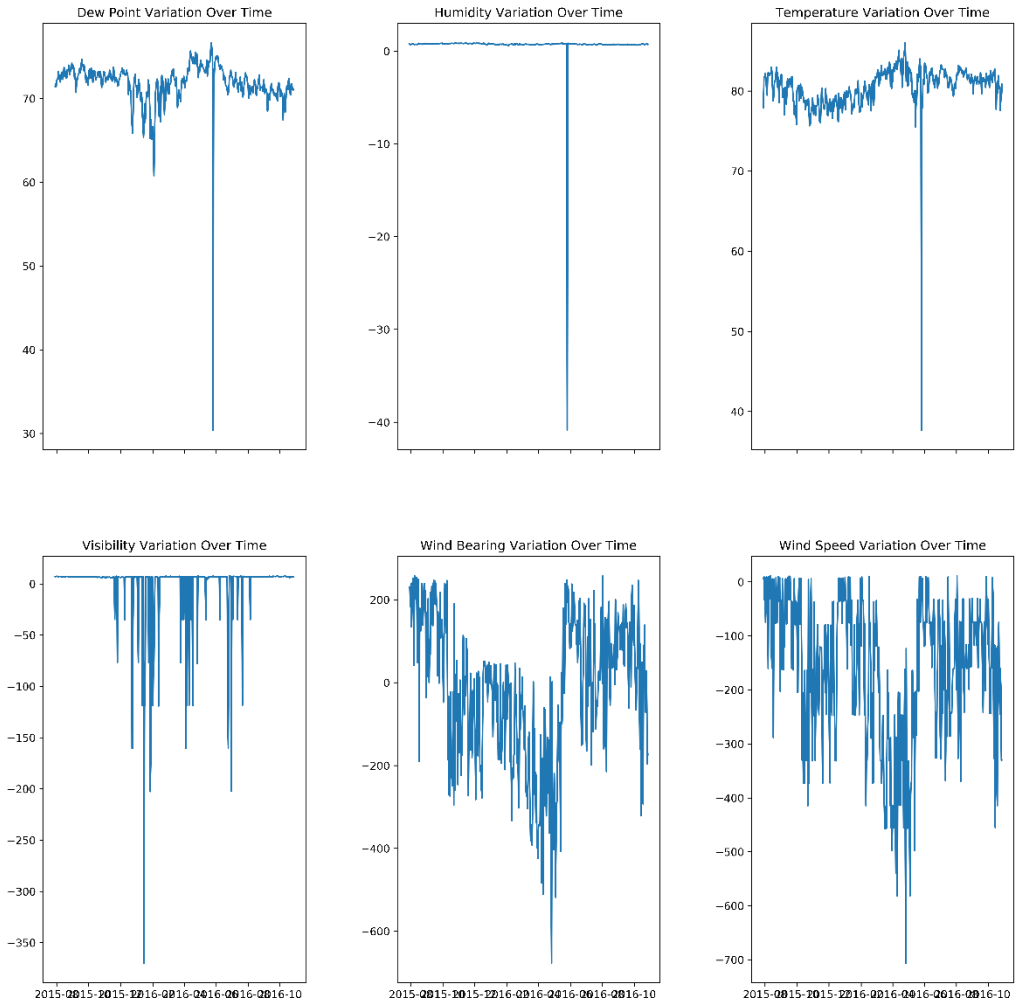

Figure A.3- Variation of Meteorological Variables with time (Norton Reservoir)

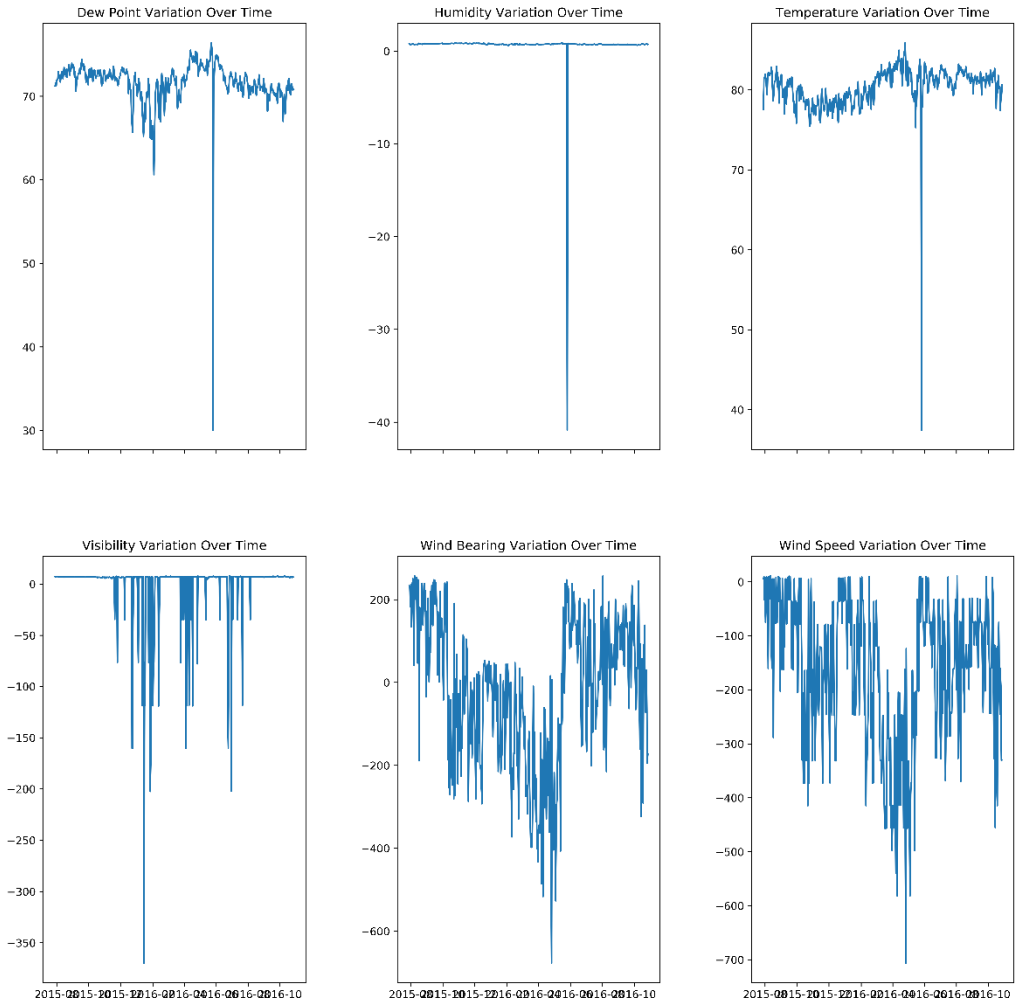

Figure A.4- Variation of Meteorological Variables with time (Upper Kotmale Reservoir)

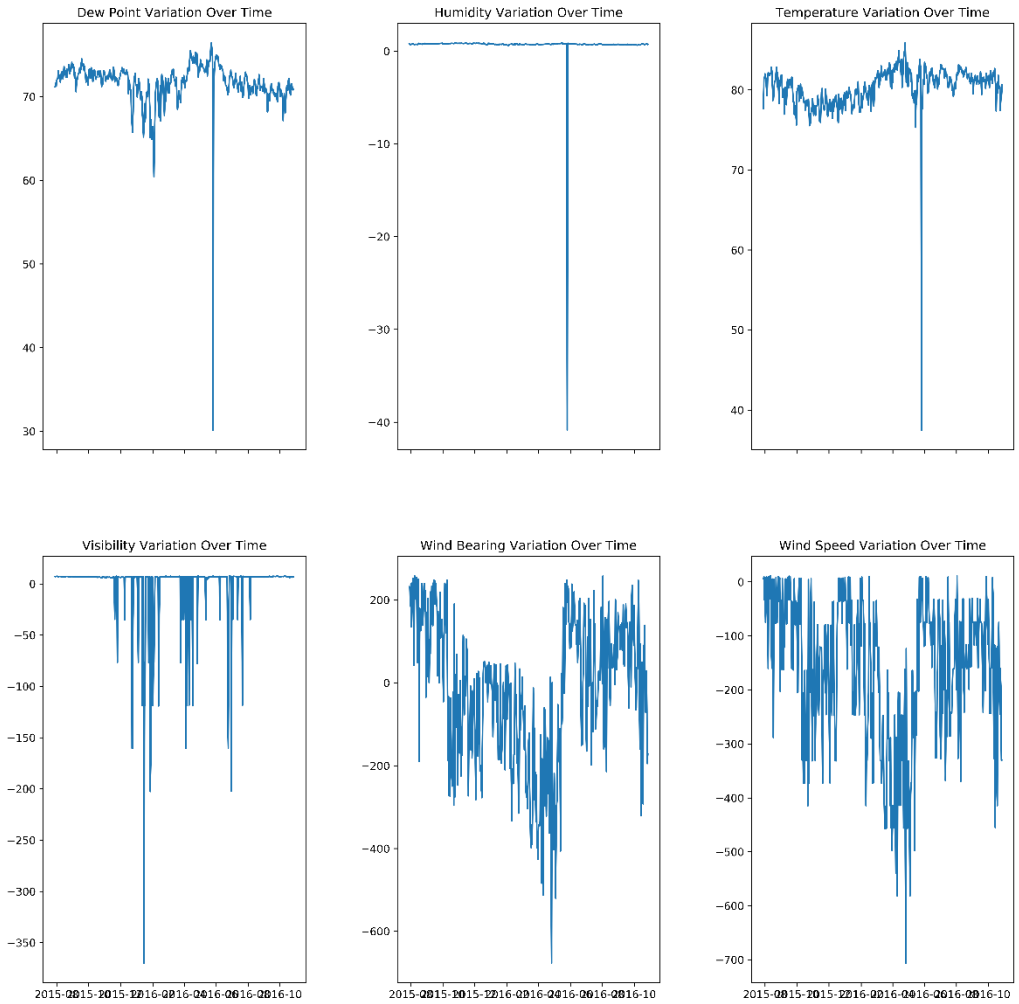

Figure A.5 - Variation of Meteorological Variables with time (Castlereigh Reservoir)

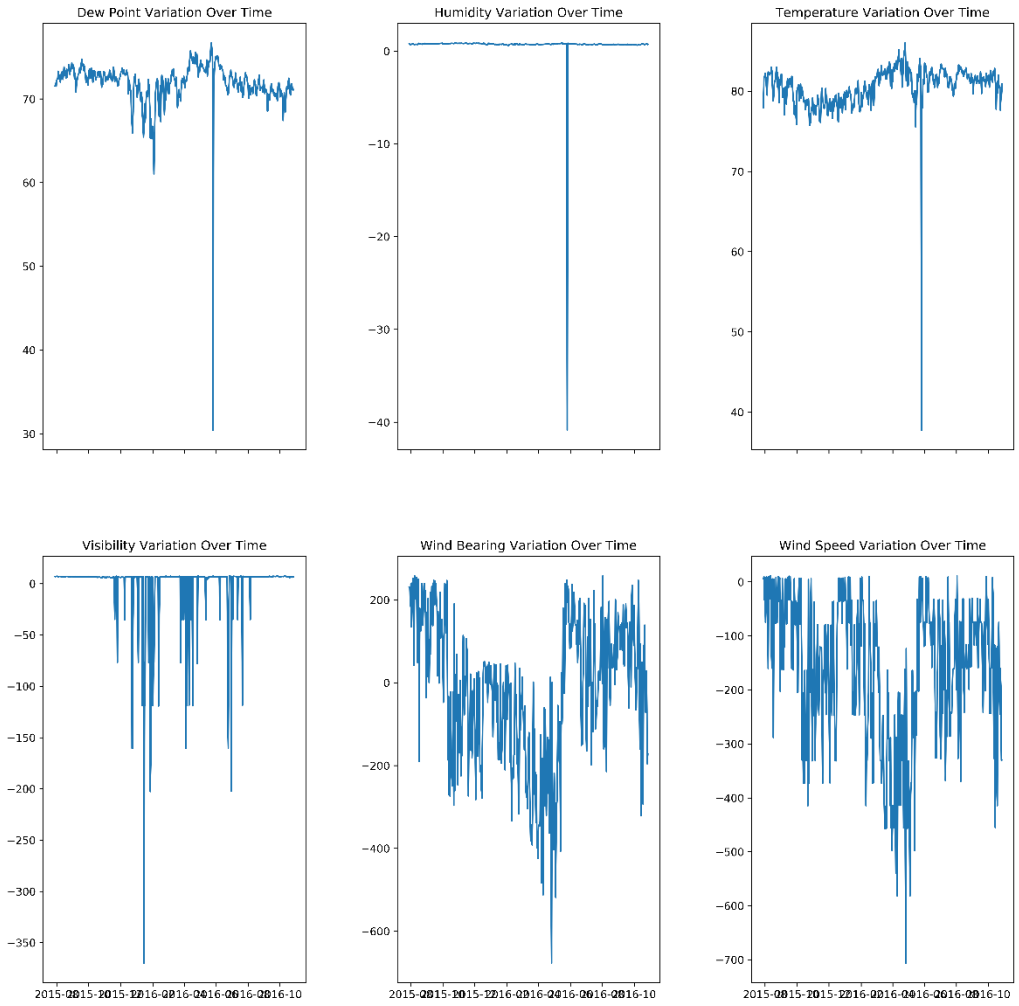

Figure A.6 - Variation of Meteorological Variables with time (Canyon Reservoir)

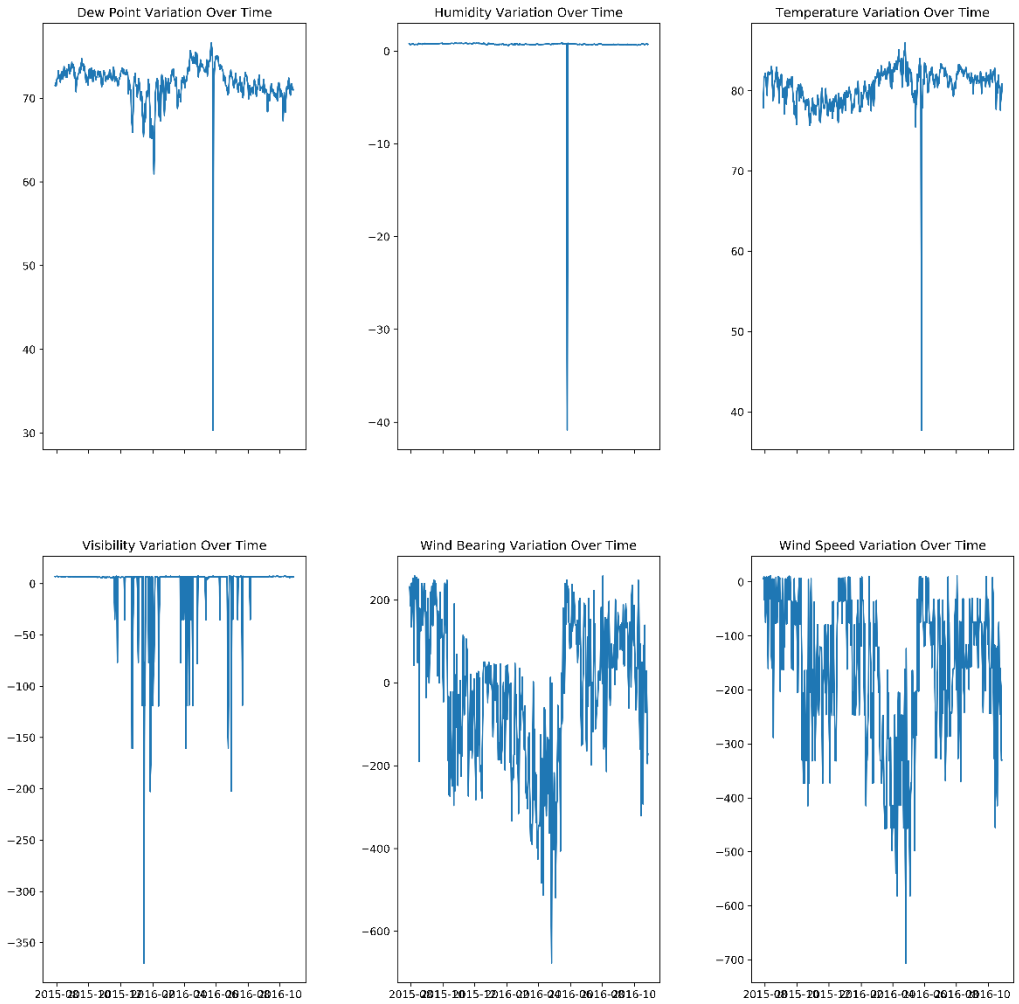

Figure A.7 - Variation of Meteorological Variables with time (Maskeliya Reservoir)

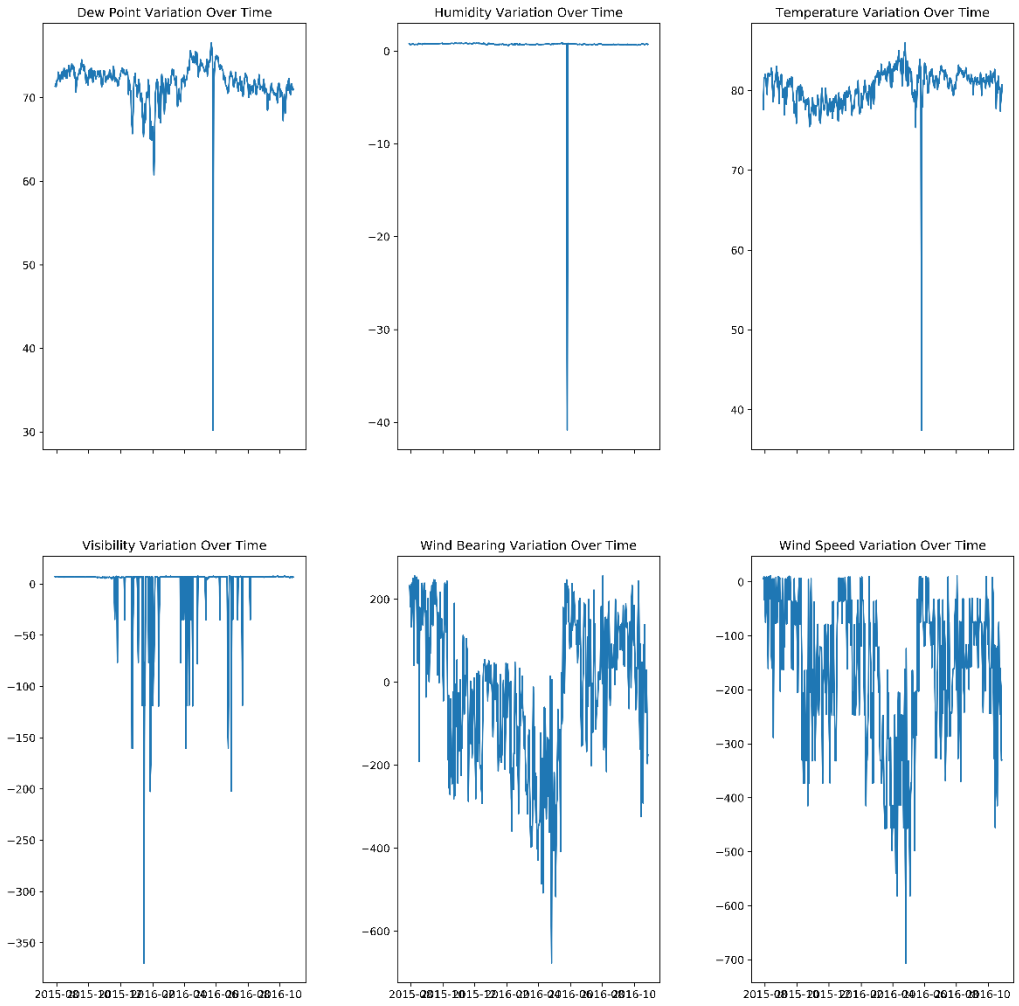

Figure A.8 - Variation of Meteorological Variables with time (Nilambe Reservoir)

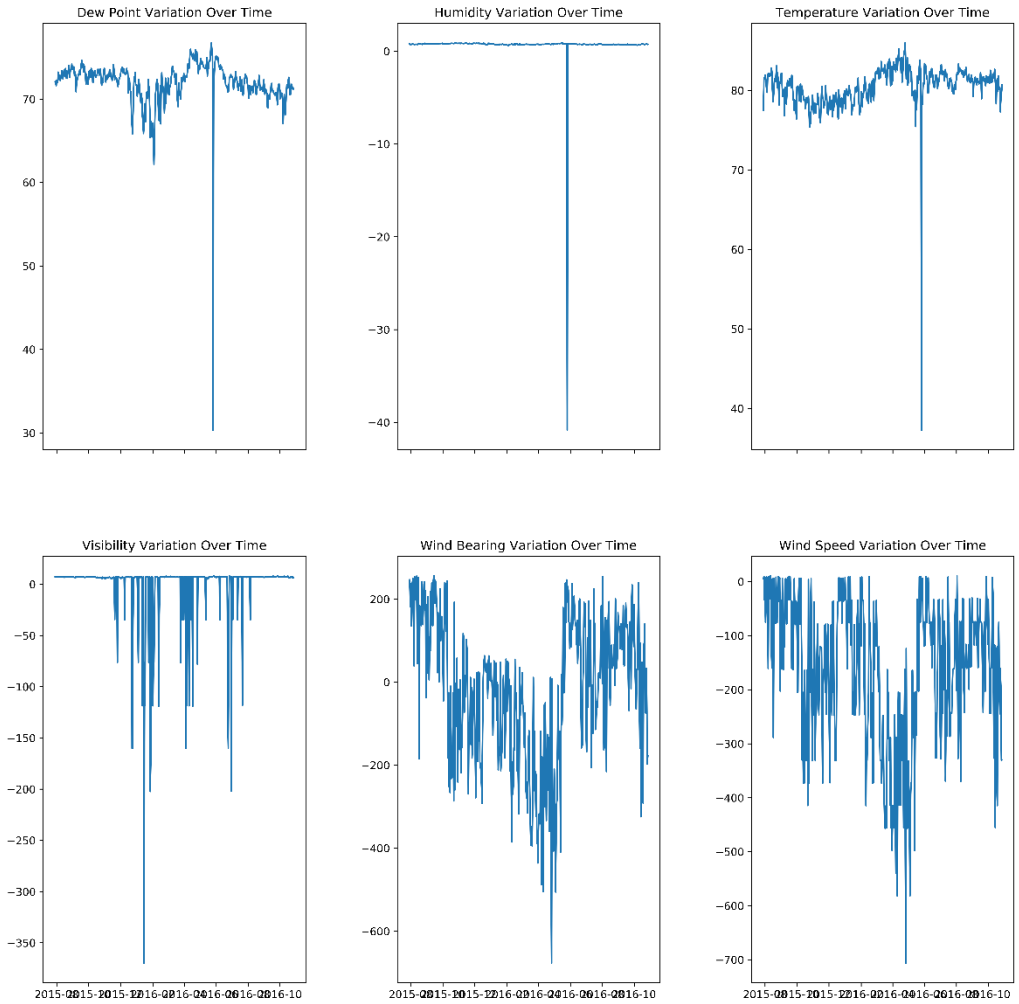

Figure A.9 - Variation of Meteorological Variables with time (Polgolla Reservoir)

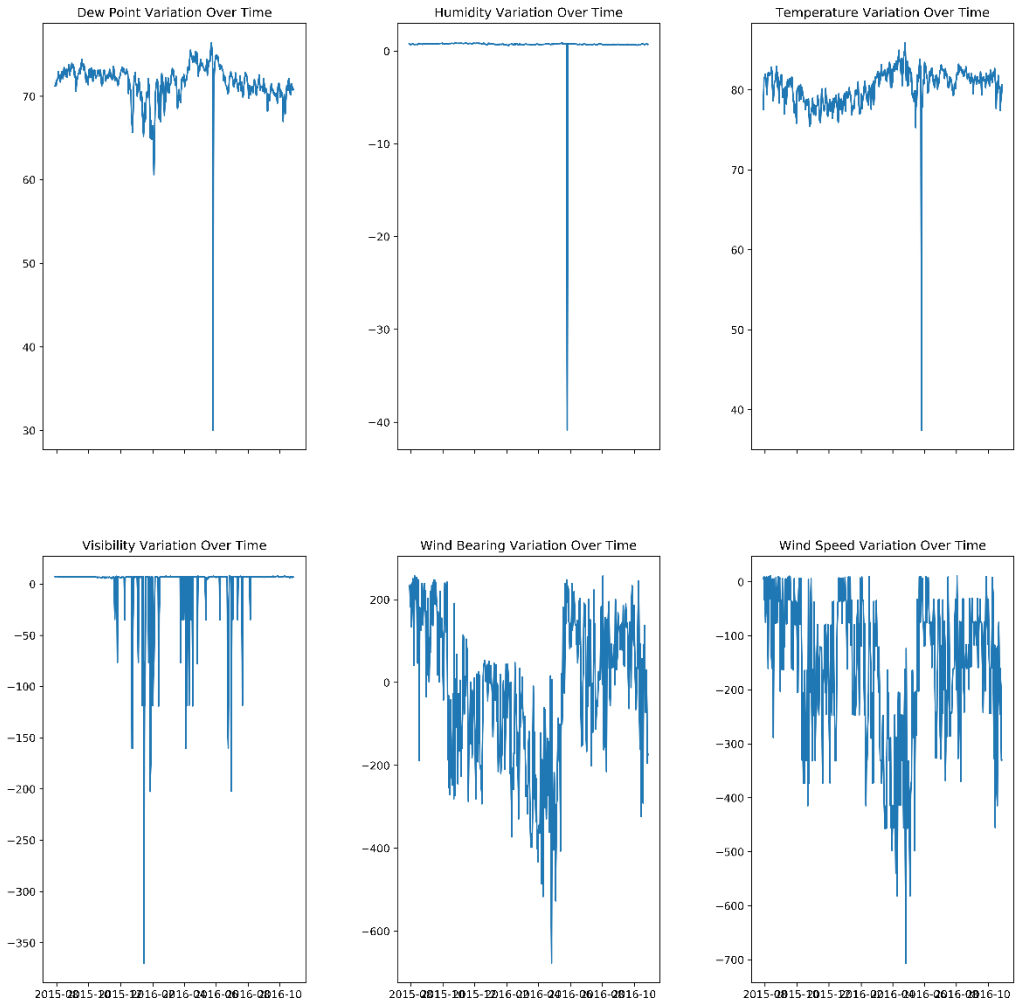

Figure A.10 - Variation of Meteorological Variables with time (Victoria Reservoir)

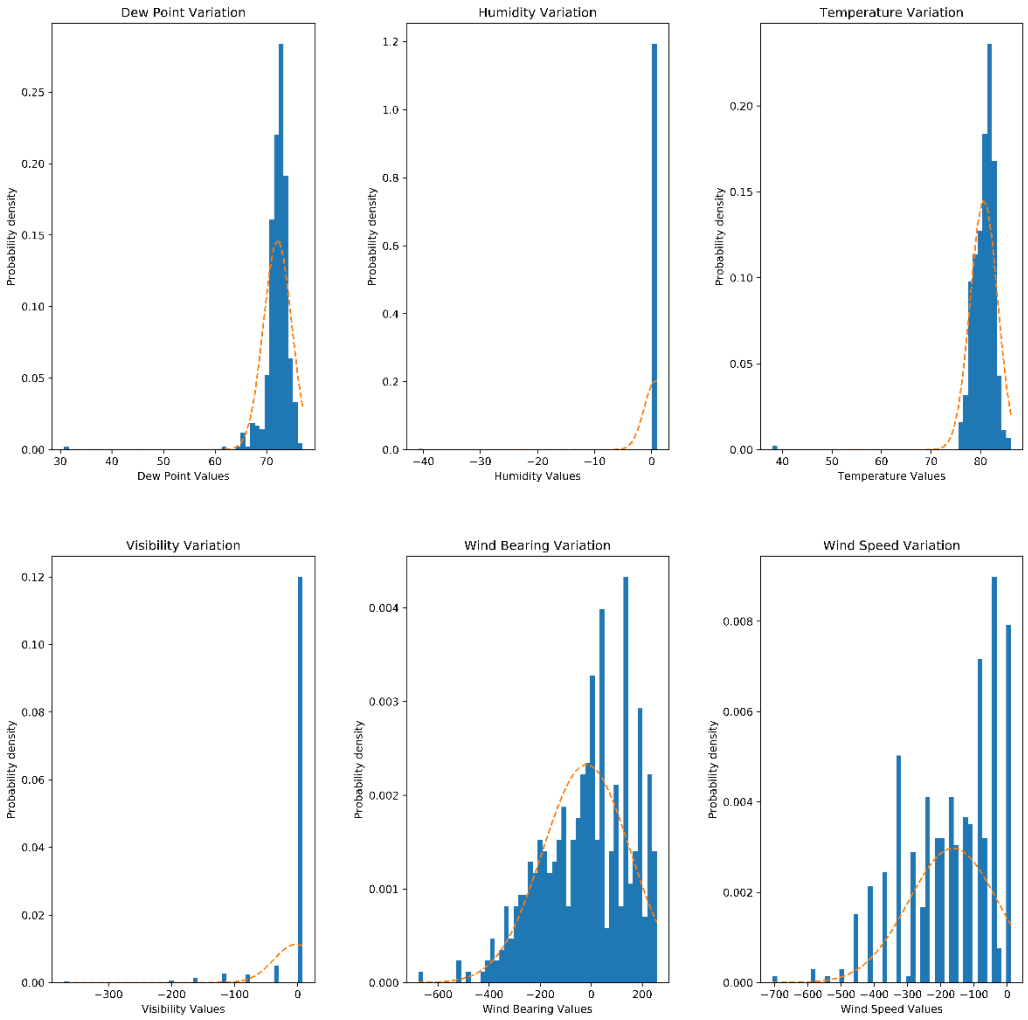

Figure A.11 - Histograms of Meteorological Variables (Broadlands Reservoir)

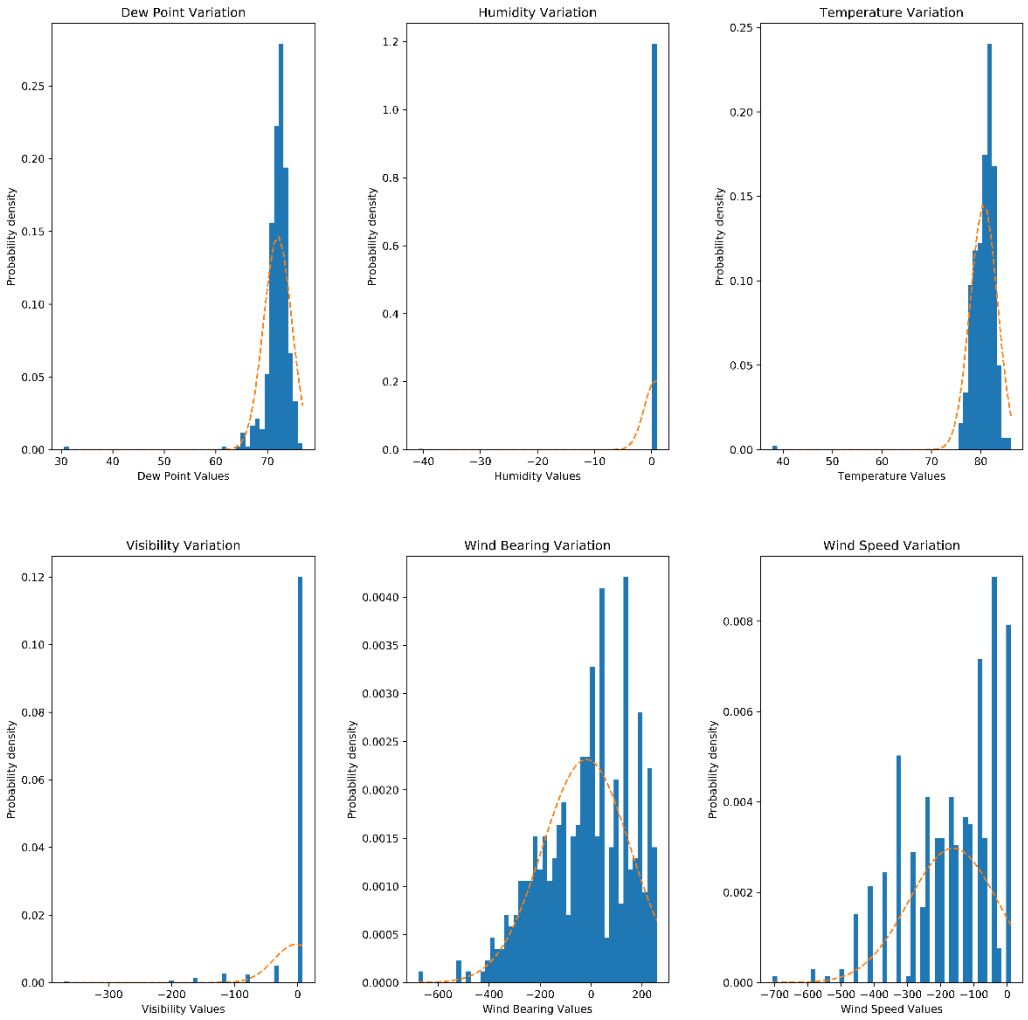

Figure A.12 - Histograms of Meteorological Variables (Laxapana Reservoir)

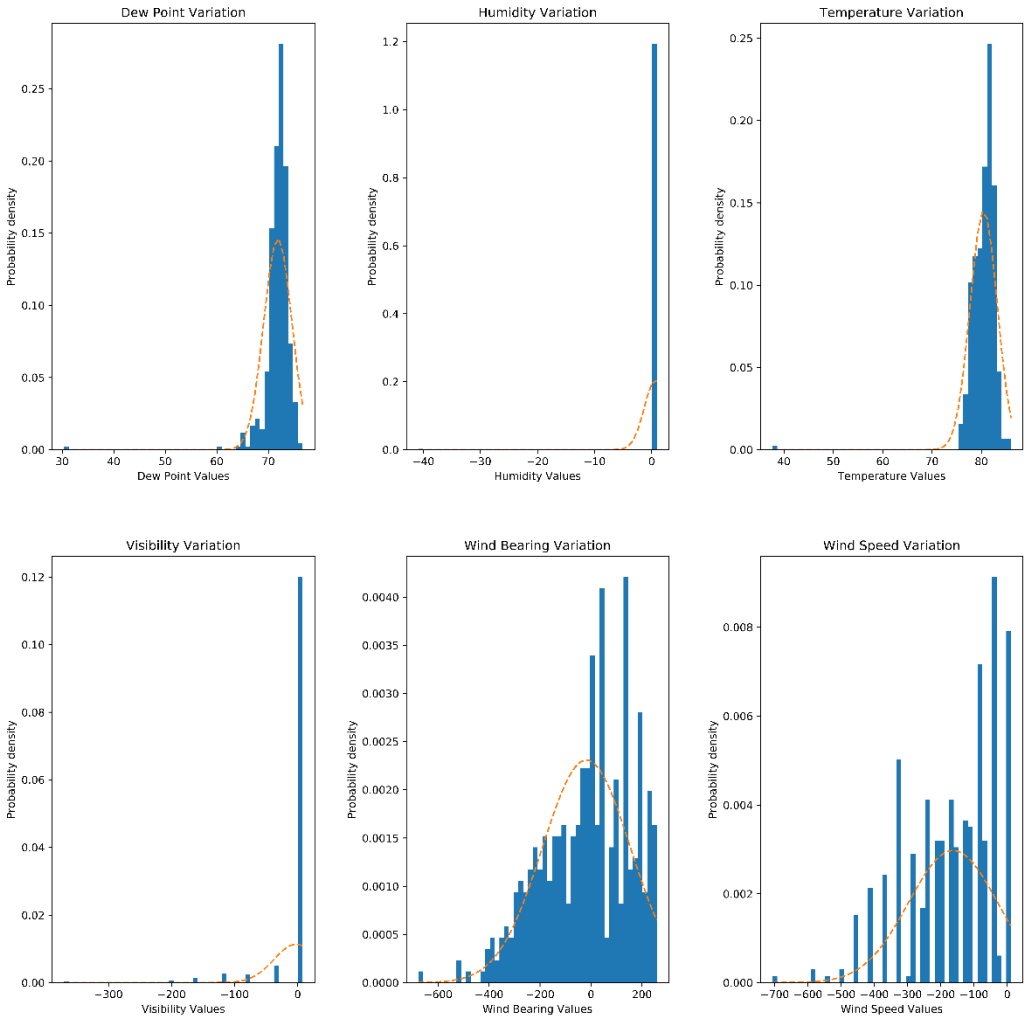

Figure A.13 - Histograms of Meteorological Variables (Norton Reservoir)

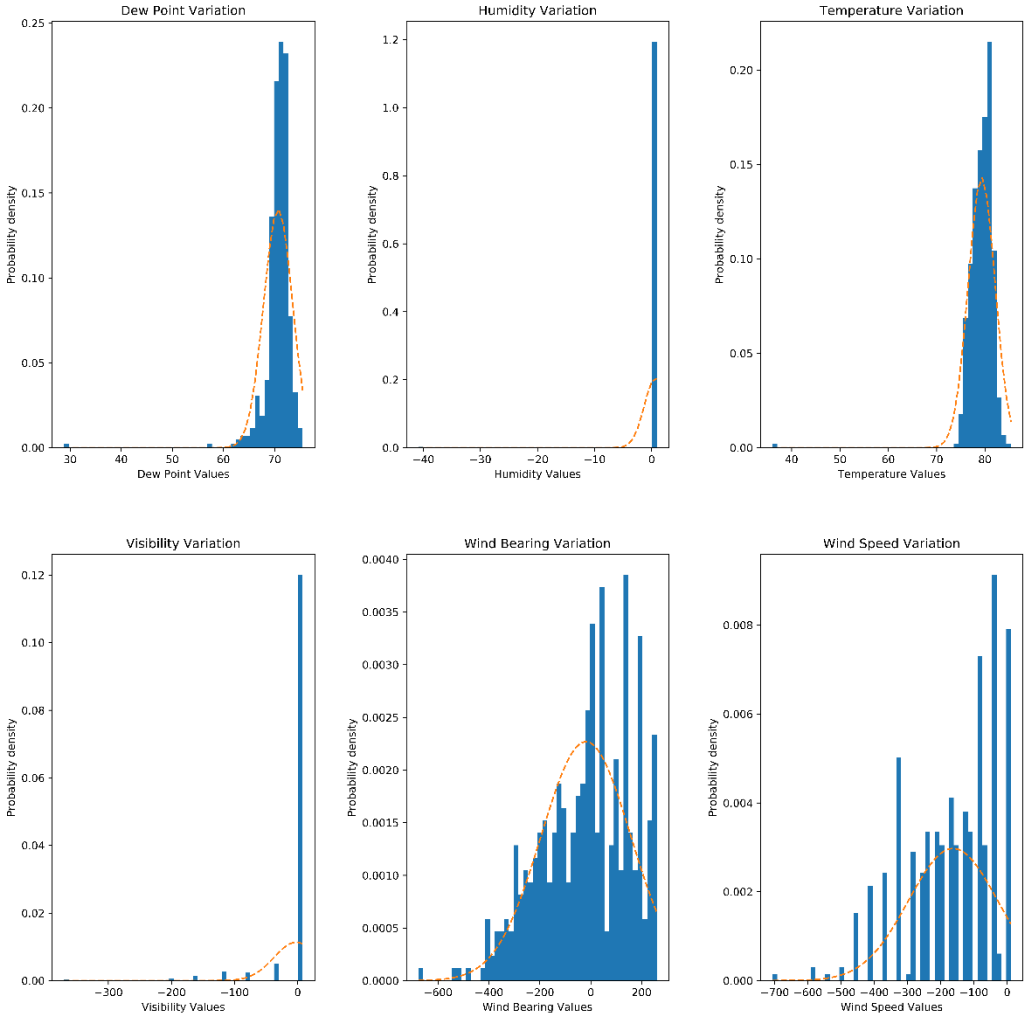

Figure A.14 - Histograms of Meteorological Variables (Upper Kotmale Reservoir)

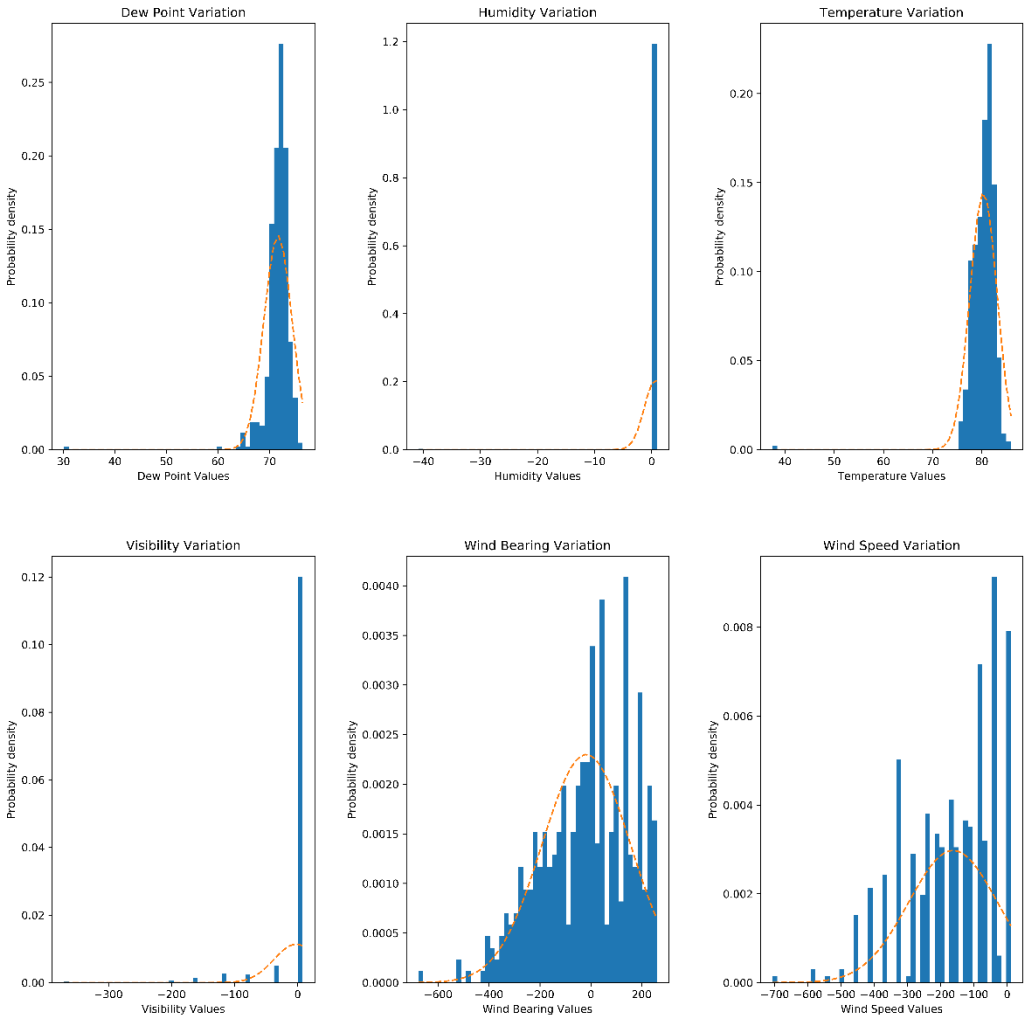

Figure A.15 - Histograms of Meteorological Variables (Castlereigh Reservoir)

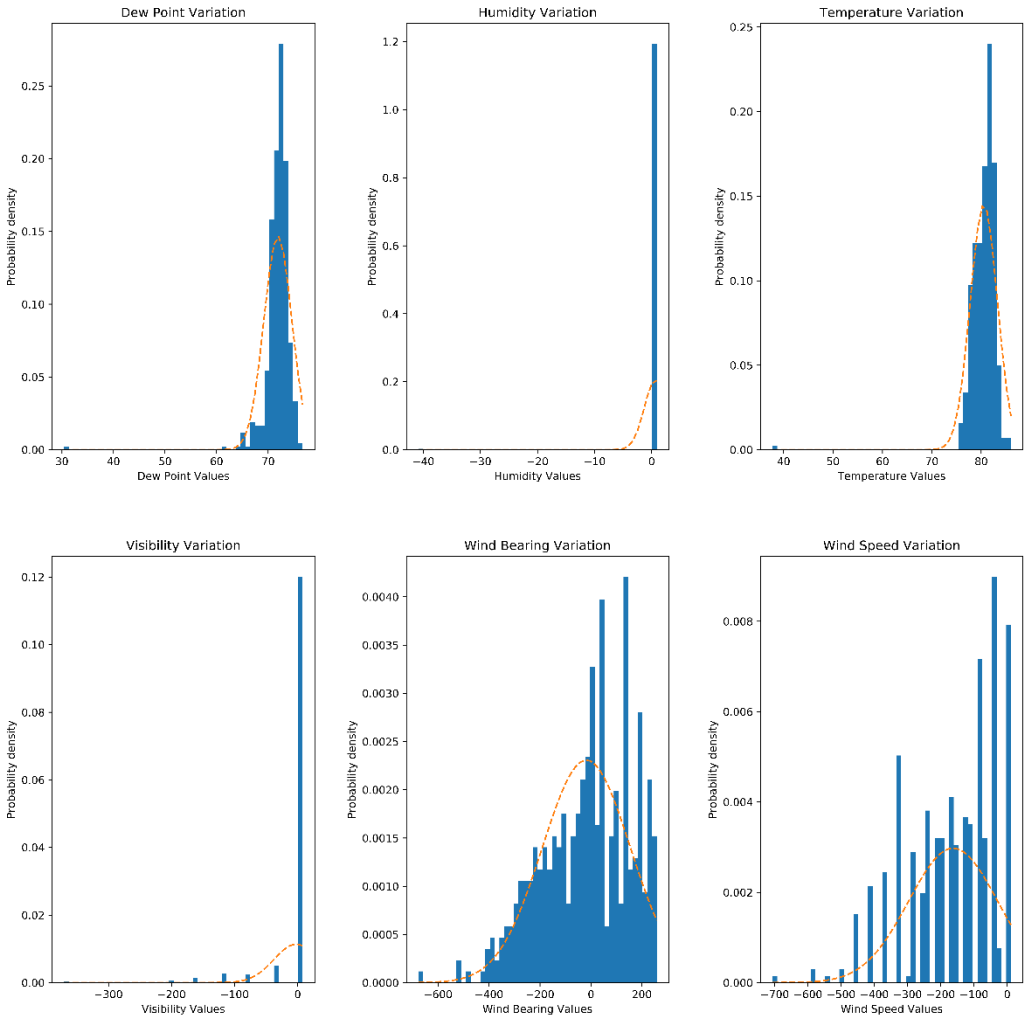

Figure A.16 - Histograms of Meteorological Variables (Canyon Reservoir)

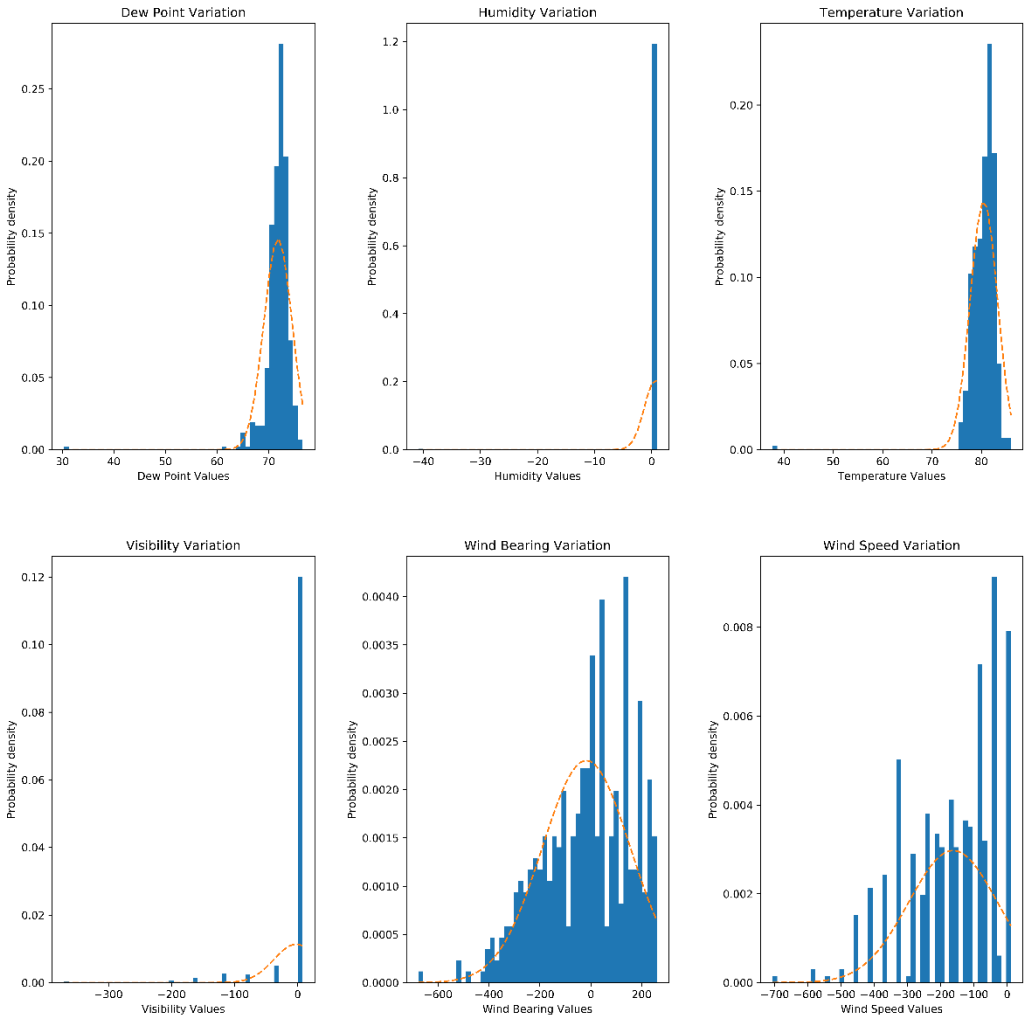

Figure A.17 - Histograms of Meteorological Variables (Maskeliya Reservoir)

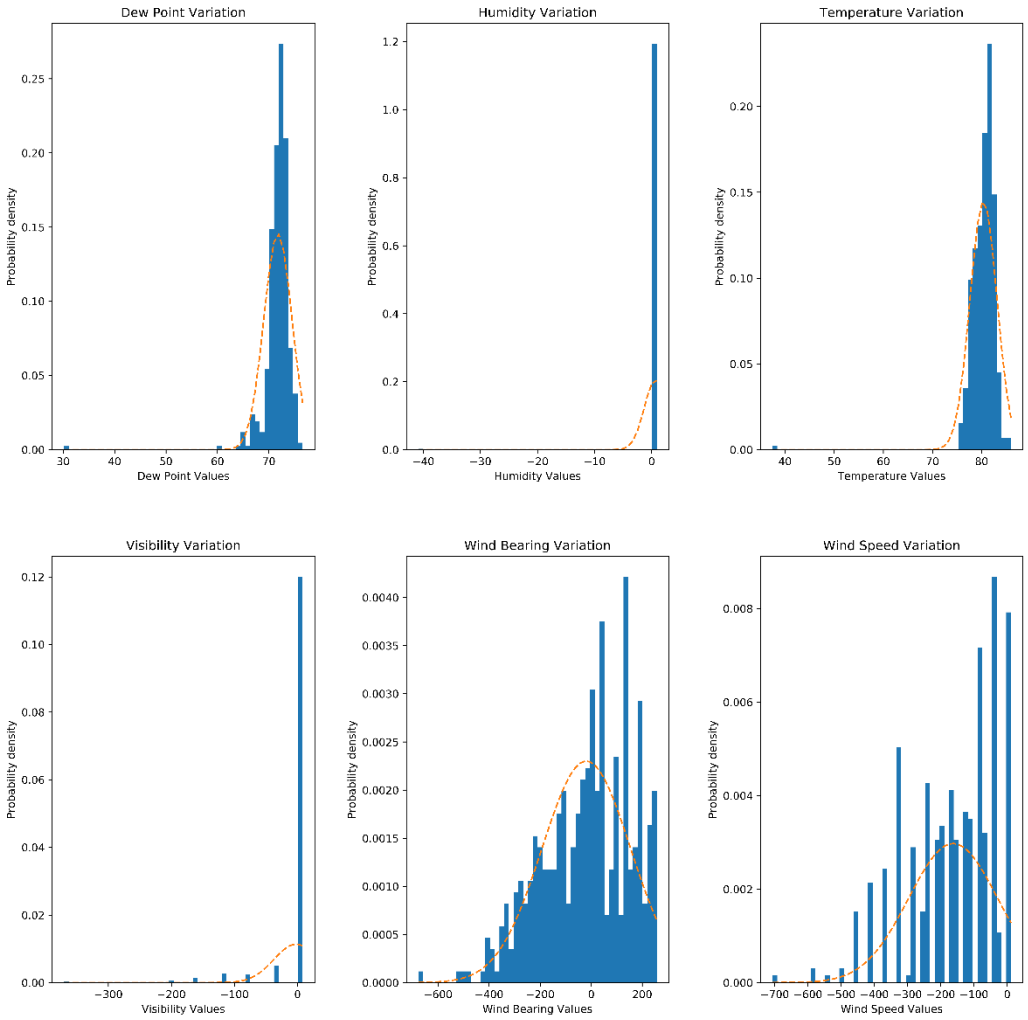

Figure A.18- Histograms of Meteorological Variables (Nilambe Reservoir)

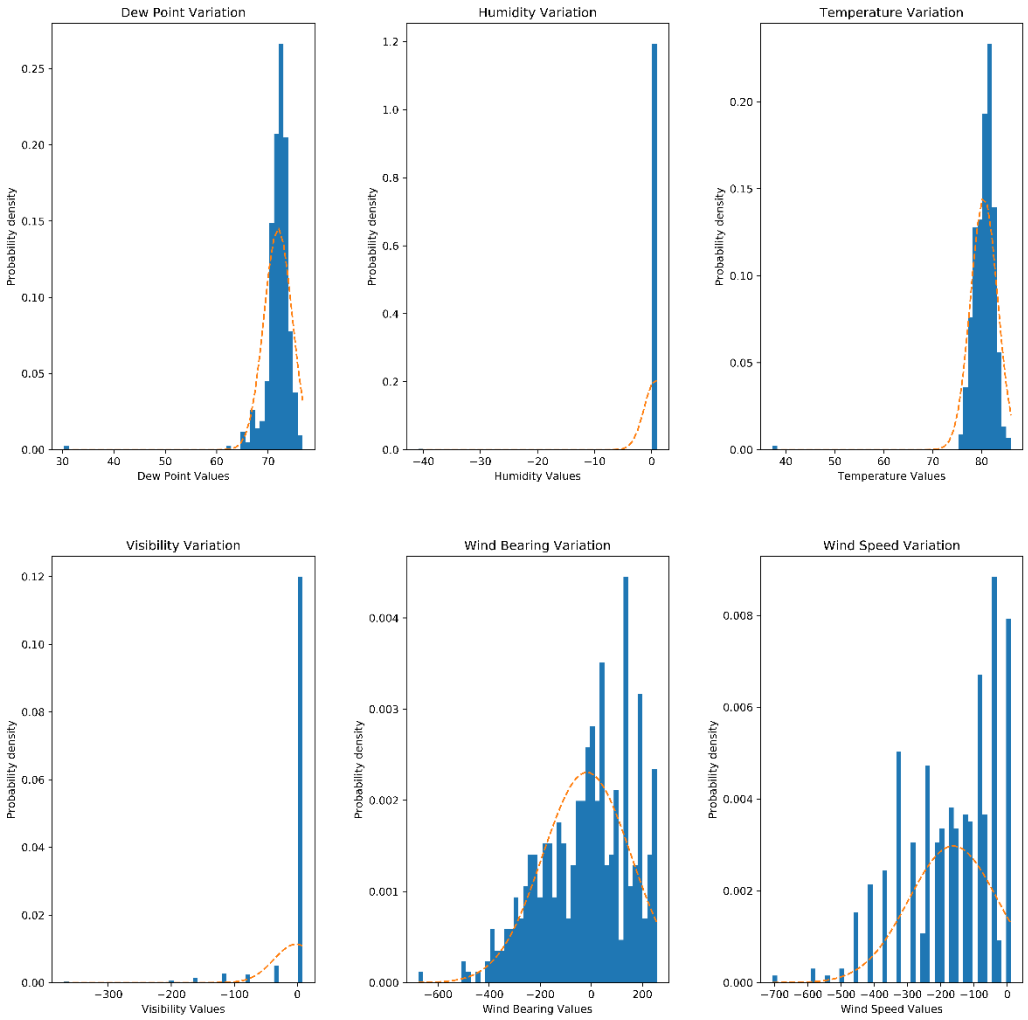

Figure A.19- Histograms of Meteorological Variables (Polgolla Reservoir)

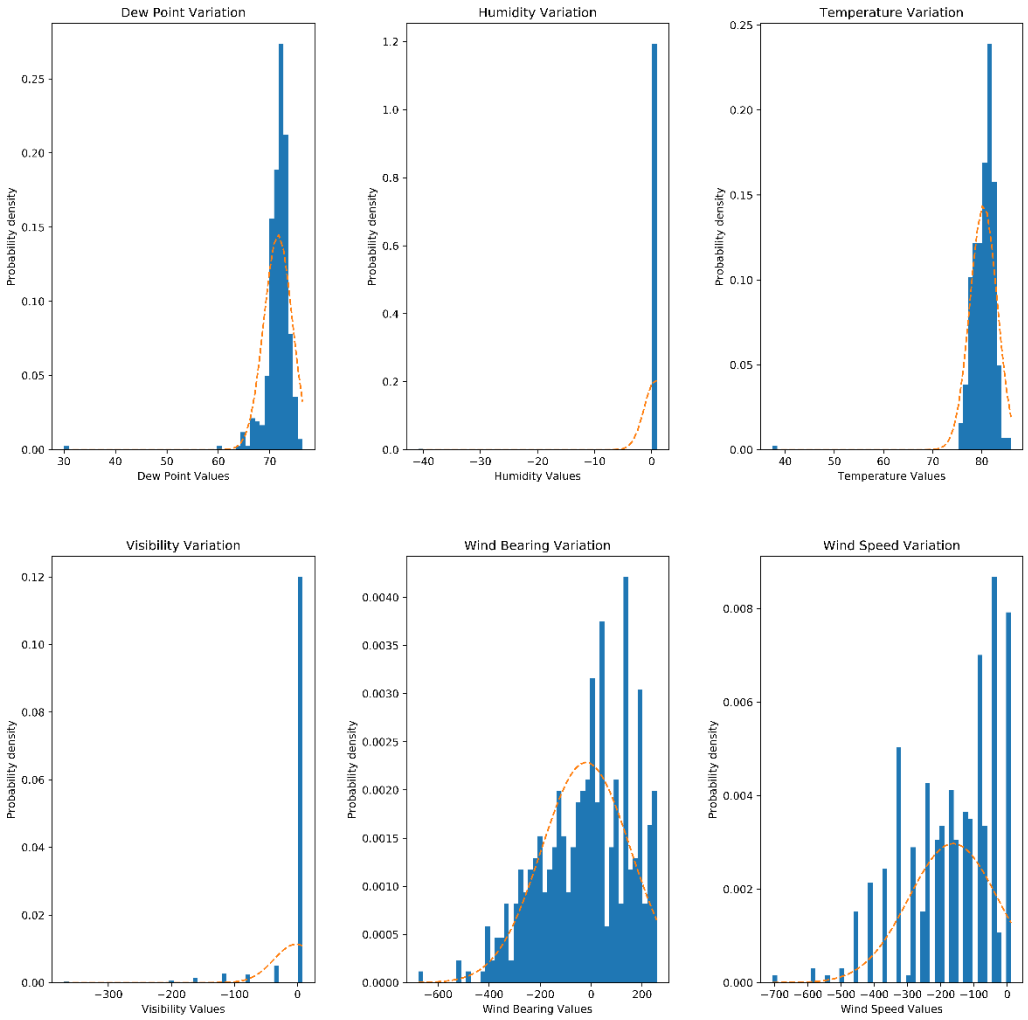

Figure A.20- Histograms of Meteorological Variables (Victoria Reservoir)

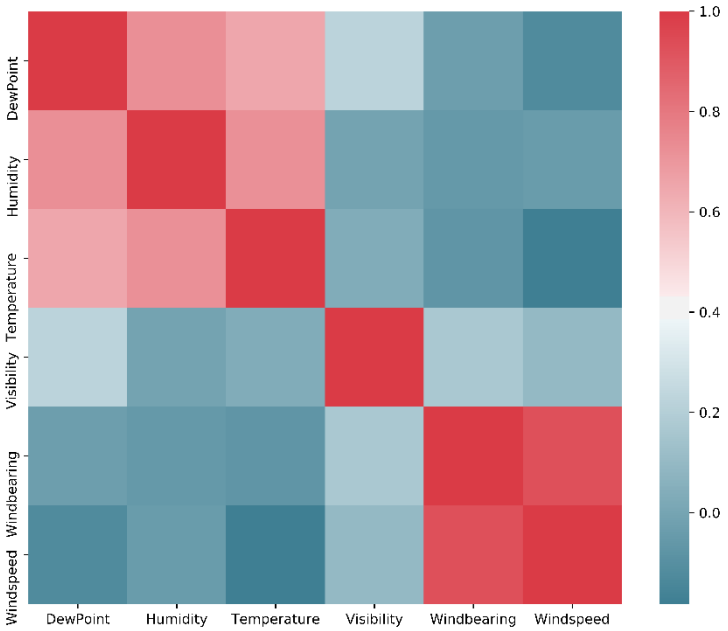

Figure A.21 - Correlation plot of Meteorological Variables (Broadlands Reservoir)

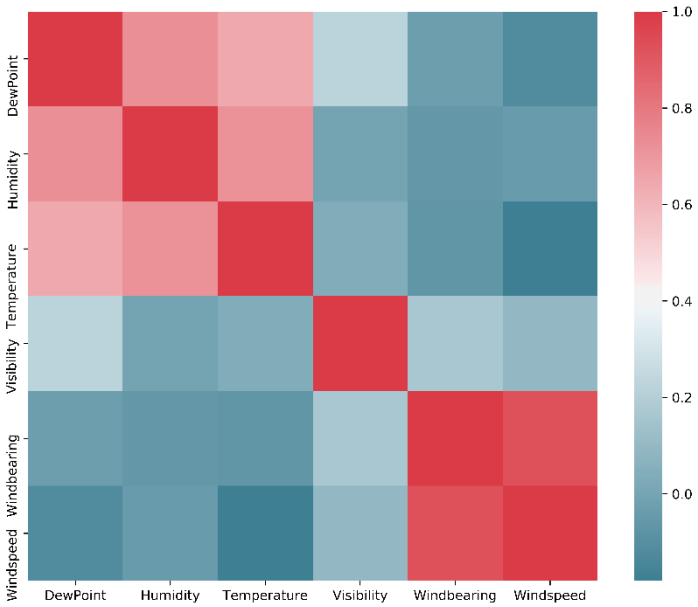

Figure A.22- Correlation plot of Meteorological Variables (Laxapana Reservoir)

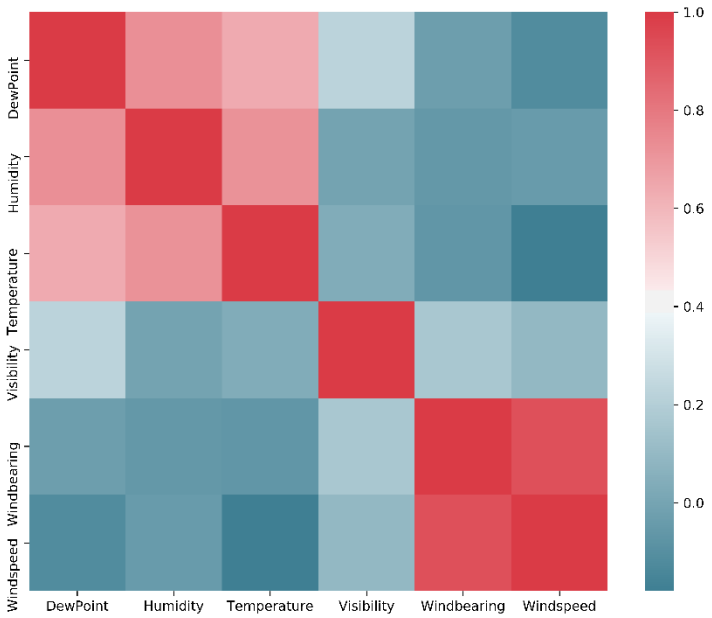

Figure A.23- Correlation plot of Meteorological Variables (Norton Reservoir)

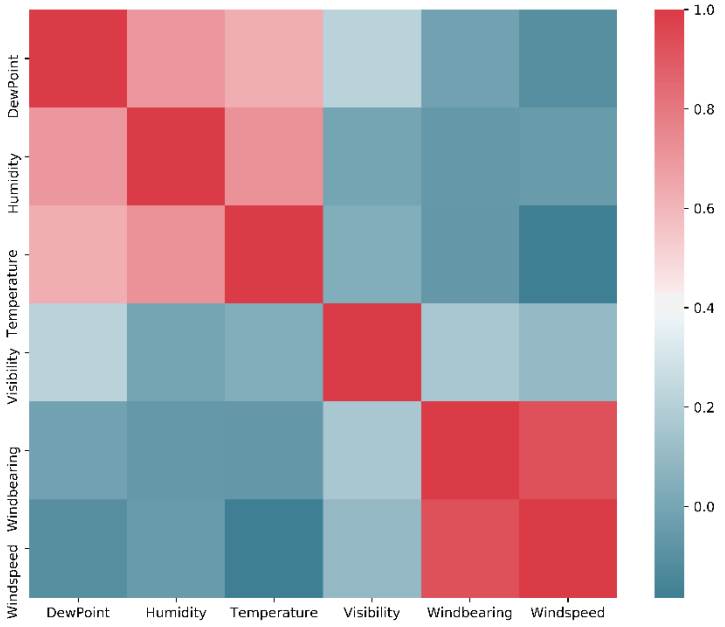

Figure A.24- Correlation plot of Meteorological Variables (Upper Kotmale Reservoir)

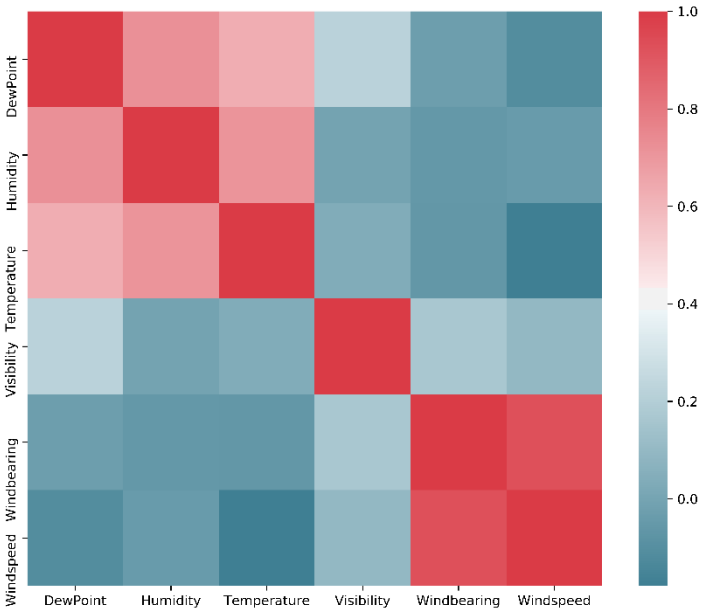

Figure A.25- Correlation plot of Meteorological Variables (Castlereigh Reservoir)

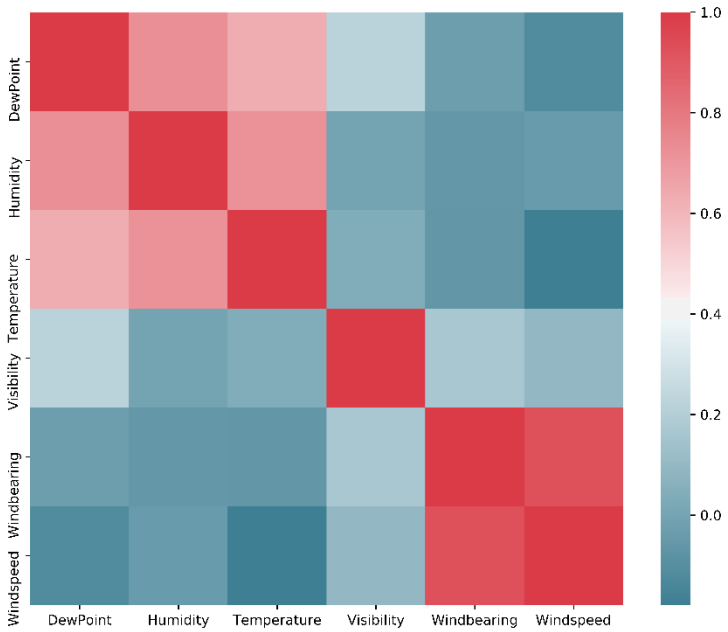

Figure A.26- Correlation plot of Meteorological Variables (Canyon Reservoir)

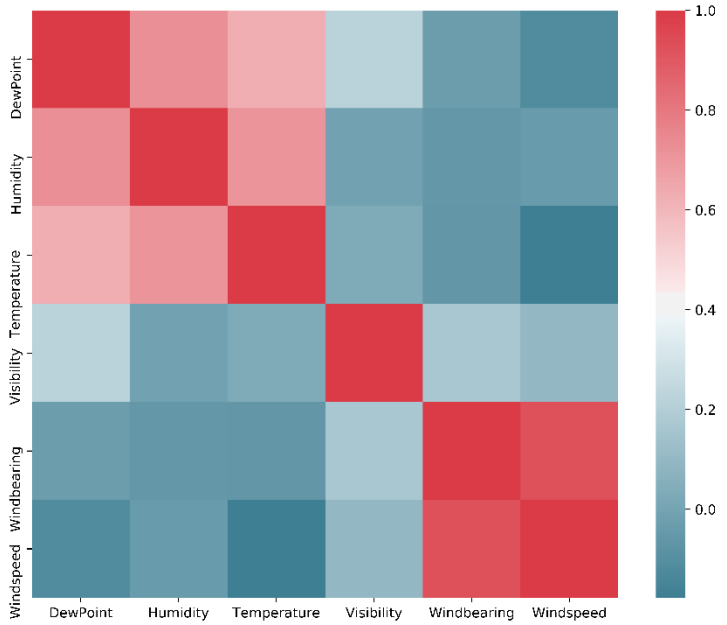

Figure A.27- Correlation plot of Meteorological Variables (Maskeliya Reservoir)

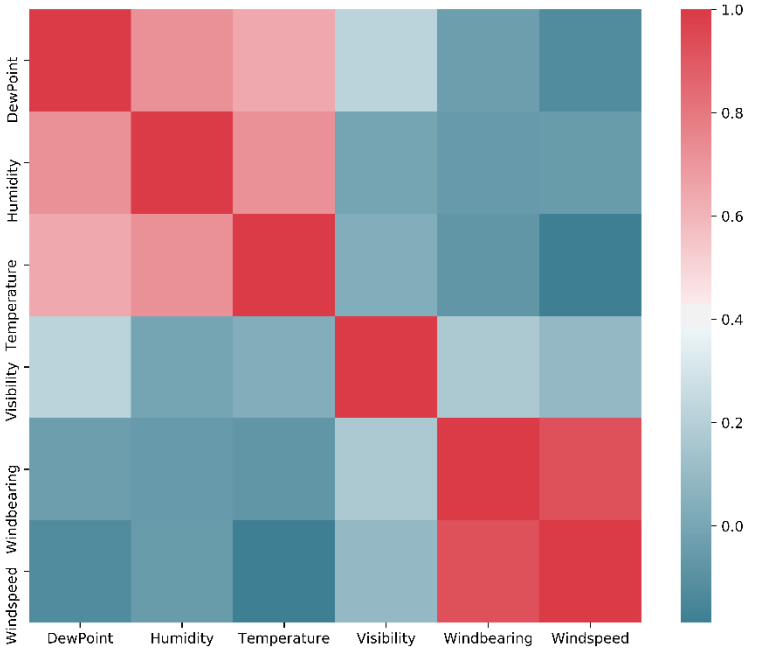

Figure A.28- Correlation plot of Meteorological Variables (Nilambe Reservoir)

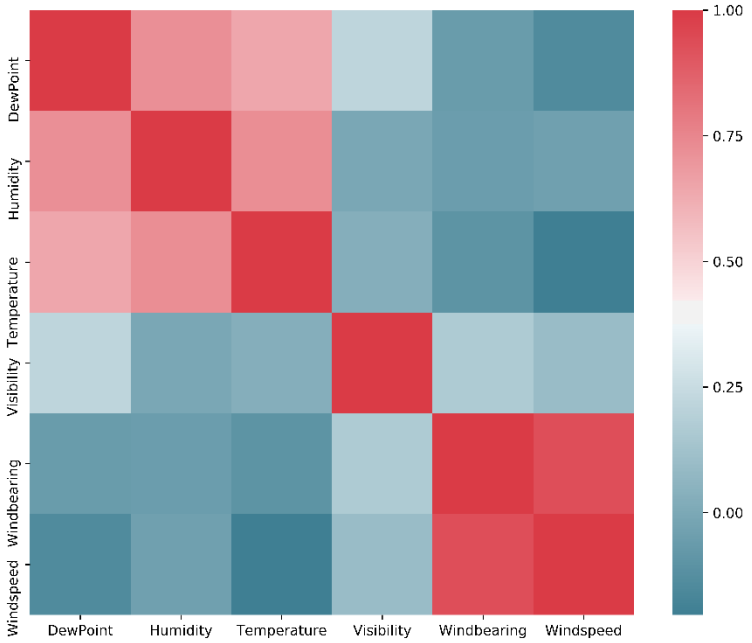

Figure A.29- Correlation plot of Meteorological Variables (Polgolla Reservoir)

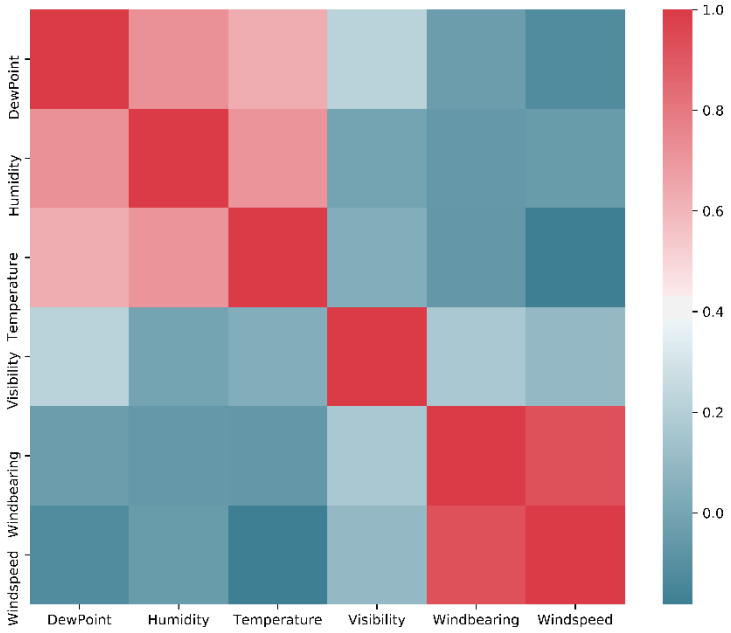

Figure A.30- Correlation plot of Meteorological Variables (Victoria Reservoir)

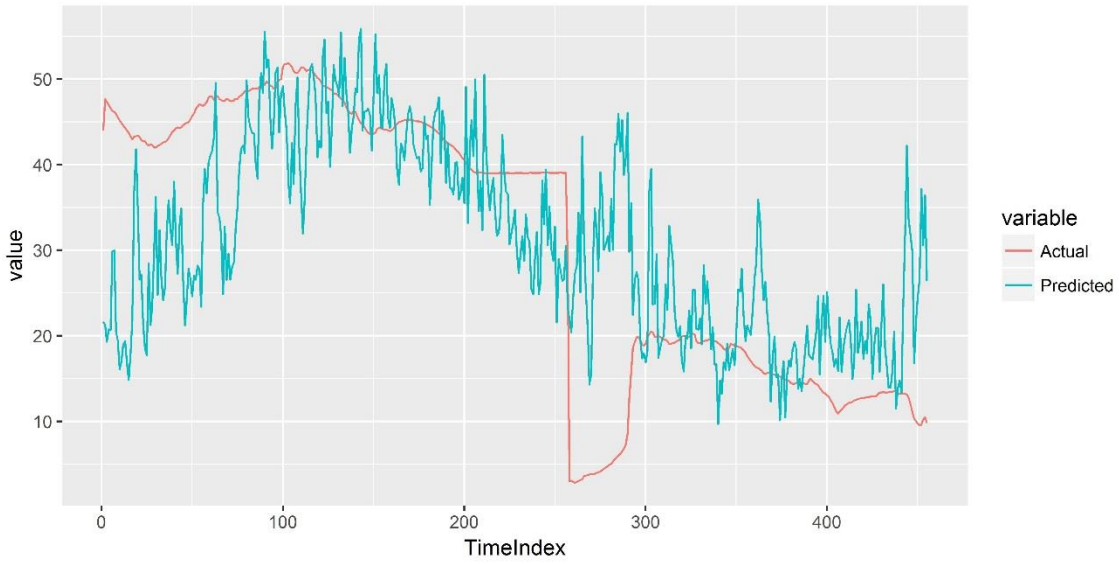

Figure A.31 - Predicted Vs Actual Values Over Time using Epsilon SVR (Linear Kernel)

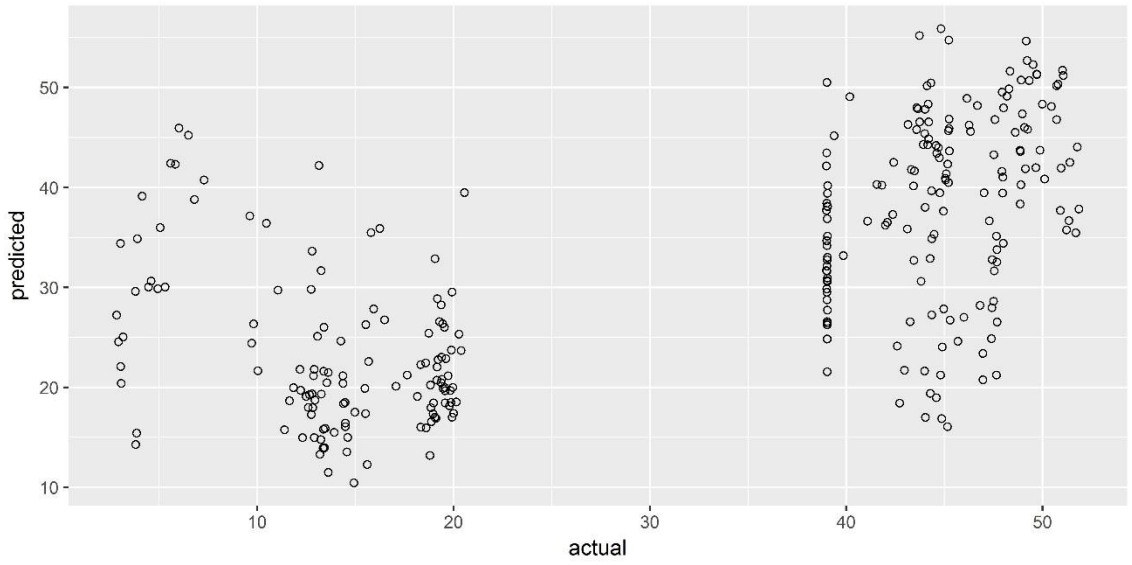

Figure A.32- Predicted Vs Actual Values Scatterplot for Training Data using Epsilon SVR (Linear Kernel)

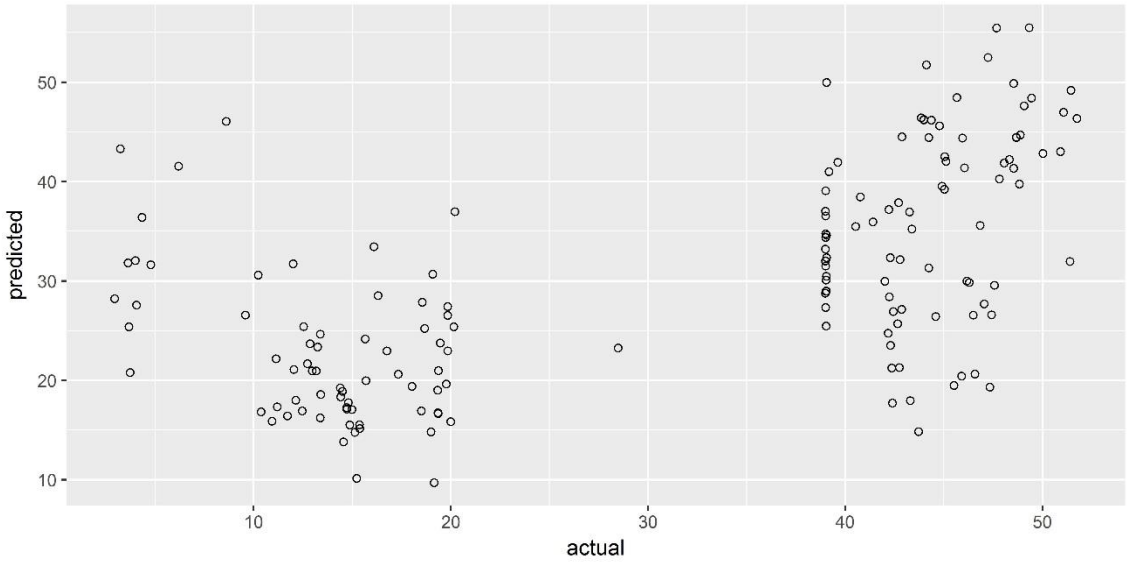

Figure A.33- Predicted Vs Actual Values Scatterplot for Test Data using Epsilon SVR (Linear Kernel)

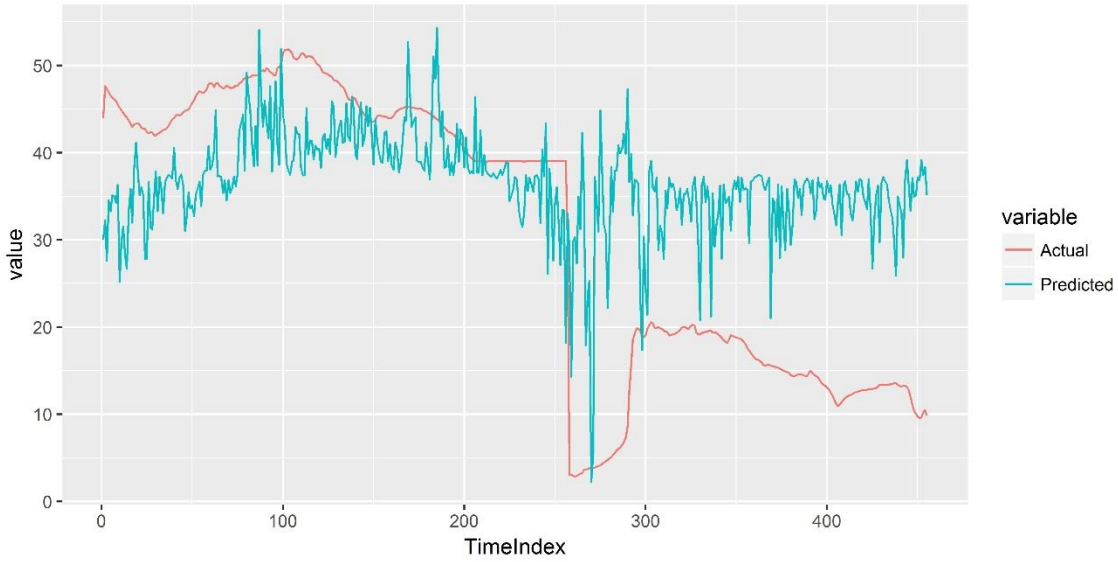

Figure A.34- Predicted Vs Actual Values Over Time using Epsilon SVR (Polynomial Kernel)

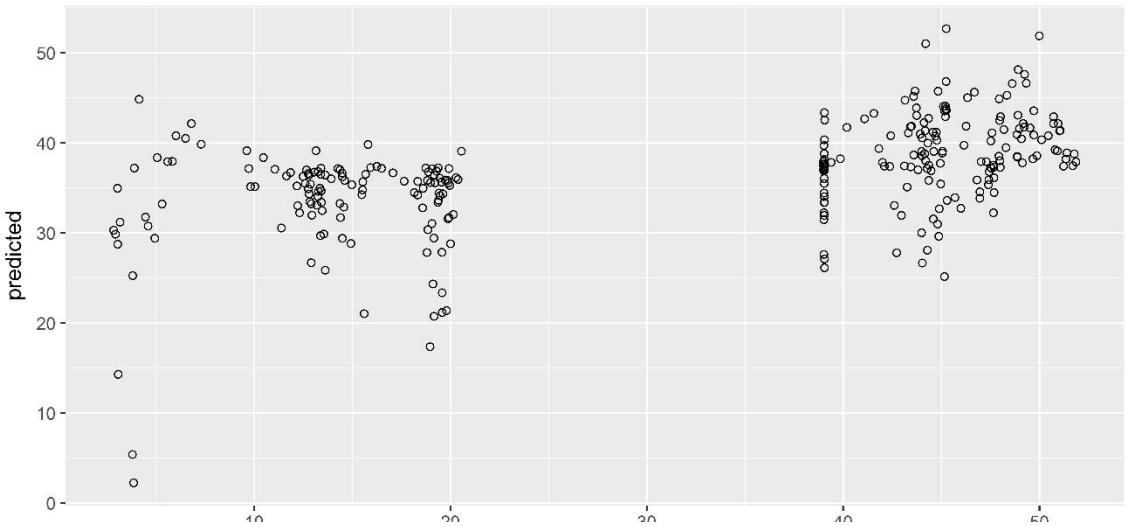

Figure A.35- Predicted Vs Actual Values Scatterplot for Training Data using Epsilon SVR (Polynomial Kernel)

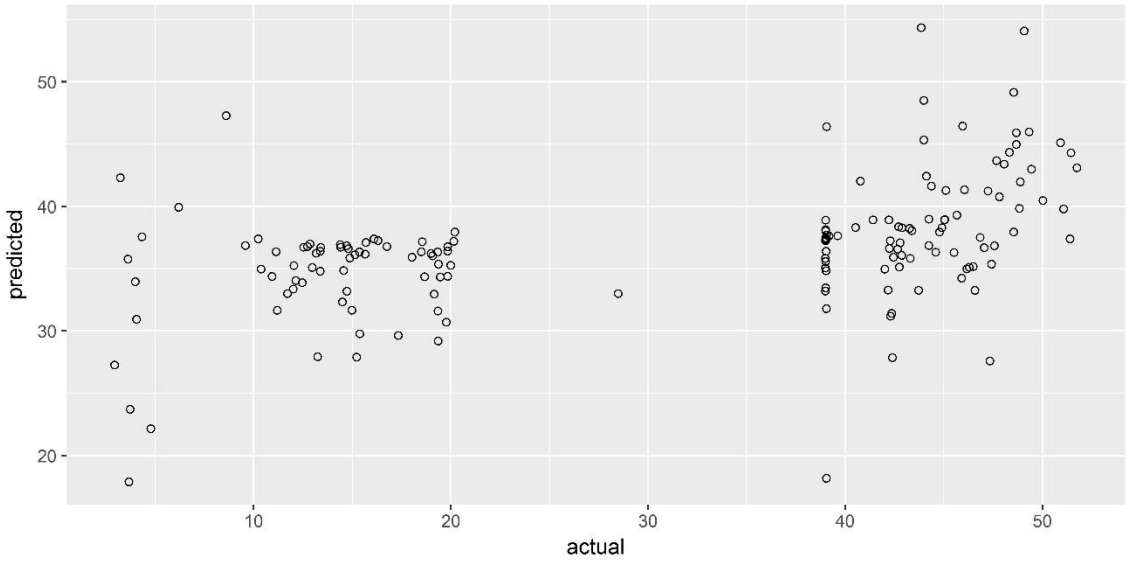

Figure A.36- Predicted Vs Actual Values Scatterplot for Test Data using Epsilon SVR (Polynomial Kernel)

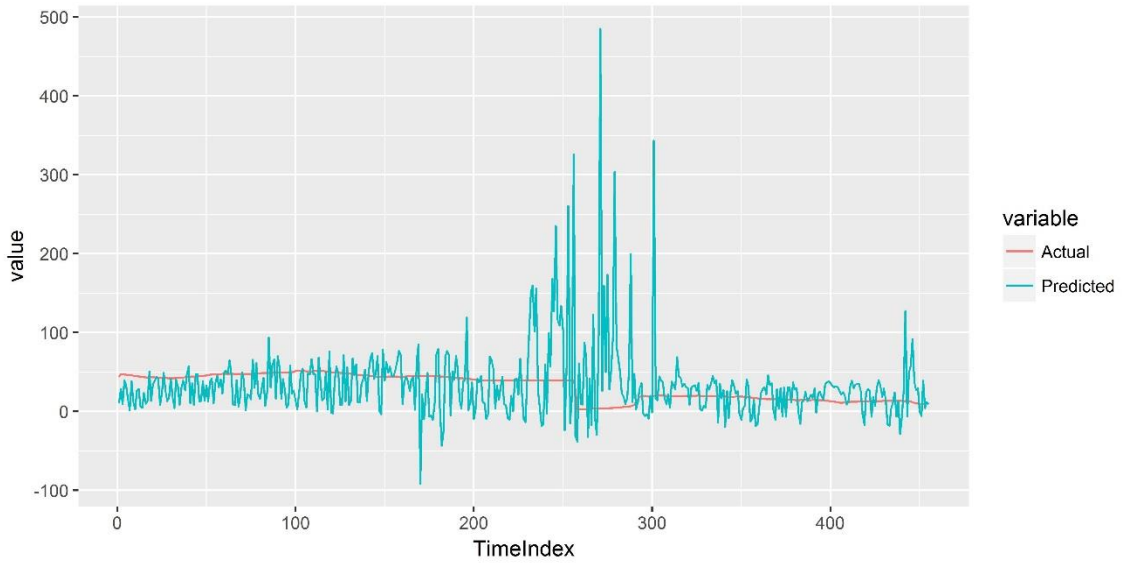

Figure A.37- Predicted Vs Actual Values Over Time using Epsilon SVR (Sigmoid Kernel)

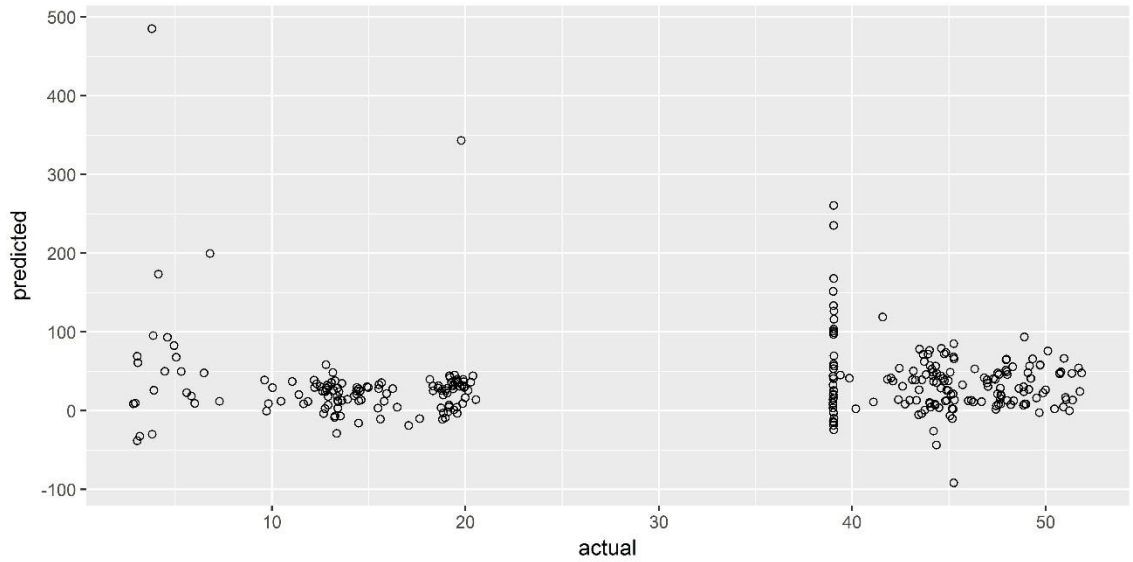

Figure A.38- Predicted Vs Actual Values Scatterplot for Training Data using Epsilon SVR (Sigmoid Kernel)

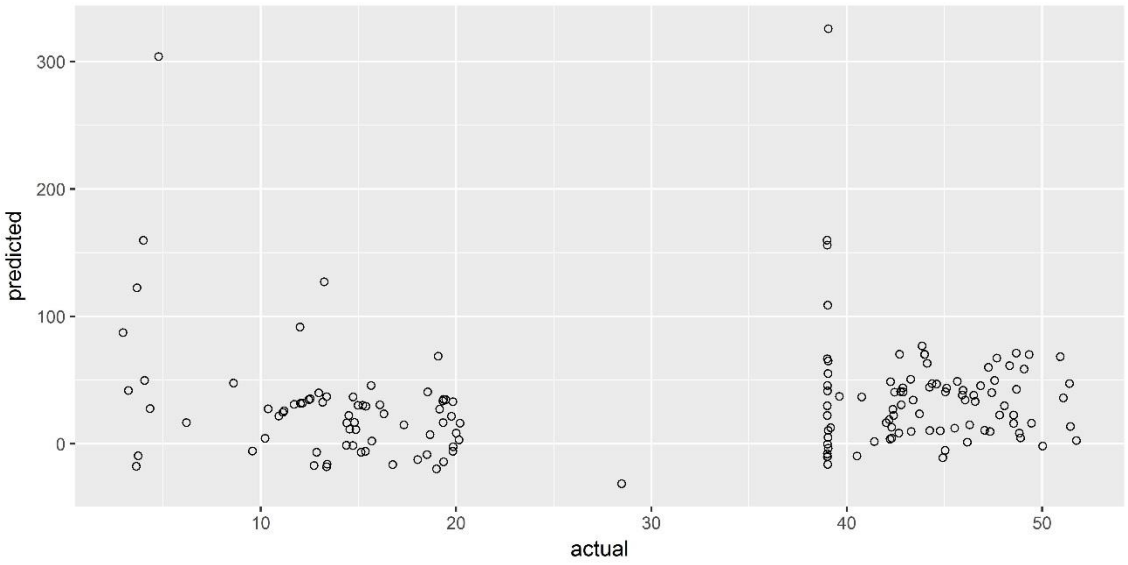

Figure A.39- Predicted Vs Actual Values Scatterplot for Test Data using Epsilon SVR (Sigmoid Kernel)

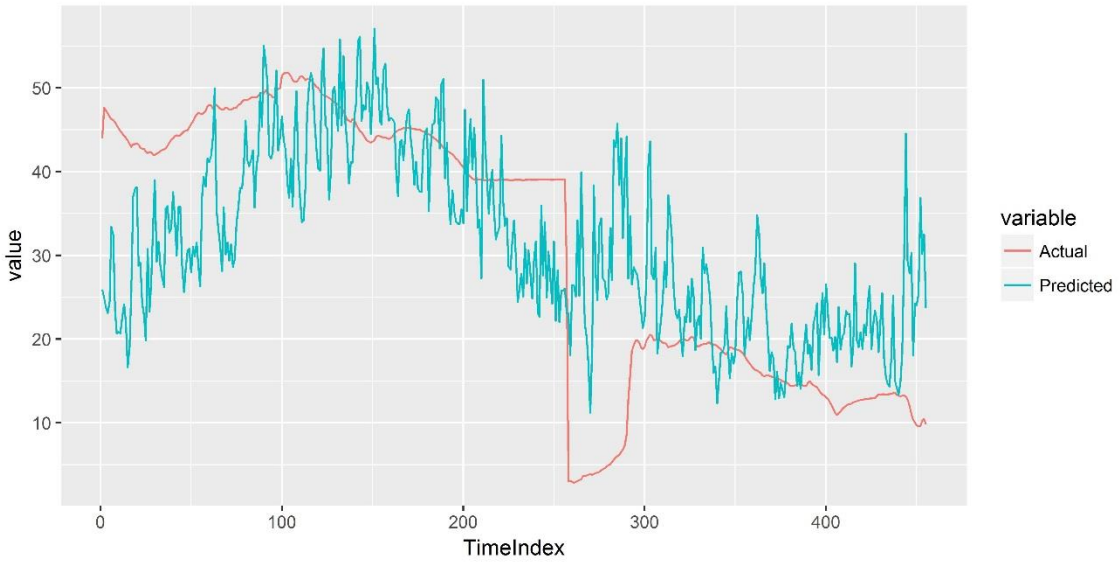

Figure A.40- Predicted Vs Actual Values Over Time using Nu SVR (Linear Kernel)

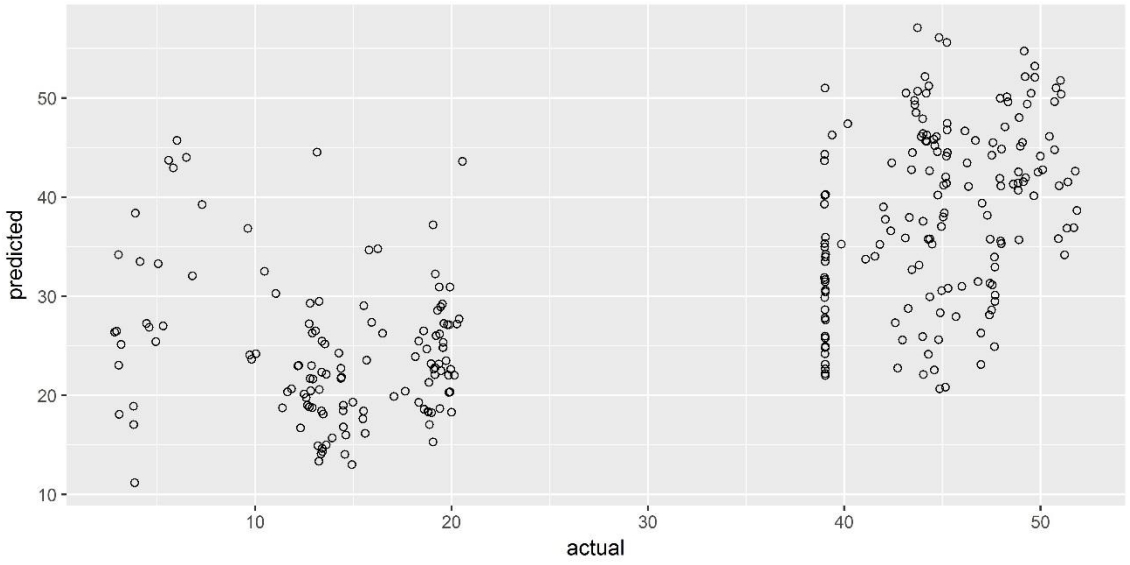

Figure A.41- Predicted Vs Actual Values Scatterplot for Training Data using Nu SVR (Linear Kernel)

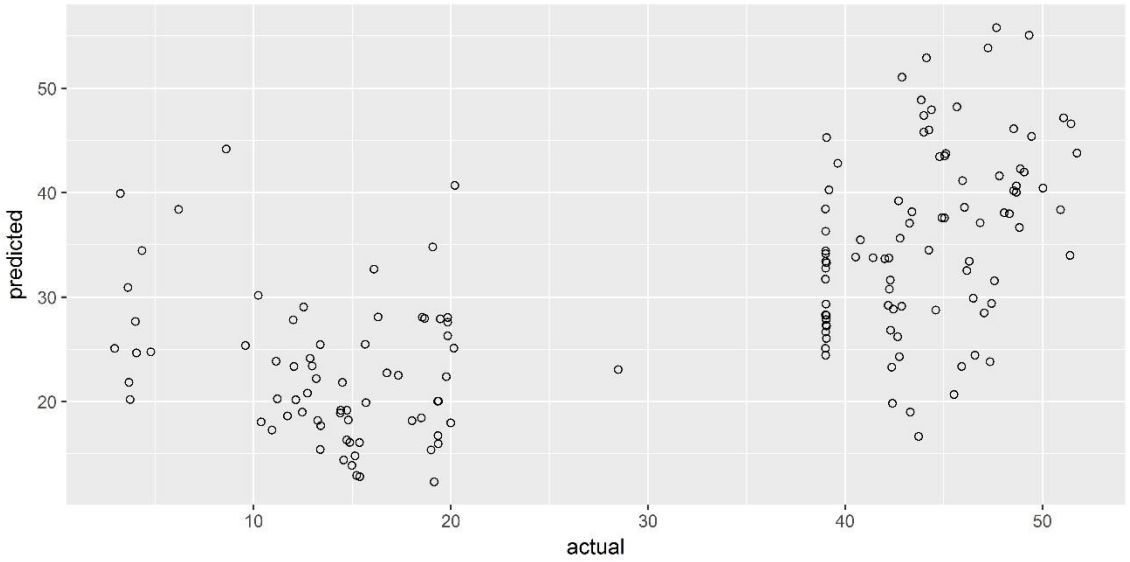

Figure A.42- Predicted Vs Actual Values Scatterplot for Test Data using Nu SVR (Linear Kernel)

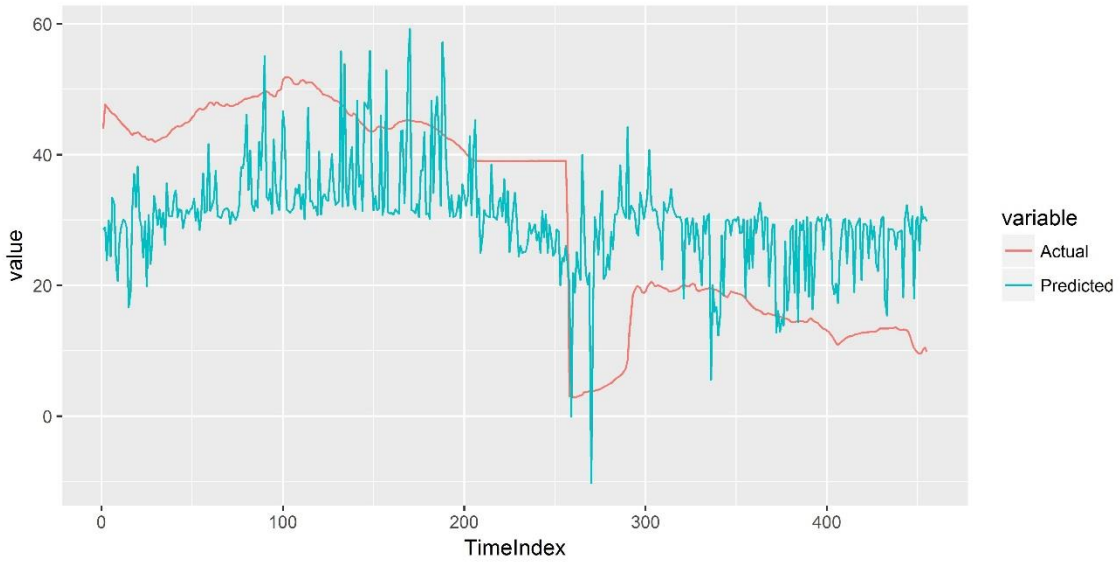

Figure A.43- Predicted Vs Actual Values Over Time using Nu SVR (Polynomial Kernel)

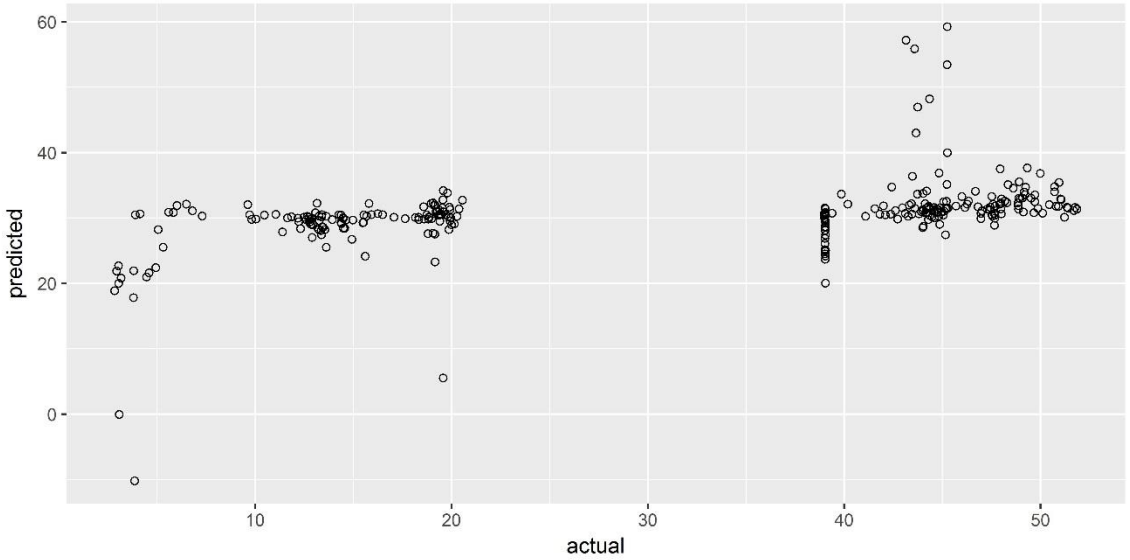

Figure A.44- Predicted Vs Actual Values Scatterplot for Training Data using Nu SVR (Polynomial Kernel)

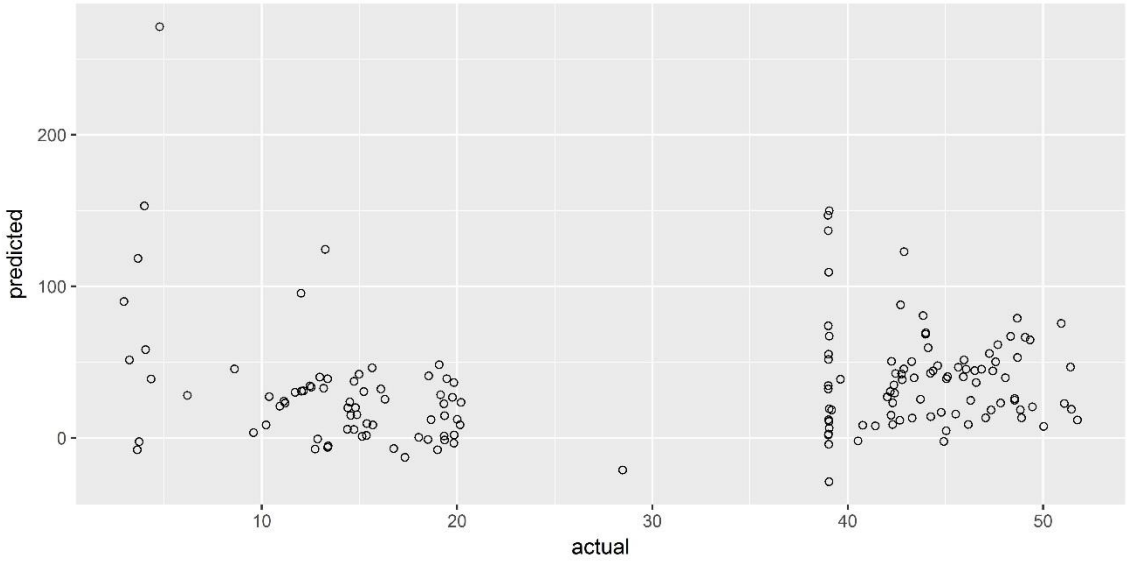

Figure A.45- Predicted Vs Actual Values Scatterplot for Test Data using Nu SVR (Polynomial Kernel)

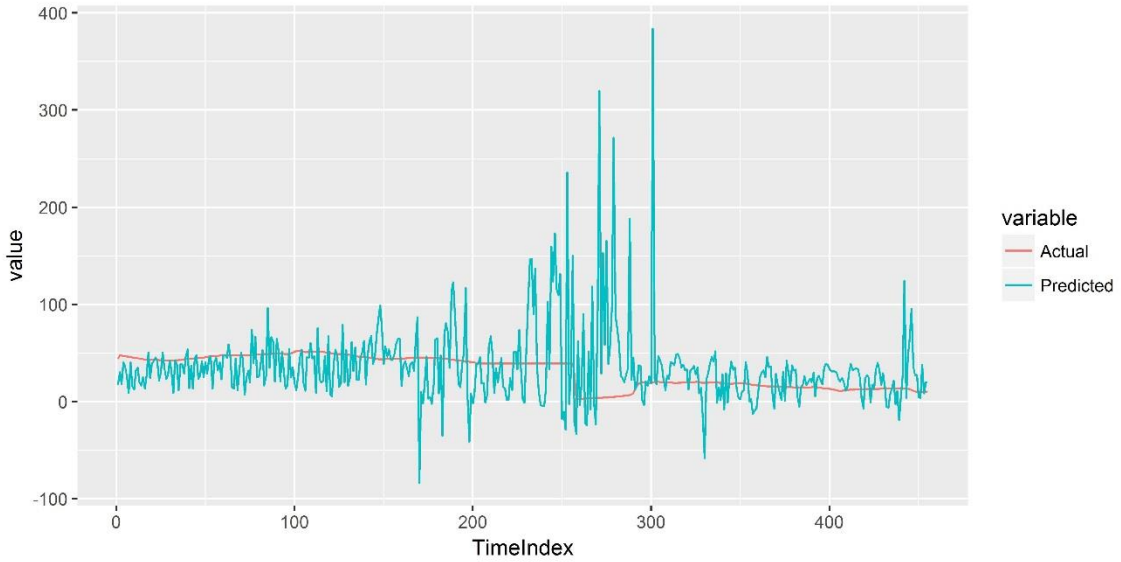

Figure A.46- Predicted Vs Actual Values Over Time using Nu SVR (Sigmoid Kernel)

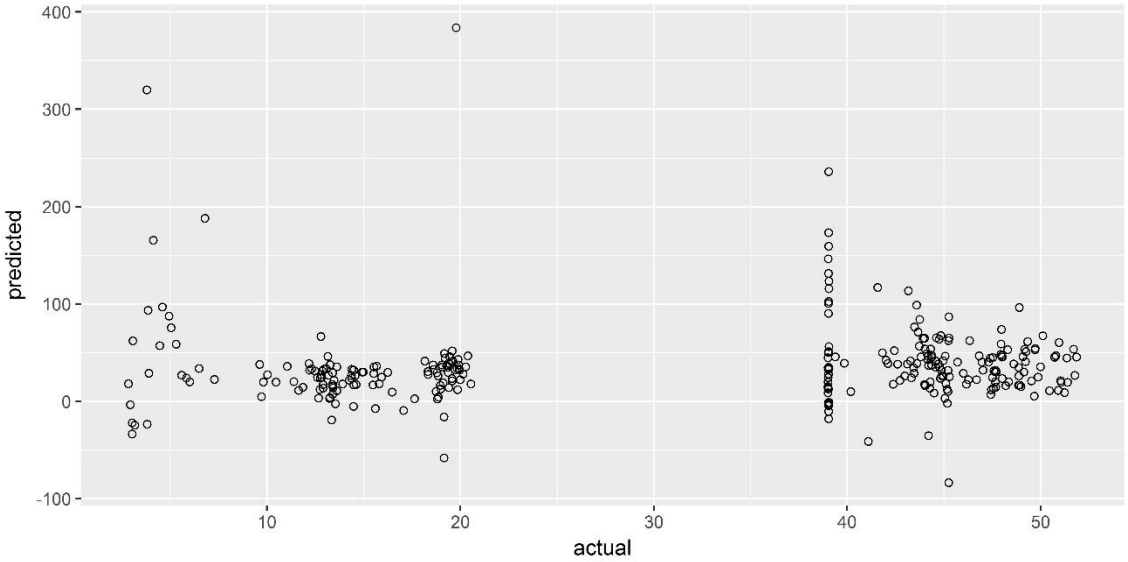

Figure A.47- Predicted Vs Actual Values Scatterplot for Training Data using Nu SVR (Sigmoid Kernel)

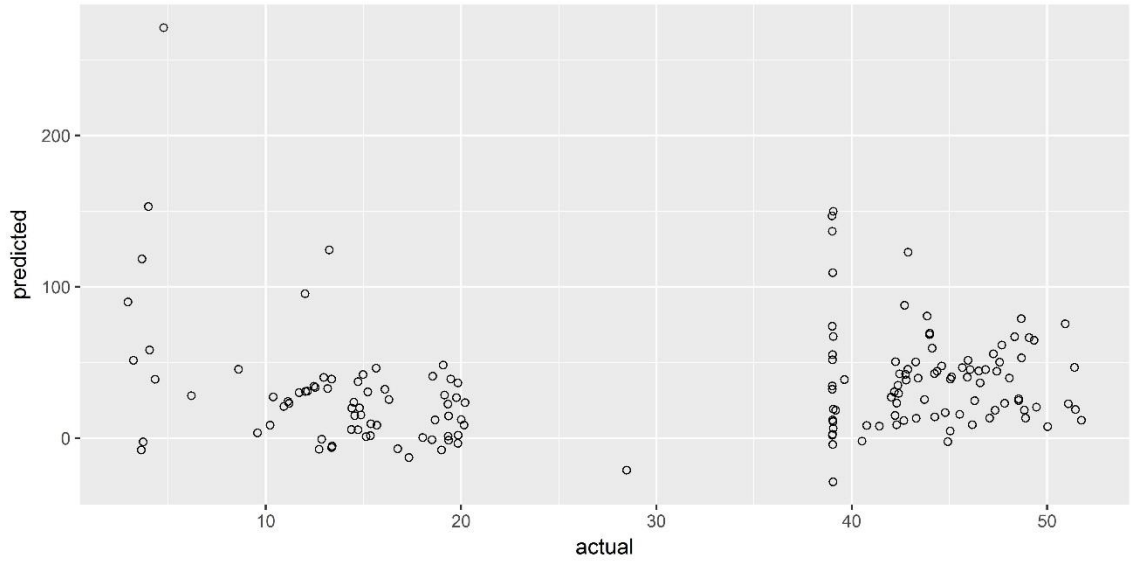

Figure A.48- Predicted Vs Actual Values Scatterplot for Test Data using Nu SVR (Sigmoid Kernel)