

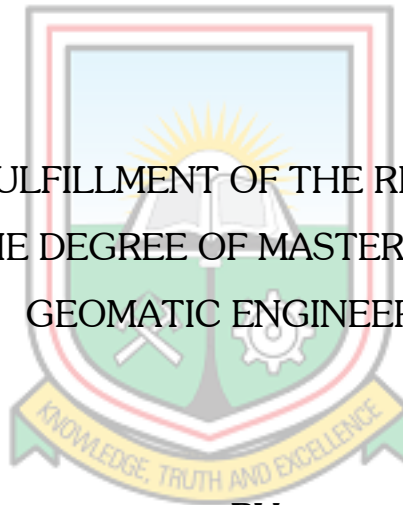
**UNIVERSITY OF MINES AND TECHNOLOGY (UMaT)  
TARKWA**

**FACULTY OF MINERAL RESOURCES TECHNOLOGY  
DEPARTMENT OF GEOMATIC ENGINEERING**

**A THESIS REPORT ENTITLED**

**PREDICTING SEA LEVEL ANOMALY IN THE GULF OF GUINEA**

**SUBMITTED IN FULFILLMENT OF THE REQUIREMENT FOR THE  
AWARD OF THE DEGREE OF MASTER OF PHILOSOPHY IN  
GEOMATIC ENGINEERING**



**BY**

**VUNASE GIDEON**

**TARKWA, GHANA**

**MAY, 2019**

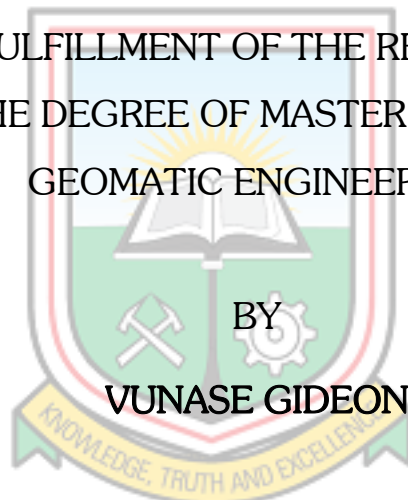
**UNIVERSITY OF MINES AND TECHNOLOGY (UMaT)  
TARKWA**

**FACULTY OF MINERAL RESOURCES TECHNOLOGY  
DEPARTMENT OF GEOMATIC ENGINEERING**

**A THESIS REPORT ENTITLED**

**PREDICTING SEA LEVEL ANOMALY IN THE GULF OF GUINEA**

**SUBMITTED IN FULFILLMENT OF THE REQUIREMENT FOR THE  
AWARD OF THE DEGREE OF MASTER OF PHILOSOPHY IN  
GEOMATIC ENGINEERING**



**VUNASE GIDEON**

**THESIS SUPERVISORS**

.....

**DR CYNTHIA B. BOYE**

.....

**DR YAO Y. ZIGGAH**

**TARKWA, GHANA**

**MAY, 2019**

## DECLARATION

I declare that this is my own work. It is being submitted for the degree of Master of Philosophy in Geomatic Engineering in the University of Mines and Technology (UMaT), Tarkwa. It has not been submitted for any degree or examination in any other University.

.....

(Signature of Candidate)

..... day of .....(Year).....



## ABSTRACT

Over the last decade Satellite Altimetry has become a viable alternative over Tidal Gauge measurements for the determination and prediction of Sea Level Anomalies (SLA) in that it is able to cover a broader area of the ocean and also due to its higher level of accuracy. It can also be used to obtain data in portions of the sea where Tidal Gauges cannot be used. This study examined various models like the Backpropagation Neural Network (BPNN) and the Generalised Regression Neural Network (GRNN) in predicting SLA in the Gulf of Guinea and the best model was determined. The variables considered were heat, humidity, wind flux and SLA in the study area. The wind flux, heat and humidity were the input variables and SLA the output variable for the various models that were developed. The best model for the training algorithms of the BPNN thus, the Bayesian Regularisation (BR), Scaled Conjugate Gradient (ScG) and the Levenberg Marquardt (LM) was 3-2-1, 3-4-1 and 3-4-1 respectively for each of the algorithms and the Correlation Coefficient (R) values respectively obtained for the models is 0.75156, 0.55852 and 0.55086 and the Mean Square Errors (MSE) are 0.000732, 0.001722 and 0.001226. Also, the optimal model of the GRNN has a spread of 0.40 which gave an R of 0.481043 and a MSE of 0.000975. The higher the correlation coefficient and the lower the mean square error of a model, the better it is for predictions, it was therefore realised from the study that the BR of the BPNN is the optimal model for predicting Sea Level Anomalies (SLA) in the Gulf of Guinea.

## DEDICATION

I dedicate this work to my loving and hard-working parents Mr Jacob Vunase and Mrs Esther Vunase who have been my source of motivation in all aspects of life.

I also dedicate this project to

My younger siblings,

Isaac Vunase

Godfred Vunase

Hager Vunase and

Benaiah Vunase

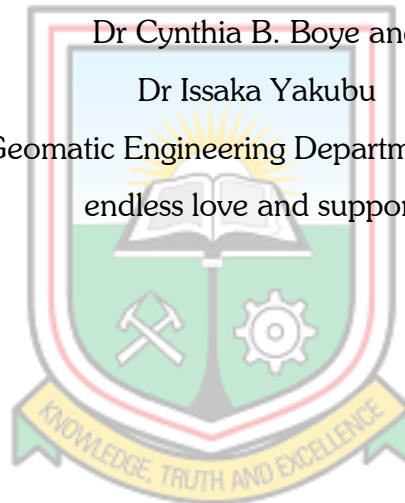
And to my academic mentors, especially

Dr Yao Y. Ziggah

Dr Cynthia B. Boye and

Dr Issaka Yakubu

All Lecturers in the Geomatic Engineering Department I say a big thank you for endless love and support.



## ACKNOWLEDGEMENTS

I thank the almighty God for by His Grace and Mercies has He granted me the knowledge and strength to undertake this academic exercise.

Help, encouragement and advice from my parents have been of most beneficial to my entire life and education. I express my sincere appreciation to you for your contribution towards my education.

I would particularly like to show gratitude to Dr Cynthia B. Boye, Dr Yao Yevenyo Ziggah and Dr Issaka Yakubu for their tireless effort, immense contribution and continual encouragement towards the success of this project. I would not also forget Mr Clement Kweku Arthur and Mr Albert Buabeng for the support they have given me. I want to say may the Almighty bless you all. Amen!

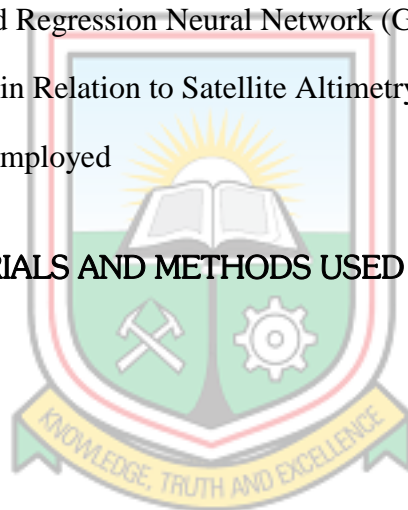


## TABLE OF CONTENTS

Contents	Page
DECLARATION	i
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	ix
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Background to Research	1
1.2 Research Problem	2
1.3 Research Objectives	3
1.4 Methods Used	4
1.5 Thesis Organisation	4
<b>CHAPTER 2 GENERAL INFORMATION ABOUT THE STUDY AREA</b>	<b>6</b>
2.1 The Gulf of Guinea	6
2.1.1 Economic Importance	7
2.1.2 Oil and Politics in the GoG	7
2.1.3 Pollution, Fisheries and Food in the GoG	8



<b>CHAPTER 3</b>	<b>LITERATURE REVIEW</b>	<b>9</b>
3.1	Altimetry	9
3.1.1	Barometric altimetry	9
3.1.2	Satellite Altimetry	10
3.2	Sea Level Anomaly	12
3.2.1	Tidal Gauges in Sea Level Anomaly Determination	13
3.3	Artificial Intelligence (AI)	14
3.3.1	Artificial Neural Network	14
3.3.2	The Feed Forward Back Propagation Neural Network (FFBPNN)	15
3.3.3	Generalised Regression Neural Network (GRNN)	19
3.4	Studies Conducted in Relation to Satellite Altimetry and Various Methods Employed	22
<b>CHAPTER 4</b>	<b>MATERIALS AND METHODS USED</b>	<b>28</b>
4.1	Overview	28
4.2	Materials	28
4.2.1	Data	28
4.3	Methods Used	32
4.3.1	Back Propagation Neural Network (BPNN)	32
4.3.2	Generalised Regression Neural Network (GRNN)	34
4.4	Statistical Performance Metrics	34
<b>CHAPTER 5</b>	<b>RESULTS AND DISCUSSION</b>	<b>37</b>
5.1	Overview	37
5.2	Collation of SLA Data in the GoG	37



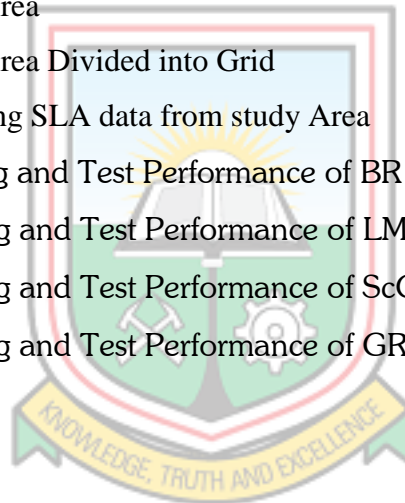


5.3	Models Developed for Prediction	39
5.3.1	Back Propagation Neural Network	39
5.3.2	Prediction	40
5.3.3	Generalised Regression Neural Network	42
5.4	Discussion	44
<b>CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS</b>		<b>46</b>
6.1	Conclusions	46
6.2	Recommendations	46
<b>REFERENCES</b>		<b>48</b>
<b>APPENDIX A RESULTS FOR THE VARIOUS MODELS DEVELOPED</b>		<b>51</b>
<b>APPENDIX B CODES</b>		<b>55</b>



## LIST OF FIGURES

Figure	Title	Page
1.1	Flowchart of Thesis Organisation	5
2.1	Map of Gulf of Guinea Region	7
3.1	A Barometric Altimeter	9
3.2	A Satellite Round Trip Signal	10
3.3	Satellite to Surface Range	12
3.4	Biological Artificial Neuron	15
3.5	Back Propagation Neural Network Architecture	18
3.6	Diagram of GRNN	19
4.1	Continent of Africa showing study area	29
4.2	Study Area	29
4.3	Study Area Divided into Grid	30
5.1	Obtaining SLA data from study Area	37
5.2	Training and Test Performance of BR Model	40
5.3	Training and Test Performance of LM Model	41
5.4	Training and Test Performance of ScG Model	42
5.5	Training and Test Performance of GRNN Model	43



## LIST OF TABLES

Table	Title	Page
4.1	Humidity Data	30
4.2	Humidity, Wind, Heat and SLA Data	31
4.3	Minimum R2 Considered Statistically Significant	35
5.1	Sample of SLA Data Averaged	38
5.2	Residual Analysis using BPNN Architectures	39
5.3	Residual Analysis using the Generalised Regression Neural Network	42



# CHAPTER 1

## INTRODUCTION

### 1.1 Background to Research

Sea level anomaly (SLA) is the difference between the actual sea surface height (SSH) and a mean sea surface height (MSSH) Anon. (2019). Sea levels vary as a result of three groups of processes including, eustatic effect (water mass supply to the ocean), steric effect (thermal increase in the volume of ocean water without a change in its mass and halosteric rise due to change in salinity) and geologic processes changing spatial geometry of ocean basins (Niedzielski and Kosek, 2008). Several studies prove this fact, for instance, a study conducted by Dadson *et al.* (2015), reveals that the shoreline along Cape Coast-Sekondi coast has been fluctuating. Another study by Kwasi (2014) also reveals that sea levels are on the rise.

Also, Gharineiat and Deng (2015) showed that north Australia is in a risk since many parts of its coastlines are in cyclone prone regions where the rate of sea level rise is also found to be 2.5 more than the globally averaged sea level rise. In addition, Kurniawan *et al.* (2014) conducted a study in the Singapore regional waters, where it was showed that the island of Singapore and part of the Sunda shelf experiences a direct impact of nonlinear dynamical interactions between the South China Sea, Andaman Sea and Java Sea. This was as a result of multiple ocean currents moving into and out of the region combined with short term meteorological effects results in a very high variability of the sea water level around the Singapore's coast.

Satellite Altimetry has become a viable alternative in the last decade over Tidal Gauge measurements for the determination of SLA because it is able to cover a large area of the ocean and also due to its higher level of accuracy. Satellite Altimetry can also be used to obtain data in portions of the sea where Tidal Gauges cannot be used (Douglas *et al.*, 2011). In view of that several scholars have conducted studies using only satellite altimetry based acquired data or a combination of the two methods of data acquisition in various studies; Morner (2017) observed that satellite altimetry measurements

started in late 1992 and is a new graceful tool to view changes in sea level over the entire globe. Also, Sterlini *et al.* (2017) studied the spatial variation of sea surface height trends in the North Sea Basin using satellite altimetry where also changes in the potential temperature and salinity of the North Sea are changed into corresponding sea surface height and regional anomalies of linear sea level trend calculated. The same is carried out for the meteorological processes which act on the sea surface.

Based on the above development, this research has used satellite altimetry data acquired in the Gulf of Guinea to predict SLA in the study area using mathematical computational intelligent algorithms.

## 1.2 Research Problem

Coastal zones are dynamic areas influenced by several factors such as wave impact, sea level rise due to climate change, human interference, among others. Understanding the behaviour of the sea and its impact on coastal morphology is of great importance to any country. Hydrodynamics is generated due to multiple ocean currents combined with meteorological effects which leads to a high variability of the local sea level. Local sea level is derived from tide record which vary from coast to coast depending on the following:

- i. Response of the ocean basin to tide producing forces;
- ii. Modification of the tide due to shallow water effects of local embayment and rivers;
- iii. Regional and local effects of weather on water levels (Holleman and Stacey, 2014).

The variability of local sea level is not accounted for during ocean weather forecasting which seriously affect coastal planning and navigation in the region (Kurniawan *et al.*, 2014). Therefore, the analysis and accurate prediction of sea level anomalies (SLA) becomes an important part of oceanographic modelling. In view of that, several scholars have also studied and ascertained ways of modelling and predicting the sea level anomaly (SLA) for various coastal zones. Authors, for example, Kurniawan *et al.*

(2014) used Genetic Programming (GP) to predict SLA in the Singapore Regional waters where it was concluded that GP helps to improve forecasting ability. In other studies, an Ocean circulation model fed with scatterometer wind was used to predict SLA and validated using TOPEX/Poseidon data (Quilfen *et al.*, 2000). Multiple regression techniques have also been used to study SLA (Doran *et al.*, 2015). Miles *et al.* (2014) also applied dynamical coupled ocean– atmosphere model to the prediction of seasonal sea level anomalies (SLA) globally for up to 7 months in advance where it was shown that Predictive Ocean Atmosphere Model for Australia demonstrates high skill in the equatorial Pacific basin and consistently exhibits more skill globally than a forecast based on persistence.

In the last few years, the use of computational intelligent algorithms for function estimation has been widely embraced in geodetic sciences. Some of the notable techniques applied specifically in SLA estimation include:

- i. Artificial neural network (ANN) methods of backpropagation (Anne *et al.*, 2017);
- ii. Support vector machine (SVM) (Moura *et al.*, 2010);
- iii. Extreme learning machine (ELM) (Imani *et al.*, 2017); and
- iv. Adaptive Neuro-Fuzzy Inference System (ANFIS) (Lin and Chang, 2007).

The general information gathered from these research works showed that the intelligent methods could predict SLA satisfactorily. This study therefore sought to determine a model that could be used specifically within the Gulf of Guinea.

### 1.3 Research Objectives

The objectives of the research are to:

- i. Collate the sea level anomaly of the study area using Satellite Altimetry data;
- ii. Predict sea level anomaly using computational intelligent algorithms.

## 1.4 Methods Used

The methods used include the following:

- i. Extensive literature Review;
- ii. Acquisition of Satellite Altimetry Data from Asia-Pacific Data-Research Center;
- iii. Processing of satellite altimetry data using computational intelligent algorithms;  
and
- iv. Analyses and interpretation of results.

## 1.5 Thesis Organisation

This thesis is structured into six chapters. Chapter 1 is the project synopsis which deals with the general introduction to the thesis. Chapter 2 is general information about the study area and this gives a description of the location of the study area and also gives information concerning the salinity and temperature of it. Chapter 3 reviews literature that is of importance to the subject thus explaining the types of computational intelligent algorithms to be used and what satellite altimetry is all about and how satellite altimetry data has been used by others in research about SLA. Chapter 4 discusses how the computational intelligent algorithms, thus BPNN and GRNN are used to process the Satellite Altimetry data. The results and discussions of the thesis are presented in Chapter 5. The conclusions and recommendations are finally discussed in Chapter 6.

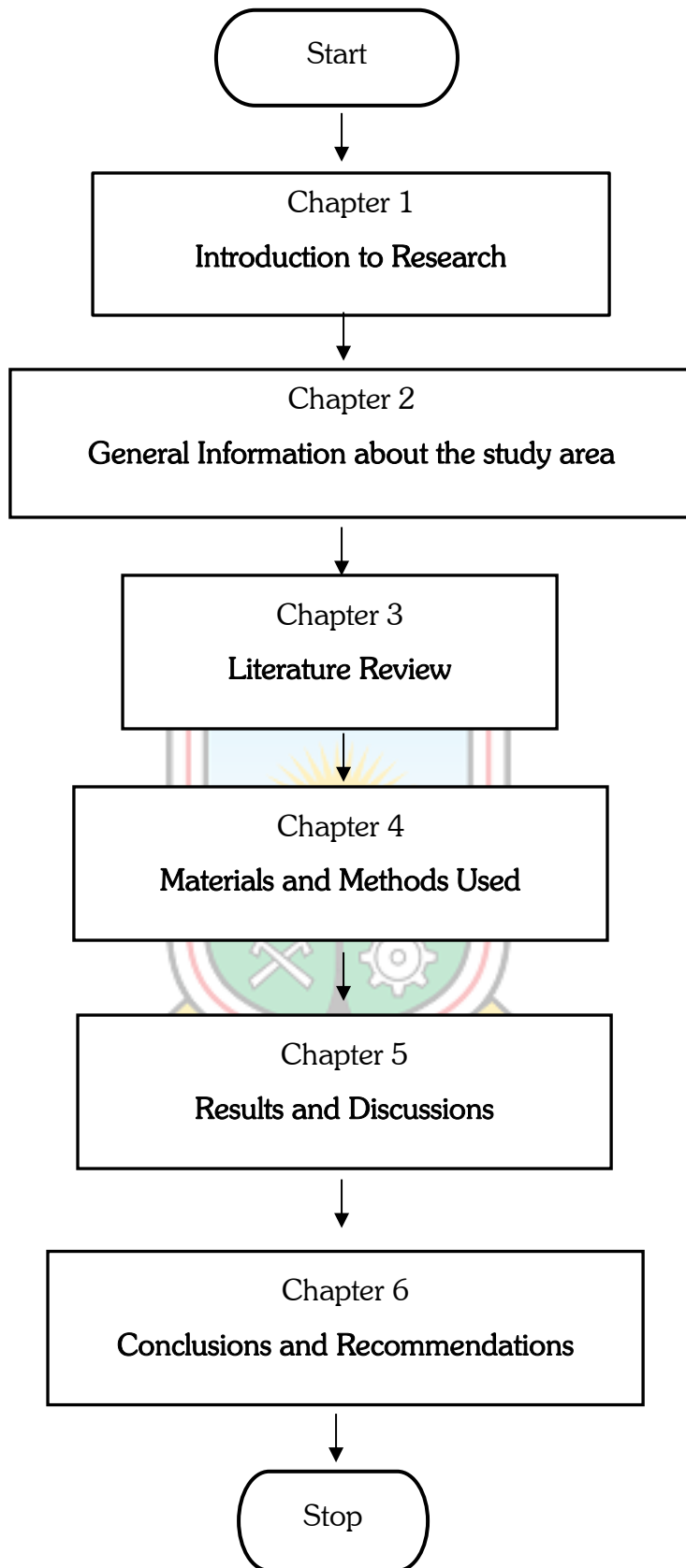


Figure 1.1 Flowchart of Thesis Organisation



## CHAPTER 2

### GENERAL INFORMATION ABOUT THE STUDY AREA

#### 2.1 The Gulf of Guinea

The name Guinea has been in use both in the Maghrib (North Africa/Morocco region) and in Europe long before Prince Henry's time. The Genoese cartographer Giovanni di Carignano got the information about Africa dated back about 1320 AD from a fellow-countryman in Sijilmas (ancient trading city in North Africa), the name Gunuia was found on his map and in the Catalan atlas of 1375 AD as Ginyia (Abubakar, 2016). The name Guinea was also applied to the south coast of West Africa, north of the Gulf of Guinea (GoG) which became known as "Upper Guinea" and the west coast of Southern Africa to the east, which became known as "Lower Guinea". The name "Guinea" is still attached to the names of three countries in Africa: Guinea, Guinea-Bissau, and Equatorial Guinea as well as Papua New Guinea in Melanesia (The Asian continent) (Abubakar, 2016).

The Gulf of Guinea (GoG) is an extremely big and important region that constitutes about 16 countries that lie along an unbroken coastline of about 6000 kilometers from the north-western coast of Africa downwards (Ukeje and Mvomo, 2013). These countries include Liberia, Sierra Leone, Senegal, Cote d'Ivoire, Ghana, Gabon, Togo, Benin, Nigeria, Cameroon, Equatorial Guinea, the island state of Sao Tome and Principe, Central African Republic, the Republic the Congo, the Democratic Republic of the Congo and Angola on the southernmost fringes (Ukeje and Mvomo, 2013).



**Figure 2.1 Map of Gulf of Guinea Region (Source: Abubakar, 2016)**

### 2.1.1 Economic Importance

The Gulf of Guinea (GoG) is a very rich area and yet under exploited. About 35% of the world's petroleum reserve is deposited in the GoG and it is also blessed with many minerals and other natural resources such as diamonds, tin, uranium, copper, manganese, gold, phosphates, granite, gas, marble, quartz, lead, zinc, wolfram, fluorite, sulfur, feldspar, silver, kaolin, mica, gypsum, oil and talc as well as a very rich rain forests accounting for 20% of the world's total rainforest and much of the oxygen-generating sources of the globe and it also keeps providing shelter and protection to a large portion of the World's biodiversity including the pygmies, animals and microorganisms (Babagana Abubakar, 2016).

### 2.1.2 Oil and Politics in the GoG

The eight oil states in West Africa around the Gulf of Guinea together produce some five million barrels of oil a day and may hold a significant amount of the world's oil reserves. Soares de Oliveira has written an important study of the impact of oil on the

region's politics. Oil, he shows, has had a powerfully negative effect on the quality of government. Even as the oil economy thrives thanks to high oil prices and significant new investment from Western oil companies, governments in the region have increasingly failed to provide welfare or security to their citizens and have instead used their states' oil revenues to protect their hold on power and enrich small elites (Richardo, 2008). Soares de Oliveira labels states that have achieved a combination of international respectability, disastrous governance, and regime stability "successful failed states." His study draws on a wealth of information to discuss the corruption of these regimes and their increasing ability to absolve themselves of the regular responsibilities of sovereignty, such as providing health care, education, and infrastructure to their citizens (Richardo, 2008). Even as they fail in these routine tasks, they can demonstrate real skill and savvy in their negotiations with oil companies and have successfully used oil wealth to buttress their international standing. That oil represents a curse is no longer a novel insight, but Soares de Oliveira's study provides a rich political sociology of the oil curse in West Africa (Richardo, 2008).

### 2.1.3 Pollution, Fisheries and Food in the GoG

Fisheries contribute to the food security of more than three billion people globally. In the Gulf of Guinea, it accounts for more than 80 percent of the animal protein and sometimes is the only source of animal protein consumed in littoral communities in Nigeria, Guinea-Bissau, Ghana and Senegal. However, the pervasiveness of unsustainable practices that are harmful to the marine environment such as pollution, overfishing - illegal, unreported and unregulated (IUU) fishing, coupled with the impact of climate change threatens the resource's ability to continue to contribute to the food security of these communities especially those in the Niger Delta area of Nigeria (Rosenblood, 2018). Pollution from oil spillages in the marine environment is exacerbating food insecurity in the region. Using the Bonga oil field spillage as an example, the shortcomings of the existing environmental regulations in Nigeria as contributing to the depletion of the marine environment in the region (Rosenblood, 2018).

## CHAPTER 3

### LITERATURE REVIEW

#### 3.1 Altimetry

An altimeter is an instrument used to measure the altitude of an object above a fixed level Anon. (2019). The measurement of altitude is called altimetry which is related to the term bathymetry, the measurement of depth underwater. The two types of altimetry are barometric altimetry and satellite altimetry Anon. (2019).

##### 3.1.1 Barometric altimetry

Barometric altimetry measures altitude by measuring air pressure. Barometric altimetry is independent in its measurements; all it needs is a constant air pressure. For any given geometric height, the indicated altitude of the altimeter (the one that defines airspace) will vary depending on the conditions of the day (Graham, 2011).

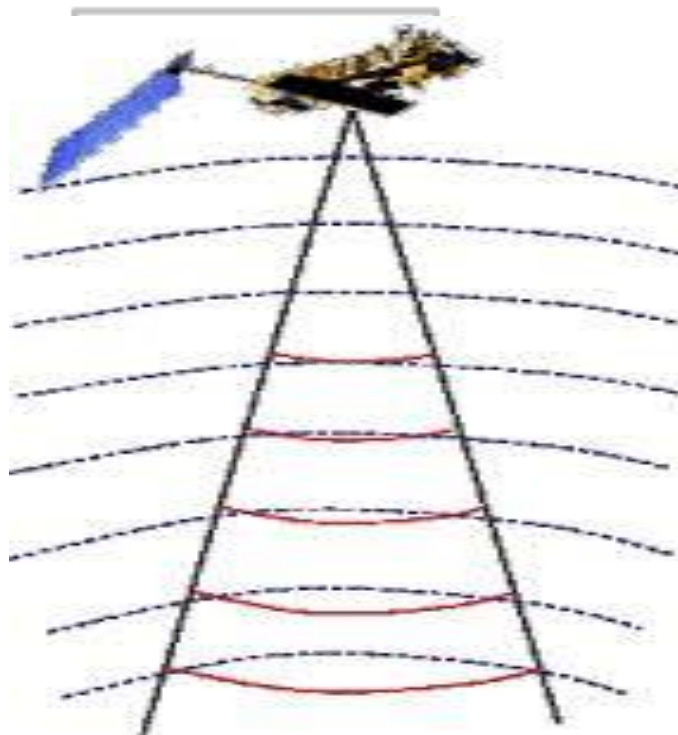


Figure 3.1 A Barometric Altimeter (Source: Anon., 2019)

Common applications are, in avionics, where barometric altimeters can help in stabilizing the vertical position and velocity of a flying object, navigation and human motion tracking and monitoring, detect moving styles of going up/down the stairs or in an elevator to determine the correct floor of a user in a multi-storey building (Sabatini and Genevese, 2013).

### 3.1.2 Satellite Altimetry

Altimetry satellites basically determine the distance from the satellite to a target surface by measuring the satellite-to-surface round-trip time of a radar pulse (Rosmorduc *et al.*, 2011).



**Figure 3.2 A Satellite Round Trip Signal (Source: Rosmorduc *et al.*, 2011)**

However, this is not the only measurement made in the process, and a lot of other information can be extracted from altimetry (Rosmorduc *et al.*, 2011). The principle is that the altimeter emits a radar wave and analyses the return signal that bounces off the surface. Surface height is the difference between the satellite's position on orbit with

respect to an arbitrary reference surface (the Earth's center or a rough approximation of the Earth's surface: the reference ellipsoid) and the satellite-to-surface range (calculated by measuring the time taken for the signal to make the round trip) (Rosmorduc *et al.*, 2011). Besides surface height, by looking at the return signal's amplitude and waveform, it can also measure wave height and wind speed over the oceans, and more generally, backscatter coefficient and Surface roughness for most surfaces off which the signal is reflected. If the altimeter emits in two frequencies, the comparison between the signals, with respect to the frequencies used, can also generate interesting results (rain rate over the oceans, detection of crevasses over ice shelves, etc.) (Rosmorduc *et al.*, 2011). To obtain measurements accurate to within a few centimeters over a range of several hundred kilometers requires an extremely precise knowledge of the satellite's orbital position. Thus, several locating systems are usually carried onboard altimetry satellites. Any interference with the radar signal also needs to be taken into account. Water vapour and electrons in the atmosphere, sea state and a range of other parameters can affect the signal roundtrip time, thus distorting range measurements. We can correct for these interference effects on the altimeter signal by measuring them with supporting instruments, or at several different frequencies, or by modelling them. Altimetry thus requires a lot of information to be taken into account before being able to use the data. Data processing is also a major part of altimetry, producing data of different levels optimized for different uses at the highest levels (Rosmorduc *et al.*, 2011).

Radar altimeters on board the satellite transmit signals at high frequencies (over 1,700 pulses per second) to Earth, and receive the echo from the surface (the "waveform") (Rosmorduc *et al.*, 2011). This is analyzed to derive a precise measurement of the time taken to make the round trip between the satellite and the surface. This time measurement, scaled to the speed of light (the speed at which electromagnetic waves travel), yields a range  $R$  measurement. However, as electromagnetic waves travel through the atmosphere, they can be decelerated by water vapor or ionization. Once these phenomena have been corrected for, the final range can be estimated with great accuracy. The ultimate aim is to measure surface height relative to a terrestrial reference frame. This requires independent measurements of the satellite's orbital

trajectory, i.e. exact latitude, longitude and altitude coordinates (Rosmorduc *et al.*, 2011).

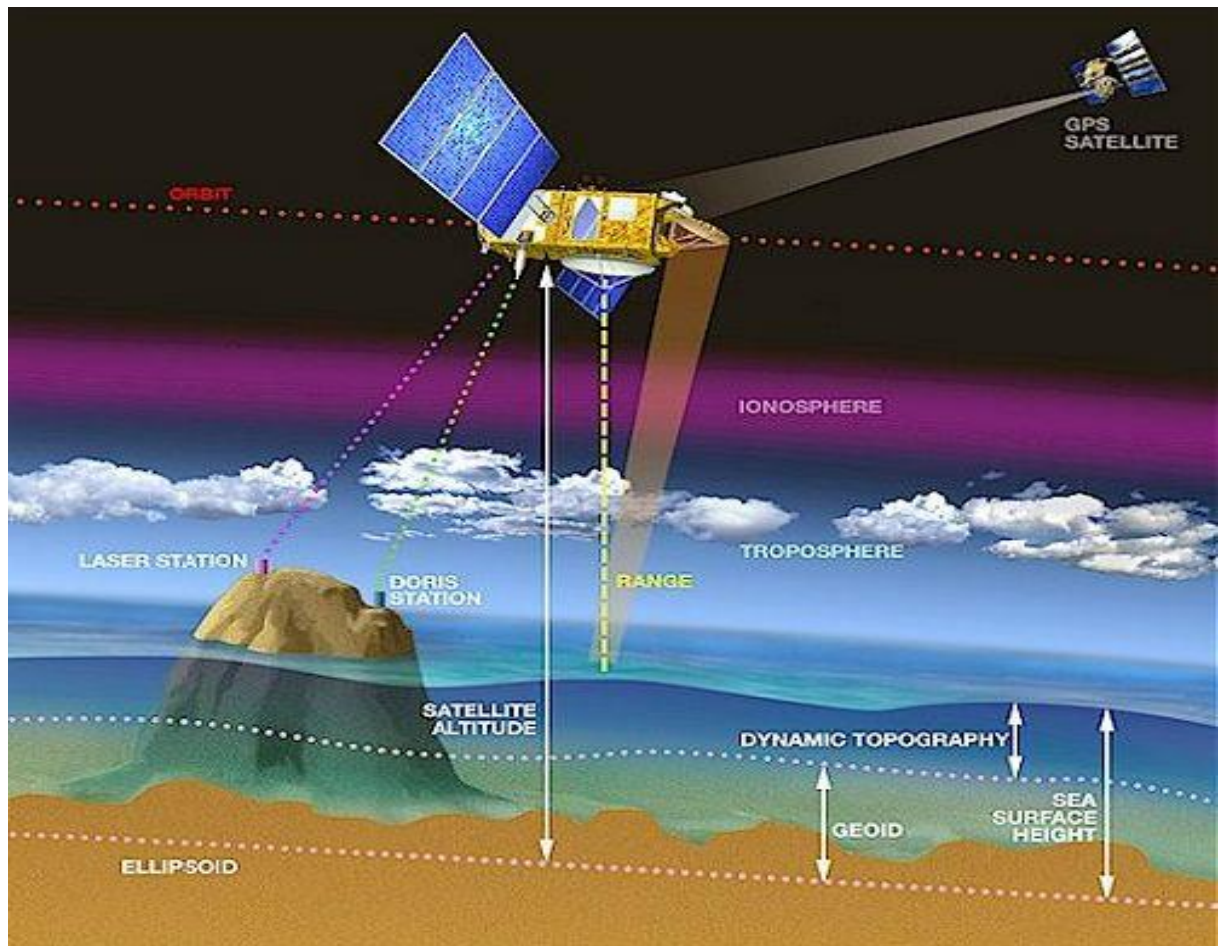


Figure 3.3 Satellite to Surface Range (Source: Rosmorduc *et al.*, 2011)

### 3.2 Sea Level Anomaly

Sea level anomaly is the difference between the actual sea surface height (SSH) and a mean sea surface height (MSSH). Sea levels vary as a result of three groups of processes including, eustatic effect (water mass supply to the ocean), steric effect (thermal increase in the volume of ocean water without a change in its mass and halosteric rise due to change in salinity) and geologic processes changing spatial geometry of ocean basins (Niedzielski and Wieslaw, 2008).

The main force that is causing significant changes in coastal regions is sea level rise and it is becoming increasingly important as a hazard to humans and urban areas in the coastal zone worldwide as global climate change takes effect (Williams, 2013). The geologic record indicates that due to past natural climate factors, sea level has been highly variable, as much as 6-8 m higher than present during the last interglacial warm period and 130 m lower during the last glacial period (Williams, 2013). According to Williams (2013), Sea level has been fairly stable for the past 3,000 years until about the mid- 19<sup>th</sup> century. It was during the 20<sup>th</sup> century that sea level began rising at a global rate of 1.7 mm/yr. The current average rise rate is 3.1 mm/yr, a 50% increase over the past two decades. Many regions are experiencing even greater rise rates due to local geophysical (*e.g.*, Louisiana, Chesapeake Bay) and oceanographic (Mid-Atlantic coast) forces. Land uplift has made few regions experience rise rates less than the global average. Observations show that increased emissions of carbon as a result of industrialisation has increased global mean temperature of the air and ocean which is responsible for sea-level rise due to ice sheet melting and thermal expansion and many related environmental changes (Williams, 2013). Sea level rise with high regional variability is exhibiting acceleration and is expected to continue for centuries unless mitigation is enacted to reduce atmospheric carbon (Williams, 2013). Low-lying coastal plain regions, deltas, and most islands are highly vulnerable. Adaptation planning on local, state and national scales for projected sea-level rise of 0.5–2 m by A.D. 2100 is advisable. Sustained global rise in sea level of 4 m to as much as 8 m is possible, but not likely until well after A.D. 2100 (Williams, 2013).

### 3.2.1 Tidal Gauges in Sea Level Anomaly Determination

In Fernandes *et al.* (2008), tidal gauges are installed at suitable positions along the coast which are used to monitor and mark the sea water variations. When there is a positive vertical variation on the gauge, two assertions can be made and thus; either the ground is subsiding or sea water levels are rising. This ambiguity is removed by installing and connecting GNSS stations as close as possible to the tidal gauges. The absolute movement of the GNSS site can be obtained directly because they are



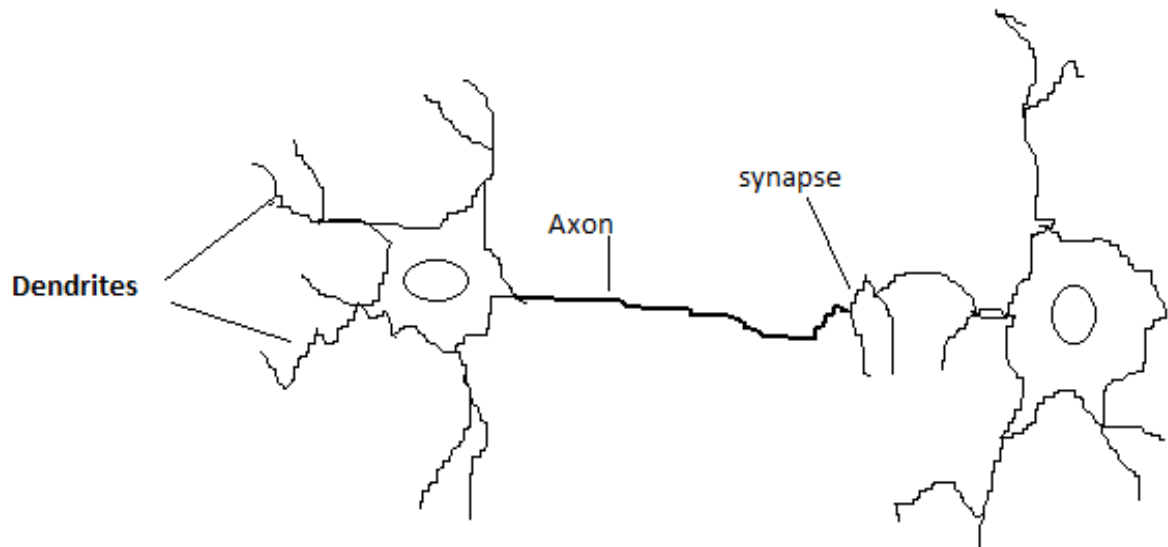
referenced to the ellipsoid. A first order levelling must then be performed in order to connect the GNSS stations with the tide gauge.

### 3.3 Artificial Intelligence (AI)

Artificial intelligence is simply the science and engineering of making intelligent machines as defined by McCarthy, the father of AI in 1955 or the capability of a digital computer to execute tasks commonly associated with the intelligence of human beings such as reasoning and learning (De *et al.*, 2016). According to Pooler (2017), artificial intelligence in today's world is progressing faster everyday with new improved innovations where today's computer systems are designed to perform small tasks, for instance, facial recognition, car driving and performance of other minor duties. Artificial intelligence is aimed to develop advanced and more complex systems that would outperform humans in whichever way and this includes the performance of more complicated tasks like playing chess and solving equations. AI in the future is aimed to perfect all human activities and provide better solutions to problems than the human can do.

#### 3.3.1 Artificial Neural Network

ANN is an oversimplified simulation of the human brain and composed of simple processing units referred to as neurons. It is able to learn and generalise from experimental data even if they are noisy and imperfect. This ability allows this computational system to learn constitutive relationships of materials directly from the results of experiments. One type of network is an artificial neuron and it is a computational model which is inspired in natural neurons which receive signals through synapses located on the dendrites or membrane of the neuron (Gershenson, 2013). When the signals received are strong enough to surpass a certain threshold then the neuron is activated and emits a signal through the axon which might be sent to another synapse and might activate other neurons (Gershenson, 2013).



**Figure 3.4 Biological Artificial Neuron (Source: Gershenson, 2013)**

### 3.3.2 The Feed Forward Back Propagation Neural Network (FFBPNN)

The ANN is a flexible mathematical structure that has taken the pattern of a biological nervous system. The common structure of an ANN is called a multilayer perceptron network (MLPN) which consists of input, hidden and output layers with nodes and activation functions. The back-propagation algorithm (BPA) can effectively train a network for nonlinear problems (Yoon *et al.*, 2011)

This neural network architecture is very popular, because it can be applied to many different tasks. To understand this neural network architecture, one must examine how it is trained and how its processes are patterned.

The first term, “feedforward” describes how this neural network processes and recalls patterns (Anon., 2008). In a feedforward neural network, neurons are only connected foreword. Each layer of the neural network contains connections to the next layer (for example, from the input to the hidden layer), but there are no connections back (Anon., 2008).

The term “backpropagation” describes how this type of neural network is trained. Backpropagation is a form of supervised training (Anon., 2008). When using a

supervised training method, the network must be provided with both sample inputs and anticipated outputs. The anticipated outputs are compared against the actual outputs for given input. Using the anticipated outputs, the backpropagation training algorithm then takes a calculated error and adjusts the weights of the various layers backwards from the output layer to the input layer (Anon., 2008).

The phases of the Backpropagation and activities in the various phases is explained by Baboo and Shereef (2010) below:

#### A. Phases in Backpropagation Technique

The backpropagation learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation: Each propagation involves the following steps:

- i. Forward propagation of a training pattern's input is given through the neural network in order to generate the propagation's output activations.
- ii. Back propagation of the output activations propagation through the neural network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight Update; For each weight-synapse:

- i. Multiply its input activation and output delta to get the gradient of the weight.
- ii. Bring the weight in the direction of the gradient by adding a ratio of it from the weight. This ratio impacts on the speed and quality of learning; it is called the learning rate.

The sign of the gradient of a weight designates where the error is increasing; this is why the weight must be updated in the opposite direction. The phase 1 and 2 is repeated until the performance of the network is satisfactory.

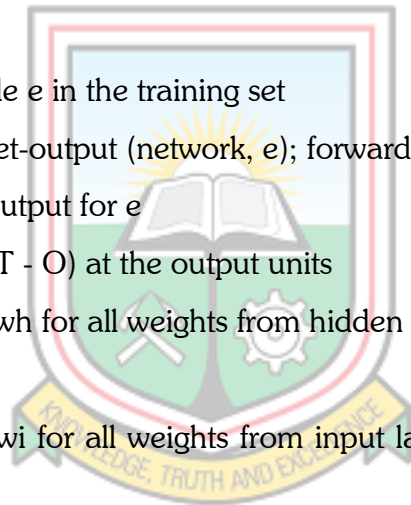
## B. Modes of Learning:

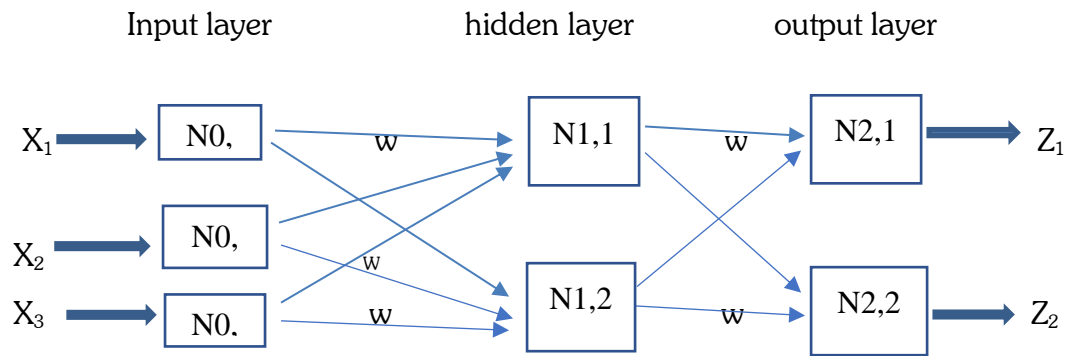
There are essentially two modes of learning to choose from, one is on-line learning and the other is batch learning. Each propagation is followed immediately by a weight update in online learning. In batch learning, much propagation occurs before weight updating occurs. Batch learning needs more memory capacity, but on-line learning requires more updates.

## C. Algorithm

Actual algorithm for a 3-layer network (only one hidden layer):

- i. Initialize the weights in the network (often randomly)
- ii. Do
- iii. For each example  $e$  in the training set
  - $O$  = neural-net-output (network,  $e$ ); forward pass
  - $T$  = teacher output for  $e$
- iv. Calculate error ( $T - O$ ) at the output units
- v. Compute  $\delta_{wh}$  for all weights from hidden layer to output layer; backward pass
- vi. Compute  $\delta_{wi}$  for all weights from input layer to hidden layer; backward pass continued
- vii. Update the weights in the network
- viii. Until all examples classified correctly or stopping criterion satisfied
- ix. Return the network





**Figure 3.5 Back Propagation Neural Network Architecture (Source: Baboo and Shereef, 2010)**

In Figure 3.5,

- i. The output of a neuron in a layer moves to all neurons in the following layer.
- ii. Each neuron has its own input weights.
- iii. The weights for the input layer are assumed (fixed) to be 1 for each input. In other words, input values are not changed.
- iv. The output of the NN is obtained by applying input values to the input layer, passing the output of each neuron to the following layer as input.
- v. The Back Propagation NN should have at least an input layer and an output layer. It could have zero or more hidden layers.

The number of neurons in the input layer is decided by the available number of possible inputs. The number of neurons in the output layer depends on the number of desired outputs. The number of hidden layers and number of neurons in each hidden layer cannot be well defined in advance, and could change per network configuration and type of data. Generally, the addition of a hidden layer could allow the network to learn more complex patterns, but at the same time decreases its performance. A network configuration can have a single hidden layer, but more hidden layers can be added if the network is not learning well (Baboo and Shereef, 2010).

### 3.3.3 Generalised Regression Neural Network (GRNN)

The Generalised Regression Neural network (GRNN) is made up of four layers; an input layer, a pattern (hidden) layer, a summation layer and an output layer.

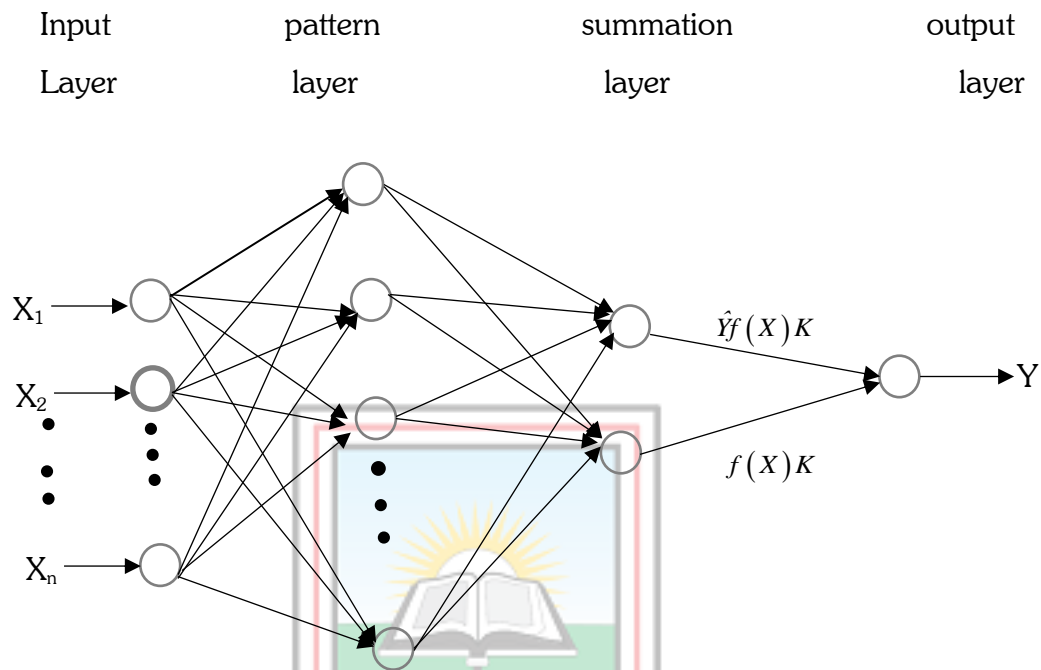


Figure 3.6 Diagram of GRNN (Source: Shinypradeepa and Vaithilingam, 2017)

The input layer which is the first is connected to the second one, the pattern layer. Each unit represents a training pattern and its output is a measure of the distance of the input from the stored patterns (Shinypradeepa and Vaithilingam, 2017). Each unit in the pattern layer is connected to the two neurons in the summation layer, thus; S-summation neuron and D-summation neuron where the former computes the sum of the weighted outputs of the pattern layer while the later calculates the outweighed outputs of the pattern neurons (Shinypradeepa and Vaithilingam, 2017). The weight connecting the  $i$ th neuron in the pattern layer and the S-summation neuron is  $y_i$  which is also the target output value corresponding to the  $i$ th input pattern. The connection weight between the  $i$ th neuron and the D-summation neuron is unity. The last layer which is the output layer merely divides the output of each S-summation neuron by that of each D-summation neuron yielding the predicted value to an unknown input vector  $x$  given in Equation 3.1

$$\hat{y}_i(x) = \frac{\sum_{i=1}^n y_i \exp[-D(x, x_i)]}{\sum_{i=1}^n \exp[-D(x, x_i)]} \quad (3.1)$$

where  $n$  is the number of training patterns and the Gaussian D function is given in Equation (3.2).

$$D(x, x_i) = \sum_{j=1}^p \left( \frac{x_j - x_{ij}}{\zeta} \right)^2 \quad (3.2)$$

where  $p$  is the number of elements of an input vector,  $x_j$  and  $x_{ij}$  are the  $j$ th element of  $x$  and  $x_i$  respectively and  $\zeta$  is generally referred to as the width parameter (spread factor).

GRNN is used to estimate continuous variables as in the case of standard regression techniques and is also based on kernel regression, as related to the radial basis function network (Shinypradeepa and Vaithilingam, 2017). The joint probability density function (pdf) of  $x$  and  $y$  is estimated during a training process in the GRNN. Because the pdf is derived from the training data with no preconceptions about its form, the system is perfectly general. The success of the GRNN method depends heavily on the width parameter. The larger that width parameter is, the smoother the function approximation. Too large a width parameter means a lot of neurons will be required to fit a fast-changing function. Too small a spread means many neurons will be required to fit a smooth function, and the network may not generalize well. GRNN needs only a fraction of the training samples that a back propagation neural network would need. GRNN is advantageous due to its ability to converging to the underlying function of the data with only few samples available. This makes the GRNN very useful and handy for the problems with inadequate data. The configuration of the neural network model is determined by the nature of the problem to be solved. The dimension of the input vector used defines the number of inputs neurons.

The various patterns of the GRNN and the processes that take place in the patterns is explained by Alomair and Garrouch (2016) below:

- i. An input layer: The number of neurons in the input layer is equal to the number of independent variables. Values for these neurons are standardized by subtracting the lowest variable value and dividing by the variable range. The input neurons feed these standardized values to each of the neurons in the hidden layer.
- ii. A hidden layer: This layer associates one neuron for each row in the training data set matrix. The neuron stores the values of the independent variables for a particular row with the corresponding dependent variable values. During training, the GRNN calculates the Euclidean distances for each input neuron of the input vector ( $X$ ). The distances from each input neuron are fed with a smoothing factor into a nonlinear exponential activation function. The resulting value is passed to the neurons in the summation layer.
- iii. A summation layer: Next, neurons of the summation layer execute the dot product between the weighted A and B coefficients and the activation results from the input neurons. The summation layer consists of two neurons: a denominator summation unit neuron and a numerator summation unit neuron. The denominator summation unit neuron sums the weight values that arrive from each of the hidden neurons. The numerator summation unit neuron sums the weight values that are multiplied by the actual dependent value for each hidden neuron.
- iv. An output layer: This layer divides the values accumulated in the numerator summation neuron unit by the values accumulated in the denominator summation neuron unit. The resulting values are presented as the predicted output value. The weights of the hidden layer neurons represent cluster centers of a multidimensional space. A and B coefficients related with these clusters are taken as the weights that connect the summation layer and the hidden layer.



The smoothing factor is perhaps the most prominent computing parameter for the GRNN's paradigm.

The optimum  $r$  for a GRNN built from a training data set is approximated using an initial guess. Next, the sum squared error (SSE) is calculated for the training data set. This process is repeated by varying  $r$  by a fixed increment. After using a series of trials for  $r$ , the smoothing factor associated with the smallest SSE is used to represent the optimum  $r$  for the training data set.

### 3.4 Studies Conducted in Relation to Satellite Altimetry and Various Methods Employed

According to Mlybari *et al.* (2014), Sea level rise threatens to increase the impact of future storms and hurricanes on coastal communities and that accurate sea level change prediction is an important factor in determining constructions and human activities in coastal and oceanic areas, support vector machines (SVM) was proposed to predict daily tidal levels along the Jeddah Coast, Saudi Arabia. The optimal parameter values of kernel function were determined using a genetic algorithm. The SVM results are compared with the field data and with back propagation (BP). Among the models, the SVM was superior to BPNN and has better generalization performance.

Biswas and Jayawardena (2014) used artificial neural network to predict water level in a flashy transboundary river of Bangladesh. The study presented the results of the Artificial Neural Network model for water level and it was found that when the hidden layer was one with two hidden neurons was when the model is adequate to train the network. The training procedure is sped up when the number of hidden neurons increases but the generalization was too bad for the application. The writers have successfully created a model that recognizes the intricate pattern of water levels.

According to Cheng *et al.* (2014), satellite altimetry is very necessary in predicting and monitoring sea level in the Arctic Ocean. Even though researchers do not have enough Satellite altimetry data because of the almost permanent ice cover in the arctic.

Increase in level of the mean global ocean that is recorded during the 20<sup>th</sup> and 21<sup>st</sup> centuries is one of the important indicators of climate change (Marijan *et al.*, 2017). The researchers studied on the Eastern part of the central Adriatic Sea in Croatia where emphasis was placed on detecting possible areas that need special attention with respect to monitoring sea level change and monitoring of geophysical processes, it was concluded that the sea level trend predicted from CryoSat-2 show the average rate of 3.44mm/yr which tallies with current global studies.

According to Makarynsky *et al.* (2004), the knowledge of near-shore sea-level variations plays an important role in the practical realisation of the height reference surface in geodesy. ANN was used to retrieve sea level data from remote data sources and it was realised that ANNs with simpler architectures do not include enough information to perform as well while more complicated nets may suffer from overfitting. Constructing the model were the previous daily tidal level observations. Evaluating the model with a different number of previous daily sea level change values led to a conclusion that the best result could be achieved when using only three previous tidal level values. Adding more of the previous data to the inputs did not change the result. To determine the optimum sizes of BPNN based on the determined input numbers, the networks with one hidden layer were used for training and testing by changing neuron sizes in a hidden layer in order to test the stability of the network. Overall, it could easily be observed from that BPNN that having architecture 3-5-1 (three neurons in input and five neurons in first hidden layers, and one neuron in output layer) produced the best result in the study.

Also, in Strassburg *et al.* (2014), two methods for sea level reconstructing which differ in the techniques for calculating basis functions thus empirical orthogonal functions (EOFs) and cyclostationary EOFs (CSEOFs) was used to calculate basis functions that were formed by decomposing data from satellite altimetry and fitting those functions to in situ tide gauge measurements with a longer temporal record. It was concluded that the CSEOF reconstructions show high skill in reproducing variations in global mean sea level (GMSL) and ocean climate indices and are affected less by both limited tide gauge distribution and added high-frequency tide gauge signal than EOF

reconstructions. Typically, CSEOF reconstructions slightly underestimate sea level amplitudes while EOF reconstructions overestimate sea level amplitudes in some cases. It was also examined that the CSEOF method of sea level reconstruction produce more robust reconstructions especially when less than perfect tide gauge data are used.

In Gharineiat and Deng (2015), twenty years of data from several satellite missions combined with tide gauges was used to determined sea level fields in the northern coast of Australia using Multi-Adaptive Regression Splines (MARS) and multiple regression-techniques. The mean R-squared error of 0.62 and Root Mean Squared (RMS) error of 0.67 cm was obtained from modelling sea levels by MARS. The MARS gives a higher account of sea level variance than the multiple regression technique thereby suggesting that the MARS can be used to efficiently monitor sea level.

According to Singh *et al.* (2013), Satellite altimetry has developed over the years to become a reliable technology for monitoring sea level. Measurement of sea level by satellite altimetry gives a near Global coverage of the oceans which is better off than in the case of coastal tidal gauges. Satellite altimetry is advantageous over tide gauge measurements since challenges of stability among others are overcome thereby is an effective sea level monitoring and coastal zone management tool. Several studies have been undertaken to show the applicability of the use of altimetric data as an alternative to tide gauge measurements (Madsen *et al.*, 2007; Ginzburg, 2010 and Robinson, 2010).

Also, Niedzielski and Kosek (2008) conducted a study where global mean sea level anomalies (SLAs) and gridded SLA in the east equatorial Pacific obtained from TOPEX/Poseidon and Jason-1 altimetric measurements were predicted using extrapolation of a polynomial-harmonic deterministic least-squares (LS) model describing a linear trend, annual and semi-annual oscillations and a combination of the extrapolation of a polynomial-harmonic model with the autoregressive forecast of LS residuals (LS+AR). Both methods allow one to compute the predictions of comparable accuracy (root mean square error) when forecasting global mean SLA data but in the case of predicting gridded SLA data, the LS+AR prediction method

gains significantly better prediction accuracy than the accuracy obtained by the LS technique during El Niño 1997/1998, La Niña 1998/1999 and during normal conditions.

In Ratheesh *et al.* (2015) also, sea level anomaly (SLA) acquired by SARAL/Altika has been examined by assimilating it along with Jason-2 and Cryosat-2 SLA in the Princeton Ocean model (POM) using ensemble optimal interpolation (EnOI) technique. The importance of SARAL SLA in a data assimilative ocean prediction system has been evaluated with special emphasis on the improvement in thermocline depth. It was observed that SARAL/Altika SLA has improved the quality of simulation.

Kurniawan *et al.* (2014) tried using Genetic Programming (GP) to model sea level anomalies data in the Singapore Regional Waters (SRW) and then used the results to modify the numerical model prediction in real time applications. It was found in the study that the error produced in using GP in SLA can be used in improving SLA prediction from primary hydrodynamic models. A good performance of non-tidal barotropic numerical modelling and GP error forecast model was also shown from the results which can be used to forecast SLA in Singapore.

Kang *et al.* (2006) also studied and found that the sea level rise in the East Japan Sea (EJS) observed from analysis of sea level anomalies (SLA) from TOPEX/Poseidon (T/P) are much larger than that of the global rates. The study went further to determine that thermal expansion is the main factor for sea level rise in the EJS but there was a nonuniform pattern in the southern EJS which comes as result of variable thermal expansion that resulted from temperature variability in the upper layer of the two basins. In addition, it was observed that there could be local non-eustatic increases in the sea level as a result of changes in water volume in other parts of the world ocean.

In Karamouz *et al.* (2014), monitoring sea level anomaly is an important factor for planning in coastal regions and it is for this reason that the parameters affecting sea level change such as sea level pressure (SLP) and sea-surface temperature (SST) were investigated in the coastal area of New York City. It was revealed that there is high

correlation between SLP, SST and sea level changes. Karamouz *et al.* (2014) used some models of ANFIS and three artificial neural networks (ANNs) including a multilayer perceptron (MLP), Elman recurrent neural network and a special class of time-delay neural network (TDNN) namely IDNN to simulate sea level changes. From the results, it was observed that the reliability of sea level predictions was improved when a hybrid model was used.

Ondimu and Murase (2006) also used neural network (NN) models to forecast reservoir levels for lake Naivasha in Kenya. The parameters that were used to develop the NN models were rainfall, water levels, evaporation rate, one pair of time harmonics and discharges for rivers Malewa and Gilgil. It was observed that the more the number of feature groups used the more accurate the neural networks are able to forecast reservoir levels.

Miles *et al.* (2014) conducted a study in which a dynamical coupled ocean atmosphere model was employed to predict seasonal sea level anomalies (SLA) for 7 months in advance globally. This was a way of foreseeing behavior of sea levels in extreme events so as to be able to put measures in place to preserve coastal life and infrastructure.

In Nishida *et al.* (2006) also, Genetic programming which is a new development was used to predict sea level anomalies at Kuchinotsu port and the results that were obtained were compared to results obtained from artificial neural network models. In the study, even though the genetic programming was trained with limited data, computational conditions such as observed data fed to model and parameters on genetic procedures were chosen with care to ensure optimum results.

Furthermore, in Davis *et al.* (2012), because of the risk rising sea levels pose to coastal zones in the Caribbean, satellite altimetry was used as a method to fill the gap in the spatially effective system for monitoring sea level in the Caribbean.

Sterlini *et al.* (2017) satellite altimetry was used to examined the variation of sea surface height trends in the North Sea Basin. The effects of variations in the salinity

and potential temperature of the study area, meteorological processes acting on the sea surface, steric and meteorological regional sea level rise anomalies summed up with those from contributions from land ice and are compared with satellite altimetry over the period 1993-2014. The results show that there is agreement between the observations and the reconstruction.

The extensive literature reviewed reveals that satellite altimetry and ANN methods are better in studying SLA in that the satellite data covers a broader area as compared to tidal gauges, also satellites can be used to acquire data in portions of the ocean where it is impossible for tidal gauges. ANN methods are better off because they are widely used and also give better predictions when compared to other methods. Even though these methods have been applied to satellite data to study SLA in other parts of the world, they are yet to be applied in the Gulf of Guinea.



## CHAPTER 4

### MATERIALS AND METHODS USED

#### 4.1 Overview

This chapter presents the materials as well as the various methods used in the study. Two major methods thus the Backpropagation Neural Network (BPNN) and the Generalised Regression Neural Network (GRNN) were used in the short-term prediction of SLA in the Gulf of Guinea (GoG) thus from 2011 to 2017.

#### 4.2 Materials

##### 4.2.1 Data

The data for the research comprising of wind flux, heat, humidity and SLA data were all acquired from the Asia-Pacific Data-Research Center (<http://apdrc.soest.hawaii...>). The data is a secondary data which has already been validated by calibration and corrections for errors at the Asia-Pacific Data Research-Center before acquisition. Errors corrected include:

- i. The Inverse Barometric (IB) correction
- ii. Ionospheric correction
- iii. Sea state correction and
- iv. Solid earth and pole tide correction

Figure 4.1 shows a map of Africa where the study area is located which is further zoomed in to the study area defined by Latitude 4°N, Latitude 0°N, Longitude 7°W and Longitude 6°E in Figure 4.2. both areas are being delineated with a yellow square/rectangle.

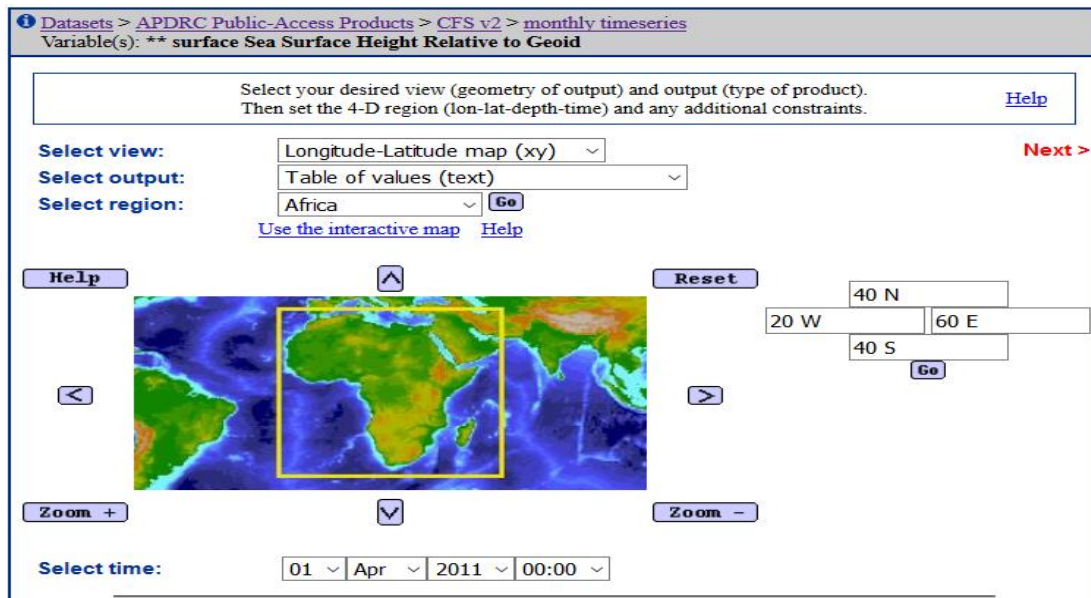


Figure 4.1 Continent of Africa showing study area (Source: Anon., 2019)

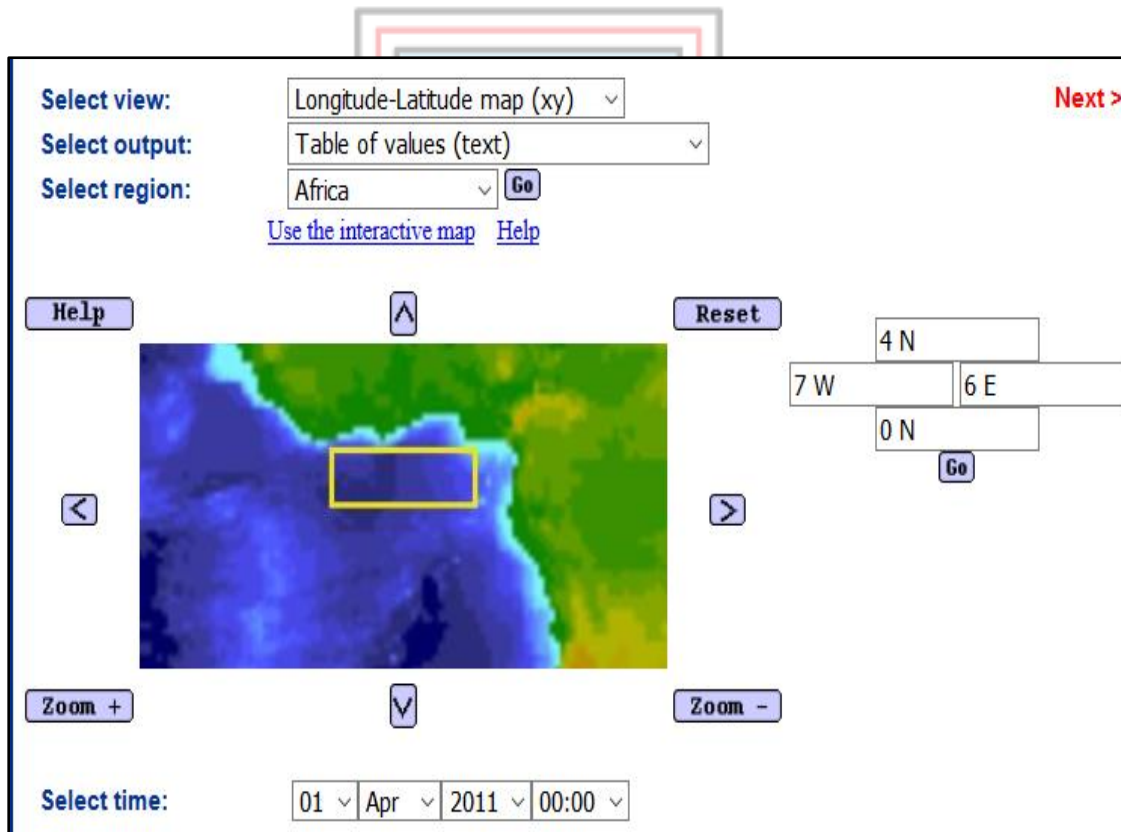


Figure 4.2 Study Area (Source: Anon., 2019)

The study area is divided into a grid as shown in Figure 4.3. where every intersection of the grid lines within the study area is the point of data acquisition.



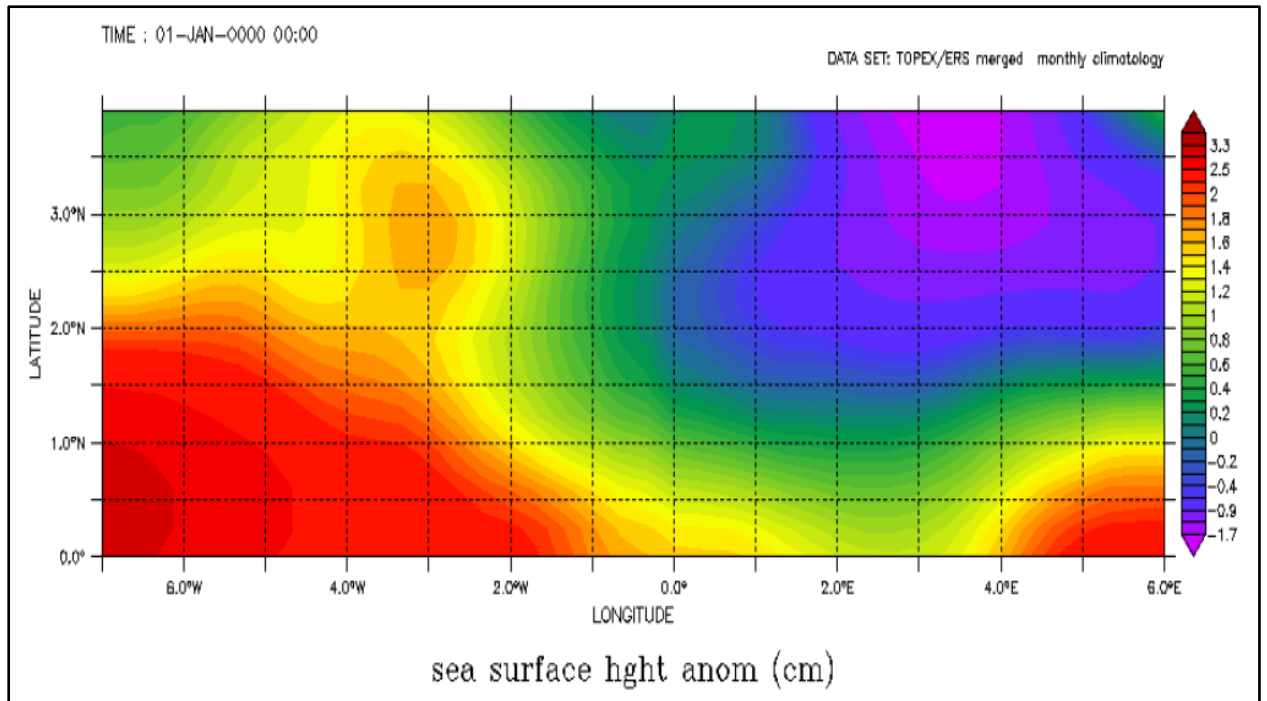


Figure 4.3 Study Area Divided into Grid (Source: Anon., 2019)

Table 4.1 displays the data for humidity which actually consists of 27 columns and 476 rows in all.

Table 4.1 Humidity Data

Time	Dates and Grid Intervals	Grid line Intersections					
		7W(-7)	6.5W	6W(-6)	5.5W	5W(-5)	4.5W
189:00:00	4N	0.01619	0.01588	0.01561	0.01568	0.01589	0.01561
188:00:00	3.5N	0.01562	0.01529	0.01514	0.0155	0.01561	0.01558
187:00:00	3N	0.01545	0.01519	0.01503	0.01522	0.01524	0.01531
186:00:00	2.5N	0.01542	0.01517	0.01502	0.01501	0.01507	0.01513
185:00:00	2N	0.01549	0.01541	0.01522	0.01516	0.01523	0.01532
184:00:00	1.5N	0.01565	0.01567	0.01551	0.01546	0.01558	0.01566
183:00:00	1N	0.01568	0.01574	0.01572	0.01575	0.01581	0.01583
182:00:00	0.5N	0.01572	0.01577	0.01581	0.01583	0.01584	0.01583
181:00:00	0	0.01576	0.01579	0.01583	0.01583	0.01583	0.0158
	1/5/2011	7W(-7)	6.5W	6W(-6)	5.5W	5W(-5)	4.5W
189:00:00	4N	0.01491	0.01399	0.01188	0.01056	0.0104	0.01103
188:00:00	3.5N	0.01464	0.01457	0.01354	0.0126	0.01218	0.01287
187:00:00	3N	0.01468	0.01505	0.01497	0.01459	0.01437	0.01488
186:00:00	2.5N	0.01488	0.01464	0.01444	0.01434	0.01473	0.01526
185:00:00	2N	0.01478	0.01427	0.01404	0.01418	0.01469	0.01517

**Table 4.1 Continued**

184:00:00	1.5N	0.0148	0.01448	0.0145	0.01477	0.015	0.01517
183:00:00	1N	0.0148	0.01468	0.01485	0.01503	0.01504	0.01509
182:00:00	0.5N	0.0149	0.0149	0.01493	0.01497	0.01502	0.01508
181:00:00	0	0.01501	0.01501	0.01493	0.01492	0.015	0.01502
	1/6/2011	7W(-7)	6.5W	6W(-6)	5.5W	5W(-5)	4.5W
189:00:00	4N	0.01479	0.01468	0.01427	0.0139	0.01422	0.01477
188:00:00	3.5N	0.01415	0.01394	0.01367	0.01405	0.01448	0.01497
187:00:00	3N	0.0134	0.01326	0.01323	0.01385	0.01437	0.01472
186:00:00	2.5N	0.01305	0.01268	0.01262	0.01306	0.01375	0.0141
185:00:00	2N	0.01246	0.01222	0.01206	0.01239	0.01289	0.01326
184:00:00	1.5N	0.0118	0.01182	0.01179	0.01198	0.01228	0.01255
183:00:00	1N	0.01138	0.01153	0.01162	0.01177	0.01196	0.01214
182:00:00	0.5N	0.01132	0.01151	0.01165	0.0118	0.01195	0.01203
181:00:00	0	0.01135	0.01157	0.01176	0.01189	0.01202	0.012
	1/7/2011	7W(-7)	6.5W	6W(-6)	5.5W	5W(-5)	4.5W
189:00:00	4N	0.01395	0.01395	0.014	0.01405	0.01409	0.01409
188:00:00	3.5N	0.0132	0.0134	0.01358	0.01371	0.01374	0.01385
187:00:00	3N	0.01251	0.01277	0.01302	0.01337	0.01353	0.01363
186:00:00	2.5N	0.01216	0.01233	0.01258	0.01298	0.01335	0.01344
185:00:00	2N	0.01185	0.01206	0.01226	0.01268	0.0131	0.0132
184:00:00	1.5N	0.01153	0.01181	0.01201	0.01246	0.01271	0.01282

(Source: Anon., 2019)

All four datasets thus humidity, wind flux, heat and SLA have the same number of columns and rows of 27 and 592 respectively of data. To be able to read the data in MATLAB, the average value of every row is calculated for the four datasets as shown in Table 4.2.

**Table 4.2 Humidity, Wind, Heat and SLA Data**

Wind (m/s)	Heat (J)	Humidity	SLA (m)
6.804815	5.658594	0.01536033	0.03213
6.719491	5.699231	0.01537259	0.030027
6.22787	5.831306	0.01523977	0.030973
5.717731	5.950256	0.01518444	0.033209
5.227315	6.049518	0.01520028	0.035379
4.812731	6.141397	0.01522389	0.037005

**Table 4.2 Continued**

4.502778	6.223431	0.01522704	0.03839
4.278472	6.307106	0.01521273	0.039126
4.118981	6.412931	0.01517759	0.032411
4.176214	6.540904	0.01516321	0.029105
4.240972	6.69759	0.01528583	0.030006
4.365787	6.872324	0.01532037	0.031381
4.544861	7.064284	0.01524889	0.033282
4.73537	7.271831	0.01510532	0.035315
4.905833	7.487582	0.01490759	0.03695
5.031574	7.706614	0.01464866	0.03811
5.128287	7.913341	0.01432079	0.031216
5.203657	8.098739	0.01393245	0.025871
5.298683	8.256245	0.01330156	0.021354
5.411713	8.394063	0.01309153	0.01736
5.397269	8.502348	0.01305338	0.013376
5.413009	8.514063	0.01303861	0.00982

(Source: Anon., 2019)

Table 4.2 above is a sample of the data used for the prediction, the number of columns are four (4) representing wind flux (m/s), heat (J), humidity and sea level anomaly (m) in the GoG. The total number of rows is equal to five hundred and ninety-two (592).

### 4.3 Methods Used

#### 4.3.1 Back Propagation Neural Network (BPNN)

The input parameters thus; humidity, wind flux and heat data were fed into the input layer of the BPNN. The output data which is the SLA data was fed into the output layer. In order to train the neural network, the data set was first normalized, to ensure constant variability in the model. That is, generally, the original data are expressed in different units with different physical meaning, and thus the absolute values order of

magnitude is highly variable. Also, the data normalization improves convergence speed. In this research, the Mapminmax scaling function that transforms input and target data by mapping it from its original range to the interval [-1, 1] was applied. The Mapminmax function is expressed as:

$$y = y_{min} + \frac{(y_{max}-y_{min}) \times (x-x_{min})}{(x_{max}-x_{min})} \quad (4.1)$$

where  $y_{max}$  is equal to 1,  $y_{min}$  is equivalent to -1,  $x$  is the current value of the data,  $x_{min}$  and  $x_{max}$  is the minimum and maximum thresholds of the observed variables,  $y$  is the scaled value

Training of the FFBPNN was then carried out after normalization of both input and output datasets. In the training, the datasets were divided into three subsets. That is, 70% for training set serving as parameterization (weight adjustment); 15% as validation set to evaluate the performance of the trained network; and 15% as testing set to assess the generalisation capabilities of the ANN model developed. The number of neurons in the input and output layers have been set as 3 and 1 respectively for each network structure.

However, the number of neurons in the hidden layer is difficult to determine because there is no universally accepted theory to clarify the number of hidden units needed to approximate a given function (Hornik *et al.*, 1989). As a result, a trial and error procedure is usually adopted during training after trying several network architectures in order to obtain the optimal number of hidden neurons thereby achieving the best network. This was done by changing the number of neurons in the hidden layer during the training from 1 to 20. Experimentally it was found that when the neurons in the hidden layer were 4 the model gave a better accuracy based on the MSE values that were obtained.

For the network training, the Levenberg-Marquardt algorithm, Scaled Conjugate Gradient and Bayesian Regularisation was used to train the FFBPNN. Based on the

data normalization function that was used, the activation function in the hidden layer used in this research was the hyperbolic function. This is given as;

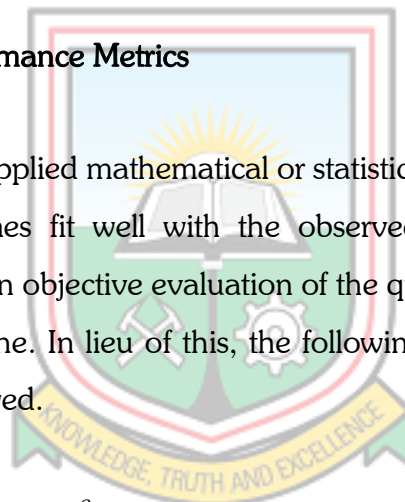
$$\text{tansig}(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \quad (4.2)$$

#### 4.3.2 Generalised Regression Neural Network (GRNN)

The humidity, wind flux and heat were also fed into the input layer of the GRNN. The output data fed into the output layer was the sea level anomaly data. The data is processed through to the pattern layer, the summation and results displayed at the output layer.

#### 4.4 Statistical Performance Metrics

The usefulness of any applied mathematical or statistical model depends on how close their predicted outcomes fit well with the observed data. By applying statistical quantitative methods, an objective evaluation of the quality of the results produced by the model could be done. In lieu of this, the following statistical performance metric (PM) tools were employed.



##### *Coefficient of Determinant ( $R^2$ ) in Regression:*

The proportion of the (corrected) total variation in the y's that can be attributed to regression on the x's is denoted by  $R^2$  which is given as Equation

$$R^2 = \frac{\text{regression sum square (SSR)}}{\text{total sum square (SST)}} = \frac{B^2 X^2 y - n y^{-2}}{y^2 y - n y^{-2}} \quad (4.3)$$

The ratio  $R^2$  is called the coefficient of multiple determination, or commonly the squared multiple correlation. The multiple correlation R is defined as the positive square root of  $R^2$ .

Table 4.3 shows the minimum  $R^2$  considered statistically significant with a power of 0.80.

**Table 4.3 Minimum  $R^2$  Considered Statistically Significant**

Sample Size	Significance Level ( $\alpha$ )=0.05 No. of Independent Variables			
	2	5	10	20
20	39	48	64	NA
50	19	23	29	42
100	10	12	15	21
250	4	5	6	8
500	3	4	5	9
1000	1	1	2	2

Note: Values represent percentage of variance explained; NA=not applicable. (source: Hair *et al.*, 2013)

#### *Mean Square Error (MSE)*

The Mean Square Error (MSE) is a frequently used measure of the differences between predicted values by a model or an estimator and the values observed. The performance index is expressed as Equation (4.13)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.4)$$

where  $y_i$  is the  $i^{th}$  observed value;  $\hat{y}_i$  is the  $i^{th}$  predicted value;  $n$  is the sample size.

#### *Root Mean Square Error (RMSE)*

The Root Mean Square Error (RMSE) is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled. These individual differences are also called residuals, and the RMSE serves to aggregate them into a single measure of predictive

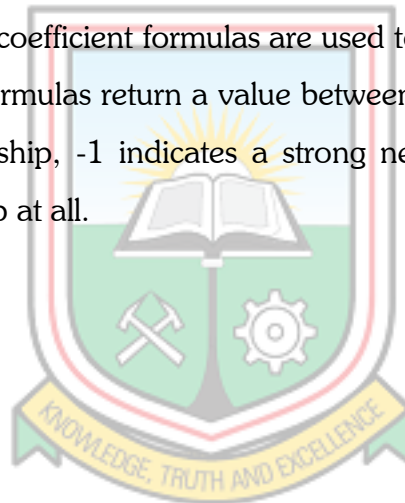
power. The RMSE of a model prediction is defined as the square root of the mean squared error as shown in Equation (4.15)

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

where  $y_i$  is the  $i^{th}$  observed value;  $\hat{y}_i$  is the  $i^{th}$  predicted value;  $n$  is the sample size.

### *Correlation Coefficient (R)*

Correlation coefficients are used in statistics to measure how strong a relationship is between two variables. There are several types of correlation coefficients: Pearson's correlation (also called Pearson's R) is a correlation coefficient commonly used in linear regression. Correlation coefficient formulas are used to find how strong a relationship is between data. The formulas return a value between -1 and 1, where: 1 indicates a strong positive relationship, -1 indicates a strong negative relationship and a zero indicates no relationship at all.



## CHAPTER 5 RESULTS AND DISCUSSION

### 5.1 Overview

This chapter presents the sequential manner through which the SLA data was collated, the models that were developed and discussion of the results derived from the two mathematical models.

### 5.2 Collation of SLA data in the GoG

The study area is divided into a grid as shown in Figure 5.1. The intersection of the grid lines are points the anomaly values are derived from which are tabulated in Table 5.1.

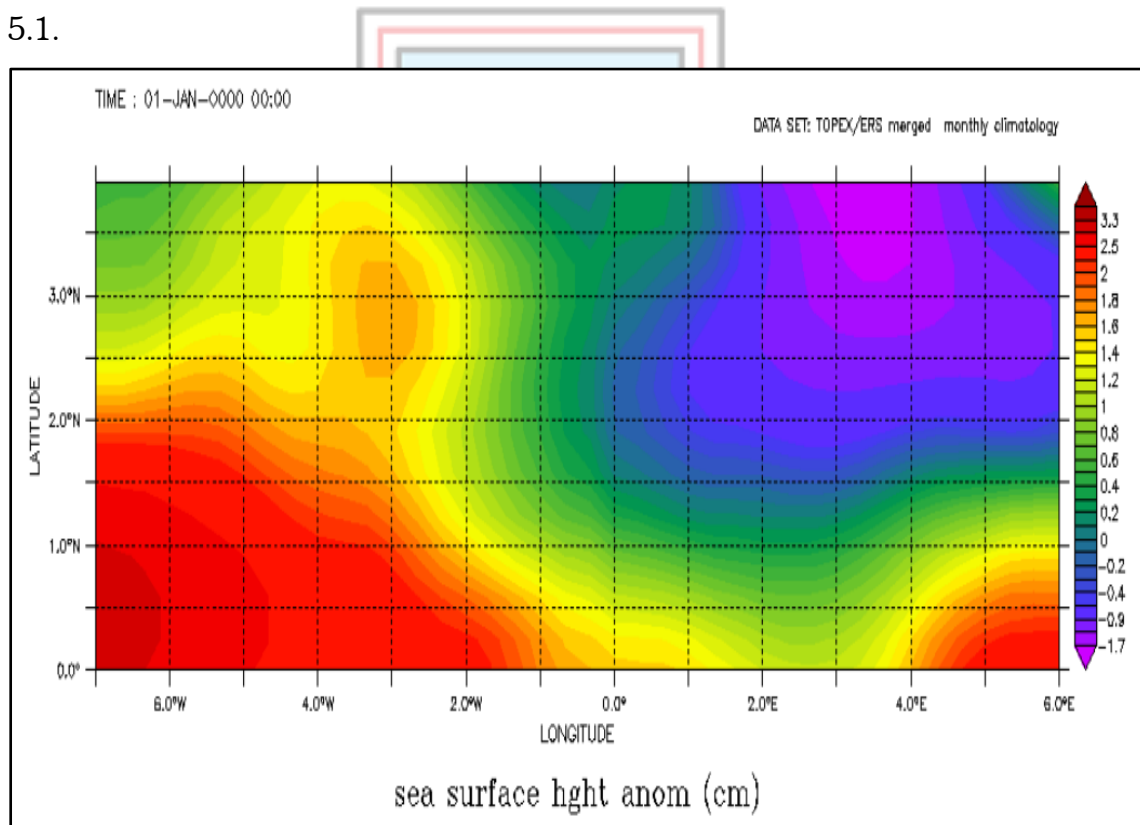


Figure 5.1 Obtaining SLA data from study Area (Source: Anon., 2019)

In order to obtain the SLA data in a single column, the average of every single row of the 592 rows of SLA data is calculated and is shown in the last column of Table 5.1.



**Table 5.1 Sample of SLA Data Averaged (m)**

2.75E	3.25E	3.75E	4.25E	4.75E	5.25E	5.75E	Average
0.056	0.055	0.056	0.057	0.067	0.055	0.064	0.0321298
0.044	0.041	0.042	0.043	0.044	0.046	0.05	0.0300275
0.033	0.034	0.032	0.035	0.037	0.037	0.04	0.0309725
0.027	0.025	0.024	0.024	0.028	0.032	0.036	0.0332088
0.019	0.016	0.015	0.016	0.02	0.028	0.036	0.0353791
0.008	0.006	0.004	0.005	0.012	0.019	0.028	0.0370055
0.001	-0.003	-0.004	-0.004	0	0.008	0.012	0.0383901
-0.004	-0.004	-0.008	-0.008	-0.005	-0.003	0.003	0.0391264
2.75E	3.25E	3.75E	4.25E	4.75E	5.25E	5.75E	
0.055	0.06	0.065	0.067	0.068	0.066	0.07	0.0324106
0.056	0.062	0.065	0.068	0.064	0.058	0.06	0.029105
0.052	0.055	0.059	0.059	0.06	0.055	0.055	0.0300055
0.045	0.047	0.049	0.051	0.051	0.052	0.054	0.0313812
0.035	0.037	0.038	0.042	0.044	0.045	0.048	0.0332818
0.018	0.021	0.024	0.026	0.031	0.036	0.038	0.0353149
0.002	0.003	0.006	0.008	0.012	0.018	0.024	0.0369503
-0.013	-0.011	-0.01	-0.007	-0.005	-0.001	0.007	0.0381105
2.75E	3.25E	3.75E	4.25E	4.75E	5.25E	5.75E	
0.064	0.067	0.067	0.066	0.064	0.054	0.052	0.0312157
0.055	0.054	0.052	0.052	0.049	0.043	0.043	0.0258708
0.042	0.042	0.04	0.038	0.037	0.037	0.039	0.0213539
0.032	0.031	0.029	0.028	0.03	0.031	0.036	0.0173596
0.02	0.02	0.02	0.02	0.023	0.026	0.03	0.0133764
0.003	0.004	0.005	0.008	0.013	0.017	0.019	0.0098202
-0.013	-0.011	-0.008	-0.006	-0.002	0.002	0.004	0.0068588
-0.025	-0.022	-0.02	-0.017	-0.017	-0.016	-0.014	0.0035506
2.75E	3.25E	3.75E	4.25E	4.75E	5.25E	5.75E	
0.043	0.045	0.051	0.053	0.052	0.063	0.075	-0.008951
0.032	0.033	0.034	0.038	0.043	0.044	0.054	-0.0153989
0.018	0.018	0.019	0.022	0.027	0.032	0.039	-0.0159382

The single column of SLA data is then used as the target in predicting with the two methods with humidity, wind flux and heat as the input datasets.

The models developed for the study were the BPNN and the GRNN.

### 5.3 Models Developed for Prediction

#### 5.3.1 Back Propagation Neural Network

This section presents the best achieved result by the BPNN algorithms, thus BR with two hidden neurons, LM with four hidden neurons and ScG also with four hidden neurons regarding SLA prediction in the Gulf of Guinea. Table 5.2 show the correlation coefficient (R), Coefficient of Determinant ( $R^2$ ), Mean Square Error (MSE) and Root Mean Square Error (RMSE) considering different network architectures for the training and test data sets.

**Table 5.2 Residual Analysis using BPNN Architectures**

Algorith	Architecture	Training				Testing			
		R	$R^2$	MSE	RMSE	R	$R^2$	MSE	RMSE
BR	3-1-1	0.114651	0.013144	0.001610	0.040125	0.544805	0.296812	0.001158	0.034029
	<b>3-2-1</b>	<b>0.382729</b>	<b>0.146481</b>	<b>0.001390</b>	<b>0.037283</b>	<b>0.75156</b>	<b>0.564842</b>	<b>0.000732</b>	<b>0.027055</b>
	3-3-1	0.114984	0.013221	0.001609	0.040112	0.543772	0.295688	0.001156	0.034
	3-4-1	0.517656	0.267967	0.001192	0.034525	0.641231	0.411177	0.001628	0.040348
	3-5-1	0.734016	0.538779	0.000751	0.027404	0.304548	0.092749	0.002245	0.047381
LM	3-1-1	0.298067	0.088844	0.001483	0.03851	0.063840	0.004076	0.001415	0.037616
	3-2-1	0.415051	0.172267	0.001347	0.036701	0.462525	0.213929	0.00147	0.038341
	3-3-1	0.503539	0.253552	0.001215	0.034857	0.394772	0.155845	0.001577	0.039711
	<b>3-4-1</b>	<b>0.538877</b>	<b>0.290388</b>	<b>0.001155</b>	<b>0.033985</b>	<b>0.550859</b>	<b>0.303446</b>	<b>0.001226</b>	<b>0.035014</b>
	3-5-1	0.587453	0.345101	0.001066	0.03265	0.308507	0.095177	0.01645	0.128258
ScG	3-1-1	0.298159	0.088899	0.001483	0.03851	0.065041	0.00423	0.001418	0.037656
	3-2-1	0.415051	0.172267	0.001347	0.036701	0.462086	0.213523	0.00147	0.038341
	3-3-1	0.503363	0.253374	0.001215	0.034857	0.396501	0.157213	0.001582	0.039774
	<b>3-4-1</b>	<b>0.51198</b>	<b>0.262124</b>	<b>0.001201</b>	<b>0.034655</b>	<b>0.558523</b>	<b>0.311948</b>	<b>0.001722</b>	<b>0.041497</b>
	3-5-1	0.578129	0.334233	0.001084	0.032924	0.307987	0.094856	0.014886	0.122008

### 5.3.2 Prediction

The validity of both models was assessed based on  $R$ ,  $R^2$ , MSE and RMSE. The higher the  $R$  and  $R^2$  the better the model for predictions and the lower the MSE and RMSE the better the model for predictions of SLA in the Gulf of Guinea.

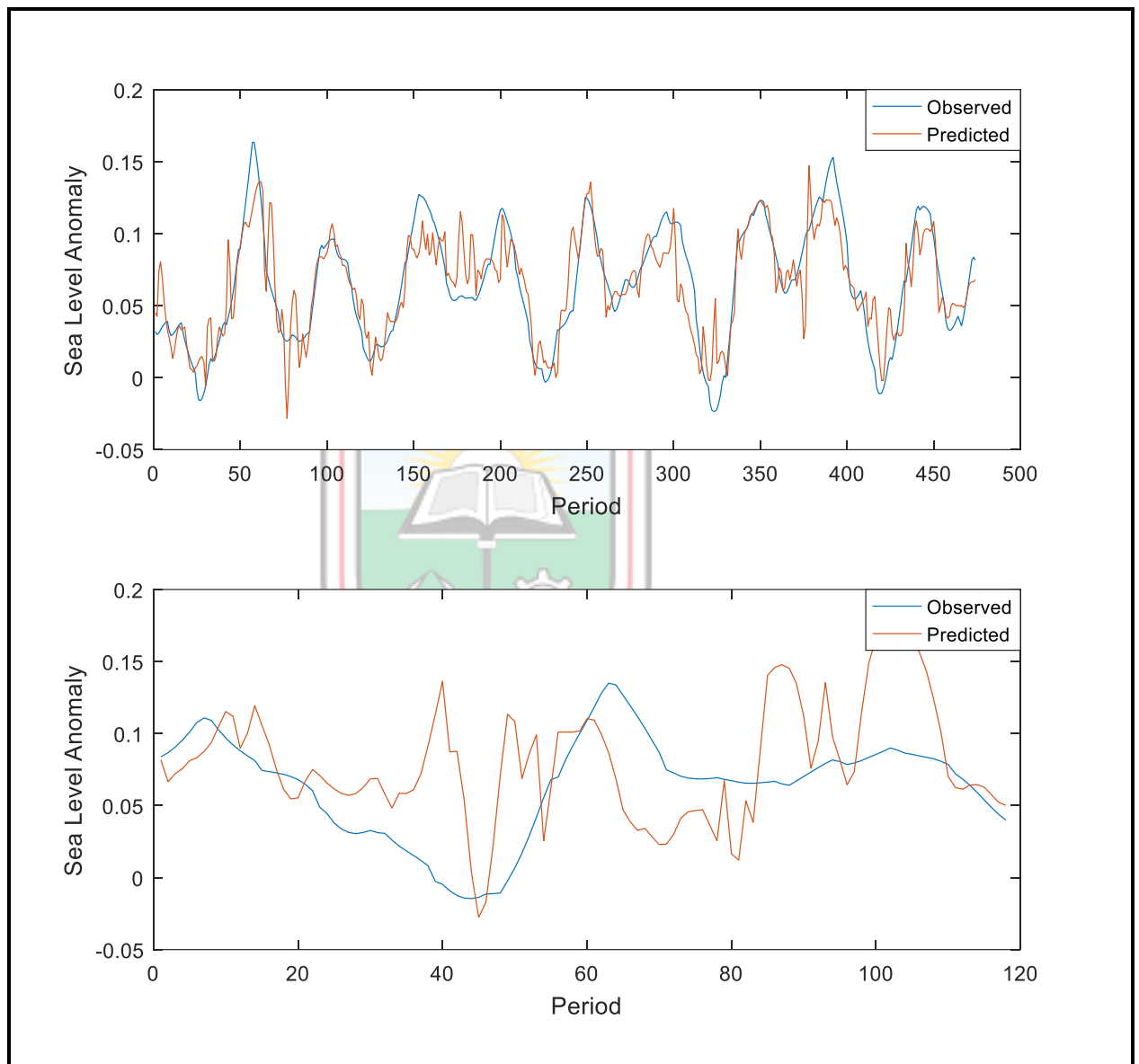


Figure 5.2 Training and Test Performance of BR Model

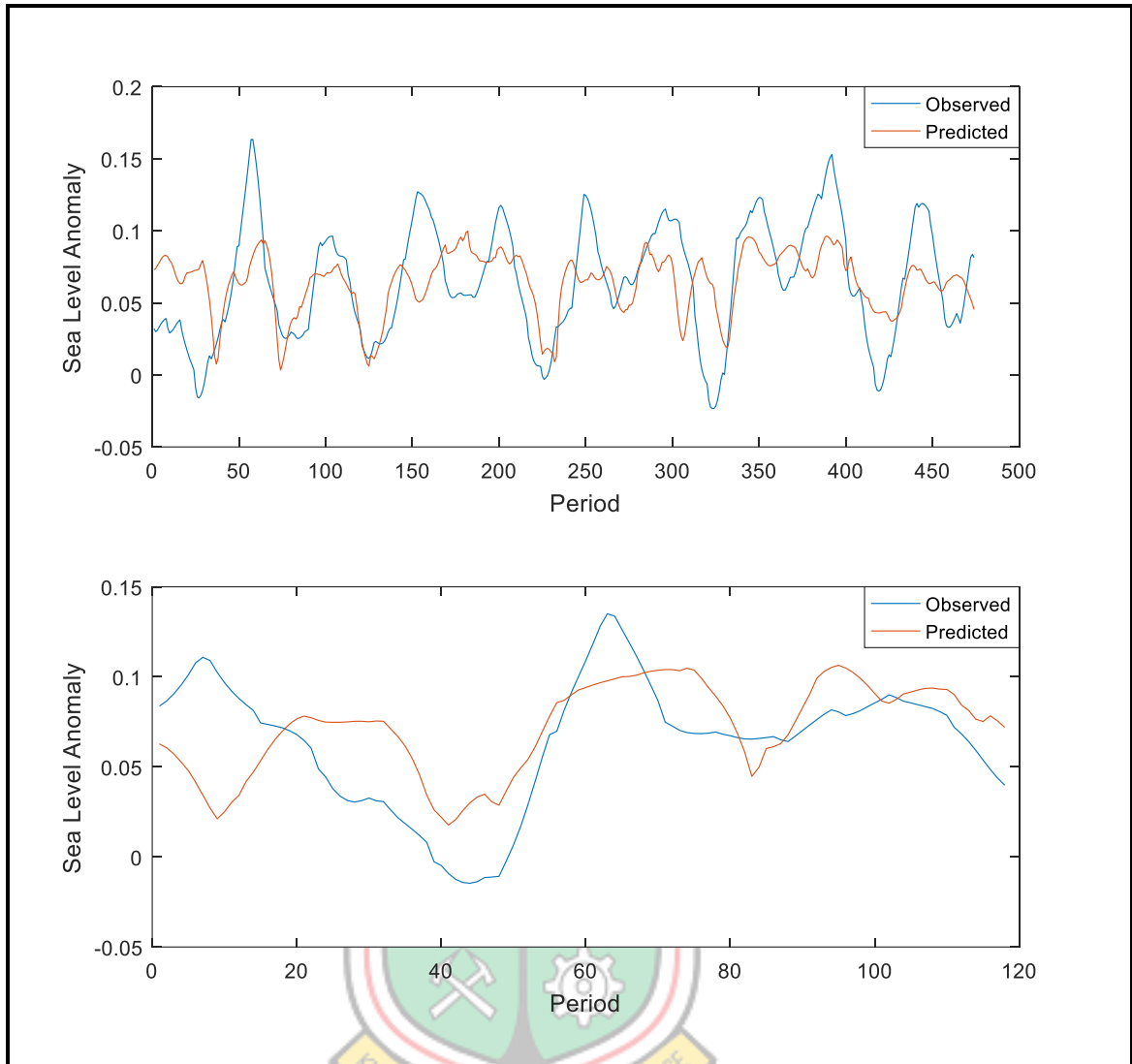
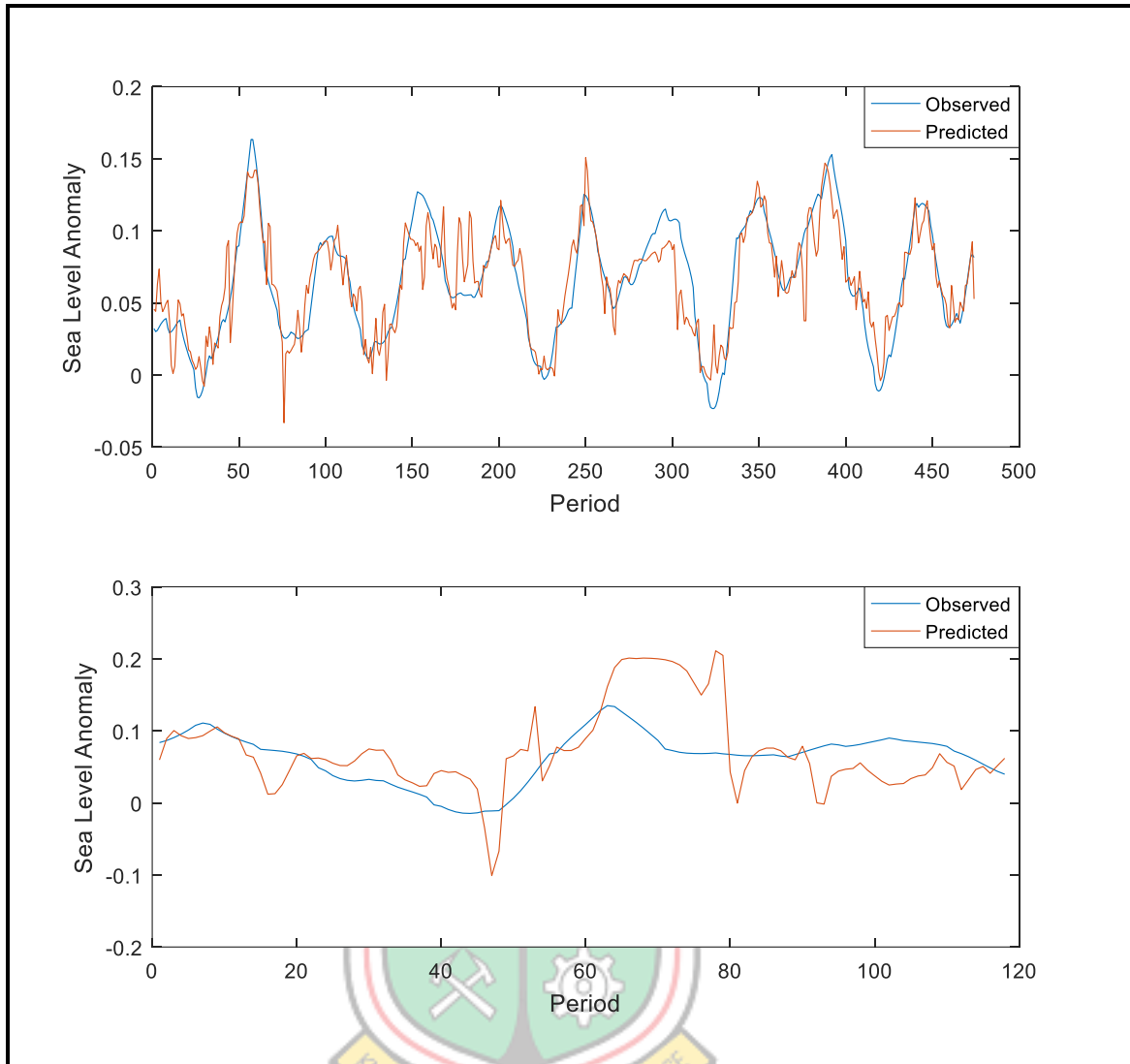


Figure 5.3 Training and Test Performance of LM Model



**Figure 5.4 Training and Test Performance of ScG Model**

### 5.3.3 Generalised Regression Neural Network

Table 5.3 shows the results from the network approximation using GRNN technique. The spread was set to 0.6 so as to allow for a closer fit of the measured data.

**Table 5.3 Residual Analysis using the Generalised Regression Neural Network Performance Statistics**

Spread	Training				Testing			
	R	R <sup>2</sup>	MSE	RMSE	R	R <sup>2</sup>	MSE	RMSE
0.10	0.919746	0.845933	0.000270	0.016432	0.492026	0.24209	0.001096	0.033106

0.20	0.763171	<b>0.58243</b>	0.000754	<b>0.027459</b>	0.412857	<b>0.170451</b>	0.001123	<b>0.033511</b>
0.30	0.644927	<b>0.415931</b>	0.001066	<b>0.03265</b>	0.412854	<b>0.170448</b>	0.001052	<b>0.032435</b>
<b>0.40</b>	<b>0.574166</b>	<b>0.329667</b>	<b>0.001261</b>	<b>0.035511</b>	<b>0.481043</b>	<b>0.231402</b>	<b>0.000975</b>	<b>0.031225</b>
0.50	0.523840	0.274408	0.001394	0.037336	0.558943	0.312417	0.000979	0.031289

From Table 5.3. the Generalised Regression Neural Network gave the optimal results when the spread is 0.4.

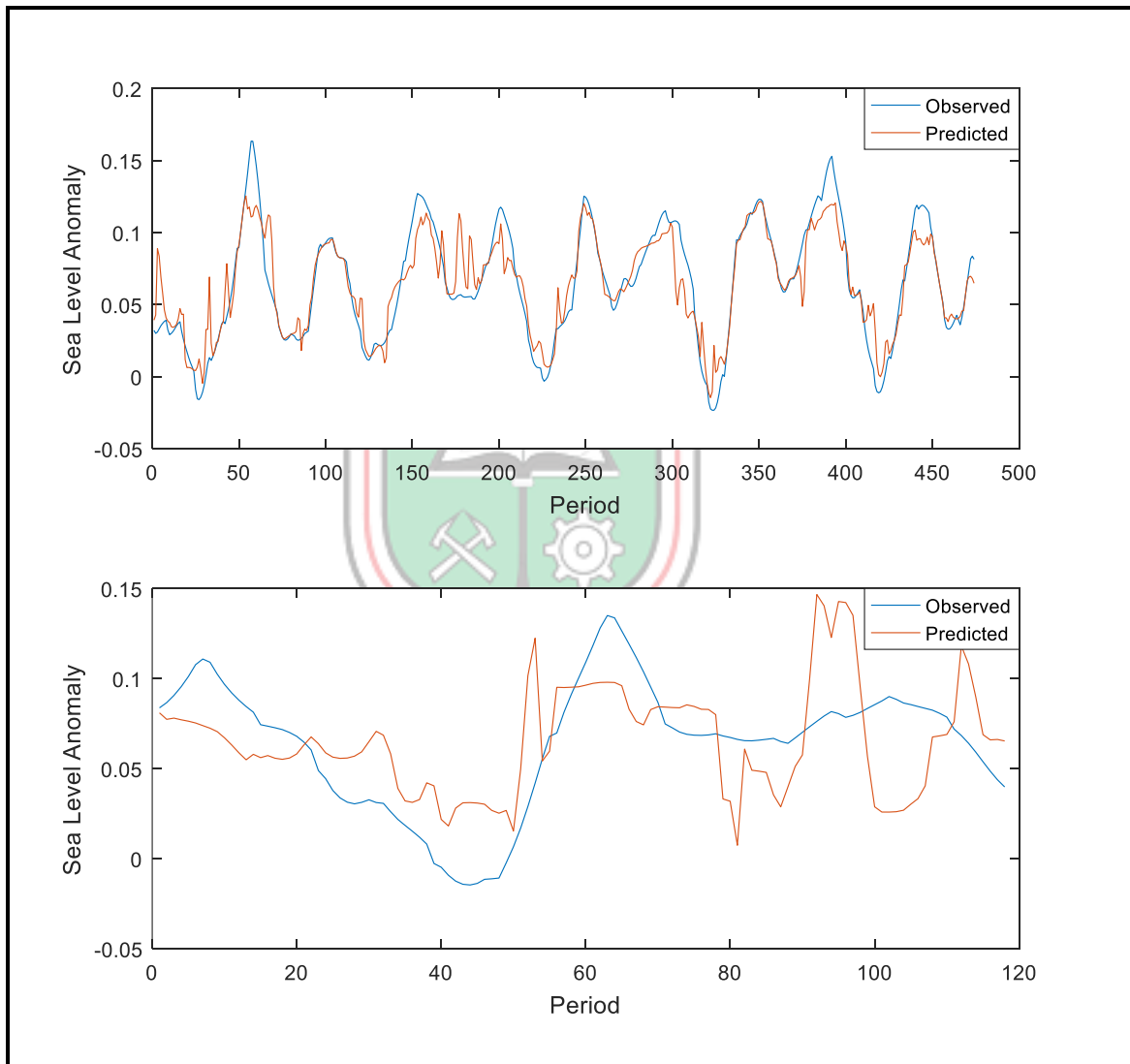


Figure 5.5 Training and Test Performance of GRNN Model

## 5.4 Discussion

The SLA data could have come in a satellite image format, colour plot, an ASCII file format or in a table of values. In this study the data came in a table of values format which was a little bit easy to deal with. Nevertheless, the data was filtered of noise such as latitude, longitude and time values by simply deleting before the averages of every row was calculated. The study considered four datasets including wind flux, heat, humidity and SLA data of the study area. The data for these four parameters was downloaded in a gridded format spanning from the years 2011 to 2017 and that was what was used to make the short-term prediction of SLA in the Gulf of Guinea. For the various datasets, the higher the wind flux, the higher ocean water precipitates thereby causing lower sea levels and vice versa. Also, higher heat energy suggests higher temperatures which brings about the melting of glaciers and ice sheets thereby causing sea levels to rise. In terms of humidity, there is an increase in sea levels when the water in the air increases. However, there are other parameters that also influence sea level anomaly such as rainfall, sea surface temperature (SST), Ocean temperature, Ocean currents, short wave radiation, long wave radiation and clouds just to mention a few that could have also been considered but for this study only the above three mentioned parameters were considered.

In this study SLA in the Gulf of Guinea was predicted using two artificial neural network methods thus the BPNN and GRNN and the best model was recommended for predictions of SLA in the study area. As observed in table 5.2, the results of the training with regards to two hidden neurons in the hidden layer for BR, thus an architecture of 3-2-1 have the minimum values in terms of error (MSE=0.000732, RMSE=0.027055) as compared to LM and ScG. An MSE of 0.001261, RMSE of 0.035511, R of 0.574166 and  $R^2$  of 0.329667 was incurred during the training stage and MSE of 0.000975, RMSE of 0.031225, R of 0.481043 and  $R^2$  of 0.231402 during the testing stage of the GRNN as shown in Table 5.3.

Figures 5.2, 5.3, 5.4 and 5.5 show the predictions between the neural network models and the measured SLA data of the Gulf of Guinea. Although all the selected optimal

architectures performed well, comparing the performance and the error measures arising from the prediction from the test data in Table 5.2 and 5.3 indicates that the GRNN architecture with spread 0.4 produced the least error values (MSE= 0.000975, RMSE= 0.031225).





## CHAPTER 6

### CONCLUSIONS AND RECOMMENDATIONS

#### 6.1 Conclusions

This research was conducted to collate the sea level anomaly of the study area using Satellite Altimetry and also to predict sea level anomaly using computational intelligent algorithms. SLA of the study area was collated successfully from the Asia-Pacific Data-Research Center. The data which came in a gridded format was averaged horizontally to be able to form a single column.

Predictions of SLA was done using the various neural network models as shown in Figures 5.2, 5.3, 5.4 and 5.5. Among the models used, the BR of the BPNN gave better predictions than the GRNN because of the larger values of R and  $R^2$  (0.75156 and 0.564842) and lower values of MSE and RMSE (0.000732 and 0.027055) respectively.

#### 6.2 Recommendations

The following recommendations are drawn from the study:

- i. Satellite altimetry and ANN in SLA prediction is a whole new dimension of research in Africa which is yet to be explored. It is recommended that ANN methods such as Radial Basis Function (RBF), support vector machine and extreme learning machine, Neural Network Autoregressive Moving Average (NNARMA), Adaptive Neuro-Fuzzy Interference System (ANFIS), Particle Swarm Optimisation (PSO), And hybrid models such ANFIS-BPNN and ANFIS-PSO be developed to make predictions of SLA in the Gulf of Guinea.
- ii. Other parameters such as sea surface temperature (SST), Ocean currents, Ocean temperature, clouds, short wave radiation, long wave radiation and rainfall data could also be used to make predictions of SLA in the Gulf of Guinea.

- iii. Further studies should also consider research about SLA along the coastal lines of the continent so that a combination of satellite altimetry and tide gauge measurements can be used.



## REFERENCES

- Abubakar, B. (2018), "Gulf of Guinea Resources, Economy and Development: Yesterday, Today and Tomorrow", pp. 1-11.
- Alomair, O. A. and Garrouch, A. A. (2015), "A General Regression Neural Network Model Offers Reliable Prediction of CO<sub>2</sub> Minimum Miscibility Pressure", *Springer*, pp. 351-365.
- Anon., (2019), "Asia-Pacific-Data-Research-Center", <http://apdrc.soest.hawaii.edu/las/v6/constrain?var=13333>, Accessed: March 4, 2019.
- Anon., (2019), "Sea Level Anomaly", [www.tsgc.utexas.edu](http://www.tsgc.utexas.edu). Accessed: April 11, 2019.
- Anon., (2019), "Altimeter", <https://en.m.wikipedia.org>, Accessed: April 10, 2019.
- Anon. (2008), "Feedforward Backpropagation Neural Network", [www.heatonresearch.com/online/...neural-networks-cs-edition](http://www.heatonresearch.com/online/...neural-networks-cs-edition), Accessed: February 29, 2019.
- Davis, D., Sutherland, M., Jaggan, S. and Singh, D. (2012), "Determining and Monitoring Sea Level in the Caribbean using Satellite Altimetry" Fig Working Week 2012, Rome, Italy, 6-10 May 2012.
- de Oliveira, R. S., (2007), "Oil and Politics in the Gulf of Guinea (Columbia/Hurst)
- Douglas, B., C., Kearney, M., S., and Leathermant, S., P. (2000), "Sea Level Rise History and Consequences", International Geophysics Series, Vol. 75.
- Gharineiat, Z. and Deng, X. (2015), "Application of the Multi-Adaptive Regression Splines to Integrate Sea Level Data from Altimetry and Tide Gauges for Monitoring Extreme Sea Level Events", *Marine Geodesy*, pp. 261-276.
- Graham, M. (2011), "GPS Versus Barometric Altitude: The Definitive Answer", *CrossCountry Magazine*, [www.xcmag.com](http://www.xcmag.com), Accessed: March 27, 2019.
- Hair, J., Black, W., Babin, B. and Anderson, R. (2013), "Multivariate Data Analysis", 7<sup>th</sup> edition, Person New International Edition, pp. 89-231.
- Hornik, K., Stinchcombe, M. and White, H. (1989), "Multilayer Feedforward Networks are Universal Approximators", <http://www.deeplearning.cs.cmu.edu>, Accessed: February 12, 2019.

- Kang, S., K., Cherniawsky, J., Y. and Foreman, M., G., G. (2006),” Patterns of Recent Sea Level Rise in the East/Japan Sea from Satellite Altimetry and in Situ Data”, *Journal of Geophysical Research*, Vol. 110.
- Karamouz, M., Kia, M. and Nazif, S. (2014),” Prediction of Sea Level Using a Hybrid Data-Driven Model: New Challenges After Hurricane Sandy”, Springer Science and Business Media Dordrecht 2014, pp. 63-71.
- Kurniawan, A., Ooi, S., K. and Babovic, V. (2014),” Improved Sea Level Anomaly Prediction Through Genetic Programming in Singapore Regional Waters”, CUNY Academic Works.
- Marijan, G., Senka, J., Nerem, R., S. and Tomislav, B. (2017),” The assessment of an Impact of Mean Sea Level Change in the Mid-Adriatic Region Based on Satellite Altimetry Records”, 17<sup>th</sup> International Multidisciplinary Scientific GeoConference SGEM 2017.
- Miles, E., R., Spillman, C., M., Church, J., A. and McIntosh, P., C. (2014),” Seasonal Prediction of Global Sea Level Anomalies using an Ocean-Atmosphere Dynamic Model”, Springer.
- Mlybari, E., A., Elbisy, M., S., Alshahri, A., H. and Albarakati, O., M. (2014),” The Use Support Vector Machine and Back Propagation Neural Network for Prediction of Daily Tidal Levels along the Jeddah Coast, Saudi Arabia”, *International Journal of Mathematical Computational, Physical and Quantum Engineering*, Vol. 8 , No.1, 2014.
- Niedzielski, T. and Kosek, W. (2008),” Forecasting sea level anomalies from TOPEX/Poseidon and Jason-1 satellite altimetry”, Springer-Verlag 2008, pp. 469–476.
- Nils-Axel, M. (2017), “Sea Level Manipulation” *International Journal of Engineering Science Invention (IJESI)*, pp. 48-51.
- Nishida, W., Matsuda, Y. and Noguchi, M. (2006), “Study on Prediction of Sea Level Anomaly Using Genetic Programming”, *Proceedings of Hydraulic Engineering* 50, pp.1429-1434.
- Ondimu, S. and Murase, H. (2006),” Reservoir Level Forecasting using Neural Networks: Lake Naivasha”, *Biosystems Engineering* 2007, pp. 135-138.

- Quilfen, Y., Bentamy, A., Delecluse, P., Katsaros, K. and Grima, N. (2000),” Prediction of Sea Level Anomalies Using Ocean Circulation Model Forced by Scatterometer Wind and Validation Using TOPEX/Poseidon Data”, IEEE transactions on Geoscience and Remote Sensing, Vol. 38. No. 4, July 2000.
- Ratheesh, S., Sharma, R., Prasad, K., V., S., R. and Basu S. (2015),” Impact of SARAL/Altika-Derived Sea Level Anomaly in a Data Assimilative Ocean Prediction System for the Indian Ocean” Marine Geodesy, pp. 354-364.
- Rosenblood, R., Gupta, A. and Webster, E. (2018),” Pollution, Fisheries and Food Insecurities in the Gulf of Guinea”, Symposium: Transnational Food Security, Transnational Legal Theory, Vol. 9, 2018, Forthcoming
- Rosmorduc, V., Benveniste, J., Bronner, E., Dinardo, S., Lauret, O., Maheu, C., Milagro, M. and Picot, N. (2011), “*Radar Altimetry Tutorial*”, Centre National d'etudes Spatiales, Benveniste, J. and Picot, N. (eds.), 306 pp.
- Sabatini, A. M. and Genovese, V. (2013), "A Stochastic Approach to Noise Modelling for Barometric Altimeters", *Sensors 2013*, pp. 15692-15707.
- ShinyPradeepa, A. and Vaithilingam, C. (2017),” Voltage Profile Assessment in Power Distribution System Using Generalized Regression Neural Network”, Elsevier, pp. 209-215.
- Singh, D., Davis, D., Sutherland, M. and Jaggan S. (2013),” Using Satellite Altimetry to Monitor and Determine Sea Level”
- Sterlini, P., Le Bars, D., de Vries, H and Ridder, N. (2017),” Understanding the Spatial Variation of Sea Level Rise in the North Sea Using Satellite Altimetry”, *Journal of Geophysical Research: Oceans*, Vol. 122, No. 8, pp. 6498-6511.
- Strassburg, M., W., Hamlington, B., D., Leben, R., R. and Kim, K., Y. (2014),” A comparative study of sea level reconstruction techniques using 20 years of satellite altimetry data”, *Journal of Geophysical Research: Oceans*, Vol. 119, No. 7, pp. 4068-4082.
- Ukeje, C. and Ela W., M. (2013), *African Approaches to Maritime Security*, Friedrich-Ebert-Stiftung, Abuja, Nigeria, 48pp.

## APPENDICES

### APPENDIX A RESULTS FOR THE VARIOUS MODELS DEVELOPED

**Table A1 Results of the Bayesian Regularisation (BR) Model**

noh n	Architec ture	Training				Testing			
		R	R <sup>2</sup>	MSE	RMSE	R	R <sup>2</sup>	MSE	RMSE
1	3-1-1	0.114651	0.013144	0.001610	0.040125	0.544805	0.296812	0.001158	0.034029
<b>2</b>	<b>3-2-1</b>	<b>0.382729</b>	<b>0.146481</b>	<b>0.001390</b>	<b>0.037283</b>	<b>0.75156</b>	<b>0.564842</b>	<b>0.000732</b>	<b>0.027055</b>
3	3-3-1	0.114984	0.013221	0.001609	0.040112	0.543772	0.295688	0.001156	0.034
4	3-4-1	0.517656	0.267967	0.001192	0.034525	0.641231	0.411177	0.001628	0.040348
5	3-5-1	0.734016	0.538779	0.000751	0.027404	0.304548	0.092749	0.002245	0.047381
6	3-6-1	0.632615	0.400201	0.000977	0.031257	0.367592	0.135124	0.002333	0.048301
7	3-7-1	0.757962	0.574506	0.000693	0.026325	0.116426	0.013555	0.003054	0.055263
8	3-8-1	0.814072	0.662713	0.000549	0.023431	0.012631	0.00016	0.015635	0.12504
9	3-9-1	0.805867	0.649421	0.000571	0.023896	0.130484	0.017026	0.012383	0.111279
10	3-10-1	0.810977	0.657683	0.000557	0.023601	0.223961	0.050159	0.00812	0.090111
11	3-11-1	0.834201	0.695891	0.000495	0.022249	0.446058	0.198968	0.014596	0.120814
12	3-12-1	0.836416	0.699591	0.000489	0.022113	0.390177	0.152238	0.02855	0.168967
13	3-13-1	0.847311	0.717935	0.000459	0.021424	0.086604	0.0075	0.024562	0.156723
14	3-14-1	0.862488	0.743885	0.000417	0.020421	0.281442	0.07921	0.009851	0.099252
15	3-15-1	0.868158	0.753698	0.000401	0.020025	0.03035	0.000921	0.013955	0.118131
16	3-16-1	0.882604	0.778989	0.000360	0.018974	0.237412	0.056364	0.0078	0.088318
17	3-17-1	0.888349	0.789163	0.000343	0.01852	0.297834	0.088705	0.003863	0.062153
18	3-18-1	0.884517	0.782370	0.000354	0.018815	0.156118	0.024373	0.087806	0.296321
19	3-19-1	0.889961	0.792030	0.000339	0.018412	0.20684	0.042783	0.005044	0.071021
20	3-20-1	0.900614	0.811105	0.000308	0.01755	-0.13259	0.01758	0.010089	0.100444

Table A2 Results of the Lavenberg Marquardt (LM) Model

noh n	Archit ecture	Training				Testing			
		R	R <sup>2</sup>	MSE	RMSE	R	R <sup>2</sup>	MSE	RMSE
1	3-1-1	0.298067	0.088844	0.001483	0.03851	0.063840	0.004076	0.001415	0.037616
2	3-2-1	0.415051	0.172267	0.001347	0.036701	0.462525	0.213929	0.00147	0.038341
3	3-3-1	0.503539	0.253552	0.001215	0.034857	0.394772	0.155845	0.001577	0.039711
4	<b>3-4-1</b>	<b>0.538877</b>	<b>0.290388</b>	<b>0.001155</b>	<b>0.033985</b>	<b>0.550859</b>	<b>0.303446</b>	<b>0.001226</b>	<b>0.035014</b>
5	3-5-1	0.587453	0.345101	0.001066	0.03265	0.308507	0.095177	0.01645	0.128258
6	3-6-1	0.639594	0.40908	0.000962	0.031016	-0.197934	0.039178	197.31143	14.04676
7	3-7-1	0.671087	0.450358	0.000895	0.029917	0.293467	0.086123	0.003558	0.059649
8	3-8-1	0.725199	0.525914	0.000772	0.027785	0.201812	0.040728	0.002719	0.052144
9	3-9-1	0.805964	0.649578	0.00057	0.023875	0.114021	0.013001	0.207	0.454973
10	3-10-1	0.832595	0.693214	0.000499	0.022338	0.062224	0.003872	0.015536	0.124643
11	3-11-1	0.766629	0.58772	0.000671	0.025904	0.389295	0.151551	0.002046	0.045233
12	3-12-1	0.852876	0.727397	0.000444	0.021071	-0.082327	0.006778	0.301104	0.548729
13	3-13-1	0.876777	0.768738	0.000376	0.019391	0.083899	0.007039	0.008653	0.093022
14	3-14-1	0.845536	0.714931	0.000464	0.021541	0.373935	0.139827	0.004168	0.06456
15	3-15-1	0.861112	0.741514	0.000421	0.020518	-0.201400	0.040562	0.031023	0.176133
16	3-16-1	0.886365	0.785643	0.000349	0.018682	0.014377	0.000207	0.115943	0.340504
17	3-17-1	0.885844	0.78472	0.00035	0.018708	0.102697	0.010547	0.005688	0.075419
18	3-18-1	0.883657	0.78085	0.000357	0.018894	-0.131092	0.017185	0.021566	0.146854
19	3-19-1	0.89786	0.806153	0.000316	0.017776	0.266629	0.071091	0.034632	0.186097
20	3-20-1	0.892983	0.797419	0.00033	0.018166	-0.201785	0.040717	0.058764	0.242413

Table A3 Results of the Scaled Conjugate Gradient (ScG) Model

noh n	Architec ture	Training				Testing			
		R	R <sup>2</sup>	MSE	RMSE	R	R <sup>2</sup>	MSE	RMSE
1	3-1-1	0.298159	0.088899	0.001483	0.03851	0.065041	0.00423	0.001418	0.037656
2	3-2-1	0.415051	0.172267	0.001347	0.036701	0.462086	0.213523	0.00147	0.038341
3	3-3-1	0.503363	0.253374	0.001215	0.034857	0.396501	0.157213	0.001582	0.039774
<b>4</b>	<b>3-4-1</b>	<b>0.51198</b>	<b>0.262124</b>	<b>0.001201</b>	<b>0.034655</b>	<b>0.558523</b>	<b>0.311948</b>	<b>0.001722</b>	<b>0.041497</b>
5	3-5-1	0.578129	0.334233	0.001084	0.032924	0.307987	0.094856	0.014886	0.122008
6	3-6-1	0.561323	0.315084	0.001115	0.033392	0.071406	0.005099	0.011023	0.10499
7	3-7-1	0.620762	0.385345	0.001	0.031623	0.448093	0.200787	0.001834	0.042825
8	3-8-1	0.787792	0.620616	0.000617	0.024839	0.358458	0.128492	0.004398	0.066317
9	3-9-1	0.803402	0.645455	0.000577	0.024021	0.247142	0.061079	0.007927	0.089034
10	3-10-1	0.839293	0.704413	0.000481	0.021932	0.410806	0.168762	0.007506	0.086637
11	3-11-1	0.791856	0.627036	0.000607	0.024637	0.116272	0.013519	0.030949	0.175923
12	3-12-1	0.84667	0.71685	0.000461	0.021471	0.187605	0.035196	0.370646	0.608807
13	3-13-1	0.865021	0.748261	0.00041	0.020248	0.245268	0.060156	0.024387	0.156163
14	3-14-1	0.859288	0.738376	0.000426	0.02064	0.079402	0.006305	0.00688	0.082946
15	3-15-1	0.865962	0.74989	0.000407	0.020174	0.05512	0.003038	0.053402	0.231089
16	3-16-1	0.86423	0.746893	0.000412	0.020298	0.022393	0.000501	0.029528	0.171837
17	3-17-1	0.895578	0.80206	0.000322	0.017944	0.218665	0.047814	0.022677	0.150589
18	3-18-1	0.886855	0.786512	0.000347	0.018628	0.011988	0.000144	0.016043	0.126661
19	3-19-1	0.888565	0.789548	0.000343	0.01852	0.253024	0.064021	0.012402	0.111364
20	3-20-1	0.901037	0.811868	0.000306	0.017493	0.146089	0.021342	0.020533	0.143293



Table A4 Results of the GRNN Model

Spread	Training				Testing			
	R	R <sup>2</sup>	MSE	RMSE	R	R <sup>2</sup>	MSE	RMSE
0.10	0.919746	0.845933	0.000270	0.016432	0.492026	0.24209	0.001096	0.033106
0.20	0.763171	0.58243	0.000754	0.027459	0.412857	0.170451	0.001123	0.033511
0.30	0.644927	0.415931	0.001066	0.03265	0.412854	0.170448	0.001052	0.032435
<b>0.40</b>	<b>0.574166</b>	<b>0.329667</b>	<b>0.001261</b>	<b>0.035511</b>	<b>0.481043</b>	<b>0.231402</b>	<b>0.000975</b>	<b>0.031225</b>
0.50	0.523840	0.274408	0.001394	0.037336	0.558943	0.312417	0.000979	0.031289
0.60	0.478402	0.228868	0.001481	0.038484	0.592048	0.350521	0.001028	0.032062
0.70	0.433783	0.188168	0.001535	0.039179	0.587517	0.345176	0.001080	0.032863
0.80	0.391586	0.15334	0.001568	0.039598	0.570636	0.325625	0.001121	0.033481
0.90	0.353739	0.125131	0.001587	0.039837	0.553148	0.305973	0.001152	0.033941
1.00	0.320894	0.102973	0.001599	0.039987	0.538716	0.290215	0.001174	0.034264
1.10	0.292854	0.085763	0.001607	0.040087	0.528031	0.278817	0.001190	0.034496
1.20	0.269112	0.072421	0.001612	0.04015	0.520758	0.271189	0.001202	0.03467
1.30	0.249088	0.062045	0.001615	0.040187	0.516257	0.266521	0.001210	0.034785
1.40	0.232221	0.053927	0.001618	0.040224	0.513870	0.264062	0.001217	0.034886
1.50	0.217997	0.047523	0.001619	0.040237	0.513026	0.263196	0.001221	0.034943
1.60	0.205970	0.042424	0.001621	0.040262	0.513269	0.263445	0.001225	0.035
1.70	0.195760	0.038322	0.001622	0.040274	0.514252	0.264455	0.001228	0.035043
1.80	0.187051	0.034988	0.001623	0.040286	0.515718	0.265965	0.001231	0.035086
1.90	0.179585	0.032251	0.001623	0.040286	0.517481	0.267787	0.001233	0.035114
2.00	0.173149	0.029981	0.001624	0.040299	0.519408	0.269785	0.001235	0.035143

## APPENDIX B CODES

### BR CODES

```
clc
```

```
format long
```

```
%% Reading the training, testing and validation data from excel
```

```
train_Data = xlsread('training_Data.xlsx');
```

```
test_Data = xlsread('testing_Data.xlsx');
```

```
% val_Data = xlsread('validation_Data.xlsx');
```

```
% whole_Data = xlsread('whole_Data.xlsx');
```

```
%% Extracting The Training Input and Target Values
```

```
train_Inputs = train_Data(1:3,:);
```

```
train_Target = train_Data(4,:);
```

```
%% Extracting the Testing Input and Target Values
```

```
test_Inputs = test_Data(1:3,:);
```

```
test_Target = test_Data(4,:);
```

```
%% Extracting the Validation Input and Target Values
```

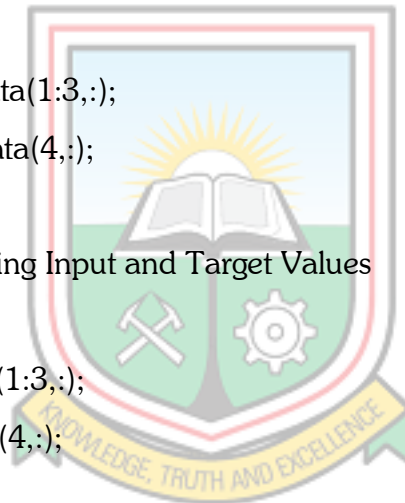
```
% val_Inputs = val_Data(1:5,:);
```

```
% val_Target = val_Data(6,:);
```

```
%% Extracting the Whole_Data Input and Target Values
```

```
% whole_Inputs = whole_Data(1:4,:);
```

```
% whole_Target = whole_Data(5,:);
```



```
%% Normalising The Training, Target and Validation Inputs
```

```
[train_X,ps]= mapminmax(train_Inputs); % Normalising the training inputs into the  
range of -1 and +1
```

```
[test_X,pt]= mapminmax(test_Inputs); % Normalising the testing inputs into the  
range of -1 and +1
```

```
% [val_X,pv] = mapminmax(val_Inputs); % Normalising the validation inputs into  
the range of -1 and +1
```

```
%% Setting the initial seed value
```

```
setdemorandstream(491218382);
```

```
disp('nohn Train_R Train_Perf Test_R Test_Perf')
```

```
%% Creating the backpropagation neural network
```

```
%%[] is where you input your number of hidden neurons, tansig is the  
%%activation function for the hidden layer, purelin is the activation  
%%function for the output layer, trainbr is the training algorithm  
for nohn = 1:20;
```

```
net = newff(train_X,train_Target,[nohn],{'tansig' 'purelin'},'trainbr');
```

```
%% set early stopping parameters
```

```
net.divideParam.trainRatio = 1.0; % training set [%]
```

```
net.divideParam.valRatio = 0.0; % validation set [%]
```

```
net.divideParam.testRatio = 0.0; % test set [%]
```

```
%% Training the backpropagation neural network
```

```
% Setting the training parameters
```

```
% net.trainParam.max_fail = 6;
```

```

net.trainParam.min_grad = 1e-07;
% net.trainParam.lr = 0.03;
% net.trainParam.mc = 0.7;
net.trainParam.epochs = 8000;
net.trainParam.goal = 0;
net.performFcn = 'mse'; % Mean Squared Error (mse)

```

```

% Training the backpropagation neural network

```

```

net = train(net,train_X,train_Target);

```

```

%% Obtaining the training, testing and validation outputs

```

```

train_Out = net(train_X);

```

```

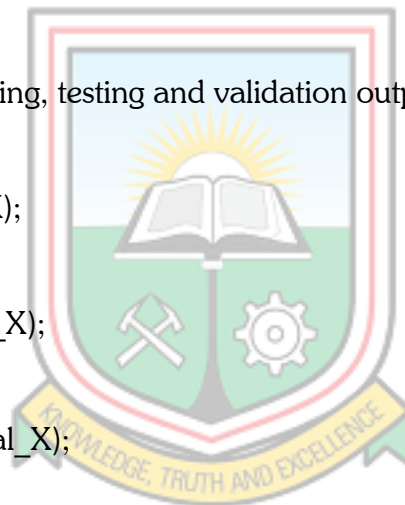
test_Out = sim(net,test_X);

```

```

% val_Out = sim(net,val_X);

```



```

%% Evaluation of the Performance of the trained back propagation neural network

```

```

% Calculating the performance of the trained back propagation neural

```

```

% network

```

```

train_e = gsubtract(train_Target,train_Out);

```

```

train_Performance = perform(net,train_Target,train_Out);

```

```

test_e = gsubtract(test_Target,test_Out);

```

```

test_Performance = perform(net,test_Target,test_Out);

```

```

% val_e = gsubtract(val_Target,val_Out);

```

```

% val_Performance = perform(net,val_Target,val_Out)

%% Calculation of the regression of the training, testing and validation Output

[r_train,m_train,b_train]= regression(train_Target,train_Out);

[r_test,m_test,b_test]= regression(test_Target,test_Out);

% [r_val,m_val,b_val]= regression(val_Target,val_Out)

%figure, plotregression(train_Target,train_Out,'Training')
%figure, plotregression(test_Target,test_Out, 'Testing')
%figure, plotregression(val_Target,val_Out, 'Validation')

fprintf('%d          %f          %f          %f\n',nohn,r_train,train_Performance,r_test,test_Performance)
end

```



## LM CODES

```

clc
format long
%% Reading the training, testing and validation data from excel

train_Data = xlsread('training_Data.xlsx');
test_Data = xlsread('testing_Data.xlsx');
% val_Data = xlsread('validation_Data.xlsx');
% whole_Data = xlsread('whole_Data.xlsx');

%% Extracting The Training Input and Target Values

train_Inputs = train_Data(1:3,:);
train_Target = train_Data(4,:);

```

```
%% Extracting the Testing Input and Target Values
```

```
test_Inputs = test_Data(1:3,:);
```

```
test_Target = test_Data(4,:);
```

```
%% Extracting the Validation Input and Target Values
```

```
% val_Inputs = val_Data(1:5,:);
```

```
% val_Target = val_Data(6,:);
```

```
%% Extracting the Whole_Data Input and Target Values
```

```
% whole_Inputs = whole_Data(1:4,:);
```

```
% whole_Target = whole_Data(5,:);
```

```
%% Normalising The Training, Target and Validation Inputs
```

```
[train_X,ps]= mapminmax(train_Inputs); % Normalising the training inputs into the  
range of -1 and +1
```

```
[test_X,pt]= mapminmax(test_Inputs); % Normalising the testing inputs into the  
range of -1 and +1
```

```
% [val_X,pv] = mapminmax(val_Inputs); % Normalising the validation inputs into  
the range of -1 and +1
```

```
%% Setting the initial seed value
```

```
setdemorandstream(491218382);
```

```

disp('nohn  Train_R  Train_Perf  Test R  Test_Perf')
%% Creating the backpropagation neural network
%%[]is where you input your number of hidden neurons, tansig is the
%%activation function for the hidden layer, purelin is the activation
%%function for the output layer, trainbr is the training algorithm
for nohn = 1:20;

```

```

net = newff(train_X,train_Target,[nohn],{'tansig' 'purelin'},'trainlm');

```

```

%% set early stopping parameters

```

```

net.divideParam.trainRatio = 1.0; % training set [%]

```

```

net.divideParam.valRatio = 0.0; % validation set [%]

```

```

net.divideParam.testRatio = 0.0; % test set [%]

```

```

%% Training the backpropagation neural network

```

```

% Setting the training parameters

```

```

net.trainParam.max_fail = 6;

```

```

net.trainParam.min_grad = 1e-07;

```

```

net.trainParam.lr = 0.03;

```

```

net.trainParam.mc = 0.7;

```

```

net.trainParam.epochs = 8000;

```

```

net.trainParam.goal = 0;

```

```

net.performFcn = 'mse'; % Mean Squared Error (mse)

```

```

% Training the backpropagation neural network

```

```

net = train(net,train_X,train_Target);

```

```

%% Obtaining the training, testing and validation outputs

```

```

train_Out = net(train_X);

test_Out = sim(net,test_X);

% val_Out = sim(net,val_X);

%% Evaluation of the Performance of the trained back propagation neural network

% Calculating the performance of the trained back propagation neural
% network
train_e = gsubtract(train_Target,train_Out);
train_Performance = perform(net,train_Target,train_Out);

test_e = gsubtract(test_Target,test_Out);
test_Performance = perform(net,test_Target,test_Out);

% val_e = gsubtract(val_Target,val_Out);
% val_Performance = perform(net,val_Target,val_Out)

%% Calculation of the regression of the training, testing and validation Output

[r_train,m_train,b_train]= regression(train_Target,train_Out);

[r_test,m_test,b_test]= regression(test_Target,test_Out);

% [r_val,m_val,b_val]= regression(val_Target,val_Out)

%figure, plotregression(train_Target,train_Out,'Training')
%figure, plotregression(test_Target,test_Out, 'Testing')
%figure, plotregression(val_Target,val_Out, 'Validation')

```



```

fprintf('%d          %f          %f          %f\n',nohn,r_train,train_Performance,r_test,test_Performance)
end

```

## ScG CODES

```
clc
```

```
format long
```

```
%% Reading the training, testing and validation data from excel
```

```
train_Data = xlsread('training_Data.xlsx');
```

```
test_Data = xlsread('testing_Data.xlsx');
```

```
% val_Data = xlsread('validation_Data.xlsx');
```

```
% whole_Data = xlsread('whole_Data.xlsx');
```

```
%% Extracting The Training Input and Target Values
```

```
train_Inputs = train_Data(1:3,:);
```

```
train_Target = train_Data(4,:);
```

```
%% Extracting the Testing Input and Target Values
```

```
test_Inputs = test_Data(1:3,:);
```

```
test_Target = test_Data(4,:);
```

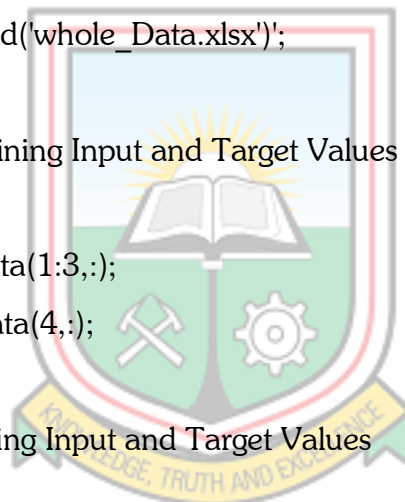
```
%% Extracting the Validation Input and Target Values
```

```
% val_Inputs = val_Data(1:5,:);
```

```
% val_Target = val_Data(6,:);
```

```
%% Extracting the Whole_Data Input and Target Values
```

```
% whole_Inputs = whole_Data(1:4,:);
```



```
% whole_Target = whole_Data(5,:);
```

```
%% Normalising The Training, Target and Validation Inputs
```

```
[train_X,ps]= mapminmax(train_Inputs); % Normalising the training inputs into the  
range of -1 and +1
```

```
[test_X,pt]= mapminmax(test_Inputs); % Normalising the testing inputs into the  
range of -1 and +1
```

```
% [val_X,pv] = mapminmax(val_Inputs); % Normalising the validation inputs into  
the range of -1 and +1
```

```
%% Setting the initial seed value
```

```
setdemorandstream(491218382);
```

```
disp('nohn Train_R Train_Perf Test R Test_Perf')
```

```
%% Creating the backpropagation neural network
```

```
%%[] is where you input your number of hidden neurons, tansig is the
```

```
%%activation function for the hidden layer, purelin is the activation
```

```
%%function for the output layer, trainbr is the training algorithm
```

```
for nohn = 1:20;
```

```
net = newff(train_X,train_Target,[nohn],{'tansig' 'purelin'},'trainscg');
```

```
%% set early stopping parameters
```

```
net.divideParam.trainRatio = 1.0; % training set [%]
```

```
net.divideParam.valRatio = 0.0; % validation set [%]
```

```
net.divideParam.testRatio = 0.0; % test set [%]
```

```
%% Training the backpropagation neural network
```

```
% Setting the training parameters
```

```
net.trainParam.max_fail = 6;  
net.trainParam.min_grad = 1e-07;  
net.trainParam.lr = 0.03;  
net.trainParam.mc = 0.7;  
net.trainParam.epochs = 8000;  
net.trainParam.goal = 0;  
net.performFcn = 'mse'; % Mean Squared Error (mse)
```

```
% Training the backpropagation neural network
```

```
net = train(net,train_X,train_Target);
```

```
%% Obtaining the training, testing and validation outputs
```

```
train_Out = net(train_X);
```

```
test_Out = sim(net,test_X);
```

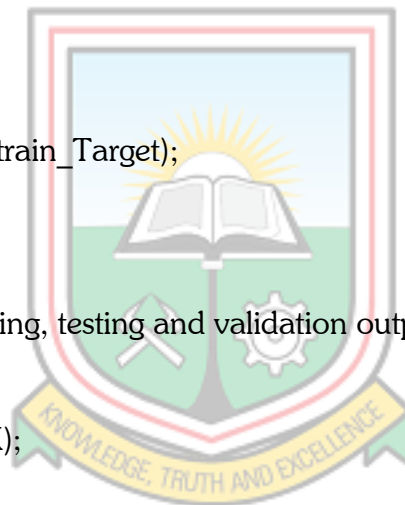
```
% val_Out = sim(net,val_X);
```

```
%% Evaluation of the Performance of the trained back propagation neural network
```

```
% Calculating the performance of the trained back propagation neural  
% network
```

```
train_e = gsubtract(train_Target,train_Out);
```

```
train_Performance = perform(net,train_Target,train_Out);
```



```

test_e = gsubtract(test_Target,test_Out);
test_Performance = perform(net,test_Target,test_Out);

% val_e = gsubtract(val_Target,val_Out);
% val_Performance = perform(net,val_Target,val_Out)

%% Calculation of the regression of the training, testing and validation Output

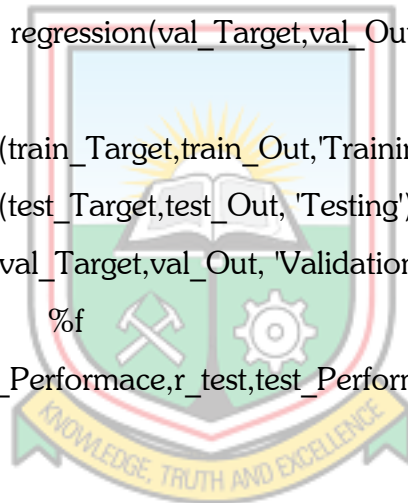
[r_train,m_train,b_train]= regression(train_Target,train_Out);

[r_test,m_test,b_test]= regression(test_Target,test_Out);

% [r_val,m_val,b_val]= regression(val_Target,val_Out)

% figure, plotregression(train_Target,train_Out,'Training');
% figure, plotregression(test_Target,test_Out, 'Testing');
%figure, plotregression(val_Target,val_Out, 'Validation')
fprintf('%d          %f          %f          %f\n',r_train,train_Performance,r_test,test_Performance)
end

```



## GRNN CODES

```

clc
format long
%% Reading the training, testing and validation data from excel

train_Data = xlsread('training_Data.xlsx');
test_Data = xlsread('testing_Data.xlsx');
% val_Data = xlsread('validation_Data.xlsx');
% whole_Data = xlsread('whole_Data.xlsx');

%% Extracting The Training Input and Target Values

```

```
train_Inputs = train_Data(1:3,:);  
train_Target = train_Data(4,:);
```

```
%% Extracting the Testing Input and Target Values
```

```
test_Inputs = test_Data(1:3,:);  
test_Target = test_Data(4,:);
```

```
%% Extracting the Validation Input and Target Values
```

```
% val_Inputs = val_Data(1:5,:);  
% val_Target = val_Data(6,:);
```

```
%% Normalising The Training, Target and Validation Inputs
```

```
[train_X,ps]= mapminmax(train_Inputs); % Normalising the training inputs into the  
range of -1 and +1
```

```
[test_X,pt]= mapminmax(test_Inputs); % Normalising the testing inputs into the  
range of -1 and +1
```

```
% [val_X,pv] = mapminmax(val_Inputs); % Normalising the validation inputs into  
the range of -1 and +1
```

```
%% Setting the initial seed value
```

```
setdemorandstream(491218382);
```

```
%% Creating the Generalised Regression Neural Network
```

```
% choose a spread constant
```

```

disp('Spread  Train_R  Train_Perf  Test R  Test_Perf')
for spread = 0.1:0.1:2;

% create a neural network
net = newgrnn(train_X,train_Target,spread);
%-----

%% Obtaining the training, testing and validation output

train_Out = net(train_X);

test_Out = sim(net,test_X);

% val_Out = sim(net,val_X);

%% Evaluation of the Performance of the trained back propagation neural network

% The performance function
net.performFcn = 'mse';
% Calculating the performance of the trained back propagation neural
% network

train_Performance = perform(net,train_Target,train_Out);

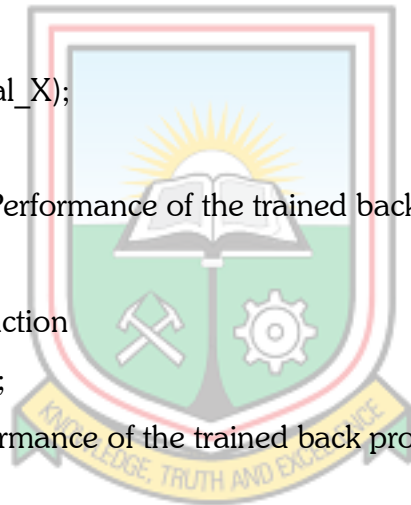
test_Performance = perform(net,test_Target,test_Out);

% val_Performance = perform(net,val_Target,val_Out)

%% Calculation of the regression of the training, testing and validation Output

[r_train,m_train,b_train]= regression(train_Target,train_Out);

```



```
[r_test,m_test,b_test]= regression(test_Target,test_Out);

fprintf('%.2f          %f          %f          %f\n',spread,r_train,train_Performance,r_test,test_Performance)
end
% [r_val,m_val,b_val]= regression(val_Target,val_Out)

%figure, plotregression(train_Target,train_Out,'Training')
% figure, plotregression(test_Target,test_Out, 'Testing')
%figure, plotregression(val_Target,val_Out, 'Validation')
```

