

A Programmable Tool-kit for a Smart Weather System Using Predictive Analytics at the Botswana International University of Science and Technology

by

Tsaone Swaabow Thapelo¹

Reg. No: 17100029

BSc in Computer Science (Central University of Las Villas)

PGDE in Mathematics (University of Botswana)

A Thesis Submitted to the Faculty of Engineering and Technology
in Partial Fulfilment of
the Requirements for the Award of the Degree of
Master of Engineering in Mechatronics and Industrial Instrumentation

Supervisor: **Dr. Molaletsa Namoshe²**

Co-Supervisor: **Dr. Oduetse Matsebe³**

Department of Mechanical, Energy and Industrial Engineering^{1,2,4}

Faculty of Engineering and Technology^{1,2,3}

Botswana International University of Science and Technology^{1,2,3}

October 8, 2020.

DECLARATION

I, Tsaone Swaabow Thapelo, hereby declare that, this thesis submitted to the Botswana International University of Science and Technology for the Degree of Masters in Mechatronics and Industrial Instrumentation titled "A Programmable Tool-kit for a Smart Weather System Using Predictive Analytics at the Botswana International University of Science and Technology" is entirely my own work; except where specific reference is highlighted. The contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University.

Tsaone Swaabow Thapelo:  : Date: October 8, 2020

COPYRIGHT ©

This thesis is copyright material protected under the Berne Convention, the Copyright and Neighbouring Rights Act, Act. No. 8 of 2000 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; critical scholarly review or discourse with an acknowledgement, without the written permission of the office of the Postgraduate School, on behalf of both the author and the Botswana International University of Science and Technology.

CERTIFICATION

The undersigned certifies that he/she has read and hereby recommends for acceptance to the Faculty of Engineering and Technology by thesis titled: *Predictive Analytics and the Internet of Things: A Programmable Tool-kit for Smart Weather*, in fulfilment of the requirements for the degree of Master of Engineering in Mechatronics and Industrial Instrumentation of the BIUST.

Supervisor: Dr. Molaletsa Namoshe _____

Date: October 8, 2020

Co-Supervisor: Dr. Oduetse Matsebe _____

Date: October 8, 2020

ACKNOWLEDGEMENTS

I thank God for the opportunity provided to realise this work. My profound gratitude are dedicated to my supervisor Dr. Molaletsa Namoshe, and my co-supervisor Dr. Oduetse Matsebe, for the opportunity and the guidance they provided to complete this work. I am immensely grateful to all the minds that contributed to this work directly or indirectly. To mention a few: Dr. Tshiamo Motshegwa from the University of Botswana and Dr. Mary-Jane M. Bopape from the South African Weather Services. I also thank Matthew Mayernik from the University Corporation for Atmospheric Research (UCAR) and two other anonymous reviewers of my contributed papers resulting from this work.

A mi Familia

Gracias a mi familia por el amor y la paciencia que me ofrecen.

DEDICATION

To my loving grand parents:

*Mrs **Semaswe Kekgethetswe** and Mr **Kekgethetswe Tumisi II.***

May their souls rest in peace.

Contents

Glossary	x
Acronyms	xi
1 Introduction, Problem Statement and Proposed Approach	1
1.1 Introduction	2
1.2 Problem Statement	4
1.2.1 Aim and Objectives	4
1.2.1.1 Specific Objectives	4
1.3 Research Approach and Methodology	5
1.4 Outline of the Thesis	7
2 Background Information and Related Literature Review	8
2.1 Background Information	8
2.2 Disruptive Technologies for Weather Services	9
2.2.1 The Frontend Service Layer	10
2.2.2 The Gateway Layer	10
2.2.3 The Backend Service Layer	11
2.3 Related Literature Review	13
2.3.1 Discussion of Related Literature	15
2.3.2 Trends in the Weather Industry	17
2.4 Deployment of the toolkit	18
2.4.1 Dashboards for Weather Services	18
3 Methodology, Hardware, Software and Methods	20
3.1 The Proposed Pipe-line of the System	21
3.2 Business Understanding	23
3.3 Data Collection and Storage	24

3.3.1	Hardware for Data Scrapping	24
3.3.2	Tools and Firmware for Data Scrapping	25
3.4	Data Understanding and Preparation	26
3.5	Data Management System	26
3.6	Modelling: Predictive Analytics	26
3.6.1	Monte Carlo Methods in Machine Learning	28
3.6.2	K-Nearest Neighbours (KNN)	29
3.6.3	Tree Based Models for Regression	29
3.6.4	Random Forest	30
3.7	Model Diagnostic Methods	30
3.7.1	Variable Importance Measures	30
3.7.2	Partial Dependence Plots	31
3.8	Deployment	31
4	Experimentations, Application Design and Results	32
4.1	Experimental Set-up	32
4.1.1	Sensor Data Scrapping	34
4.1.2	Web Scrapping	36
4.2	Weather Monitoring: Application Design	36
4.3	Temperature Time Series Forecasting using KNN	40
4.3.1	Results on Temperature Time Series Forecasting	41
4.4	Enhancing Access and Making Use of SASSCAL's Data	46
4.4.1	Study Area and Modelling Datasets	47
4.4.2	Data Collection, Data Understanding and Preparation	48
4.4.3	Construction of the Random Forest Model	51
4.4.4	Determining Influential Variables for Maximum Air Temperature	51
4.4.5	Determining Marginal Effects of Predictors on Air Temperature	54
4.4.6	Discussions on Data Access and Utilisation of SASSCAL Data	56
5	Conclusions, Recommendations and Future Work	57
5.1	Conclusions	57
5.2	Recommendations	59
5.3	Limitations and Future Work	59
6	Situational Analysis	60

List of Tables

2.1	Common NWP models and their respective programming tools.	9
2.2	Summary of hardware (sensors) for a low-cost automatic weather station	11
2.3	Summary of related works, their origins, year published, and the deployed IoT and predictive analytics tools and methods.	16
2.4	Tools for climate data analytics in southern Africa.	17
3.1	CRISP-DM methodology for data driven modelling.	23
4.1	Variables contained in the collected data set the BIUST AWS.	39
4.2	Predictive performance of MIMO vs Recursive multi-step ahead strategy.	41
4.3	Accuracy (in %) of the strategies for various configurations of MIMO (M_i) and Recursive (R_i) evaluated for the horizons: 18, 20, 22, 24, 48, 72, 96 and 120 hours.	41
4.4	Maximum, minimum and mean temperatures (in °C) recorded at Shakawe AWS.	48

List of Figures

1.1 The BIUST AWS installed by Sutron for weather monitoring.	3
2.1 The general flow of weather data from the frontend layer to the backend layer.	12
3.1 The general pipe line of the proposed methodology	21
3.2 Mind map summarising the tools and methods deployed in this work.	22
4.1 The ESP8266 module used for prototyping.	33
4.2 The proposed prototype to collect temperature and relative humidity.	34
4.3 The uPyCraft environment displaying the measured values.	35
4.4 Visualising the distribution of AWSs in Botswana using the developed App.	38
4.5 Visualising the BIUST AWS data using the developed dashboard App.	39
4.6 Scatter plots and Pearson's correlation coefficients illustrating the linear relationship between the actual values of maximum air temperature and the predicted maximum air temperature using the Recursive strategy.	42
4.7 Scatter plots and Pearson's correlation coefficients illustrating the linear relationship between the actual values of maximum air temperature and the predicted maximum air temperature using the MIMO strategy.	43
4.8 Graphs showing the multi step ahead forecasts using the MIMO (blue) and Recursive (red) strategies with some perfect fit by both strategies. The blue trend depicts the MIMO; the red one depicts the Recursive, and the green trend for the observed values.	44
4.9 The SASSCAL platform for dissemination of weather data	46
4.10 Location of the study area.	47
4.11 Ridgeline plots depicting the mean daily temperatures in °C at Shakawe AWS.	49
4.12 Daily average air temperature (ATAVG) distribution for the year 2015.	50
4.13 Variable importance for maximum air temperature for the Tubu AWS.	52
4.14 Variable importance for maximum air temperature for the Tubu AWS.	53
4.15 Partial dependency of maximum air temperature on average relative humidity.	54
4.16 Partial dependency of maximum air temperature on other top predictors.	55

Glossary

Analytics is the use of computational methods to discover patterns in data.

Data is a measure of historic information.

Data Mining is the extraction of implicit patterns and knowledge from data.

Data Munging is the manual process of cleansing data prior to data analytics.

Data Science entails all tools and techniques applicable to data to make sense out of it.

Data Scraping is the automated computer processing to extract data from data sources for efficacious interfacing by other computer programs.

Data Wrangling is the automated process of examining and transforming data from one data format into another format digestible by various computer programs.

Interpretable Machine Learning is the deployment of machine learning approaches in understandable manners to a lay man.

Machine Learning is the design of data driven algorithms for extraction of structural descriptions from data instances to generate new knowledge.

Predictive analytics is the art and science of discovering and reporting interesting and meaningful patterns in data using computational methods.

Smart Weather the use of disruptive technologies like the Internet of Things and data driven systems to enhance weather services.

Web Scraping is a data scraping technique to extract data from text-based mark-up applications like web HTML and XHTML.

Acronyms

AMQP Advanced Message Queuing Protocol

ANN Artificial Neural Network

App Application, software for smart devices

AWS Automatic Weather Station

Bagging Bootstrap Aggregating

BDMS Botswana Department of Meteorological Services

BITRI Botswana Institute for Science and Technology

BIUST Botswana International University of Science and Technology

CI Cyber-Infrastructure

CoAP Constrained Application Protocol

COSMO Consortium for Smallscale Modelling

CRISP-DM Cross-Industry Standard Process for Data Mining

ECMWF European Centre for Medium-Range Weather Forecasts

GCM General Circulation Model

GrADS Grid Analysis and Display System

HPC High Performance Computing

HTTP Hyper-Text Transport Protocol

ICON Icosahedral Non-hydrostatic

IDE Integrated Development Environment

iid Independent and Identically Distributed

IoT Internet of Things

KNN K-Nearest Neighbours

LLN Law of Large Numbers

MAE Mean Absolute Error

MIMO Multi-Input Multi-Output

ML Machine Learning

MLP Multiple Layer Perceptron

MOGREPS-UK The Met Office convective-scale ensemble

MOS Model Output Statistics

MQTT Message Queuing Telemetry Transport

NWP Numerical Weather Prediction

PDA Predictive Data Analytics

Perl Practical Extraction and Retrieval Language

RF Random Forests

RMSE Root Mean Square Error

SADC Southern African Development Community

SASSCAL Southern African Science Service Centre for Climate Change and Adaptive Land Management

SCADA Supervisory Control and Data Acquisition

UKMO UK Met Office

UM Unified Model

WASCAL West African Science Service Centre for Climate Change and Adaptive Land Management

WMO World Meteorological Organization

WRF Weather Research and Forecasting

Abstract

This work presents a programmable toolkit that allows the design and implementation of a smart weather system (SWS) at a local level. The uPyCraft software development kit (SDK) was used to load the MicroPython code and scrap three weather values (surface temperature, relative humidity and atmospheric pressure) from the DHT11 and BME280 sensors connected to the ESP8266 microcontroller. The NetBeans SDK was used to develop the database management system using SQLite and Java programing to store the collected data. The RStudio SDK was used for statistical data analysis, time series forecasting and in the development of a dashboard. The toolkit also facilitates expansion of functionalities of an existing automatic weather station (AWS) like that at the Botswana International University of Science and Technology (BIUST) into a fully-fledged SWS. The K-Nearest Neighbour (KNN) was used for multi-step ahead forecasting of maximum air temperature as a proof of concept in using the weather data collected from the BIUST AWS. The Multi-Input-Multi-Output (MIMO) strategy and the Recursive strategy were compared through the KNN algorithm. The performance of these two predictive models were assessed using the mean absolute error (MAE) and correlation coefficient (r). The optimum predictive performance of MIMO outperformed that of the Recursive strategy with a 95% accuracy compared to 67% of the latter. This work is the first of its kind in developing a programmable toolkit to bridge existing gaps in the availability, access and use of weather data generated by the BIUST AWS. The developed dashboard can be used for dissemination of weather outputs to end users. The generated datasets can be used by students, researchers and data analysts to evaluate machine learning algorithms and determine optimal solutions based on real world weather data instead of using artificial data. The proposed programmable toolkit is scalable, with potential applications in smart houses, smart villages, and smart irrigations.

Key Words: Programmable Toolkit, Weather Sensors, Microcontrollers, Predictive Analytics, Forecasting, Dashboard, Smart Weather System

CHAPTER 1

Introduction, Problem Statement and Proposed Approach

Chapter Introduction

Meteorological data is a valuable global commodity that is catalysing the development of robust responses to some of the climate related problems pressing our societies. In fact, weather data collected using environmental sensors adds value to decision making and refinement of policies to improve human lives [69]. These include: applications in smart irrigations [44, 86] and predictions in meteorological sectors. This chapter explores current meteorological challenges [104], while identifying data driven solutions [31, 84] to bridge the gap [19] between weather data or information and end users.

Motivation

Smart Weather Systems (SWSs) are expected to play an important role in the smart homes, smart villages and smart farms in the near future. Increasing complexity in the integration of weather data from various repositories, electronic weather sensors and smart weather applications requires microcontrollers and data driven techniques. This integration enables to address interoperability problems while making use of the available yet scarce resources. Climate informatics [16] integrates climate science and environmental data science [36] to develop predictive analytics [22] and statistical machine learning [34] solutions to facilitate the understanding of our environment.

1.1 Introduction

Weather and climate monitoring services are among the key thematic areas at the Botswana International University of Science and Technology (BIUST). On another hand, one of BIUST's priority projects is that of a "Smart Village", which kick started with the BIUST "Smart House" [74]. Such initiatives are expected to have small-scale renewable energy resources [3, 50] and energy management systems. These kind of projects require advanced technologies for control and management like: monitoring weather events, monitoring room temperatures [75], managing energy consumption; and integrating, visualising or disseminating data from various sources to generate knowledge for decision making.

Small to medium private enterprises and research institutions like BIUST, SASSCAL¹ and WASCAL² have introduced automatic weather stations (AWSs) [2] to monitor weather at finer temporal resolutions. Meanwhile, most of the AWSs installed in developing countries are underutilized. For instance, the BIUST AWS in Figure 1.1 is equipped with a local machine that runs a SCADA software (XConnect) and a data logger that stores historical weather measurements. The current system is maintained by the Sutron Company, making it difficult for local technicians to integrate new electronic devices to collect weather data. This particular AWS also lacks the backend service layer for data analytics as well as the functionality for dissemination of weather outputs to end users (see the situational analysis in appendices [6]). Another example is that of the Botswana Department of Meteorological Services (BDMS); it lacks a designated platform for dissemination of the generated weather analytics and forecasts to end users, relying on local stations (i.e., radio, TV) and Facebook.

Apart from the aforementioned gaps, Botswana still lacks a number of AWS for a wide coverage of monitoring. Moreover, there is limited research in SADC to address specific challenges relating to the installation of IoT based Cyber-Infrastructure (CI) systems to enable the collection of weather data. Another gap is in documentation on interoperability of devices to connect and exchange useful weather information [105]. This necessitates the implementation of smart weather systems to enable interoperability and integration of new devices to enhance repetitive operations and services related to meteorological data.

¹www.sasscalweathernet.org

²<https://wascal.org/>

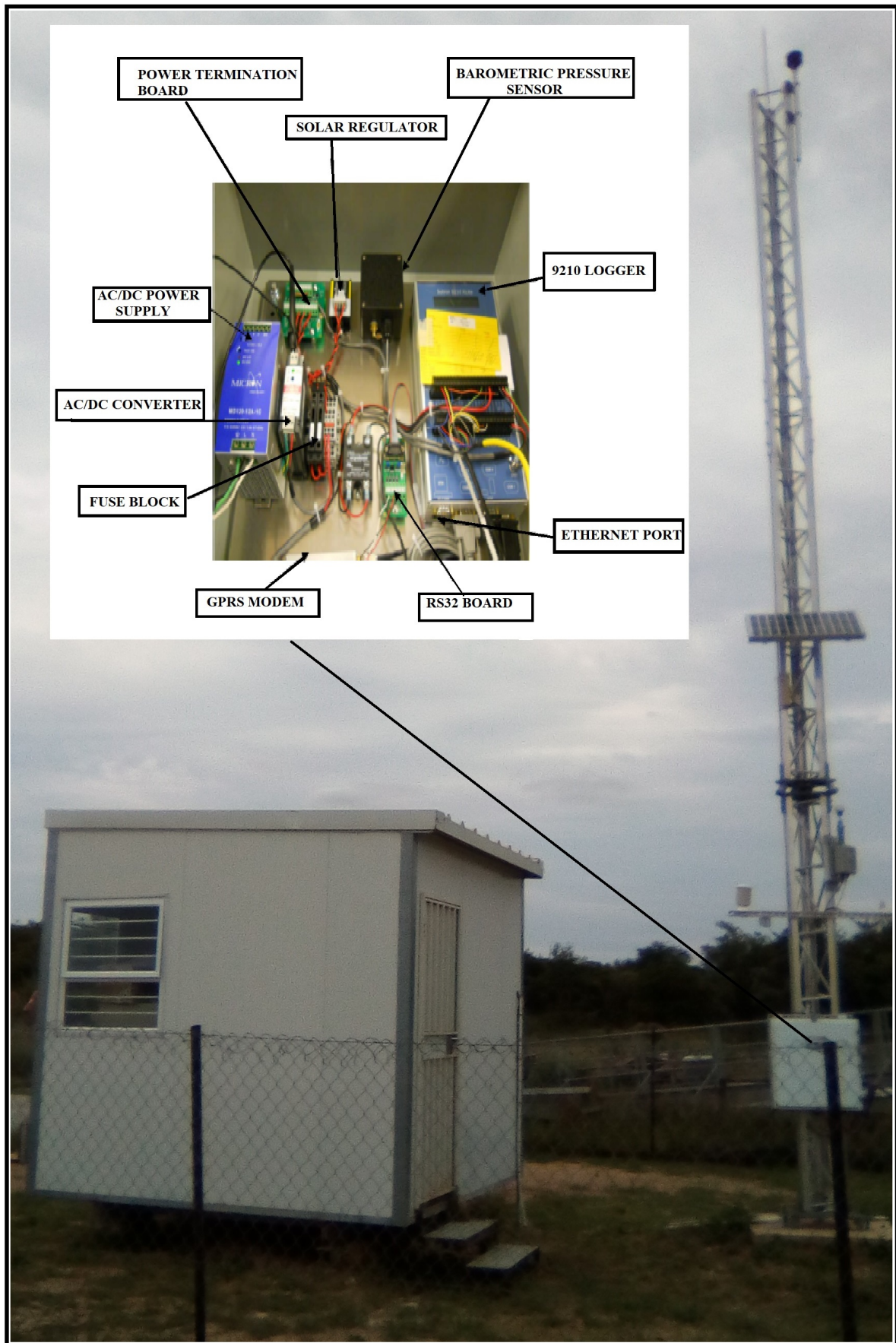


Figure 1.1: The BIUST AWS installed by Sutron for weather monitoring.

1.2 Problem Statement

Small to medium enterprises like BIUST, SASSCAL [1] and WASCAL [2] have AWS for weather monitoring. Meanwhile, there is still limited coverage of AWSs in Botswana. Moreover, most end users like farmers and students lack technical skills [84] and tools to collect and use meteorological data in their business operations. Therefore, most of the data collected by the installed AWS are archived and go under utilised. All these hinder active participation of citizens to contribute to climate research. This is a case study at BIUST. Currently the BIUST AWS lacks the backend service layer for predictive analytics and dissemination of weather products to end users. In this context, the implementation of a smart weather system at BIUST is a necessary task with a two-fold outcome, namely: an enhanced weather system for the BIUST AWS; and added value in end user comfort level.

1.2.1 Aim and Objectives

The aim is to contribute to the local weather monitoring community and improve the efficiency and efficacious use of products from automatic weather stations. The general objective is to develop a programmable toolkit for a smart weather system in BIUST.

1.2.1.1 Specific Objectives

1. To design algorithms for data scraping and data wrangling by:
 - (a) Designing a prototype to collect air temperature, humidity and atmospheric pressure using MicroPython; with available electronic sensors and microcontrollers,
 - (b) Implementing R algorithms for web scraping, then generate modelling datasets;
2. To use the KNN for short term forecasting using univariate temperature series, i.e.,
 - (a) To compare KNN's MIMO and Recursive multi-step ahead forecasting strategies using the maximum air temperature time series collected from the BIUST AWS,
3. To develop a dashboard application for deployment of the proposed toolkit, i.e.,
 - (a) To deploy the R Libraries for visualisation of the generated weather outputs.

1.3 Research Approach and Methodology

This is an applied research to bridge the identified gaps in section 1.1 following a situational analysis conducted through oral interviews and printed questionnaires (see Appendices 6). Data scraping [117], data wrangling [92] and dashboards techniques were used to make use of the consolidated data. The quantitative approach was deployed for experimentations with non-parametric algorithms (KNN and Random Forests) that do not make assumptions about the distributions of data, saving time for analysts and developers [22]. The KNN algorithm is easy to interpret compared to black box models like neural networks.

The MIMO, Recursive and Direct strategies are commonly used for multistep forecasts [1]. The Direct strategy does not use any predicted values to make forecasts, hence less prone to propagation of errors; but it requires large computational time and higher functional complexity than the other two [1]. The MIMO strategy can better learn the dependency between the predictor and target to make the entire forecast sequence at ago. Meanwhile, it constructs independent models with heavier computational load than the Recursive strategy while using more data to avoid overfitting. The recursive strategy requires the construction of a single model, hence saving computational time [108]. However, it allows propagation of errors from one step to another, hence the performance can quickly degrade with an increase of the forecasting horizon [6]. The Recursive strategy can be biased for nonlinear problems (i.e., weather forecasting) which can be mitigated by using hybrid strategies [108]. Therefore, a trade-off between computational costs, bias and estimation variance [6, 108] was considered to select MIMO and the Recursive strategies to implement the KNN model.

The data deployed in this work is constrained to point observations collected using AWSs. The focus was on maximum air temperatures which are associated to heat waves and high evapo-transpiration that lead to droughts. The BIUST AWS was selected as a case study since it experiences the gaps in the provision of weather data to consumers. The Shakawe and Tubu AWSs were used as a proof of concept for pragmatic applications of predictive analytics in meteorology. The data is mainly limited to short term forecasting activities, and cannot generalise for climatological studies. Meanwhile, this work gives insights of how these disruptive technologies can enhance meteorological service delivering a local levels.

Contribution

This toolkit is the first of its kind at BIUST, and it is expected to open many other design and development opportunities. The toolkit, written in MicroPython, Java and R, provides the data collection and pre-processing tools; as well as the mathematical programming functionalities to leverage existent AWSs into fully-fledged SWS. It also facilitates the integration of other national weather data repositories to enhance further research. The generated datasets and the developed dashboard application can be used for agricultural and educational purposes. In fact, the developed dashboard API can be adapted to be used as a SCADA system by farmers to monitor weather parameters and weather patterns for effective remote operations (i.e., smart irrigations) [52].

It should be noted that having a SWS with data driven technologies that generate real world datasets can enable further studies on local environmental phenomena. From an academic point of view, the toolkit allows applications of more sophisticated data driven techniques (i.e., filling missing gaps in weather data banks). The data can also enable students and other researchers to evaluate machine learning model performance based on real world datasets. The toolkit uses statistical machine learning techniques to perform multi-step ahead forecasts of maximum air temperature using data from the BIUST AWS. For instance, the generated forecasts can be used by students and researchers who seek to investigate the potential benefits of weather data (i.e., sunshine duration and temperature) in energy management strategies. Another potential extension of this toolkit could be in the design and implementation of an automatic room heater control system [75] for the smart house [74] in BIUST.

Based on the aforementioned facts, the proposed programmable toolkit using low-cost microcontrollers and predictive analytics can accomplish the necessary actions to bridge the identified gaps in sharing and making use of meteorological data collected at local levels in developing countries; particularly for the BIUST environment which has been allocated to develop prototypes for smart villages in Botswana.

Chapter Summary

This chapter motivated the implementation of disruptive technologies in the meteorological sector. It then articulated the present challenges imposing limitations regarding the delivering of meteorological services at local bases; particularly in developing countries like Botswana. The chapter also highlighted the potential value of open-source technologies such as the Internet of IoTs, predictive data analytics, machine learning and dashboards. It should be noted that the proposed non-parametric methods can extract useful hidden patterns from raw data that often go unexplored; and that human-engineered processing steps can be automated using computer programming techniques such as data scraping and data wrangling to minimise human-machine interaction. The developed dashboard can be scaled up to fill the existing gaps in dissemination of weather outputs to end users in Botswana, i.e., in BIUST.

1.4 Outline of the Thesis

- ❑ **Chapter 1** introduced the study, identified the gaps, and stated the problem, including research aim and objectives, limitations and contribution of this research work.
- ❑ **Chapter 2** presents the background information on Numerical Weather Prediction (NWP) models and literature review, covering the disruptive technologies of IoT and predictive analytics and their applications in weather analytics and forecasting. The chapter also presents the application of dashboards for weather service delivery.
- ❑ **Chapter 3** presents the framework of data driven modelling from the perspective of IoT and predictive data analytics; articulating the methodology, tools and methods.
- ❑ **Chapter 4** presents experimental design, dashboard application design and results. Algorithms for web scraping are presented for the collection of weather data from data sources. Then data scraping is deployed to manipulate data from sensors. Data wrangling is then deployed to generate datasets that are then injected into the data mining workbenches. The chapter also articulates the design of a dashboard application as a proof of concept for pragmatic application.
- ❑ **Chapter 5** presents the conclusions, recommendations and future work. It summarises the findings, results, limitations, contribution, recommendations and future work resulting from this research work.

Background Information and Related Literature Review

Chapter Introduction

This chapter presents a background information and related literature review on conventional methods of meteorological services. It articulates present challenges imposing limitations regarding the delivering of weather services at local levels; particularly in developing countries. It then presents the potential value of low-cost IoT devices like electronic sensors and microcontrollers. It also highlights open-source data driven approaches and computer applications that can be deployed for added value in weather monitoring and service delivery to end users in developing countries like Botswana.

2.1 Background Information

Physically based Numerical Weather Prediction (NWP) models (i.e., COSMO [4], ECMWF [5], ICON [118], UKMO [90, p. 7] and the WRF [87], see Table 2.1) have been in use to monitor and model weather. Despite the wide usage of these systems, they have been reported to be less efficacious at finer spatio-temporal resolutions [116, p. 6] with severe biases due to inadequate parametrisation. Several post processing methods have been used to circumvent this challenge including: the Perfect Prognosis [115], Model Output Statistics (MOS) [38] and Reanalysis [61]. The common post processing model in statistical downscaling is MOS which takes in NWP model outputs and climate data as inputs.

Table 2.1: Common NWP models and their respective programming tools.

Model	Origin	Programming tools
COSMO	Europe	MPI/OpenMP, OpenACC, CUDA
WRF	United States	C++, F90, OpenMP and MPI
ICON	Europe	Fortran, C, MPI, LAPACK, BLAS
ECMWF-Model	Europe	Atlas: Fortran, C/C++, CUDA
MOGREPS-UK	Europe	Fortran, MPI, OpenMP

MOS deploys multiple linear regressions [61] to find the relationship between observations and global model data. It is known that linear regressions models make many assumptions about the data [22, p. 276]. In addition to that, implementing and maintaining these NWP models is also expensive regarding the required resource. In that case, non-parametric distribution free models like Random Forests and Neural Networks [45] can be deployed for modelling as well as in: 1. blending Earth Observations with outputs from NWP models, 2. post-processing of ensemble weather outputs [91], and 3. correcting model biases.

2.2 Disruptive Technologies for Weather Services

IoT describes the interconnection of electronic devices [111, p. 11] such as sensors and microcontrollers. IoT and data driven technologies enable innovative synergies to design affordable and flexible solutions without incurring licensing expenses [48, p. 277]. Meteorology is one of the sectors that applies IoT [102, 48]. In fact, AWSs are embedded with sensors [48] that capture various weather data like temperature and humidity. The collected data is manipulated using supervisory control and data acquisition (SCADA) systems [21].

The integration of IoT and predictive analytics (i.e., low-cost AWSs and machine learning) lead to great potential for many operations that rely on weather data by lowering the cost of proto-typing and implementation. For instance, electronic sensors embedded in AWSs [24] can harvest more fine-grained data about soil moisture at farms. The data is then processed further for use in decision making (i.e., watering only a targeted area). These lead to effective use of resources that are scarce in many African countries like Botswana. The integrated framework of a holistic data-driven IoT CI system can be partitioned into three: 1. the Frontend Service Layer; 2. the Gateway Layer; and 3. the Backend Service Layer.

2.2.1 The Frontend Service Layer

This layer is comprised of digital sensors capable of detecting and measuring environmental phenomena. There exist various weather sensors, and they have been articulated well in a number of articles (see [23, 81, 53, 23]) as illustrated in Table 2.2. Electronic sensors are mainly characterised by manufacturers, accuracy, resolution, range, precision and cost. For instance, the Bosch BMP180 measures atmospheric pressure [300,1100] hPa, and temperature [-40,85] in °C, with an accuracy of $\pm 1.0^\circ\text{C}$. It can work as an altimeter with an accuracy of $\pm 1.0\text{m}$. The DHT22 measures relative humidity (RH) [0,100] (in %), with an accuracy of [2-5]%, and temperature [-40,125] °C, with an accuracy of $\pm 0.5^\circ\text{C}$. The DHT11 measures temperature [0,50] in °C, and RH [20,80] in % and an accuracy of 5%.

Other common weather sensors include: the wind vane and the anemometer to measure the wind direction (in °) and wind speed respectively. The pyrometer sensor measures the electromagnetic (solar) radiation emitted by the sun. The AS3935 Franklin sensor detects lightning. The rain detector measures rain droplets using: a collecting funnel, tipping bucket and a magnetic reed switch. This rain gauge generates output pulses which are fed to the microcontroller's clock input. Likewise, the soil moisture sensor measures water content of the soil. The designs of these electronic sensors keep on changing; with some incorporating visual displays (see the 1B6 MS rain meter with a thermometer in Table 2.2).

2.2.2 The Gateway Layer

This layer facilitates the interaction of electronic devices and their external environments by a controller or gateway. The gateways distribute data from sensors to base stations for further operations. The controller need to be persistent to data, with enough memory and bandwidth to facilitate data filtering before transmission. The sensors are attached to the controller through interfaces such as: Inter-Integrated Circuit (I²C), Serial Peripheral Interface (SPI) and General Purpose Input Output (GPIO) pins. I²C is easy to implement, and can host multiple master devices and multiple slave devices using Serial Data Line (SDA) and Serial Clock Line (SCL). Though SPI requires more connections, it is more common due to its lower power consumption and higher throughput than I²C [48, p. 280]. The communication can also be wired or through wireless standards protocols like ZigBee.

Table 2.2: Summary of hardware (sensors) for a low-cost automatic weather station

Sensor	Image	More Details	Accuracy	Interface
<u>DHT11</u> : Temperature and Relative Humidity Size: 28 x 12 x 8 mm		Temp. Range: [0-50] in °C RHumi. Range: [20:90] in %	Temp: ± 0.5 °C RHumi: $\pm 5\%$	3 DuPont wires
<u>DHT22</u> : Temperature and Relative Humidity Size: 40 x 23 mm		Temp. Range: [-40,80] in °C RHumi. Range: [0,100] in %	Temp: ± 0.5 °C RHumi: $\pm 2\%$	1-wire bus
<u>BMP180</u> : For Atmospheric Pressure & Temperature		Pres. Range: [300,1100] hPA Temp. Range: [-40,85] °C	± 1.0 °C	I ² C
<u>Rain Meter</u> : w/thermometer Model: 1B6 MS		For rainfall & temperature	2 mm	PC817
<u>Anemometer</u> :		Calculates the wind speed Uses reed switch	-	GPIO
<u>Wind Vane</u> : Measures wind direction using: Reed switches		Its an analog circuit with enabled resistors to act as voltage divider	22.5°	Analog

2.2.3 The Backend Service Layer

Unlike the frontend and the gateway layers that mainly involve engineers (electrical and electronics), this layer is mainly comprised of (computer) scientists and developers. It mainly involves the management of data including: data cleaning, data analysis, data visualisation, data storage and dissemination of outputs. Meanwhile, the rapid increase of new sensors with high frequencies has brought varying challenges to environmental scientists.

For instance, weather forecasters are exposed to a vast amount of weather data [1,2] and information to digest before they can issue forecasts. Furthermore, most weather data operations such as: feature engineering and imputation of missing values [60] cannot be handled effectively via the human power. These invite system automation [117] and machine learning to minimise human intervention in the overall pipeline of weather services. Through this layer, forecasters can deploy algorithms in pre and post processing of weather data, model outputs [35], as well as in the detection of changes of weather patterns [62].

IoT Hardware, Software and Firmware

The IoT cyber infrastructure system (see Figure 2.1) comprises of electronic micro-controllers such as Arduino's ATmega328 and ATtiny828 [111], [102]. Other low-cost microcontroller boards include: Raspberry Pi, PyBoard, BBC micro:bit, Adafruit's Circuit Playground Express, and the ESP8266/ESP32 family. The application of IoT devices requires dedicated software and firmware, with tailored libraries and modules to efficaciously manage resources like power, input devices, data and memory.

Microcontrollers have been traditionally coded with languages like C or C++ that are difficult to learn. Recent applications deploy MicroPython using integrated development environments (IDE) like uPyCraft or Thorny. The transmission of data to backend layers can be conducted through direct wiring, WiFi, GSM/GPRS, or satellite connections. These are mainly facilitated by machine-to-machine transmission protocols such as: Advanced Message Queuing Protocol (AMQP) [82], Constrained Application Protocol (CoAP) [14], Message Queuing Telemetry Transport (MQTT) [111] and Hypertext Transfer Protocol (HTTP) [7]. The communications are bidirectional to enable operations like updating gateways and system configurations.

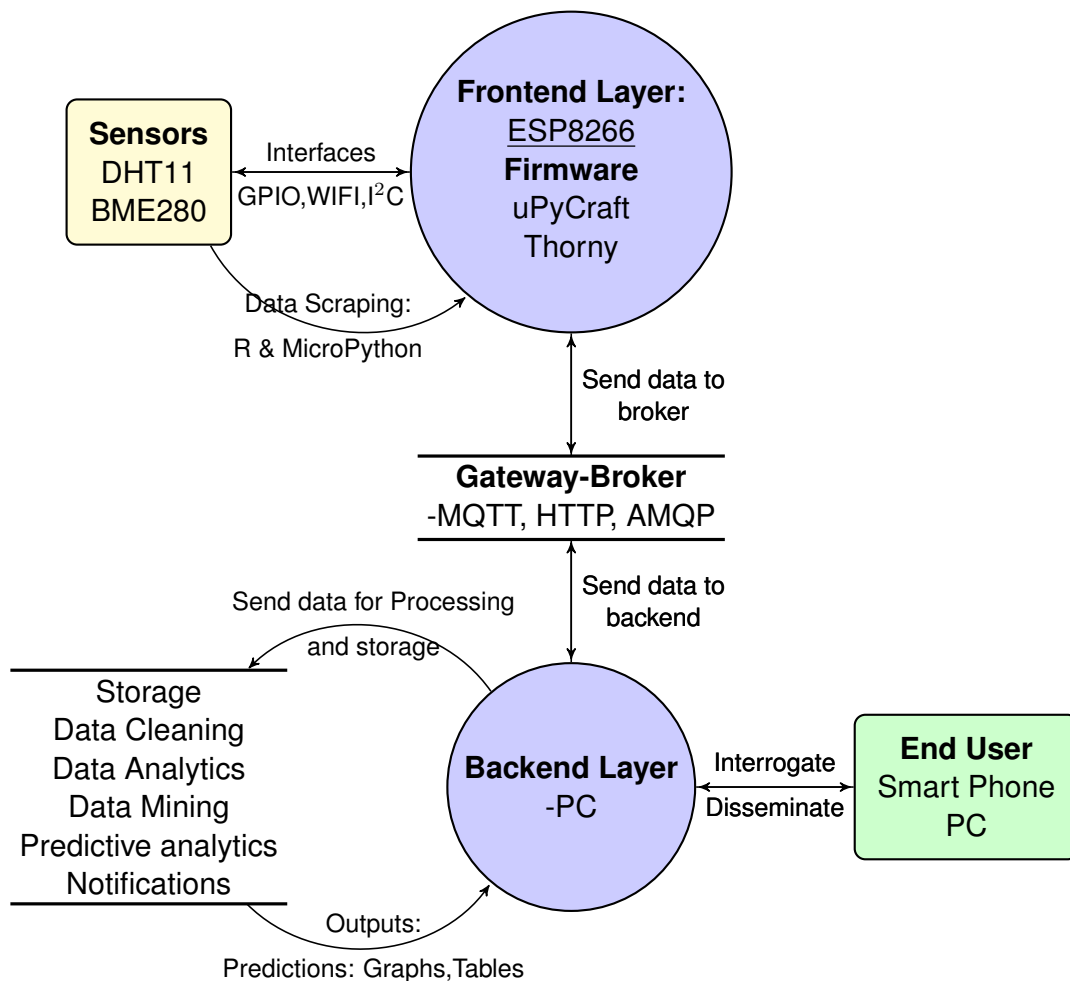


Figure 2.1: The general flow of weather data from the frontend layer to the backend layer.

2.3 Related Literature Review

As noted by Mass *et. al.* in [62], advances in IoT and data collection instruments have catalysed the use of short-term forecasts to facilitate adaptations to characteristics of local terrains. The development of a low-cost AWS depends on various factors such as: the business problem, availability of resources and human technical skill set. Devaraju *et. al.* [23] implemented a low-cost AWS with four sensors to collect weather data at intervals of thirty seconds. They used the SHT11 sensor for temperature and relative humidity, while the 6410 anemometer and the 6450 pyrometer were used to detect wind direction, precipitation and solar radiation respectively. The BMP180 was used for atmospheric pressure.

Aponte-Roa *et. al.* [24] used: AM2315 to collect humidity and temperature; the BMP280 for atmospheric pressure; an anemometer and a wind vane for wind speed and wind direction. Their system was powered using four 6V solar panels and a Lithium-Polymer battery. The data was transmitted via Wi-Fi for storage in MySQL database. Likewise, Felipe *et. al.* [95] developed a low-cost hydro meteorological AWS. Their system integrated a 5MP camera to capture and store images used to record different types of atmospheric phenomena like clouds. The data was transmitted via wireless to Hydras3, a visualization system used in the institution of hydro-meteorology and environmental studies in Colombia. MySQL was used for data storage. They also used MonoDevelop and C# to develop a GUI to visualise the collected data. These two works deployed Raspberry Pi for computing. Meanwhile, neither of the two works deployed predictive models for data mining in the application layer.

Similarly, Karvelis *et. al.* [53] deployed the Zerynth platform and the ESP32 microcontroller to record weather data using a ship at 15 minutes intervals; from 22-01-2018 to 09-02-2018. The data was used to predict short-term wind speed using a linear regression. They used a 4 day data set for training, and the rest for testing. The models were evaluated using mean absolute error (MAE) and root mean squared error (RMSE) to get 1.591 and 2.565 respectively. The protocols deployed include: UART, SPI and I²C. This was another good contribution, but the experiments used more data for testing than training which might lead to over-fitting. Though they did not specify the programming language they used, the development environment used (Zerynth) in their work supports C and Python programming.

It can be noted so far that some applications of these IoT devices are climate specific, and with some deployment in typical environments. For instance, Netto *et. al.* [78] proposed an open-source application controlled using Arduino Uno to monitor the impacts of climate change in southern Patagonia. They had a challenge though, the experimental region was remote with no coverage of telephone communications except the expensive satellite networks. This particular solar powered system comprised of an AWS and an electronic ablation station incorporated with Radio Frequency Identification to measure the melting of glacier. These were all embedded with low-cost sensors: BMP280, DHT22, DS18B20, water proof humidity sensor, Real Time Clock (RTC DS3231) and a micro card shield.

Nsabagwa *et. al.* [81] proposed an AWS using a Raspberry Pi as a gateway to connect the sink nodes. The AWS was close to a networked building, so they used Ethernet for internet connection. The nodes were equipped with radio transceivers each to transmit data to the sink node. The gateway transmitted the AWS data to a repository, and a downloadable text file via HTTP was used as a database. This limited efficient querying, necessitating some form of relational database for efficacious operations. The work claims to have reduced implementation costs of an AWS from \$7000.00 (commercial) to \$1800.00 (open-source). Unfortunately, their system does not account for the dissemination of data to end users. They also did not deploy data mining to extract hidden patterns from the weather data.

Muck *et. al.* [72] harvested weather values (at intervals of 2 minutes) using SparkFun Shield sensors. The sensors were interfaced to Arduino Uno R3. They used the Raspberry Pi 3 Model B as a gateway to stream the harvested data for storage in Google Cloud using SQL. They also developed a website and a mobile application using Google Data Studio and Android Studio (using Java) to display the real-time weather conditions to end users. Their application had the functionality for short-term weather prediction in intervals of thirty minutes. Notifications were disseminated via social media. Though their work did not explicitly state more details on the modes of communication and transmission protocols, it provided a good example of a data flow in an IoT framework. Meanwhile, this approach used simple predictive algorithms with less data. Other algorithms such as Recurrent Neural Networks or ensemble predictive models like Random Forests could have given better results.

Trifun *et. al.* [101] also used a Raspberry Pi based platform and low-cost electronic components (i.e, DHT11 and BMP180 sensors). The C language was used to read data from the sensors. The harvested data was transmitted through a wireless access point and USB Wi-Fi dongle from a measuring point to the Android client. Their work used UDP instead of TCP protocol to avoid establishment of connection at each data request. This also avoided the provision of a separate channel for data transfer between the GUI client (Android App using Java) and the server. Their work lacked the data science functionalities. Another work by Hardeep *et. al.* [99] deployed an Arduino Uno (ATmega328) and 3 sensors (SHT25, BMP180 and wind vane) to collect weather parameters. They transmitted the data via wireless (Zigbee) connection to a computer with a GUI developed using LabVIEW.

2.3.1 Discussion of Related Literature

Most of the proposed systems reviewed so far integrated low-cost electronic components like sensors and microcontrollers like ESP-32, Arduino Uno, PIC16F887 and Raspberry Pi (see Reddy *et. al.* [100], Brito *et. al.* [11] and Palak *et. al.* [51]). The collected data was either stored on a local repository or uploaded on to a cloud [100]. Some of these works performed short-term weather forecasting, some did not. Some works configured system notifications of weather conditions [99] to local users [51] via Amazon SNS web service [1].

Based on the information displayed in Table 2.3, different researchers deploy different sensors and different microcontrollers to harvest environmental data. It can also be noted that most of these works did not implement the data driven models to extract patterns from the collected data (with exceptions of [51] in India and [53] in Greece), and only few ([51, 23, 99, 95]) implemented GUI for dissemination of data to end users. Another interesting observation is that non of these work deployed the R programming for: web scraping, web crawling and dashboard design. MicroPython was neither used in these works for data scraping from sensors as well as in microprogramming. Thus, this work will combine R and MicroPython for data scraping, data logging, visualization and dashboard design. R² and Shiny applications have been deployed for related weather projects. These include but not limited to: Appsilon³, and the UK Drought and Water Scarcity Research Programme [17].

¹<http://aws.amazon.com/sns>.

²<http://www.r-project.org/>

³<https://appsilon.com>

Table 2.3: Summary of related works, their origins, year published, and the deployed IoT and predictive analytics tools and methods.

Paper & Origin	Year	Sensors	Controllers	Interfaces	Software & Firmware	Data Storage	Predictive Analytics	GUI
[23] India	2015	SHT11,BMP180, DI {6450,7852}	PIC16F887	MAX-{232&485,} XBee-Pro	C	Excell & MySQL	✗	Visual Basic
[99] India	2016	SHT25,BMP180, Wind Sensor	ATMega-328	Zigbee	C/C++	Data Logger	✗	LabVIEW
[95] Colombia	2018	SHT11,BMP180, DI {6450,7852}, Camera	RPI 2:B	Wireless	Python3	MySQL	✗	C#
[53] Greece	2018	DHT22,BMP180, BMP280,DS18B20 HC-SR04,LDR	ESP-32 DevKitC	RFID	C	DataLogger	✓	✗
[72] Malaysia	2018	SHT11,BMP180, DI {6450,7852}	PIC16F887	-	-	MySQL	✗	✗
[24] Puerto Rico	2018	SHT11,BMP180, DI {6450,7852}	PIC16F887	-	-	MySQL	✗	✗
[78] Brazil	2019	DHT22,BMP180, BMP280,DS18B20 HC-SR04,LDR	Arduino Uno	RFID	-	DataLogger	✗	✗
[81] Uganda	2019	SHT11,BMP180, DI {6450,7852}	Raspberry Pi	-	C	CSV	✗	✗
[51] India	2019	DHT22,BMP180, BME280, Rain Sensor	PIC16F887	-	-	MySQL	✓	Django
This Work: Botswana	2020	DHT11,BME280	ESP8236	GPIO	R, RStudio, MicroPython, uPyCraft	SQLite	✓	Shiny App

2.3.2 Trends in the Weather Industry

Though the study in [106] was conducted to better understand the state of art technologies and future landscape of the weather industry in the United States, it also serves as a guide and mode of inspiration to developing countries like Botswana. The report indicated some trends and changes in weather enterprises due to the growing demand for weather services as weather events become more volatile and expensive to address. The report supported innovative tools that deploy disruptive technologies (Artificial Intelligence, Big Data, and the IoTs) to improve businesses in meteorological sectors. A point of interest to note here is that "these new technologies are catalysing advances in weather prediction and decision support by harnessing the value hidden within the collected weather data [106, p. 4]."

Predictive analytics and machine learning technologies offer indispensable contribution to the meteorological sector. Statistical tools [2] and visualization techniques [76], coupled with smart devices make it easy to provide decision support and weather solutions. A lot of companies have recently adopted advanced statistical machine learning techniques and surrogate machine learning models to post-process outputs from NWP models for improved forecasts at finer spatio-temporal resolutions. These have caught attentions of companies like IBM and Monsanto to acquire weather enterprises in the past decade. Meanwhile, there is still a lot to do in developing countries like Botswana in terms of capacity building to foster pragmatic implementation of disruptive technologies in meteorology. The ticks in Table 2.4 indicate programming languages for climate modelling in some southern African countries, while the crosses indicate practical limitations (adapted from Meque [67, p. 6] [4]).

Table 2.4: Tools for climate data analytics in southern Africa.

Model	R	GrADS	Python	CDO	Fortran
Lesotho	✓	✗	✗	✗	✗
Botswana	✗	✗	✗	✗	✗
Malawi	✓	✓	✗	✗	✗
South Africa	✓	✓	✗	✓	✓
Tanzania	✓	✓	✓	✓	✗
Zambia	✓	✓	✗	✗	✗

⁴<ftp://cscftp.sadc.int/regcm-training/Day1/>

2.4 Deployment of the toolkit

As articulated by Abbott [22, p. 16], the construction of machine learning models is not that complicated. One of the challenges associated to them is the collection and preparation of data into format that can be ingested into a data mining workbench. Another problem is that of developing the data driven systems such that the generated outputs can be utilised by end users. Though the applications of machine learning is over researched, the same are under deployed at practical level. One way to tackle these is through the development of weather applications [31] (i.e., web, dashboard) to disseminate weather information.

2.4.1 Dashboards for Weather Services

A dashboard is a visual display of information, consolidated and arranged on a single screen to facilitate exchange of information between key stakeholders [63]. The use of dashboards has been traditionally done in private companies. It is important to note that the use of dashboards in government organisations has increased in the past years [33]. These computer applications have been used to empower the public in smart cities [63]. Since IoT facilitates the consolidation of weather data, its integration with reactive applications like dashboards could facilitate crowd-sourcing [28] of social networks. Crowd-sourcing can facilitate exchange of weather information between weather service providers and the public. These added value promotes transparency, participation and collaboration.

Several studies and literature reviews exist to support data science and dashboards in government operations [33]. Matheus [63] performed a thorough literature review, coupled with case studies to emphasise the value of data, the advantages of dashboards as well as the risks and principles to be considered in the design of dashboards for public use. A good example is that by Speedwell Weather⁵ company (USA and Europe) which facilitates efficacious visualisation of model forecasts for precipitation and temperature values based on outputs from NWP models such as: ECMWF Ensemble, ECMWF Operational, GFS Ensemble and GFS Operational. The Speedwell dashboard serves clients by visualising weather forecasts using tables, forecast maps, as well as the box-and-Wiskers plots.

⁵<https://www.speedwellweather.com/>

Other applications that integrate predictive analytics and dashboards include the Waze App ⁶ which deploys big data analytics and dashboards for instantaneous monitoring and display of road traffic and accidents via electronic panels in Rio de Janeiro City, in Brazil. This obviously helps in decision making as drivers may avoid routes that may lead to traffic jams. Another application, the Moovit ⁷ social application, for public transport deploys real-time GPS information on various events (i.e., break downs, traffic jams, accidents, weather conditions) as generated by citizens during the trajectory from one location to another.

Despite the deployment of dashboard applications for empowerment of the public [63, 49] in several sectors mentioned above, though in different countries, no measures have been exercised locally in Botswana to address the gaps in delivery weather outputs to the public. There is a gap in literature linking engineering and scientific studies with application development to meet the problems pressing societies in developing countries, particularly in Botswana. Noting that gap, this thesis is geared towards enhancing weather services; from data collection, dataset generation, analysis; and sharing of weather outputs to the public. The thesis attempts to fill these gaps by examining applications of IoT and predictive analytics in Botswana using BIUST and Okavango (Tubu and Shakawe) as case studies.

Chapter Summary

The rapid growth of IoT and low-cost AWS [81, 23] have paved a way for predictive analytics techniques like machine learning which have demonstrated to be effective in modelling weather [36]. They have found applications in modelling weather phenomena due to their simplicity and flexibility, a desirable feature in complimenting physically based models discussed in Section 2.1. For that reason, they are found to be more competitive and complimentary to the conventional approaches of environmental data science [36]. It should be noted that the main aim at current is not to replace the existing climate and weather models but rather to complement their outputs. A number of academic research have been documented in the past decades regarding the applications of data driven models in weather forecasting. Research from these demonstrates that which method to use depends on the available data, the available electronic equipments and the data mining workbenches [36]. The next chapter presents the approach followed to design a prototype applicable for weather monitoring. It will also articulate the methods used to make use of the collected data from any automatic weather station maintained by SASSCAL with the objective of making use of the generated data.

⁶<https://www.waze.com/>

⁷<https://moovitapp.com/>

CHAPTER 3

Methodology, Hardware, Software and Methods

Chapter Introduction

This chapter presents the methodological approach deployed to solve the identified problems in meteorological services at local levels. The chapter describes the methods of predictive analytics as well as the IoT devices that can enhance weather services. The work takes an applied approach that is more pragmatic than theoretical, with the objective of solving challenges associated to data collection, modelling, forecasting and dissemination of weather outputs to the public in developing countries like Botswana. It also enunciates the mathematical formulation proposed for predictive analytics, articulating the non-parametric machine learning models used in this research.

The Toolkit Insight

The toolkit for SWS comprises of three main layers: Frontend, Application and Backend layers as shown by the diagram in Figure 2.1. The general system circuit comprises of the ESP8266 micro-controller connected to a sensor node to measure three weather values (i.e., temperature and humidity using the DHT11 sensor and the atmospheric pressure using the BME280 sensor). A pull-up resistor was used to manage the amount of current passing to the DHT11 sensor. The measurements collected by the different sensors and meters are stored a standalone database developed using Java and SQLite. Figure 3.1 presents the flow chart (logical operation) of the proposed SWS. The complete list of materials deployed in this work is discussed in section 3.3.2.

3.1 The Proposed Pipe-line of the System

The generic end-to-end data driven pipe line used in this work is presented in Figure 3.1. It involves the consolidation of data (data scraping and web scraping) from data sources, data preparation (i.e., data wrangling), the generation of datasets for modelling and data analysis. The KNN model was used to impute missing values. Non parametric models (KNN and Random Forests) were used for modelling. Automated algorithms were also developed to allow real time injection of data from SASSCAL's automatic weather stations. Tools and methods deployed in this work are summarised using a mind map in Figure 3.3.2.

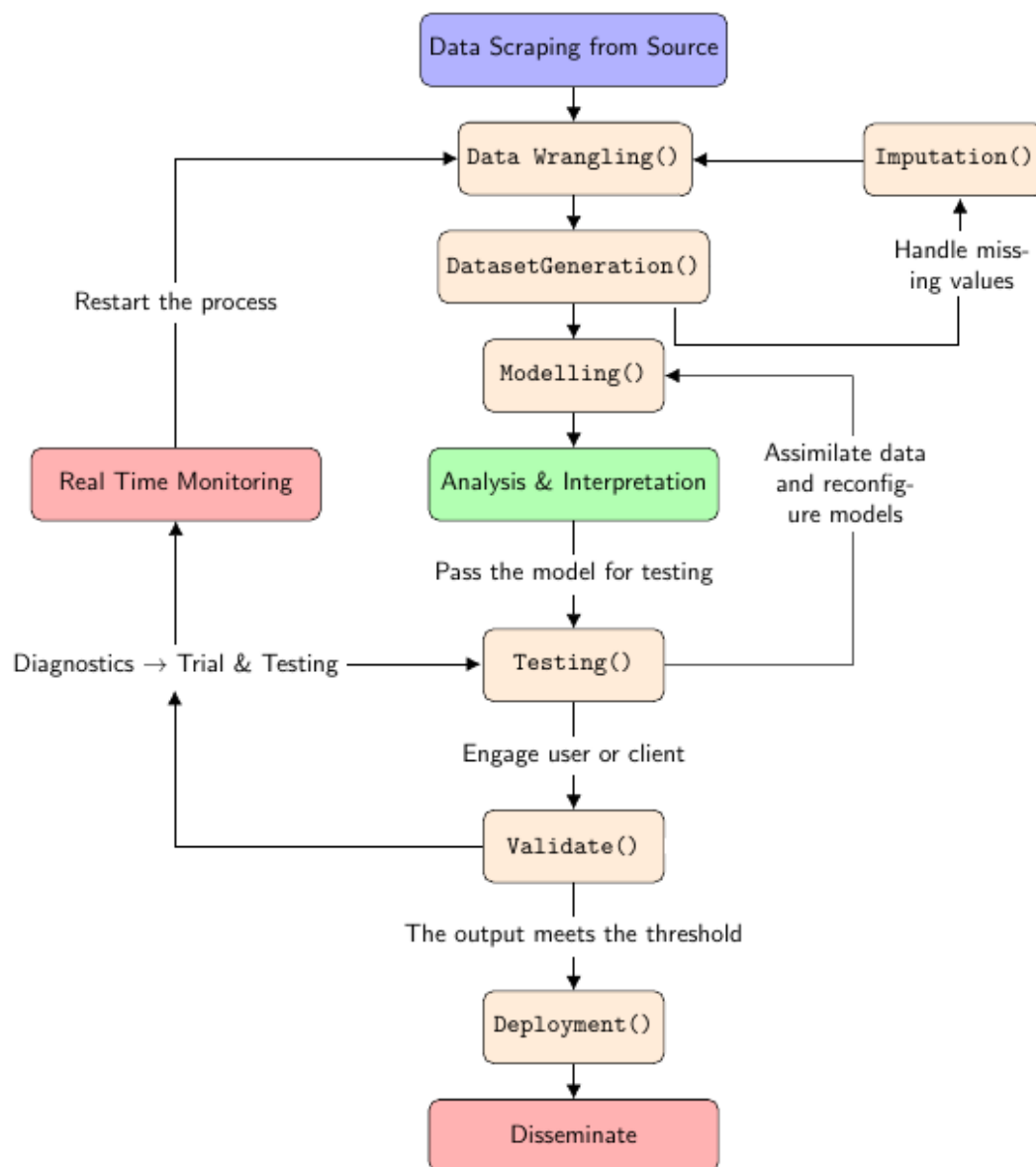


Figure 3.1: The general pipe line of the proposed methodology

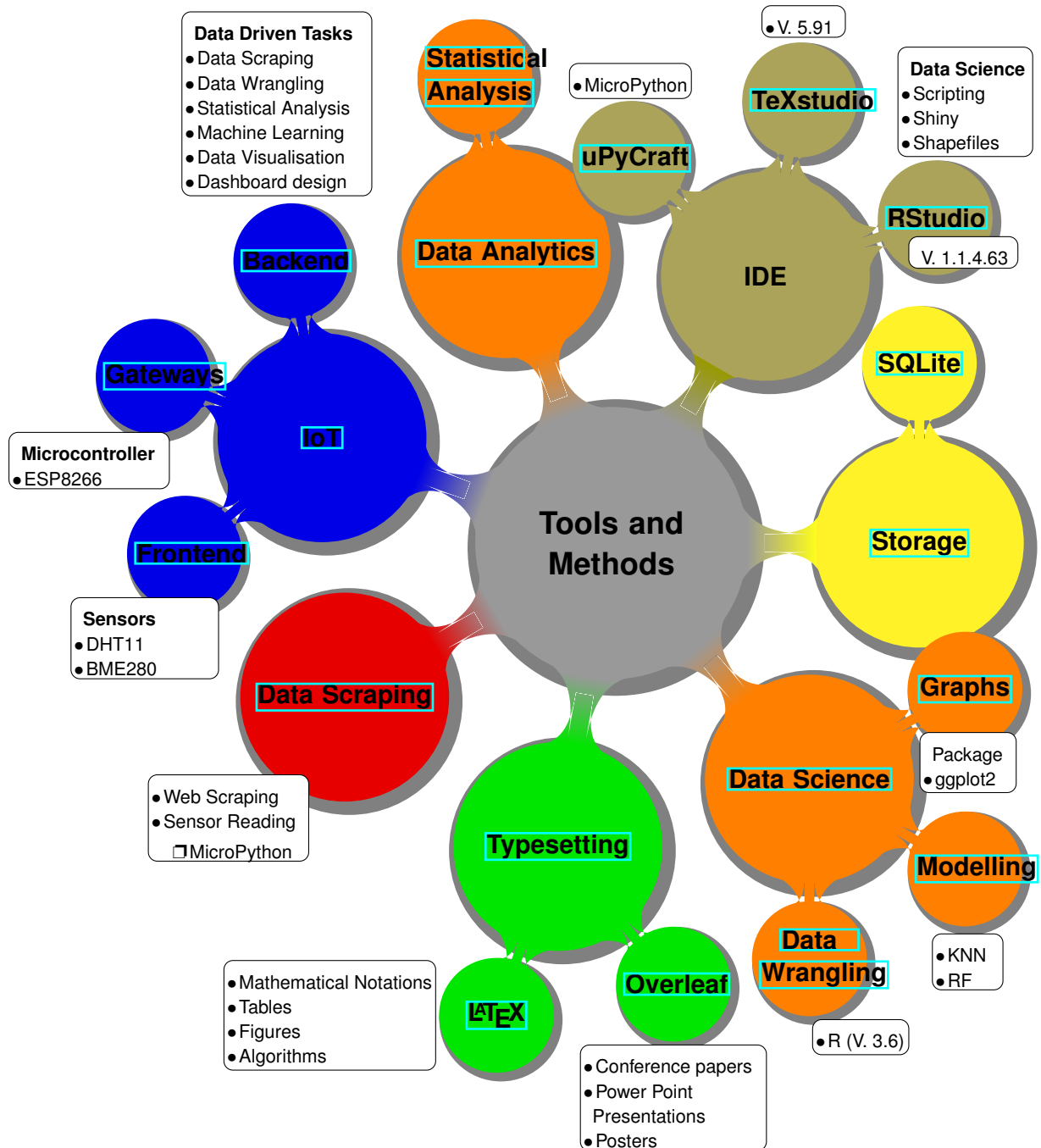


Figure 3.2: Mind map summarising the tools and methods deployed in this work.

This work deployed the *Cross-Industry Standard Process Model for Data Mining* (CRISP-DM) ¹ methodology (see Table 3.1) [22, p. 20], [54] to guide the development of the project from start to end. It should be noted that the original CRISP-DM as articulated in various literature does not include the model diagnostic phase included here, which is crucial for interpretation [70] and communication of model results to key stakeholders.

Table 3.1: CRISP-DM methodology for data driven modelling.

Phase	Description
Business Understanding	Problem statement, variables and conceptual mapping
Data Collection and Storage	Scraping data from sensors and websites and storage
Data Understanding	Examine the data; identify problems in the data
Data Preparation	Data pre-processing; feature engineering, deriving variables
Modelling	Construction of models and predictions
Assessment and Evaluation	Evaluation, assessments; and reporting of results
Model Diagnostics	Interpreting results from black box models
Model Deployment	Use of models (i.e., multi-step ahead forecasts; web application to disseminate outputs)

3.2 Business Understanding

The business problem for this research involved the use of data generated by automatic weather stations for predictive modelling. The main objective was to extract valuable information that can enhance the general understanding about the evolution of weather patterns at finer temporal resolutions. This particular phase also includes the evaluation of various IoT technologies as well as the analysis of the required tools and data driven methods to implement the proposed low-cost AWS prototype. These entail 1.) Identification of techniques for consolidation of weather data from environmental sensors and websites, 2.) Identification of the IoT tools for transmission and storage of the harvested data, as well as 3.) Identification of applications for dissemination of the weather data to end users. More about this phase can be explored further in this books [22], [54, p. 65] available online.

¹Some of the phases can overlap; while other phases that involve data collection and preparation can be condensed into data scraping and data wrangling through scripting.

3.3 Data Collection and Storage

Data acquisition is the first step in predictive analytics. It is irrefutable that without data, no predictive model can be constructed. This work deployed data scraping (i.e., reading temperature and humidity data from the DHT11 and BME280 sensors, and extracting data from a web site) techniques for weather data collection. Data scraping [73] is a general term used to refer to the process of deploying computer software to extract values from data repository like electronic weather sensors and websites for further data operations. The measurements collected by the weather sensors are archived in the SQLite database.

3.3.1 Hardware for Data Scrapping

NodeMCU ESP8266

This is a low-cost microcontroller that supports Wi-Fi networking, efficient power usage and reliable performance for practical applications. It is actually a simplified Programmable Logic Controller (PLC). PLC controls major industrial hardware like conveyor belts and power equipment. Likewise, microcontrollers like the ESP family facilitate access to peripheral devices like sensors (DHT11 and BME280) through interfaces such as GPIO, I²C, SPI, ADC and PWM. The ESP8266 can also be deployed as a standalone application, or as a slave. It can, as well, be deployed as a Wi-Fi adaptor.

Electronic Equipments

- ❑ The **DHT11** sensor deployed in this work has four pins (see 2.2.1 and Table 2.2). The sensor has a thermistor to accurately measure the surrounding temperature and a capacitive module to measure humidity. It contains a small chip for analog to digital conversion and does not include a pull-up resistor for the signal line. This sensor operates with 3-5.5V DC with a current supply of 0.5-2.5 mA.
- ❑ **BME280**: The BME280 sensor is the successor of the BMP180. It is a small and low power consumption sensor that measures absolute barometric pressure at a range of [300,1100] hPa; and temperature at a range of [0,65]°C. It operates at a voltage of [1.2,3.6]V and can be interfaced via I²C or SPI.
- ❑ **Resistor**: Pull up resistor (10K Ω) for the DHT11 sensor.
- ❑ **Jumper wires**: Male to male and male to female jumpers were used to connect the components of the proposed prototype (i.e., sensors to the ESP8266 board).

3.3.2 Tools and Firmware for Data Scrapping

R

R was used for web scraping [47, 73, 117] and data wrangling [92]. The choice was motivated by the fact that R is an open source platform for statistical machine learning [68] and visualisations. R is pedagogically simple, flexible, extensible and versatile; with an active community of developers contributing with various R packages. Meanwhile, it has limitations in scraping data from sensors, and this invited MicroPython.

MicroPython

This work deployed MicroPython [111], a reimplementation of Python 3 written in C, to scrap [47] data from sensors. It was preferred because of its simpler syntax compared to that of C/C++. C/C++ compile directly to CPU instructions while MicroPython is an interpreted language that can run directly on microcontroller boards without any compilations. Meanwhile, MicroPython can use more memory with less performance compared to C/C++, but the performance hit is not that significant in practical projects.

IDE

Several integrated development environment (IDE) can be used to program microcontrollers as well as in performing data analysis operations. These include: uPyCraft and Thorny IDE for sensor scraping. The following IDEs were selected for use in this work.

- ❑ **UPyCraft:** This is an easy-to-use IDE to program the ESP8266 microcontroller. It was used in this work to code and upload MicroPython onto the controller [65].
- ❑ **RStudio** (Version 1.1.463): This was used for statistical coding, web scraping [47, 73, 117], data analytics, and the design of an interactive dashboard using the Shiny package. Shapefiles of Botswana were created using the Raster package, then used to visualise the locations of the respective AWSs in Botswana.
- ❑ **NetBeans IDE 8.2:** To implement the SQLite code local data storage.

It should be noted that due to the limitation of sensors and microcontrollers, this experimental set up was used only for prototyping and illustration purposes. A decision was made therefore to make use of the data collected by the AWSs maintained by SASSCAL and BIUST. The reason was simply to use enough data with several weather variables adequate for real world applications of predictive analytics as articulated in the next sections.

3.4 Data Understanding and Preparation

Three data sources were explored in this work: the BIUST AWS, Shakawe AWS and Tubu AWS. In all the cases, the unit of analysis was a snapshot (instance) of an event of weather measured by weather sensors. Each instance (row) contained a unique weather measurement and the columns contained descriptions of the weather variables. There were more than enough data instances available for modelling. The variables for each observed weather event were recorded (see Table 4.1). The target variable was Maximum Air Temperature. Data preparation was done to examine the recorded values, identifying and imputing missing values. The TimeStamp variable was transformed to a date type consumable by R, Rattle and RWeka. A combination of two R functions was used in this task. The function `as.Date()` converts the TimeStamp to a date format; then the outcome was fed into R's `seq()` function to create a sequence of dates at a resolution of twenty-four hours.

3.5 Data Management System

SQLite was used as a standalone database management system (DBMS) since it is simple to setup and use; and does not require administration. Thus, SQL is suitable for IoT and embedded devices that minimise human-machine interaction in a SWS. All the commands were coded in NetBeans using Java. SQLite DBMS facilitates bundling diverse content into a self-contained and self-describing packages for transferring data across platforms. The current version of the toolkit implements three basic operations: Creating a database, inserting an observation and retrieving the desired weather instances.

3.6 Modelling: Predictive Analytics

Predictive analytics [22, 54] is an inter-disciplinary research that integrates different fields [41, p. 6] such as: cybernetics, mathematical optimization, statistical machine learning and other data driven technologies such as visualisation, analysis, knowledge discovery, predictions and dissemination. The integration of these tools and methods facilitate scientific understanding and the gaining of knowledge [25, p. 2], [25, p. 3] from data. Data-driven algorithms induce models from data by identifying the optimal topology of a given model.

Mathematical Problem Statement

Let $\mathbb{D}$ in [3.1] be a matrix with n covariates $x_{i,j} \in X$ and an endogenous variable $y_i \in Y$. These can be instances harvested using weather sensors discussed in section [2.2.1]:

$$\mathbb{D} = \{x_{i,j}, y_i | x_{i,j} \in \mathbb{R}^n \wedge y_i \in \mathbb{R}\}, \quad (3.1)$$

The goal is to construct a learner (model, approximator, function or a rule) to map X to Y using only the data at hand. The regression problem is thus defined as F :

$$F : \mathbb{R}^n \rightarrow \mathbb{R}. \quad (3.2)$$

This can be defined as a data driven model targeting at generating a set of objective functions (learners) ℓ [43, p. 21] in [3.3]: that replicate the best configuration for F :

$$\ell = F(\mathbb{D}|x, y) + \epsilon, \quad (3.3)$$

where ϵ is a noise (error) random variable (satisfying Equation [3.4]) to account for the fact that X may not contain enough information to perfectly predict Y :

$$\mathbb{E}[\epsilon|X] = 0. \quad (3.4)$$

The optimal learner ℓ is useful in inferences and predictions [34, p. 32] in the real world. In inferences, the objective is to understand the relationship between the predictor variables in $x_{i,j} \in X$ and the dependent $y_i \in Y$ with some error ϵ [57, p. 1], [113]. The assumption here is that, the more we know about the factors related to some target outcome, the better we can predict that target upon the presence of new data. In predictions, a function $F(\cdot)$ is induced to make predictions of the target Y using:

$$\hat{Y} = \hat{F}(\mathbb{D}|x, y) + \epsilon, \quad (3.5)$$

where $\hat{F}$ is the model to estimate for F , and $\hat{Y}$ is the predicted output for Y . Meanwhile, some scenarios integrate both inference and prediction into one. The choice of F may depend on the domain knowledge, experience or just the setting of the business problem. As articulated by Abbott [22, p. 13], predictive analytics has much in common with data mining, hence sharing the same algorithms and approaches. Gibert *et. al.* [36] presented a detailed guideline of data mining for environmental data science. Other researchers [70, 26, 42, 25, 55] have contributed with strategies of interpretability for better understanding in machine learning algorithms for data mining. The common loss functions for machine learning models are derived using calculus of variations; and they include: the mean squared error (*MSE*), mean absolute error (*MAE*) and the Root Mean Squared Error (*RMSE*); see [34, p. 70], [12, p. 40] and [37, p. 13]:

$$MSE = F^* = \operatorname{argmin}_F \mathbb{E}_{x,y} \mathbb{P}_{\mathbb{D}} \|y - F(x)\|^2, \quad (3.6)$$

$$MAE = F^* = \operatorname{argmin}_F \mathbb{E}_{x,y} \mathbb{P}_{\mathbb{D}} \|y - F(x)\|_1, \quad (3.7)$$

where y is the target value, while $F(x)$ is the predicted value. Possible predictive models include Neural Networks, Support Vector Machines, and Gaussian Processes. Meanwhile, KNN and Random Forests were chosen since they are interpretable.

Before embarking upon the proposed methods deployed in this work, it is very important to note that machine learning problems can be either solved using randomised algorithms like the Las Vegas (LV) and Monte Carlo (MC) Methods [94]. LV algorithms either return precise solutions or an acknowledgement of failure, while MC algorithms return approximate solutions with a random amount of error. LV algorithms are practically intricate to achieve, leaving a room for MC algorithms to be applied in most applications as seen in this work.

3.6.1 Monte Carlo Methods in Machine Learning

A MC estimate of a particular quantity of interest is formed by drawing samples from some probability distribution [93]. Sampling provides a flexible way to approximate many sums and integrals at reduced costs [93]. MC sampling is suitable in approximating sums or integrals that cannot be computed exactly (i.e., a sum with an exponential number of terms and with an unknown exact simplification). In such cases, the sum or an integral can be viewed as if it was an expectation under some distribution and then approximate the expectation by a corresponding average. In the case of a regression, we have the formula:

$$\ell = \int p(x)F(x)dx = \mathbb{E}_p[F(x)], \quad (3.8)$$

where p denotes a probability density over a random variable x . Thus, ℓ can be approximated by drawing n samples of x_i^n from p and forming the empirical average

$$\hat{\ell}_n = n^{-1} \sum_{i=1}^n F(x_i). \quad (3.9)$$

The approximation in [3.9] is justified by the fact that the estimator $\hat{\ell}$ is unbiased since

$$\mathbb{E}_p[\hat{\ell}_n] = n^{-1} \sum_{i=1}^n \mathbb{E}_p[F(x_i)] = n^{-1} \sum_{i=1}^n \ell = \ell. \quad (3.10)$$

Moreover, the average converges almost surely to the expected value if x_i are i.i.d (by LLN),

$$\lim_{n \rightarrow \infty} \hat{\ell}_n = \ell, \quad (3.11)$$

provided that the variance of the individual terms, $\sigma^2[F(x_i)]$, is bounded.

3.6.2 K-Nearest Neighbours (KNN)

This instance-based type of lazy learners [46, p. 128] deploys a *lookup table* of historical instances to model new ones [22, p. 254], [43, p. 33]. It is a local and interpretable model [70, p. 76] where a new instance $\hat{y}$ is modelled based on the outcome of the Function 3.12 (see [54, p. 264]) applied to k selected nearest neighbours $\mathbb{N}_k(x^*)$ ² to be considered around the data point of interest [22, p. 254]. The KNN is not sensitive to outliers.

$$\hat{y} = k^{-1} \sum_{x_i \in \mathbb{N}_k(x^*)}^k x_i. \quad (3.12)$$

Here, x is a set of regressors, and $\hat{y}$ is the vector of predicted variables. The task involves choosing the optimal k and the similarity measure for the distance from a data point to its neighbours. The Euclidean distance is commonly used [22, p. 258], [54, p. 238]. The value of k can be determined through an iterative search like starting with a small number of k , and evaluating the loss function while increasing k until the testing data error rate increases beyond the set threshold [22, p. 256]. The predictions generated by the models can be explained using the retrieved k -neighbours deployed for the prediction. This is simple with fewer predictor variables but it can be intricate when the dimensional data increases [70, p. 77]. KNN has been widely deployed in a variety of forecasting applications [96], [20].

3.6.3 Tree Based Models for Regression

Decision trees are non-parametric local estimators that predict the value of a target variable using the divide and conquer method [22, 46] to minimise the entropy (objective function):

$$H(Y|X) = \text{minimize}\{SSE = n^{-1} \sum_{j=1}^k \sum_{i=1}^{n_j} (y_i - \mu_j|_{\mu_j \in X_j})^2\}. \quad (3.13)$$

Given the dataset X with n observations, the idea is to identify the variable x which leads to the highest information gain when using it as a split criterion. Splits are performed along the dimensions with the most reduction of error within areas of the target space j conditioned on the predictors X [12], propagating down the leafs. The final value assignment is calculated as a weighted average of the leaf node values μ_k . The outer sum goes over all k

²A neighbourhood $\mathbb{N}_k(x^*)$ is an open ball of radius ϵ centred at the minimiser x^* , and it is defined as $\mathbb{N}_k(x^*) = \{x \in \mathbb{R}^n : ||x - x^*|| < \epsilon\}$ excluding the boundary of the ball.

data areas $\{Y_i, X_i\}$ of size m_j . The inner sum runs over squared differences between the target $y_i \forall X_i \in J$. Decision trees are easy to visualise and interpret when few trees are used. They can scale well as the number of variables increase [22]. They have a built-in variable selection for dimensionality reduction through variable importance measures.

3.6.4 Random Forest

A Random Forest (RF) model [9] is a robust non-parametric method that is considered to be an instance of Bagging [88, p. 2]; and can be implemented using Equation 3.14:

$$\hat{F}_{RF}(x) = n^{-1} \sum_{i=1}^n \ell_i(\mathbb{D}_i; x), \quad (3.14)$$

where the dataset $\mathbb{D}$ defined by x is the input to the model function ℓ . They are distributional free models that require no functional assumptions on the data variables [27, p. 1].

This approach optimizes predictive accuracy by fitting an ensemble of trees to stabilize model estimates. This is done by growing a set of different and uncorrelated standalone tree models then average predictions of all trees to assess the overall performance. Each tree is grown (using a set of n bootstrap samples) by recursively partitioning the population based on optimization of a split rule over the m -dimensional variable space. At each split, a subset of $\hat{m} \leq m$ candidate variables are tested for the split rule optimization, dividing each node into two daughter nodes, recursively, until the process reaches the stopping criteria (i.e., using node purity or node member size). The idea is to reduce variance σ^2 and improving the performance of the model [10, p. 17]; which ultimately avoids over-fitting.

3.7 Model Diagnostic Methods

3.7.1 Variable Importance Measures

Variable importance (VIP) [9] measures quantifies the global effect of input variables to the predictions generated by a model [85, p. 32]. They are computed by averaging the increase in squared errors ("%IncMSE") when the variable is permuted [41, p. 16]. VIP facilitate explanations based on the weight and magnitude of the modelling variables. The challenge with these measures is that they are tied to the error of the model [70, p. 102].

3.7.2 Partial Dependence Plots

Partial dependence plots (PDP) are graphical renderings that visualise the marginal effect of input-output relationships (i.e linear, monotonic or more complex) [32] in a reduced feature space [41, p. 16], [27, p. 14]. The function $F_S(x_s)$ for regression is estimated using the Monte Carlo method in section 3.6.1 [55, p. 4], and implemented by Greenwell [40]:

$$F_S(x_s) = \int p_c(x_c) \hat{F}(x_s, x_c) dx_c = \mathbb{E}_{x_c}[\hat{F}(x_s, x_c)], \quad (3.15)$$

$$p_c(x_c) = \int p(x) dx_s, \quad (3.16)$$

where F_S is the output from a ML model; $x_i^n = x_s \cup x_c$; x_s is a set of interest, and x_c is its compliment, while $p_c(x_c)$ is the marginal probability density of x_c . The assumption here is that x_s and x_c are uncorrelated. The set x_s usually contains one variable (for 2D plots) or two variables (for 3D plots). Roshanak *et. al.* [77] used PDP to identify the most influential variables in explaining the variability of hurricane-induced electric power outage durations.

3.8 Deployment

This is the final stage of the CRISP-DM methodology proposed in this research. The approach taken in here was that of using the developed application for visualisation and dissemination of weather data and model outputs to final consumers. A dashboard application was developed (using the *Shiny* package in RStudio) for dissemination of final outputs.

Chapter Summary

This chapter aimed at articulating the step-by-step activities towards the achievement of the research objectives. Data scraping was proposed to extract the weather data from the sources, while data wrangling was proposed to generate datasets for modelling. Statistical metrics were proposed to analyse and interpret the data before and after modelling. The proposed automated scripts are crucial for collection of real-time data from electronic sensors and weather data repositories. On the other hand, the non-parametric models have the advantage that they are simple and more flexible since they do not require assumptions about the data distributions, requiring less data preparation, hence saving time [22, p. 6]. Random Forest have built-in functions for feature engineering to facilitate dimensionality reduction. Random Forests also facilitate the functionality of partial dependence plots for machine learning interpretability.

Experimentations, Application Design and Results

Chapter Introduction

This chapter presents a detailed description of the experimental set up and procedures for harvesting the data used for predictive modelling. The data is collected from electronic sensors as well as from the SASSCAL website. The data collected from the BIUST AWS is used for comparison of multi-step ahead forecasting using time series for regression. The weather data collected from the SASSCAL repository is used for analysis of maximum air temperatures in the Okavango Delta. The experimental outputs are deployed using a dashboard application developed through this work.

4.1 Experimental Set-up

The main task here was to scrap the data from data sources such as: websites or sensors, then feed it into data mining workbenches for predictive data analytics. Figure 2.1 presents the IoT AWS pipe line. The general flow of weather data from the experimental set-up consisted of: sensors (BME280 and DHT11), breadboard, resistors, jumpers and the ESP8266 MCU. The first task was to flash the firmware on to the ESP8266 microcontroller in Figure 4.1 for programming with MicroPython. The back-end layer is used for data storage, processing, predictive modelling and dissemination of end products to consumers.

Figure 4.1 shows the *GPIO* maps for the ESP8266 module; a low-powered computer system-on chip integrated with a 32-bit processor. The ESP8266 was selected for prototyping since it is a simple-to-use and ready made board with low-costs, suitable for prototyping in smart weather systems. It has 64 KB of RAM and a 64 KB boot ROM, and a data RAM of 96 KB. It comprises of input/output pins to attach external electronics components like sensors, relays and resistors for control and management of other electronics devices.

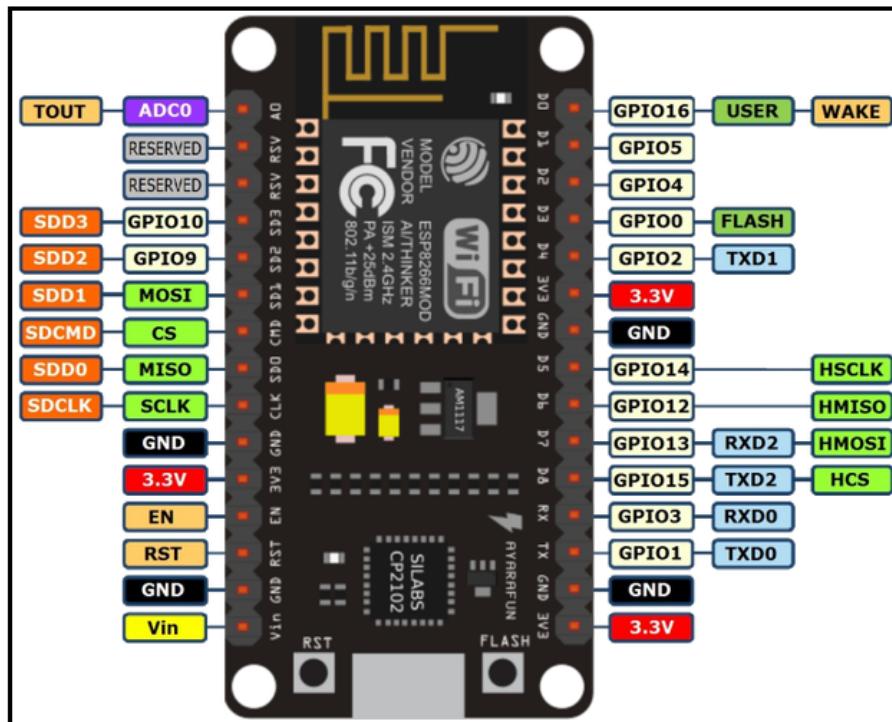


Figure 4.1: The ESP8266 module used for prototyping.

To make it work, one just need to connect the some wires: the GND and VCC from the sensor to GND and 3.3V in ESP8266. The BMP180 and DHT11 use 3.3-5V, so they were connected to the 3.3 V pins on the ESP8266 board. Though the proposed sensors measure temperature, humidity, and barometric pressure, this work only focused on temperature and humidity, since most of the illustrations are constrained to analysing and predicting maximum air temperature. The uPyCraft software (see Figure 4.3) was used for microprogramming since it simple and easy to learn; with a friendly graphical user interface. It should be noted that the most motive behind this project was to investigate the tools that can facilitate the extraction of data from electronic sensors. The idea is that once mastered, the techniques can be transferred to any application of mechatronics and instrumentation. For instance, industrial plants and scientific laboratories deploy thermocouples and infrared temperature sensors to capture light or temperature given off by certain objects.

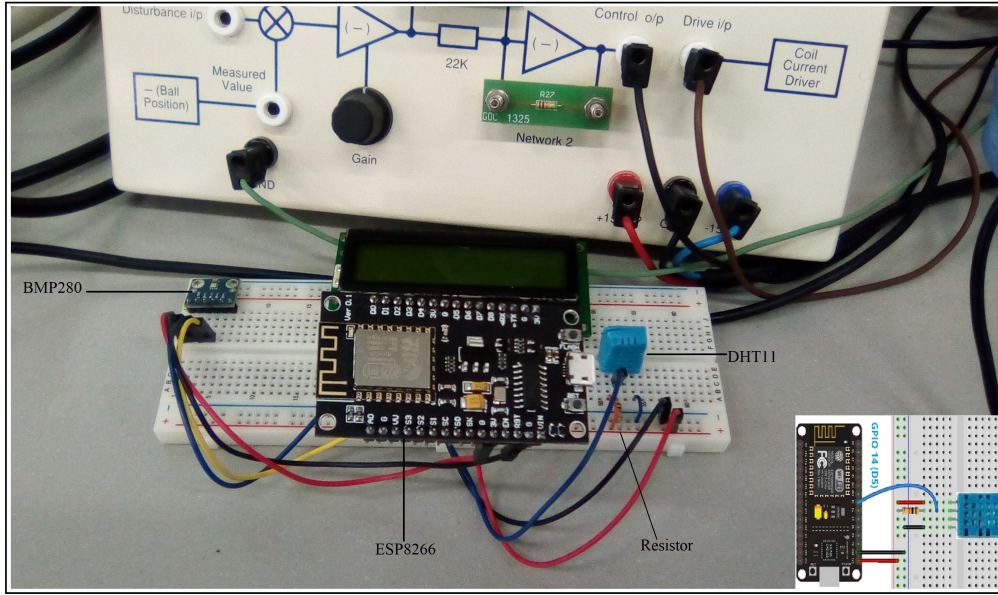


Figure 4.2: The proposed prototype to collect temperature and relative humidity.

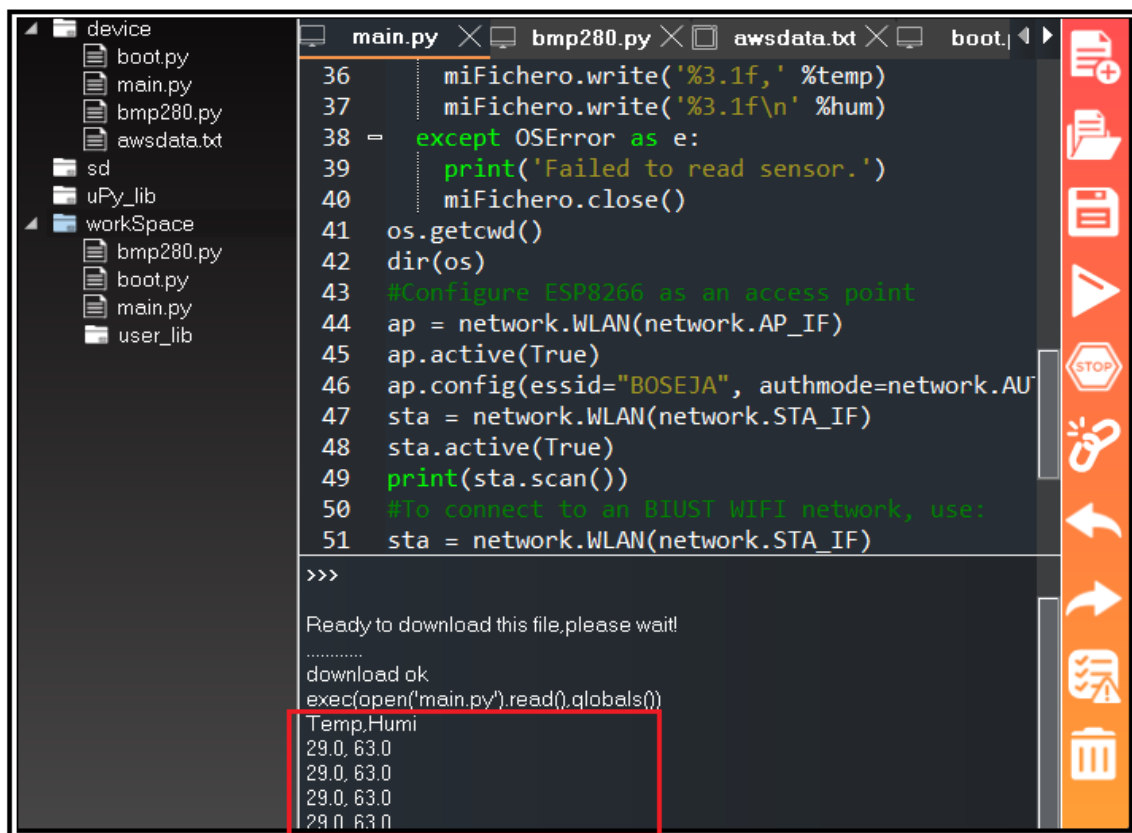
4.1.1 Sensor Data Scraping

The uPyCraft IDE (see Figure 4.3) was used to flash the firmware to the ESP8266 board. Sensors were connected to the breadboard using jumpers. This work deployed the DHT11 sensor with four pins. Therefore, a pull-up resistor ($10K\Omega$) was connected between the VCC pin and the data line to connect the unused input pin to keep the VCC high. Likewise, the BMP280 sensor was firstly soldered, then connected to the breadboard as well.

Algorithm 1 was used to implement the MicroPython code for data scraping. The class `Pin` was imported from the `machine` library (line 1), a low-level abstraction containing functions related to the hardware (i.e., timers, communication protocols, CPU, etc.,) to define the pins to be used for connections. The `sleep` method (line 7) was imported from the `time` module to facilitate configuration of time delays to sample the data. The `dht` and `bmp` modules were imported (line 2 and 3) to be used by the functions that read data from the DHT11 and BMP280 sensors. The `dhtSensor` object was created to read the data and transmit it through the `GPIO 14` pin on the ESP8266 module. The functions in line 12-13 were used to read temperature and relative humidity. The values were written onto a file (as seen in Figure 4.3) to be exported for further analysis. The algorithm was then uploaded to the ESP8266 which was then reset to ensure that it promptly boots up from the external flash. This allowed the measured weather data to be saved at every configured time intervals.

Algorithm 1: Sensor data scraping using MicroPython

```
/* Set the path to the webpage file */6 while True: do
1 from machine import Pin /* define pins */7   Try:
2 from time import sleep /* for sampling */8     sleep(delay)
3 import dht /* invoke the dht driver */9     dhtSensor = dht.DHT22(Pin(14))
4 import bmp /* invoke the bmp driver */10    dhtSensor.measure()
5 delay = 15                               11    bmpSensor.measure()
/* Note that if the sampling rate is high,12    temp = sensor.temperature()
more data will be measured. This might13    hum = sensor.humidity()
not be ideal in some cases, for        14    miFichero.write('%3.1f, %temp)
instance, if the interval is small,    15    miFichero.write('%3.1f' %hum)
most of the measurements will be equal 16
as seen in Fig 4.3.                    */17 except OSError as e:
                                         print('Failed to read sensor.')
```



```
device
├── boot.py
├── main.py
├── bmp280.py
├── awsdata.txt
├── sd
├── uPy_lib
└── workspace
    ├── bmp280.py
    ├── boot.py
    ├── main.py
    └── user_lib
```

```
main.py X bmp280.py X awsdata.txt X boot.py
36 miFichero.write('%3.1f, ' %temp)
37 miFichero.write('%3.1f\n' %hum)
38 except OSError as e:
39     print('Failed to read sensor.')
40     miFichero.close()
41 os.getcwd()
42 dir(os)
43 #Configure ESP8266 as an access point
44 ap = network.WLAN(network.AP_IF)
45 ap.active(True)
46 ap.config(essid="BOSEJA", authmode=network.AUTH_WPA_WPA2_PSK)
47 sta = network.WLAN(network.STA_IF)
48 sta.active(True)
49 print(sta.scan())
50 #To connect to an BLUETOOTH network, use:
51 sta = network.WLAN(network.STA_IF)

>>>
Ready to download this file, please wait!
.....
download ok
exec(open('main.py').read(), globals())
Temp, Humi
29.0, 63.0
29.0, 63.0
29.0, 63.0
29.0, 63.0
```

Figure 4.3: The uPyCraft environment displaying the measured values.

4.1.2 Web Scraping

The following algorithm was designed to automatically scrape data from the SASSCAL weathernet for a given automatic weather station. The user can specify the desired location and the desired temporal resolution (hourly, daily or monthly); all of which are dictated by the file path and the index of the table containing the weather observations.

Algorithm 2: Web scraping using R scripting

```
/* Set the path to the webpage file */  
Input: File Path: of type String  
Output: A matrix of climatological values  
*/1 while (FilePath is Valid) do  
2   miCamino = "FilePath";  
   /* Read the HTML page */  
3   mistablas ←  
       readHTMLTable(miCamino)  
       /* Select the desired table */  
4   miFichero ← mistablas[Indexo]  
5 return miFichero;      // Return the table
```

The argument to the algorithm is the path to the file to be scrapped. The source is read using the *readHTMLTable* to scrape all the tables containing the meteorological values in that file into a temporary file *mistablas* on a local host for further processing. The final task involves manoeuvring through the various tables contained in *mistablas* to select the desired matrix *miFichero* of weather elements to be used for modelling. In case of a file located on a local disk, the *read.table()* or *read.csv()* function can be used to load a csv or an excel file. The resultant object is converted to a data-frame using the *as.data.frame()*. The next step will be to explore the dataset to get insights about the imperfection of data.

4.2 Weather Monitoring: Application Design

This work deployed the Shiny package [13] to develop a dashboard App in RStudio (see Figures 4.4 and 4.5) and turn statistical analysis into interactive web App [15, p. 22] using Algorithm 3. A Graphical User Interface (GUI) was created using the *fluidPage()* function to assemble some HTML functionalities for the App. The App's functionalities are instantiated via the dashboard page. It deploys an AdminLite as a theme built on top of a Bootstrap grid layout that can be occupied by HTML elements. A *dashboardSidebar()* function facilitates a quick navigation through the functionalities of the developed App such as: visualisation of data, weather analytics, weather forecasting and the generation of weather reports.

Algorithm 3: For the weather dashboard. The file path FP is used to fetch and parse the data into a table (Tablas), containing the modelling dataset y_i .

Input: Data set path, n, BS,h,LGS,k, y_i
Result: Interface for analytics and visualisation

```

1 while Interactive do
2   gui ← fluidPage( miCamino ← "FP"
3   Tablas ← parseTheTables(miCamino)
4   miFichero ← Tablas[Indexo]
5   myTitle← dashboardHeader(...),
6   mySideBar← dashboardSidebar(...),
7   myBody←dashboardBody(
8     fluidRow(...) );
9   server ← function(input,output) {
10    X = Actual← tail(MAXTMP[(n-(LB)):n],h)
11    ModelKNN ←  $\hat{y} = \frac{1}{k} \sum_{i=1}^k x_i$ 
12    Y = PredictedV ←
13      ModelKNN$prediction
14    dif ← abs(X-Y); pos ← dif <= 2;
15    dif[pos];
16    %Accuracy ←
17      ((size(dif[pos]))/size(dif))*100;
18    myPlots ←
19      PlotFunction(Preds,LosFunc,...) };
20    shinyApp(gui, server);
21 end

```

The *dashboardBody()* function was used to design the dashboard page to contain among others: regular Shiny output contents such as the tables containing the weather data, scatter plots and the summary statistics generated via the predictive models. The *server()* function constructs a list of R objects to manipulate all the outputs to be displayed by the dashboard App. The vector variable X contains the actual time series of the maximum air temperature. This variable is to be used for uni-variate time series forecasting using the KNN ($Model_{KNN}$), and the predictions are stored in vector Y. The next instructions determine the accuracy regarding the predictive performance as articulated below:

$$\%Accuracy = \frac{Total_{HITS}}{Total_{Predictions}} * 100. \quad (4.1)$$

Thus, the accuracy percentage was computed as the ratio of correctly predicted observations ($\leq \pm 2$) for the summer season multiplied by 100. Finally, the *shinyApp* function combines the GUI and Server into a functional dashboard App in Figure 4.4. The App can then be used by end users to access or visualise weather outputs as seen in Figure 4.4.

The raster package was used in conjunction with the maptools package to manipulate shape-files and generate geographic data like the Botswana map depicted in Figure 4.4.

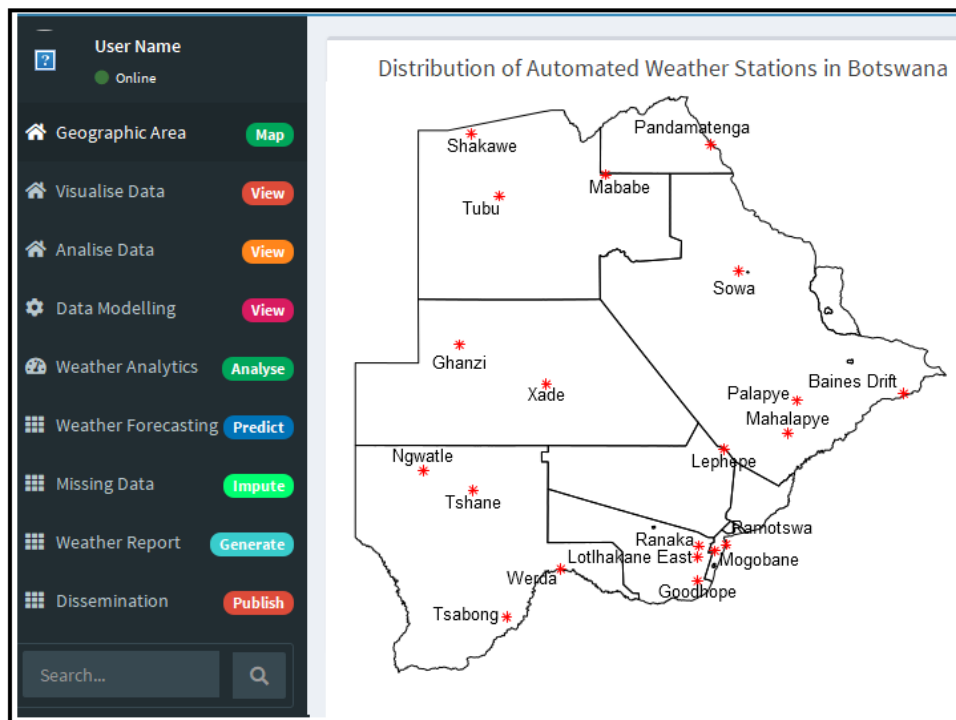


Figure 4.4: Visualising the distribution of AWSs in Botswana using the developed App.

Lets take the BIUST AWS¹ for instance, and run it through the proposed methodology.

The BIUST AWS

The BIUST AWS is located at 22.5946° S, 27.1233° E in Palapye, Botswana (see Figure 1.1). It was installed and commissioned in August 2016, with sensors polling data using the XConnect software at hourly intervals (can be reconfigured to minutes). The harvested data include (see Table 4.1): minimum and maximum (wind speed in m/s , relative humidity in %, air temperature in $^{\circ}C$), average wind direction in $^{\circ}$, precipitation (mm), Barometric Pressure ($mBar$), Global Horizontal Irradiation $1HR$ (W/m^2), Global Horizontal Irradiation $24HR$ in W/m^2 , Solar Radiation, and Total Suspended Particulate. The modelling data used for this particular experiment ranges from 12-01-2018 to 28-01-2018 at intervals of 60 minutes, with 425 instances (observations).

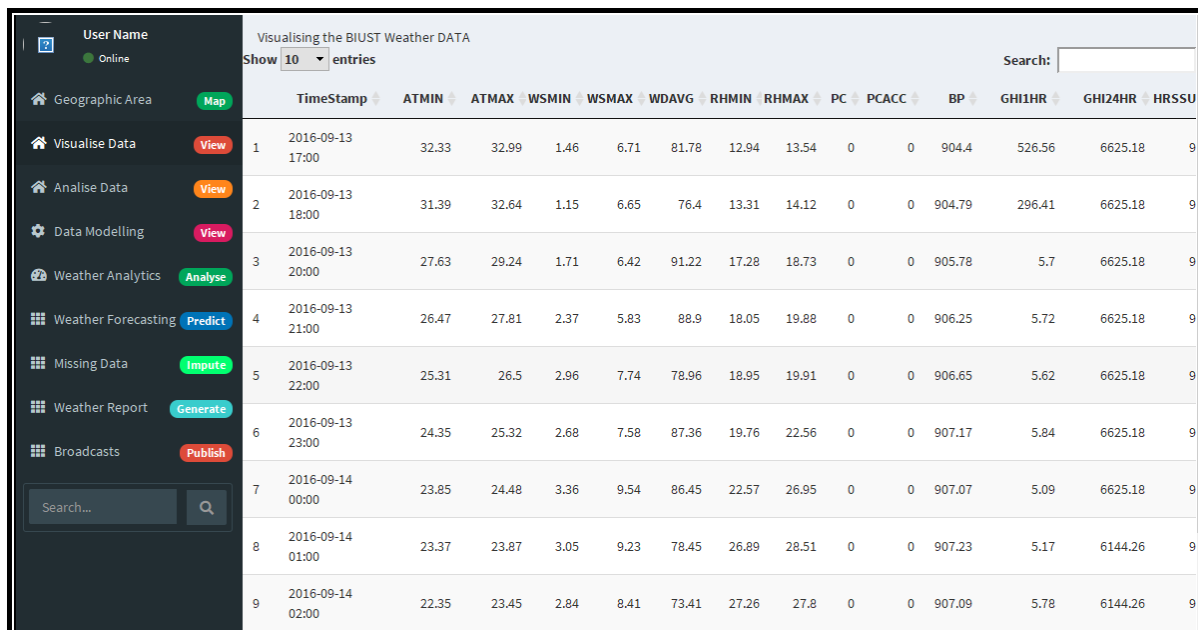
¹The image of this AWS also appears on the cover page of this thesis.

Notations for the measurements collected from the BIUST AWS are shown in Table 4.1.

Table 4.1: Variables contained in the collected data set the BIUST AWS.

Measurement	TEMPERATURE			WIND SPEED		
Level	Maximum	Minimum	Average	Maximum	Minimum	Average
Notation	ATMAX	ATMIN		WSMAX	WSMIN	
Measurement	RELATIVE HUMIDITY			PRECIPITATION		
Level	Maximum	Minimum	Average	Hourly	Daily	Average
Notation	HMAX	HMIN		PC	PCAA	
Measurement	ATMOSPHERIC PRESSURE			IRRADIATION		
Level	Maximum	Minimum	Average	Hourly	Daily	
Notation			BP	GH1HR	GH24HR	
Measurement	Total Suspended Particulate			Solar Radiation		
		TSP			SOLRAD	
Measurement	BATTERY			AVERAGE WIND DIRECTION		
		BATT			WDAVG	

The data can be loaded and visualised using the dashboard App as seen in Figure 4.5.



User Name		Visualising the BIUST Weather DATA												
Online		Show 10 entries	Search:											
	TimeStamp	ATMIN	ATMAX	WSMIN	WSMAX	WDAVG	RHMIN	RHMAX	PC	PCACC	BP	GH1HR	GH24HR	HRSSU
1	2016-09-13 17:00	32.33	32.99	1.46	6.71	81.78	12.94	13.54	0	0	904.4	526.56	6625.18	9
2	2016-09-13 18:00	31.39	32.64	1.15	6.65	76.4	13.31	14.12	0	0	904.79	296.41	6625.18	9
3	2016-09-13 20:00	27.63	29.24	1.71	6.42	91.22	17.28	18.73	0	0	905.78	5.7	6625.18	9
4	2016-09-13 21:00	26.47	27.81	2.37	5.83	88.9	18.05	19.88	0	0	906.25	5.72	6625.18	9
5	2016-09-13 22:00	25.31	26.5	2.96	7.74	78.96	18.95	19.91	0	0	906.65	5.62	6625.18	9
6	2016-09-13 23:00	24.35	25.32	2.68	7.58	87.36	19.76	22.56	0	0	907.17	5.84	6625.18	9
7	2016-09-14 00:00	23.85	24.48	3.36	9.54	86.45	22.57	26.95	0	0	907.07	5.09	6625.18	9
8	2016-09-14 01:00	23.37	23.87	3.05	9.23	78.45	26.89	28.51	0	0	907.23	5.17	6144.26	9
9	2016-09-14 02:00	22.35	23.45	2.84	8.41	73.41	27.26	27.8	0	0	907.09	5.78	6144.26	9

Figure 4.5: Visualising the BIUST AWS data using the developed dashboard App.

This is crucial for meteorological service providers as well as other end users that might be interested in monitoring weather, or those interested in using the historical weather data that is stored in a database. In essence, the implemented functionality can be put into action to disseminate AWS weather data to end users; thus bridging the gap experienced by various organisations like the BIUST and BDMS to meet the demands of their clients. Section 4.3 in continuation deploys the BIUST data for univariate temperature time series.

4.3 Temperature Time Series Forecasting using KNN

As highlighted in section 1.1 of the introduction, the BIUST AWS lacks the backend layer. This section articulates the proof of concept regarding the use of these data for weather forecasting. The experiments deploy a methodology developed by Martínez *et. al.* [59] for regression on a time series forecasting using KNN (see Equation 3.12). The work compares two approaches for multi-step ahead forecasting. The Euclidean distance [70, p. 76] was used as the similarity measure. More details are documented in [108, 8, 107, 109].

1. The *Recursive Multi-Step Ahead* strategy in Equations 4.2 and 4.3 iteratively uses a one-step ahead model multiple times to make future projections [1, p. 79]. The forecast for the prior time step is input to the next time step and the process repeats in each step:

$$F_{t+1} = F(x_{t-1}, x_{t-2}, \dots, x_{t-n}), \quad (4.2)$$

$$F_{t+2} = F(F_{t+1}, x_{t-1}, x_{t-2}, \dots, x_{t-(n+1)}), \quad (4.3)$$

where F_{t+i} are forecasts. This strategy has the drawback of propagating errors from one time step to another. Furthermore, it can be biased for nonlinear systems [108, p. 3] and the model performance can quickly degrade as the forecasting horizon increases [1, p. 79].

2. The *Multi-Input Multi-Output (MIMO)* strategy in Equation 4.4 develops one model that is capable of generating the entire forecast sequence at a go [109, p. 5], [1, p. 72] as:

$$F_{t+1}, F_{t+2}, \dots = F(x_{t-1}, x_{t-2}, \dots, x_{t-n}). \quad (4.4)$$

This strategy is characterized by the use of a vector of target values with length equal to the number of horizons to make forecasts using the same model structure. This particular constraint diminishes the flexibility and the consistency of the single-output approaches [109, p. 5] which might lead to possibility of biasing the final performance of the model. The value of k was determined through an iterative search starting with $k = 2$, and evaluating the loss function while increasing k to 62 until the testing error rate increases beyond the set threshold [22, p. 256]. The proposed strategies were compared in making multi-step ahead forecasts at different forecasting horizons (18, 20, 22, 24, 48, 72, 96 and 120).

4.3.1 Results on Temperature Time Series Forecasting

The models were evaluated using different batch sizes in hours: 48, 72, 80, 100, through 240 as seen in the first column of Table 4.2, where: BS is the batch size, LGS is the number of lags, while K is the number of nearest neighbours to be used for predictions. Several experiments were conducted in a trial and error fashion for each strategy. The MAE and Correlation Coefficient (r) were recorded to evaluate the different configurations.

Table 4.2: Predictive performance of **MIMO** vs **Recursive** multi-step ahead strategy.

Config	Horizon Size: 18 Hours		Horizon Size: 20 Hours		Horizon Size: 22 Hours		Horizon Size: 24 Hours	
[BS LGS K]	MAE	r	MAE	r	MAE	r	MAE	r
[48 20 2]	[2.53] [2.88]	[0.44] [0.32]	[1.70] [2.18]	[0.70] [0.56]	[0.42] [1.13]	[0.98] [0.86]	[0.42] [0.40]	[0.98] [0.98]
[72 24 3]	[2.70] [2.82]	[0.39] [0.31]	[1.92] [2.03]	[0.65] [0.61]	[0.75] [0.81]	[0.94] [0.92]	[0.76] [0.63]	[0.93] [0.94]
[80 20 2]	[2.53] [2.88]	[0.44] [0.32]	[1.70] [2.18]	[0.70] [0.56]	[0.42] [1.23]	[0.98] [0.86]	[0.42] [0.40]	[0.98] [0.98]
[100 20 2]	[2.53] [2.88]	[0.44] [0.32]	[1.70] [2.18]	[0.70] [0.56]	[0.42] [1.23]	[0.98] [0.86]	[0.42] [0.40]	[0.98] [0.98]
[120 62 2]	[3.24] [2.98]	[0.19] [0.23]	[2.56] [2.29]	[0.43] [0.51]	[1.68] [1.18]	[0.70] [0.84]	[1.02] [0.40]	[0.97] [0.98]

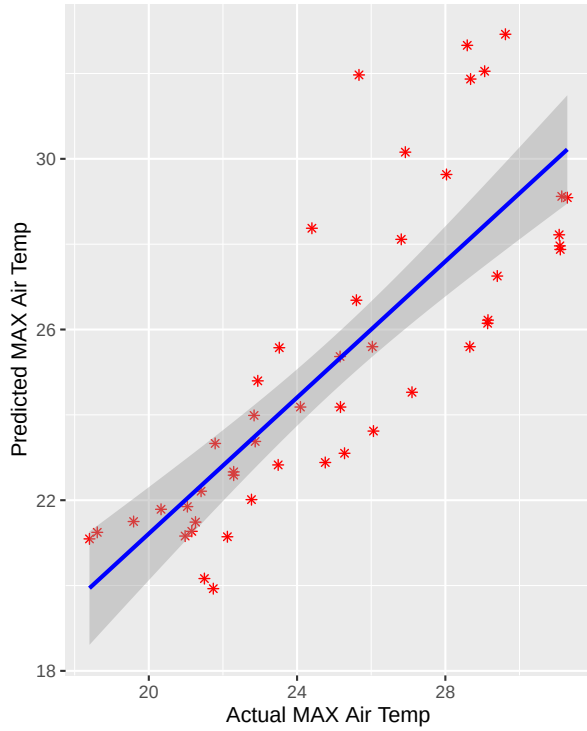
Config	Horizon Size: 48 Hours		Horizon Size: 72 Hours		Horizon Size: 96 Hours		Horizon Size: 120 Hours	
[BS LGS K]	MAE	r	MAE	r	MAE	r	MAE	r
[144 20 3]	[2.18] [0.99]	[0.85] [0.95]	[2.11] [1.62]	[0.86] [0.93]	[1.03] [1.75]	[0.95] [0.91]	[1.02] [1.81]	[0.96] [0.88]
[168 40 3]	[1.94] [1.24]	[0.93] [0.92]	[0.94] [1.73]	[0.96] [0.90]	[0.99] [1.80]	[0.96] [0.91]	[1.02] [1.82]	[0.96] [0.90]
[192 62 3]	[0.66] [1.89]	[0.99] [0.80]	[0.75] [2.04]	[0.98] [0.79]	[0.80] [1.20]	[0.98] [0.93]	[1.02] [1.37]	[0.96] [0.93]
[216 60 8]	[1.44] [1.66]	[0.95] [0.93]	[2.61] [3.02]	[0.84] [0.92]	[2.17] [3.01]	[0.84] [0.85]	[1.50] [2.53]	[0.92] [0.84]
[240 24 6]	[1.70] [1.15]	[0.94] [0.93]	[2.66] [1.74]	[0.93] [0.89]	[2.70] [1.95]	[0.90] [0.88]	[2.83] [2.14]	[0.77] [0.84]

The first column of Table 4.2 shows that the MAE is high between 2.53°C and 3.24°C at initial phase when forecasting 18 hours in advance using the batch size ranging from 48 to 120 hours. The MAE suddenly reduces when increasing the forecasting horizon, while the r increases to show confidence in the model predictions. Increasing the number of batches for the look-up table seems to yield desirable error. Meanwhile, there are no significant improvements regarding both MAE and r statistics when making multi-step ahead forecasts up to 24 hours of horizon size and beyond that. Equation 4.1 was used to score the respective configurations for each forecasting horizon and results are shown in Table 4.3.

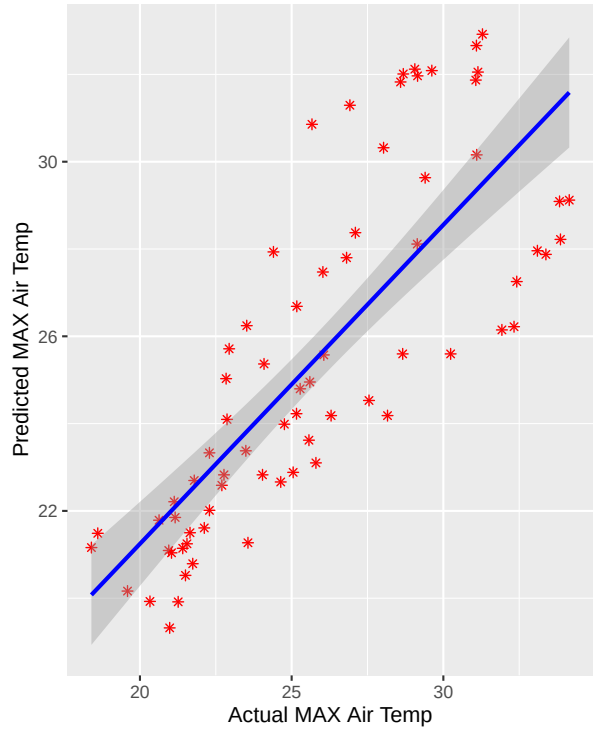
Table 4.3: Accuracy (in %) of the strategies for various configurations of **MIMO** (M_i) and **Recursive** (R_i) evaluated for the horizons: 18, 20, 22, 24, 48, 72, 96 and 120 hours.

Horizon	M_1, R_1	M_2, R_2	M_3, R_3	M_4, R_4	M_5, R_5	Horizon	M_6, R_6	M_7, R_7	M_8, R_8	M_9, R_9	M_{10}, R_{10}
18	50,50	50,50	50,50	50,39	50,50	48	48,81	58,69	98,54	75,35	60,83
20	70,55	70,60	70,55	70,55	70,55	72	44,63	93,61	97,58	42,29	33,68
22	100,82	91,86	100,82	100,82	100,82	96	89,60	86,58	96,80	53,27	38,56
24	100,60	92,58	100,80	100,27	96,56	120	89,59	89,58	89,75	68,42	45,50
Average	80,62	76,64	80,67	80,54	79,61	Average	68,66	82,62	95,67	60,33	44,64

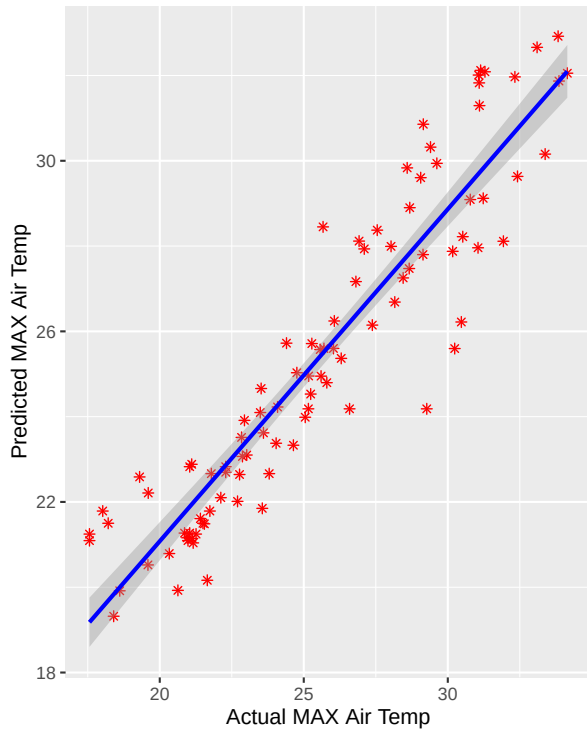
Horizon = 48 hours, $R = 0.80$, $y = 0.798x + 5.26$



Horizon = 72 hours, $R = 0.79$, $y = 0.731x + 6.63$



Horizon = 96 hours, $R = 0.93$, $y = 0.779x + 5.51$



Horizon = 120 hours, $R = 0.93$, $y = 0.739x + 6.7$

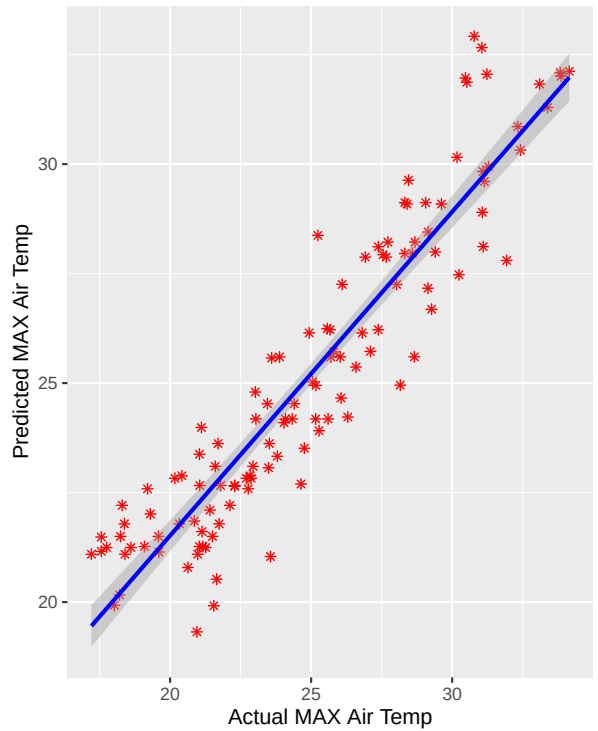


Figure 4.6: Scatter plots and Pearson's correlation coefficients illustrating the linear relationship between the actual values of maximum air temperature and the predicted maximum air temperature using the Recursive strategy.

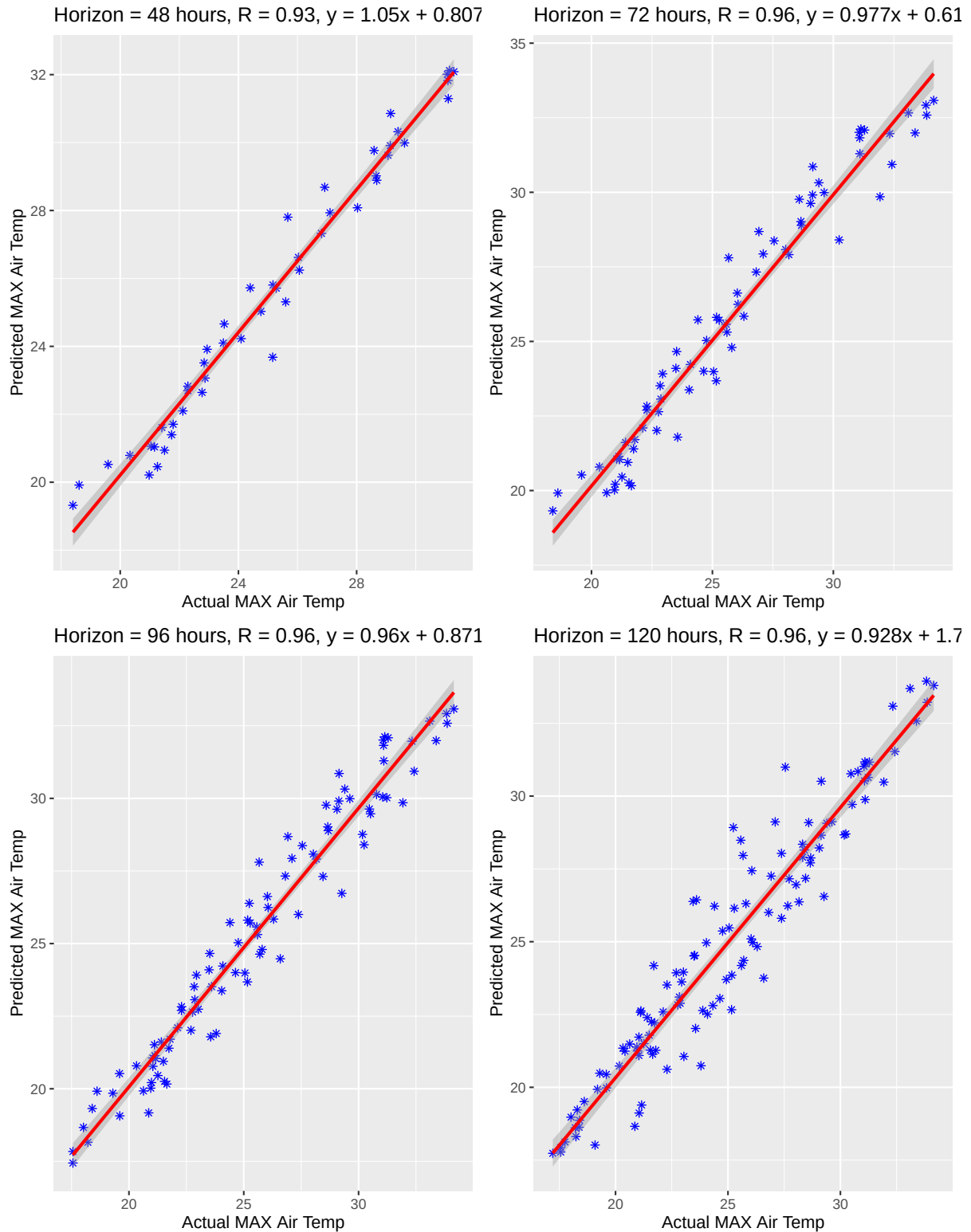


Figure 4.7: Scatter plots and Pearson's correlation coefficients illustrating the linear relationship between the actual values of maximum air temperature and the predicted maximum air temperature using the MIMO strategy.

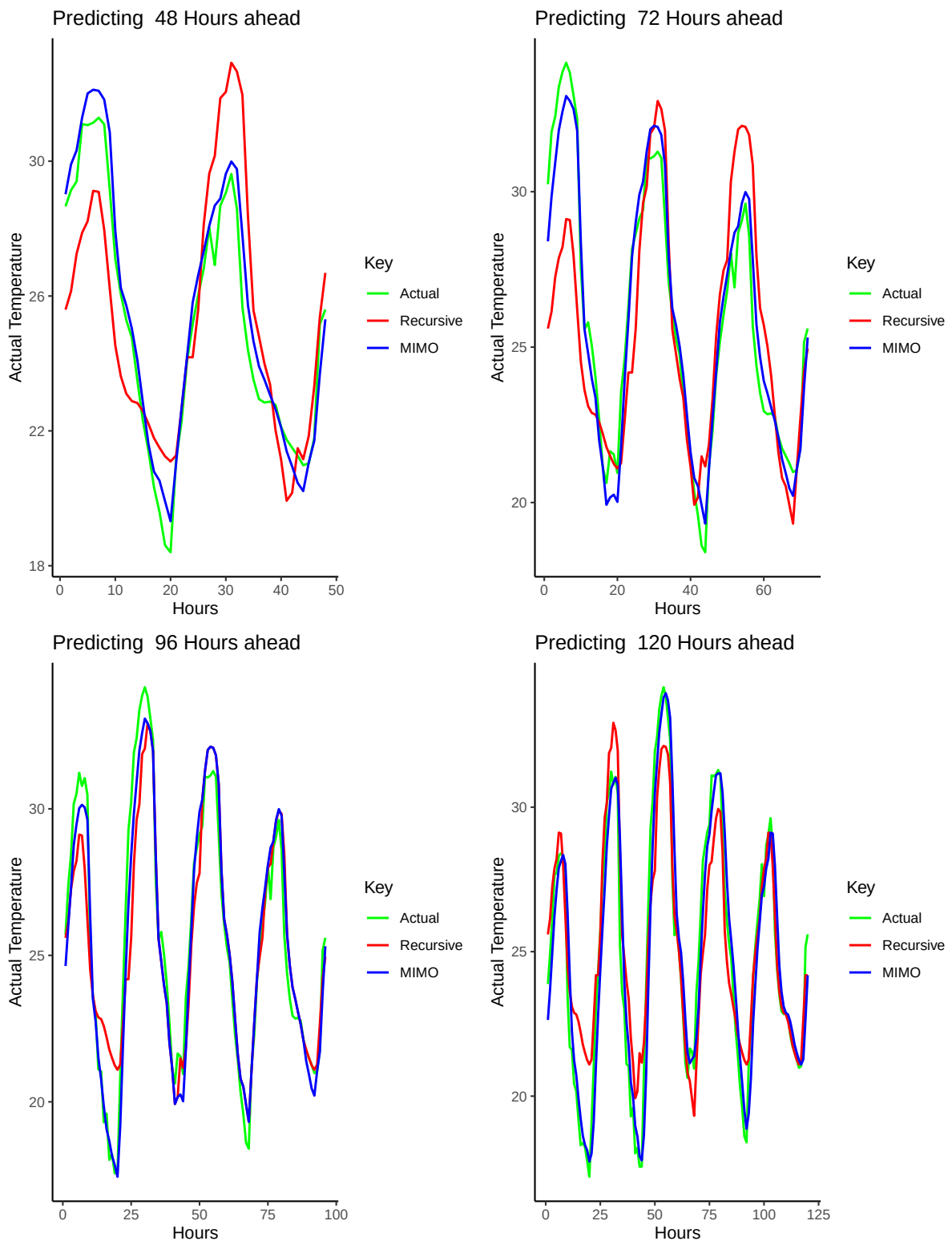


Figure 4.8: Graphs showing the multi step ahead forecasts using the MIMO (blue) and Recursive (red) strategies with some perfect fit by both strategies. The blue trend depicts the MIMO; the red one depicts the Recursive, and the green trend for the observed values.

Sub-discussions on Temperature Time Series

One of the tasks of this work was to embed predictive models into a dashboard application (see Figures 4.4 and 4.5) for dissemination of weather products to make use of the data generated by the automated weather stations. Thus, this section articulated that proof of concept for deployment of predictive models, which is the last phase of the CRISP-DM process [103] and hence predictive analytics as articulated by Dean in Chapter 12 [22]. A decision was made to use a non-parametric algorithm (KNN) for weather forecasting. Two strategies were compared for multi-step ahead forecasting of maximum air temperature, which are: MIMO and the Recursive strategy.

The predictions for the horizons below 20 hours were unstable and not within the acceptable threshold of ± 2 (see column 1 of the upper part of Table 4.2) for January which is summer. Meanwhile, the predictions stabilised from around 22 hours of forecasting horizon, with a reduction in MAE and an increase in r . The MIMO outperformed the Recursive strategy for most of the forecasting horizons in terms of MAE and correlation coefficient. These can be seen well in Table 4.3, which displays the predictive performance expressed in terms of percentages using Equation 4.1.

The MAE, r and the average (AVG) percentage of accuracy all showed that the best mutli-step ahead forecasting strategy was MIMO, which achieved a total average of 74%; while the Recursive strategy achieved a total percentage average of 65%. The 3rd configuration gave the best results for predictions in the forecasting horizon range from 18 to 24 achieving scores of 80% and 67% for both MIMO and Recursive respectively. The 8th configuration gave the best results in the forecasting horizon range from 48 to 120 hours, with scores of 95% for MIMO and 67% for Recursive.

The scatter plots and Pearson's correlation coefficients agree (i.e., $r > 0.75$) with the linear relationship between the actual values and the predicted values of maximum air temperature using KNN for both MIMO (Figure 4.7) and the Recursive strategy (Figure 4.6). The plots are labelled with the horizon in hours, the correlation coefficient, and the equation for the best line of fit respectively. Figure 4.8 presents graphs showing a typical configuration (BS = 192, LGS = 62, and K = 3) using the MIMO and Recursive strategies with some perfect fit by both strategies. The blue trend depicts the MIMO; the red one represent the Recursive strategy, and the green for observed values. All these demonstrate the value added by IoTs and predictive analytics (data scraping, data wrangling and data analysis) for weather nowcasting.

On a different note, it was observed that when dealing with batch sizes containing more instances, the results in this case are good, with lower MAE and higher r , demonstrating that the KNN model is able to mimic the trend of the maximum air temperature sufficiently, especially when taking K = 3, and using 62 lags. It was noted that the predictive performance for both model configurations improved very much as the batch size increases. Another interesting point noted here is that the performance also increased when increasing the forecasting horizon beyond 18 hours. Meanwhile, the performance deteriorated beyond 120 hours for both strategies which can be attributed to the chaotic nature of the weather system at finer temporal resolutions.

4.4 Enhancing Access and Making Use of SASSCAL's Data

In 2012, the German Federal Ministry of Education and Research collaborated with five southern African countries (Angola, Botswana, Namibia, South Africa and Zambia) to initiate SASSCAL: The Southern African Science Service Centre for Climate and Land Management. The main goal was to foster regional networking for scientific research, innovation and knowledge generation to mitigate challenges brought by climate variability to the local people. SASSCAL therefore became a service provider to meet the demand of various key stakeholders. These include: researchers, farmers, students and other clients who need advisory services, information support, innovation management and capacity building to support sustainable economic development in the southern African region.

To achieve its goals, SASSCAL installed AWSs for environmental monitoring in the region. This was complimented by the development of an online platform² to disseminate data to end users (see Figure 4.9). This is a significant contribution, considering the value of weather observations in weather forecasting and sectors such as: agriculture, water management and energy. However, efficacious access to this data is not trivial.

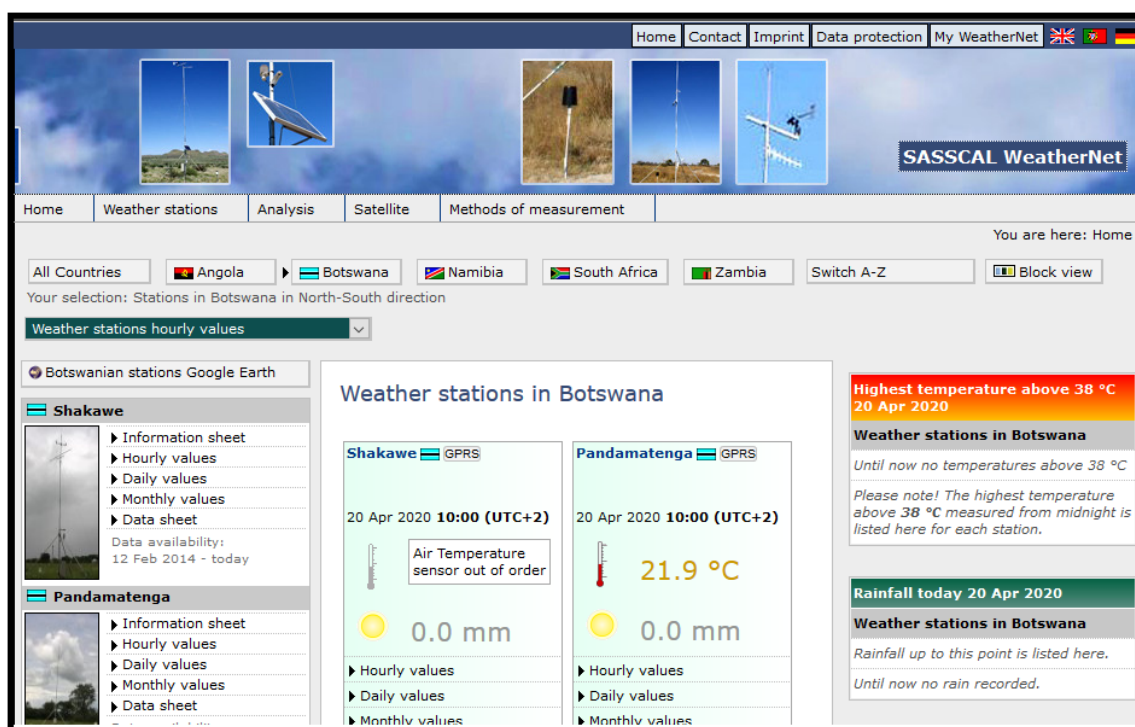


Figure 4.9: The SASSCAL platform for dissemination of weather data

²www.sasscalweathernet.org

Most of the end users find the Earth Observation (EO) data inaccessible and highly technical [64, p. 28] [110, p. 15] [84, p. 21]. This is a crucial barrier in data science and modelling. In essence, no data driven project can start without data [22, p. 14]. This section thus articulates the process of tackling such a limitation using techniques of predictive analytics to enhance access and make use of the data from the SASSCAL weathernet.

4.4.1 Study Area and Modelling Datasets

The modelling data was collected from two weather stations: Shakawe (WMO No.: 68026, Station ID: 68026) and Tubu (WMO No.: 68027, Station ID: 67591) located at the Okavango Delta [2] in Botswana. The Okavango Delta was selected due to its interesting nature; it is a largely vegetated water resource sustaining several livelihood strategies such as tourism, agriculture, fishing and collection of veld products. Previous studies have reported high temperatures in the north west region of Botswana [79, 71] compared to other regions of the country. Silitshena *et. al.* [89] supported this fact basing closeness of the northern regions to the equator. Meanwhile none of these studies linked the occurrence of high summer temperatures to other weather variables observed in the Okavango Delta.

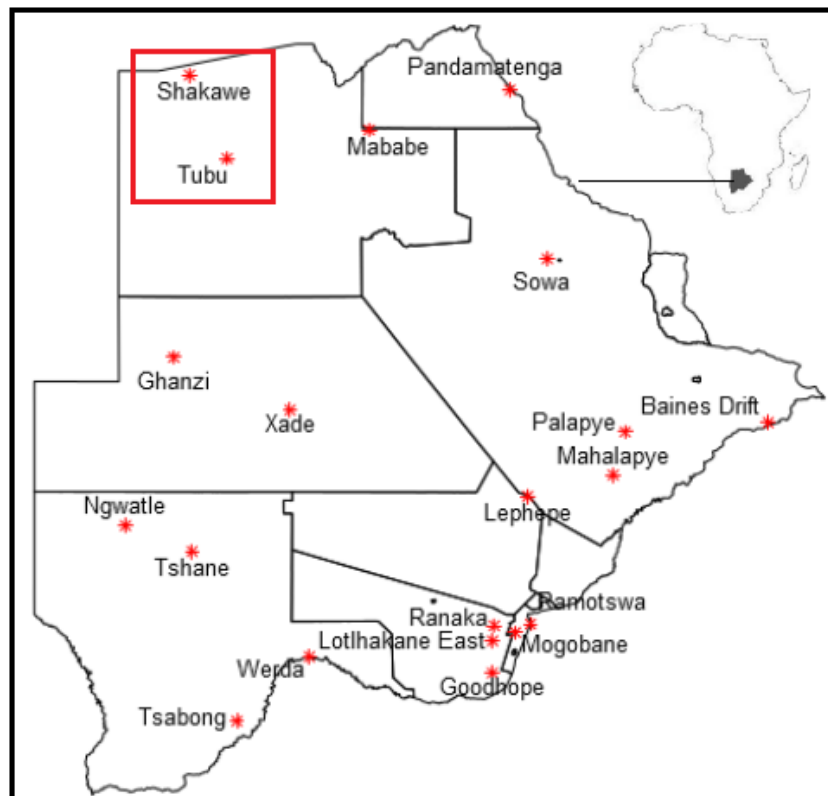


Figure 4.10: Location of the study area.

4.4.2 Data Collection, Data Understanding and Preparation

Previous documentations [79, 83] reported that Botswana's highest temperatures, particularly in the North Western [89, p. 33], are experienced during austral summers from October to March. To illustrate that, this work deploys data from the Shakawe AWS for the period ranging from 01-01-2015 to 31-12-2018. Algorithm 2 was used for data scraping [73] and data wrangling [29]. The collected weather variables include: temperatures in $^{\circ}\text{C}$, i.e., average air temperature (ATAVG), minimum air temperature (ATMIN), maximum air temperature (ATMAX), average surface temperature (STAVG); Total Precipitation (P) in mm ; wind speed in m/s , i.e., average wind speed (WSAVG), maximum wind speed (WSMAX), average wind direction (WDAVG) in $^{\circ}$; average relative humidity (HAVG) in %, average barometric pressure (BPAVG) in hPa ; solar irradiance in W/m^2 , i.e., average solar irradiance (SIAVG), sum of solar irradiance (SISUM) in MJ/m^2 ; sunshine duration (SunD) in minutes, average leaf wetness (LWAVG); average dew point (DPAVG) and average wet bulb (WBAVG) in $^{\circ}\text{C}$. Missing values were replaced using the KNN approach in section 4.3.

Table 4.4: Maximum, minimum and mean temperatures (in $^{\circ}\text{C}$) recorded at Shakawe AWS.

Year	Maximum	Minimum	Mean
2015	41.5	-0.5	23.1
2016	41.5	0.4	22.8
2017	40.6	0.8	21.8
2018	39.8	0.9	21.6

The maximum air temperature (ATMAX) was 41.5°C on the 13th of November in 2015 (see Table 4.4), while a minimum of -0.5°C was captured on the 19th of June. The mean temperature ranged between 11.6°C and 32.5°C , with an average of 23.1°C . In 2016, the maximum temperature reached was 41.5°C on the 31st of October, while a minimum of 0.4°C was recorded on the 23th of July. The mean temperature ranged between 12.9°C and 32.8°C (2016), with an average of 22.8°C . In 2017, ATMAX went up to 40.6°C on the 19th of October, while a minimum of 0.8°C was recorded on the 18th of August. The mean temperature ranged between 13.5°C and 31.6°C for this particular year, with an average of 21.8°C . Likewise, the ATMAX reached in 2018 was 39.8°C on the 18th of November, while a minimum of 0.9°C was recorded on the 10th of September. The mean temperature series approximately lie between 12.2°C and 31°C , with an average of 21.6°C for the year 2018.

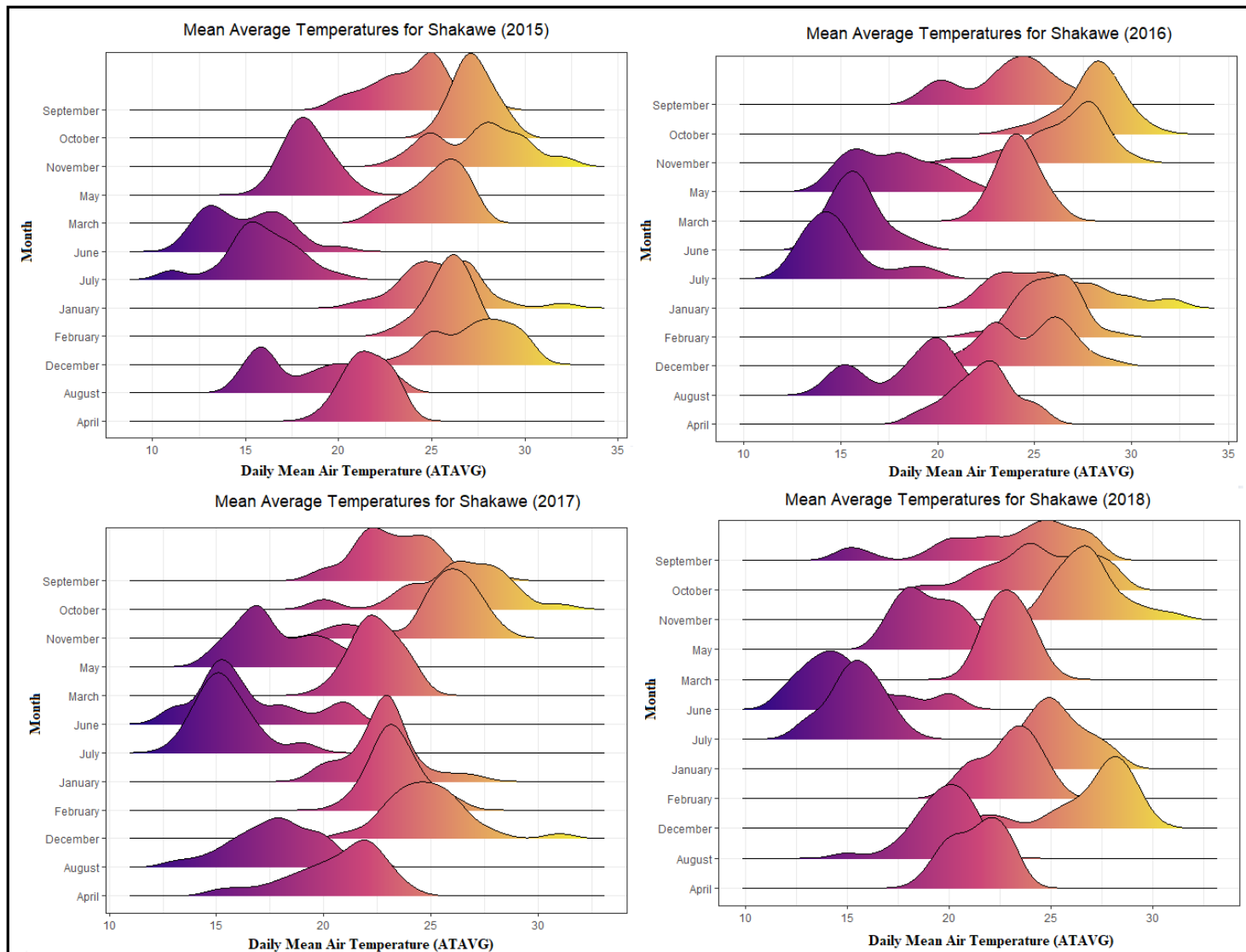


Figure 4.11: Ridgeline plots depicting the mean daily temperatures in °C at Shakawe AWS.

It can be noted that temperatures start increasing at the ending of September, culminating around October-to-January when the sun is overhead at the tropic of Capricorn [89, p. 33], then they decline gradually through March. A snap shot of daily mean air temperature distributions (the 25th percentile, median, and 75th percentile) observed at Shakawe AWS for the year 2015 is illustrated using the side-by-side violin/box plots in Figure 4.12. It can be appreciated that the ATAVG diminishes gradually from around April until reaching the lowest temperatures around June and July, leading to a colder season (May to August).

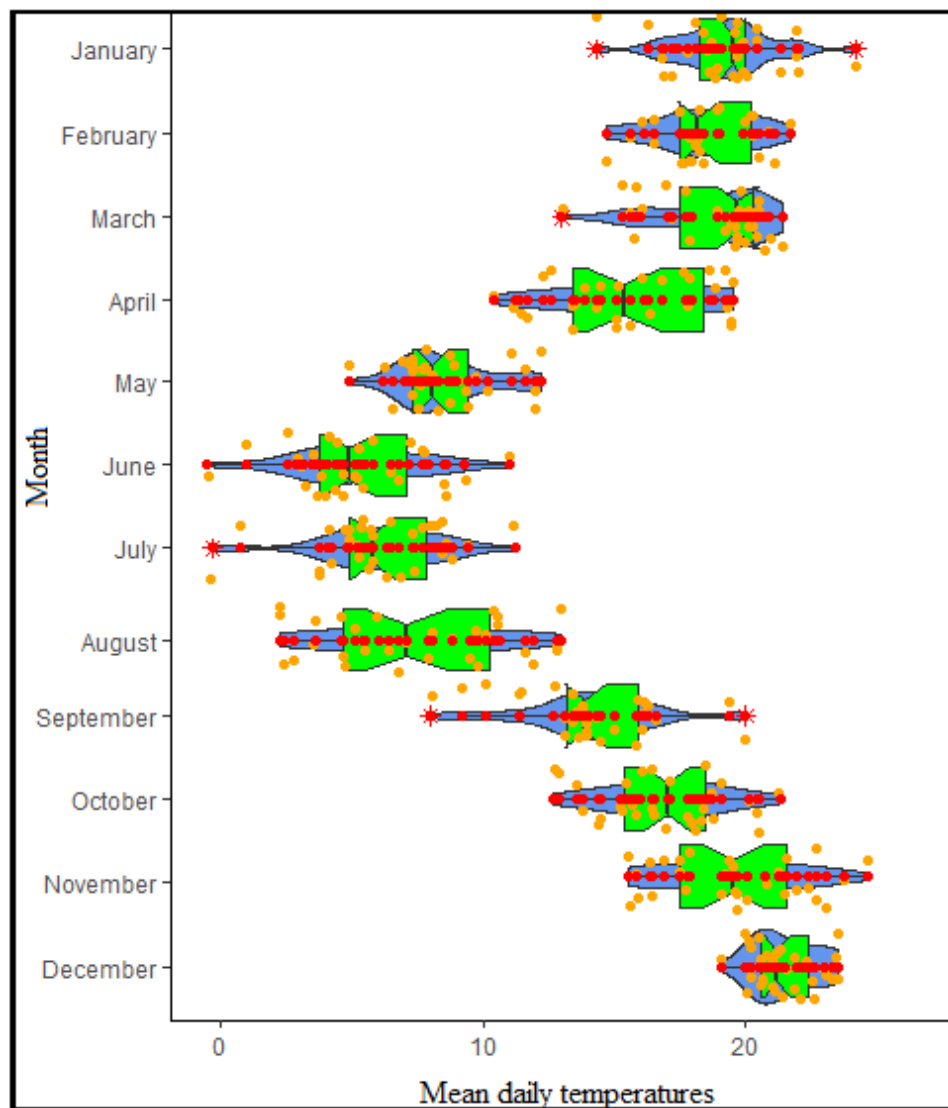


Figure 4.12: Daily average air temperature (ATAVG) distribution for the year 2015.

The asterisks (*) in Figure 4.12 identify outlier temperatures (in January, March, July, and September), depicting observations outside the 99% range of a normal distribution represented by the whiskers. The percentile sizes of the quartiles highlight skewness for the ATAVG distributions (January, March, June, July, September, November and December).

4.4.3 Construction of the Random Forest Model

The seed was set to 132 (for reproducibility). Following the approach deployed in [112, p. 5], the Random Forest (RF) model was implemented using the function in Equation 4.5:

$$RF = f(\hat{x}, \mathbb{D}_s, T_{\mathbb{D}}, ntrees, mtry, N_s, I), \quad (4.5)$$

where: $\hat{x}$ is the formula interface defining the model and target variable ATMAX; $\mathbb{D}_s$ is the modelling dataset, $T_{\mathbb{D}}$ is an index vector indicating instances to be used for training. The logical indicator (I) in Equation 4.5 determines whether to visualise the VIP scores of the predictor variables (see Figure 4.14 and 4.13). This is helpful in the generation of hypothesis when studying the relationships between variables. The optimal model was configured using the number of trees ($ntrees$) and the number of variables ($mtry$) randomly selected at each split to minimise the MSE [34, p. 29]. Here, $\hat{f}(x_i)$ is the prediction of $\hat{f}$ for the i^{th} instance. The RF ensemble was assessed by averaging predictions from each tree.

4.4.4 Determining Influential Variables for Maximum Air Temperature

The variable importance method (well articulated in this book [70, p. 100]) was used to determine influential variables for maximum air temperatures observed at the Tubu AWS. The RF model in 4.4.3 was fed as input to the $vip()$ function in Equation 4.6:

$$VIP = vip(Object, arg), \quad (4.6)$$

where, the argument *Object* is the RF model, and *arg* represents auxiliary arguments for graphical display such as: the number of variables to display on the plot, the colours, and other aesthetics used for visual presentation. Setting the number of trees out of the range [70,100] does not yield a significant impact on the performance of the RF model, but rather increases the computational cost. Each tree in a RF ensemble uses a different set of variables, thus allowing determination of the most important variables that influence the target under consideration (i.e., ATMAX) based on the available data only. The case wise variable importance was used as the average percentage increase in mean squared errors (on the test set) when the variables are permuted. The RF was fit to predict the ATMAX for the Tubu AWS. The variable importance measures are displayed in Figures 4.13 and 4.14.

The mean increase accuracy ("%IncMSE") was used (see Figures 4.13 and 4.14) to analyse the relationship between the maximum air temperature and other variables observed at Tubu. Only the data corresponding to months of the summer seasons were considered (October-March) [83]. Four datasets were generated from the Tubu AWS using climatological data collected between October 2016 and March 2020. The first dataset runs from October 2016 to March 2017. The second one runs from October 2017 to March 2018. The third runs from October 2018 to March 2019, and the fourth runs from October 2019 to March 2020. These datasets had no missing values, requiring minimal data preparations. Two additional variables (Timestamp and Month) were created to enhance the analysis.

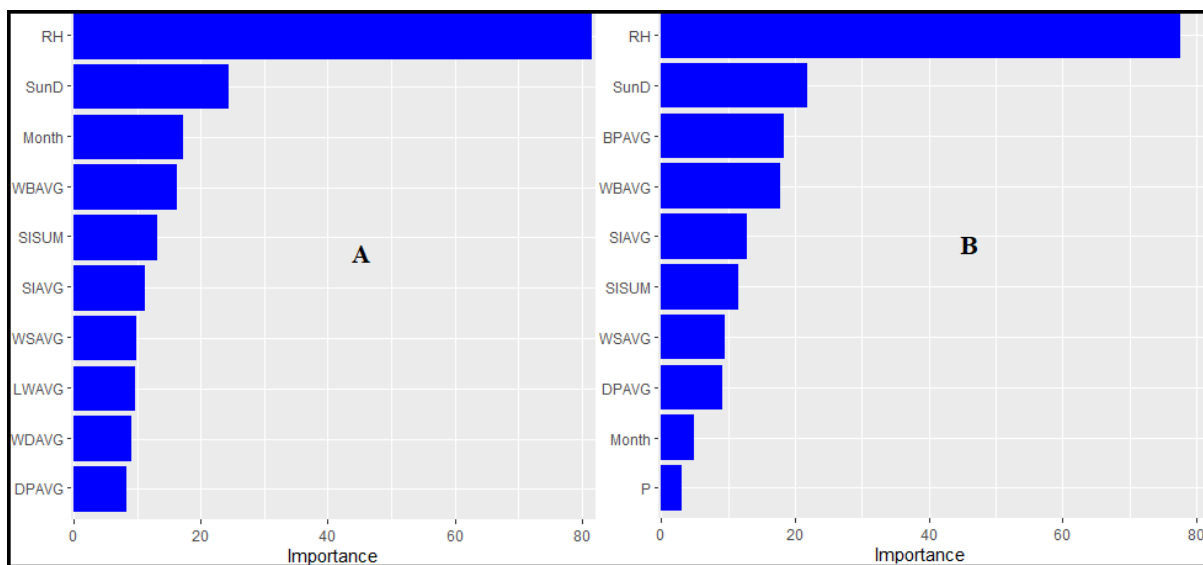


Figure 4.13: Variable importance for maximum air temperature for the Tubu AWS.

Sub-figure A in Figure 4.13 corresponds to the period from October 2016 to March 2017. It can be noted that for that particular summer period, the average relative humidity (RH) was more influential in characterising or determining the maximum air temperature with a percentage of ≈ 82 . This was followed by sunshine duration (SunD), recorded in minutes, and the month with 25% and 17% respectively. Sub-figure B in Figure 4.13 corresponds to the period running from October 2017 to March 2018. Similarly, it can be seen that the relative humidity and sunshine duration are still found to be influential around this weather station with 76% and 22% respectively, followed by the atmospheric pressure (BPAVG), with 18%. From these figures (4.13), it can be appreciated that relative humidity is directly related to maximum air temperature the most compared to other weather variables observed at Tubu.

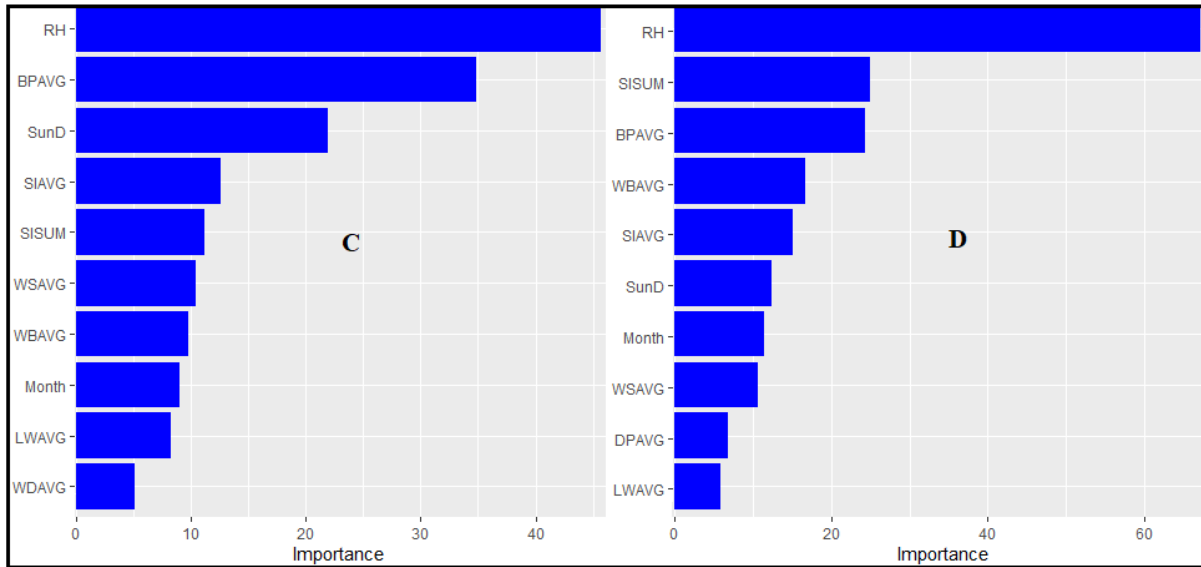


Figure 4.14: Variable importance for maximum air temperature for the Tubu AWS.

Likewise, sub-figure C in Figure 4.14 corresponds to the period from October 2018 to March 2019 while sub-figure D in Figure 4.14 corresponds to the period ranging from October 2019 to March 2020. Though the average relative humidity (RH) was also more influential in both periods (51% and 69% respectively) compared to other variables, it can be noted here that the percentage of contribution was minimal compared to the previous summer periods depicted in Figure 4.13. Sub-figure C clearly shows that average atmospheric pressure (BPAVG) with 35%, and sunshine duration (SunD) with 22% are also important in determining maximum air temperature, followed by solar irradiance (SIAVG) with a 12.5%.

These experiments summarised in Figures 4.14 and Figure 4.13 clearly support Leong in [39, p. 132] that relative humidity is directly related to air temperature (maximum). It should be noted that the relationship between weather variables is chaotic, varying from one location to another at specific times due to several factors, both local and external. For instance, it can be seen that for the first period in sub-figure A in 4.13, the top three most influential variables were: relative humidity, sunshine duration and the month respectively, while in sub-figure B of Figure 4.13 the top three were relative humidity, sunshine duration (SunD) and the average atmospheric pressure (BPAVG). The scenery is not the same in the last two experiments depicted in Figure 4.14, where the top three most influential variables were: relative humidity, atmospheric pressure and sunshine duration in sub-figure C; and relative humidity and sum of solar irradiance (SISUM) in sub-figure D respectively.

4.4.5 Determining Marginal Effects of Predictors on Air Temperature

The Random Forest model was also fitted to the datasets articulated in subsection 4.4.4 to analyse the partial dependencies of maximum air temperature on the rest of the observed variables at the Tubu AWS during those respective summer periods. The following graphs only show PDPs for the top ranked predictor variables based on variable importance measures shown in subsection 4.4.4. Figure 4.15 shows how maximum air temperature relates to average relative humidity (as also indicated by Leong in [39, p. 132]). It can be appreciated that maximum air temperature (ATMAX) decreases with respect to an increase in average relative humidity (RH) in all the four summer periods for the Tubu AWS.

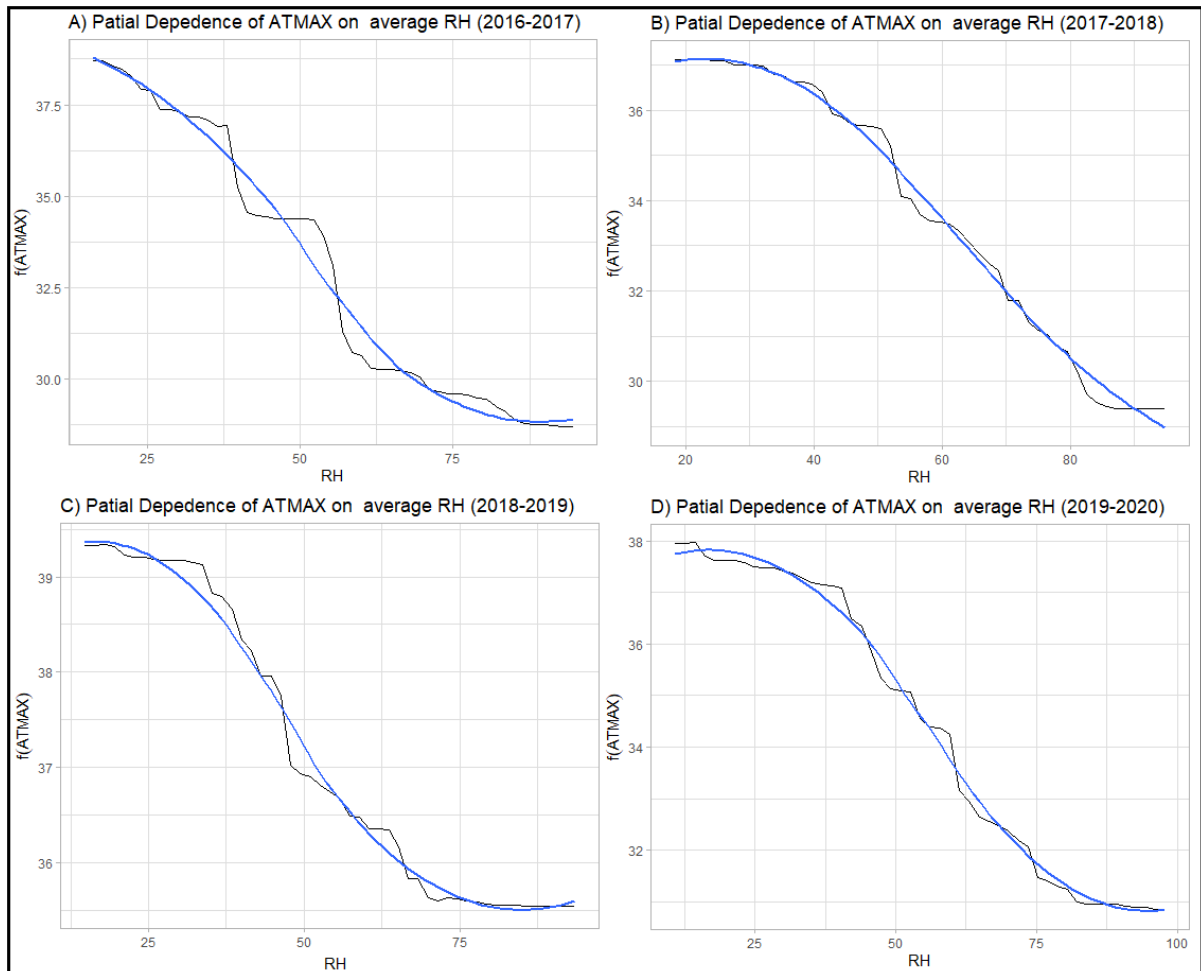


Figure 4.15: Partial dependency of maximum air temperature on average relative humidity.

In sub-figure A of Figure 4.15, ATMAX decreases from 38.8°C to 28°C as the RH increases from 10% to 89%. In B, ATMAX diminishes from 37°C to 31.5°C as RH increases from 18% to 95%. In C, ATMAX diminishes from 39.4°C to 35.5°C as the RH increases from 13% to 93%. In D, ATMAX diminishes from 39°C to 31°C as RH increases from 12% to 98%.

The partial dependence plots in Figure 4.15 clearly demonstrate that the dependency of a particular predictor variable (i.e., average relative humidity) on the target variable (i.e., maximum air temperature) maintains the same characteristics despite the change in temporal settings (change of time). That is to say, the maximum air temperature reduces as the average relative humidity increases though there are some slight perturbations within the respective summer periods. For that reason, the PDP in Figure 4.16 are solely based on a particular summer (2017-2018) in contrast to the illustrations displayed in Figure 4.15.

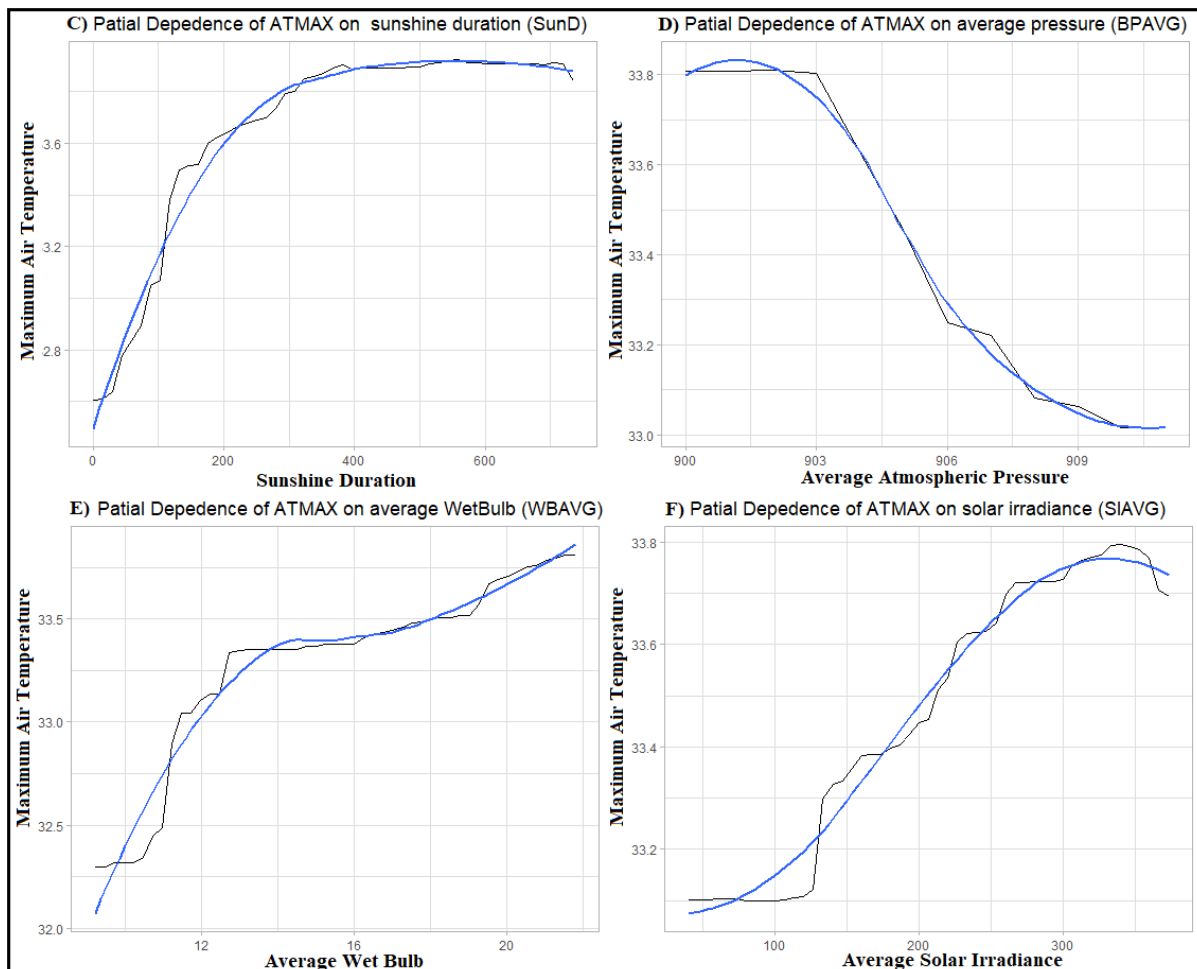


Figure 4.16: Partial dependency of maximum air temperature on other top predictors.

In sub-figure E of Figure 4.16, ATMAX increases from 32.6°C to 34°C as the sunshine duration (SunD) increases from 0 to 750 minutes. In sub-figure F, ATMAX diminishes from 33.8°C to 33.1°C as average pressure (BPAVG) increases from 900 mbars to 911 mbars. In sub-figure G, ATMAX increases from 32.3°C to 33.8°C as the average wetbulb (WBAVG) increases from 9°C to 21.6°C. In sub-figure H, ATMAX increases from 33.3°C to 33.8°C, then reduces as average solar irradiance (SIAVG) increases from 40 W/m^2 to 350 W/m^2 .

4.4.6 Discussions on Data Access and Utilisation of SASSCAL Data

Many organizations provide Earth Observational data using proprietary software or domain specific user interfaces that require registrations and licences. All these add to myriad barriers to data access and thus limiting the use of weather and climate data in research. This work illustrated that these can be circumvented by integrating techniques of data science [22] such as: data scraping [73] and data wrangling [29] as demonstrated using Algorithm 2. As a proof of concept, the collected weather data was used to analyse extreme temperatures around the Okavango Delta; using Shakawe and Tubu AWSs as case studies.

The results from this work concur with previous studies [83, 80] regarding an increasing trend of maximum temperatures as seen in Table 4.4 about the Shakawe AWS. In fact, this particular AWS recorded high maximum air temperatures in the year 2015 and 2016, which is consistent with other results reported in literature [114]³. Meanwhile, the year 2017 and 2018 experienced a slight gradual decrease of maximum air temperatures from 41.5 °C (in 2015/2016) to 40 °C (in 2017) and 39 °C (2018). Interestingly, the minimum air temperature at the same weather station also increases from -0.5 °C (in 2015), through 0.4 °C (in 2016), 0.8 °C (in 2017) and 0.9 °C (in 2018); while the mean temperature diminishes from 23.1 °C to 21.6 °C. In 2019, the maximum and mean air temperatures increased to 41 °C and 23.3 °C respectively. It should be noted that the year 2019 has been reported as the second hottest (after 2015) in the past decades as confirmed by World Meteorological Services (WMO) [3].

These results indicate that although there is a positive trend in general as shown in previous studies [30, 18, 58, 97, 56] worldwide, there is variability on year by year basis. The variability may also be attributed to the fact that weather and climate events are location specific. The variable importance measures of the Random Forest model found the average relative humidity to be more influential in determining maximum air temperature (see Figures 4.13 and 4.14). Likewise, using the partial dependence plots, the results demonstrate that the maximum air temperature and average relative humidity are negatively correlated around the Okavango Delta, particularly, around the Tubu AWS (see Figures 4.15).

³<https://public.wmo.int/en/media/press-release/wmo-confirms-2019-second-hottest-year-record>

Conclusions, Recommendations and Future Work

5.1 Conclusions

This work presented a programmable toolkit for a smart weather system to bridge existing gaps in the availability, access and use of weather data generated by the BIUST AWS. Several researchers have deployed microcontrollers and electronic weather sensors to develop low-cost AWS. This current toolkit differs from these previous efforts in unique ways. First, this work deploys MicroPython and the uPyCraft IDK to scrap data (surface temperature, relative humidity and atmospheric pressure) from two sensors (DHT11 and BME280). The ESP8266 module was used as a microcontroller. The NetBeans SDK was used to develop the database management system using SQLite and Java to store the collected weather data. Moreover, the toolkit provides functionalities for harvesting data from other online data repositories like the SASSCAL Weathernet using the RStudio SDK. This is ideal for crowdsourcing of weather data [66]. RStudio was also used for: statistical data analysis, time series forecasting and the development of the dashboard. The developed dashboard can be used for dissemination of weather outputs to end users. Thus, this toolkit provides a complete framework for smart weather systems that support data collection, data management, data analysis, predictions and the delivering of end products to end users.

The variable importance measures and partial dependence plots were implemented using the Random Forest. It was found that the average relative humidity is more influential in determining maximum air temperature (MAXAT). Moreover, an increase in relative humidity resulted in a decrease in the observed MAXAT in the Okavango Delta. The analysis of MAXAT is very necessary, more especially to the health sector, since they are considered to be more related to heat waves as reported by Russo *et. al.* [98]. The study also examined predictive performance of two strategies implemented through the KNN algorithm for multi-step ahead forecasting of maximum air temperature using data from the BIUST AWS:

1. The MAE, r and the average (AVG) percentage of accuracy all showed that the best multi-step ahead forecasting strategy was MIMO, which achieved a total average of 74%; while the Recursive strategy achieved a total percentage average of 65%.
2. The optimum predictive performance of MIMO outperformed that of the Recursive strategy with a 95% accuracy compared to 67% of the latter.
3. It was observed that when dealing with batch sizes containing more instances, the results in this case are good for both strategies, with lower MAE and higher r . This demonstrates that the KNN model is able to mimic the trend of the maximum air temperature sufficiently, especially when taking $K = 3$, and using 62 lags.
4. The predictive performance for both model configurations improved very much as the batch size increases; and when the forecasting horizon increases beyond 18 hours.
5. Meanwhile, the performance deteriorated beyond 120 hours for both strategies which can be attributed to the chaotic nature of the weather at finer temporal resolutions.

It is noted that once the programmable toolkit and the dashboard API have been designed, the development of a smart weather system for weather monitoring operations and services is straightforward. The toolkit can lead to the ultimate design of effective and low-cost system for weather monitoring, analytics and forecasting at BIUST. The toolkit and the generated weather datasets can be used by students, researchers and data analysts to evaluate machine learning algorithms to determine optimal solutions based on real world weather data instead of using artificial data. This toolkit has potential for the integration of smart weather devices in related applications like smart houses, smart villages, and smart irrigations systems. The toolkit is still under development and open to contributions.

5.2 Recommendations

Several initiatives like SASSCAL and WASCAL have been launched to provide high quality local weather data. These data alone is useless, unless it is interpreted and transformed into useful information and knowledge that can result in better weather services that are relevant and actionable for the end user. This work recommends for collaboration of the meteorological service providers, engineers and other external key stakeholders (academia and private firms) through consistent activities and interactions. The weather service providers should continuously define the roles of data scientists, machine learning experts and application developers (for user interfaces) in the meteorological sector. These would catalyse the use of the available tools and resources like the generated data, data mining algorithms and embedded systems [48]. These disruptive technologies are fast to develop and can be very competitive. The idea is not to replace but to compliment the convectional weather systems (i.e., to configure model outputs; variable selection and improve model forecasts).

5.3 Limitations and Future Work

This programmable toolkit is the first of its kind for smart weather operations and services in BIUST. The toolkit integrated several development kits for various programming languages. However, future work can focus on developing a unified package for the whole pipeline). The main experiments in this work were conducted using only the data from AWSs. Meanwhile, results based on these point observations can lack the regional or global aspect of external factors like cut off lows that can influence the internal variability of local weather conditions. Therefore, more work needs to be done to experiment with more AWSs and more data types (i.e., satellite images). All these could be coupled with NWP model outputs and data harvested using drones to improve weather forecasts at local levels. Moreover, a GSM module can be integrated the toolkit to enable remote monitoring of environmental locations of interest while be able to automate events of interest like irrigations in farms; or automate temperature control in smart houses. From an academic point of view, more advanced machine learning techniques like deep learning, including ensemble models, could be applied on the generated weather data, i.e., to fill the missing values in local, national and regional weather data banks, or to model environmental scenarios of interest.

CHAPTER 6

Situational Analysis

A situational analysis was conducted through printed questionnaire and oral interviews with the meteorological service providers and technicians responsible for AWSs in tertiary institutions. The objective was to comprehend their current technical expertise, challenges and perspectives; taking into consideration the current state of art and disruptive technologies in the world of work.

The attached questionnaire is a typical sample received from the officers responsible for the AWS at the Botswana International University of Science and Technology. The questions are in black, while the responses are highlighted in red. All the respondents indicated an interest in collaboration for enhancement of weather services delivery to end users at all levels. This work was typeset using $\text{\LaTeX}$ and TeXstudio. The situational analysis was prepared separately from the main text. To avoid the breaking of page numbers, the situational analysis and the poster document are appended at the end after the bibliography.

Bibliography

- [1] Marie-Aude Aufaure and Esteban Zimányi. *Business Intelligence*. Springer, 2012. [5](#), [40](#)
- [2] Murat Seckin Ayhan, Vijay Raghavan, et al. SASSCAL WeatherNet to support regional weather monitoring and climate-related research in Southern Africa. *arXiv preprint arXiv:1706.00856*, 2017. [2](#)
- [3] AbuBakr Bahaj. Transforming rural communities through mini-grids. *Smart Villages: New Thinking for Off-Grid Communities Worldwide*, page 29, 2015. [2](#)
- [4] Michael Baldauf, Axel Seifert, Jochen Förstner, Detlev Majewski, Matthias Raschendorfer, and Thorsten Reinhardt. Operational convective-scale numerical weather prediction with the COSMO model: description and sensitivities. *Monthly Weather Review*, 139(12):3887–3905, 2011. [8](#)
- [5] Magdalena A. Balmaseda, Arthur Vidard, and David L. T. Anderson. The ECMWF Ocean Analysis System: ORA-S3. *Monthly Weather Review*, 136(8):3018–3034, 2008. [8](#)
- [6] Taieb Souhaib Ben and Atiya Amir F. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems*, 27(1):62–76, 2015. [5](#)

- [7] Tim Berners-Lee, Roy Fielding, and Henrik Frystyk. Hypertext transfer protocol–HTTP/1.0, 1996. 12
- [8] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine learning strategies for time series forecasting. In *European business intelligence summer school*, pages 62–77. Springer, 2012. 40
- [9] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 30
- [10] Suggests Brewer and Maintainer Andy Liaw. Package ‘randomforest’. 2018. 30
- [11] Robison Cris Brito, Fábio Favarim, Guilherme Calin, and Eduardo Todt. Development of a low cost weather station using free hardware and software. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, pages 1–6. IEEE, 2017. 15
- [12] Chiranjit Chakraborty and Andreas Joseph. Staff Working Paper No. 674: Machine learning at central banks. 2017. 27, 29
- [13] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. shiny: Web Application Framework for R. R package version 1.0. 0. *R Found. Stat. Comput., Vienna*. [https://CRAN.R-project.org/package= shiny](https://CRAN.R-project.org/package=shiny) (accessed 12 Feb. 2018), 2017. 36
- [14] Xi Chen. Constrained application protocol for Internet of Things. URL: <https://www.cse.wustl.edu/~jain/cse574-14/ftp/coap>, 2014. 12
- [15] Beeley Chris. *Hands-On Dashboard Development with Shiny*. Packt Publishing Ltd.s, 2018. 36
- [16] Monteleoni Claire, Schmidt Gavin A, and McQuade Scott. Climate informatics: accelerating discovering in climate science with machine learning. *Computing in Science & Engineering*, 15(5):32–40, 2013. 1
- [17] Maria J Cruz and Egbert Gramsbergen. Adding Value and Facilitating Data Reuse: the Case of the 4TU.Centre for Research Data. 2018. 15

- [18] J.S. D'Aleo and M. Khandekar. Chapter 6 - Weather Extremes. In Don J. Easterbrook, editor, *Evidence-Based Climate Science (Second Edition)*, pages 103 – 120. Elsevier, second edition edition, 2016. [56](#)
- [19] Chris Darden. Bridging the gap between research and operations in the National Weather Service: Collaborative activities among the Huntsville meteorological community. 2002. [1](#)
- [20] Yajnaseni Dash, Saroj K. Mishra, and Bijaya K. Panigrahi. Rainfall prediction for the Kerala state of India using Artificial Intelligence approaches. *Computers & Electrical Engineering*, 70:66 – 73, 2018. [29](#)
- [21] Bailey David and Wright Edwin. *Practical SCADA for industry*. Elsevier, 2003. [9](#)
- [22] Abbott Dean. *Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst*. Wiley, 1 edition, 2014. [1](#), [5](#), [9](#), [18](#), [23](#), [26](#), [27](#), [29](#), [30](#), [31](#), [40](#), [45](#), [47](#), [56](#)
- [23] JT Devaraju, KR Suhas, HK Mohana, and Vijaykumar A Patil. Wireless portable microcontroller based weather monitoring station. *Measurement*, 76:189–200, 2015. [10](#), [13](#), [15](#), [16](#), [19](#)
- [24] Christopher Velazquez Albert A. Espinoza Luis Feliciano Velazquez Ricardo Serano Diego A. Aponte-Roa, Luis Benitez Montalvan. Evaluation of a low-cost, solar-powered weather station for small-scale wind farm site selection. 2018. [9](#), [13](#), [16](#)
- [25] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. [26](#), [27](#)
- [26] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *arXiv preprint arXiv:1808.00033*, 2018. [27](#)
- [27] John Ehrlinger. ggRandomForests: RandomForests for Regression. 2015. [30](#), [31](#)
- [28] Kimberly L Elmore, ZL Flamig, V Lakshmanan, BT Kaney, V Farmer, Heather D Reeves, and Lans P Rothfusz. mping: Crowd-sourcing weather reports for research. *Bulletin of the American Meteorological Society*, 95(9):1335–1342, 2014. [18](#)

- [29] Florian Endel and Harald Piringer. Data Wrangling: Making data useful again. *IFAC-PapersOnLine*, 48(1):111–112, 2015. [48](#), [56](#)
- [30] Francois Engelbrecht, Jimmy Adegoke, Mary-Jane Bopape, Mogesh Naidoo, Rebecca Garland, Marcus Thatcher, John McGregor, Jack Katzfey, Micha Werner, Charles Ichoku, and Charles Gatebe. Projections of rapidly rising surface temperatures over Africa under low mitigation. *Environmental Research Letters*, 10(8):085004, aug 2015. [56](#)
- [31] Zabini Federica. Mobile weather apps or the illusion of certainty. *Meteorological Applications*, 23(4):663–670, 2016. [1](#), [18](#)
- [32] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. [31](#)
- [33] Sukumar Ganapati. Use of dashboards in government. *CLEARING*, 260(312k):157–608, 2011. [18](#)
- [34] James Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning*, volume 112. Springer, 2013. [1](#), [27](#), [51](#)
- [35] Karina Gibert, Joaquín Izquierdo, Geoffrey Holmes, Ioannis Athanasiadis, Joaquim Comas, and Miquel Sànchez-Marrè. On the role of pre and post-processing in environmental data mining. 2008. [11](#)
- [36] Karina Gibert, Joaquín Izquierdo, Miquel Sànchez-Marrè, Serena H. Hamilton, Ignasi Rodríguez-Roda, and Geoff Holmes. Which method to use? An assessment of data mining methods in Environmental Data Science. *Environmental Modelling and Software*, 2018. [1](#), [19](#), [27](#)
- [37] Paulo Giudici and Silvia Figini. *Applied Data Mining for Business and Industry*. A John Wiley and Sons, Ltd., second edition, 2009. [27](#)
- [38] Harry R Glahn and Dale A Lowry. The use of model output statistics (mos) in objective weather forecasting. *Journal of applied meteorology*, 11(8):1203–1211, 1972. [8](#)

- [39] Leong Goh, Cheng. *Certificate Physical and Human Geography*. Oxford University Press, 1983. [53](#), [54](#)
- [40] Brandon M. Greenwell. *pdp: An R Package for Constructing Partial Dependence Plots*. 2017. [31](#)
- [41] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Gianotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):93, 2018. [26](#), [30](#), [31](#)
- [42] Patrick Hall. Predictive modeling: Striking a balance between accuracy and interpretability. [27](#)
- [43] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Science & Business Media, 2009. [27](#), [29](#)
- [44] TA Howell, DW Meek, CJ Phene, KR Davis, and RL McCormick. Automated weather data collection for research on irrigation scheduling. *Trans. ASAE*, 27(2):386–391, 1984. [1](#)
- [45] William W Hsieh. An adaptive nonlinear MOS scheme for precipitation forecasts using neural networks. *Weather and Forecasting*, 18(2):303–310, 2003. [9](#)
- [46] Eibe Frank Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 2 edition, 2005. [29](#)
- [47] Frank Jennings and John Yates. Scrapping over data: are the data scrapers’ days numbered? *Journal of Intellectual Property Law & Practice*, 4(2):120–129, 2008. [25](#)
- [48] S.J. Johnston, M. Apetroaie-Cristea, M. Scott, and S.J. Cox. Chapter 15 - Applied Internet of Things. In Rajkumar Buyya and Amir Vahid Dastjerdi, editors, *Internet of Things*, pages 277 – 298. Morgan Kaufmann, 2016. [9](#), [10](#), [59](#)
- [49] Allio Michael K. Strategic dashboards: designing and deploying them to improve implementation. *Strategy & Leadership*, 40(5):24–31, 2012. [19](#)
- [50] Daniel M Kammen. Energy innovation for smart villages. *Smart Villages: New Thinking for Off-Grid Communities Worldwide*, page 21, 2015. [2](#)

- [51] Palak Kapoor and Ferdous Ahmed Barbhuiya. Cloud based weather station using iot devices. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 2357–2362. IEEE, 2019. [15](#), [16](#)
- [52] L. Karthikeyan, Ila Chawla, and Ashok K. Mishra. A review of remote sensing applications in agriculture for food security: Crop growth and yield, irrigation, and crop losses. *Journal of Hydrology*, 586:124905, 2020. [6](#)
- [53] Petros Karvelis, Theofanis-Aristofanis Michail, Daniele Mazzei, Stefanos Petsios, Andrea Bau, Gabriele Montelisciani, and Chrysostomos Stylios. Adopting and embedding machine learning algorithms in microcontroller for weather prediction. In *2018 International Conference on Intelligent Systems (IS)*, pages 474–478. IEEE, 2018. [10](#), [13](#), [15](#), [16](#)
- [54] John D Kelleher, Brian Mac Namee, and Aoife D’Arcy. *Fundamentals of Machine Learning for Predictive Data Analytics*. The MIT Press, Cambridge, Massachusetts, London, England, 2015. [23](#), [26](#), [29](#)
- [55] Josua Krause, Adam Perer, and Enrico Bertini. Using visual analytics to interpret predictive machine learning models. *arXiv preprint arXiv:1606.05685*, 2016. [27](#), [31](#)
- [56] Ngar-Cheung Lau and Mary Jo Nath. A Model Study of Heat Waves over North America: Meteorological Aspects and Projections for the Twenty-First Century. *Journal of Climate*, 25(14):4761–4784, 2012. [56](#)
- [57] Breiman Leo. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. [27](#)
- [58] Wenjun Ma, Lijun Wang, Hualiang Lin, Tao Liu, Yonghui Zhang, Shannon Rutherford, Yuan Luo, Weilin Zeng, Yewu Zhang, Xiaofeng Wang, Xin Gu, Cordia Chu, Jianpeng Xiao, and Maigeng Zhou. The temperature–mortality relationship in china: An analysis from 66 chinese communities. *Environmental Research*, 137:72 – 77, 2015. [56](#)
- [59] Francisco Martinez, Maria P Frias, Francisco Charte, and Antonio J Rivera. Time series forecasting with knn in r: the tsfknn package. 2019. [40](#)

- [60] Francisco Martínez, María Pilar Frías, María Dolores Pérez-Godoy, and Antonio Jesús Rivera. Dealing with seasonality by narrowing the training set in time series forecasting with kNN. *Expert Systems with Applications*, 103:38–48, 2018. [11](#)
- [61] Caren Marzban, Scott Sandgathe, and Eugenia Kalnay. MOS, perfect prog, and reanalysis. *Monthly weather review*, 134(2):657–663, 2006. [8](#), [9](#)
- [62] Clifford Mass, Clifford F Mass, et al. Nowcasting: The next revolution in weather prediction. *Bulletin of the American Meteorological Society*, pages 1–23, 2011. [11](#), [13](#)
- [63] Ricardo Matheus, Marijn Janssen, and Devender Maheshwari. Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities. *Government Information Quarterly*, 2018. [18](#), [19](#)
- [64] Pierre-Philippe Mathieu and Christoph Aubrecht. *Earth observation open science and innovation*, volume 15. Springer, 2018. [47](#)
- [65] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.", 2012. [25](#)
- [66] Fred Meier, Daniel Fenner, Tom Grassmann, Marco Otto, and Dieter Scherer. Crowdsourcing air temperature from citizen weather stations for urban climate research. *Urban Climate*, 19:170–191, 2017. [57](#)
- [67] Arlindo Meque. Introduction to Climate Modelling: Southern African Regional Climate Information Services for Disaster Resilience Development (SARCIS-DR) Project. 2019. [17](#)
- [68] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997. [25](#)
- [69] Patricia K Mogomotsi, Amogelang Sekelemani, and Goemeone EJ Mogomotsi. Climate change adaptation strategies of small-scale farmers in Ngamiland East, Botswana. *Climatic Change*, pages 1–20, 2020. [1](#)
- [70] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2019. [23](#), [27](#), [29](#), [30](#), [40](#), [51](#)

- [71] Oliver Moses and Wame L Hambira. Effects of climate change on evapotranspiration over the Okavango Delta water resources. *Physics and Chemistry of the Earth, Parts A/B/C*, 105:98–103, 2018. [47](#)
- [72] P Muck and M Homam. IoT Based Weather Station Using Raspberry Pi 3. *International Journal of Engineering & Technology*, 7:145, 11 2018. [14](#), [16](#)
- [73] Simon Munzert, Christian Rubba, Peter Meißner, and Dominic Nyhuis. *Automated data collection with R: A practical guide to web scraping and text mining*. John Wiley & Sons, 2014. [24](#), [25](#), [48](#), [56](#)
- [74] Zungeru Adamu Murtala, Gaboitaolelwe Jwaone, Diarra Bakary, Chuma Joseph M, Ang Li-Minn, Kolobe Lone, David Mpho, and Zibani Ishmael. A secured smart home switching system based on wireless communications and self-energy harvesting. *IEEE Access*, 7:25063–25085, 2019. [2](#), [6](#)
- [75] Zungeru Adamu Murtala, Mangwala Mmoloki, Chuma Joseph, Gaebolae Baboloki, and Basutli Bokamoso. Design and simulation of an automatic room heater control system. *Heliyon*, 4(6):e00655, 2018. [2](#), [6](#)
- [76] Lekha Nair, Sujala Shetty, and Siddhanth Shetty. Interactive visual analytics on big data: Tableau vs d3. js. *Journal of e-Learning and Knowledge Society*, 12(4), 2016. [17](#)
- [77] Roshanak Nateghi, Seth D Guikema, and Steven M Quiring. Forecasting hurricane-induced power outage durations. *Natural hazards*, 74(3):1795–1811, 2014. [31](#)
- [78] Guilherme Tomaschewski Netto and Jorge Arigony-Neto. Open-source automatic weather station and electronic ablation station for measuring the impacts of climate change on glaciers. *HardwareX*, 5:e00053, 2019. [14](#), [16](#)
- [79] Tiro Nkemelang. *Evaluating temperature and precipitation extremes under 1.5 ° C and 2.0 ° C warming above pre-industrial levels: Botswana case study*. PhD thesis, University of Cape Town, 2018. [47](#), [48](#)
- [80] Tiro Nkemelang, Mark New, and Modathir Zaroug. Temperature and precipitation extremes under current, 1.5 ° C and 2.0 ° C global warming above pre-industrial lev-

- els over botswana, and implications for climate change vulnerability. *Environmental Research Letters*, 13(6):065016, jun 2018. [56](#)
- [81] Mary Nsabagwa, Maximus Byamukama, Emmanuel Kondela, and Julianne Sansa Otim. Towards a robust and affordable Automatic Weather Station. *Development Engineering*, 4:100040, 2019. [10](#), [14](#), [16](#), [19](#)
- [82] John O’Hara. Toward a commodity enterprise middleware. *Queue*, 5(4):48–55, 2007. [12](#)
- [83] Moses Oliver. Heat wave characteristics in the context of climate change over past 50 years in Botswana. 2017. [48](#), [52](#), [56](#)
- [84] Nicholas Ozor and Alfred Nyambane. Bridging Climate Information Gaps to Strengthen Capacities for Climate Informed Decision-making: Climate Information and Needs Assessment Report: Cameroon, Kenya, Malawi, Nigeria and Tunisia. 2018. [1](#), [4](#), [47](#)
- [85] Hall Patrick and Gill Navdeep. *An Introduction to Machine Learning Interpretability*. O’Reilly Media, 2018. [30](#)
- [86] LL Pfitscher, DP Bernardon, LM Kopp, AAB Ferreira, MVT Heckler, BA Thome, PDB Montani, and DR Fagundes. An automated irrigation system for rice cropping with remote supervision. In *2011 International Conference on Power Engineering, Energy and Electrical Drives*, pages 1–6. IEEE, 2011. [1](#)
- [87] Jordan G Powers, Joseph B Klemp, William C Skamarock, Christopher A Davis, Jimy Dudhia, David O Gill, Janice L Coen, David J Gochis, Ravan Ahmadov, Steven E Peckham, et al. The weather research and forecasting model: Overview, system efforts, and future directions. *Bulletin of the American Meteorological Society*, 98(8):1717–1737, 2017. [8](#)
- [88] J Ross Quinlan et al. Bagging, boosting, and c4. 5. In *AAAI/IAAI, Vol. 1*, pages 725–730, 1996. [30](#)
- [89] K Silitshena R, M and G. McLeod. *Botswana, A Physical, Social and Economic Geography*. Longman Botswana (PTY) LTD, 1998. [47](#), [48](#), [50](#)

- [90] EN Rajagopal, GR Iyengar, John P George, M Das Gupta, Saji Mohandas, Renu Siddharth, Anjari Gupta, Manjusha Chourasia, VS Prasad, K Sharma Aditi, et al. Implementation of unified model based analysis-forecast system at NCMRWF. *NM-RF/TR/2/2012*, 45, 2012. [8](#)
- [91] Stephan Rasp and Sebastian Lerch. Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 146(11):3885–3900, 2018. [9](#)
- [92] Tye Rattenbury, Joseph M Hellerstein, Jeffrey Heer, Sean Kandel, and Connor Carreras. *Principles of data wrangling: Practical techniques for data preparation*. "O'Reilly Media, Inc.", 2017. [5](#), [25](#)
- [93] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013. [28](#)
- [94] Christian P Robert and Casella George. *Introducing Monte Carlo Methods with R*, volume 18. Springer, 2010. [28](#)
- [95] Manuel Felipe Rodriguez Perez. Diseño y construcción del prototipo de un sistema de adquisición, transmisión y visualización de datos hidrometeorológicos. 2018. [13](#), [15](#), [16](#)
- [96] Zhang Rui, Xu Yan, Dong Zhao, Yang, Kong Weicong, and Wong Kit, Po. A Composite k-Nearest Neighbor Model for Day-Ahead Load Forecasting with Limited Temperature Forecasts, 2016. [29](#)
- [97] Simone Russo, Alessandro Dosio, Rune G. Graversen, Jana Sillmann, Hugo Carrao, Martha B. Dunbar, Andrew Singleton, Paolo Montagna, Paulo Barbola, and Jürgen V. Vogt. Magnitude of extreme heat waves in present climate and their projection in a warming world. *Journal of Geophysical Research: Atmospheres*, 119(22):12,500–12,512, 2014. [56](#)
- [98] Simone Russo, Jana Sillmann, and Andreas Sterl. Humid heat waves at different warming levels. *Scientific reports*, 7(1):1–7, 2017. [58](#)
- [99] Hardeep Saini, Abhishek Thakur, Satinderpar Ahuja, Nitant Sabharwal, and Naveen Kumar. Arduino based automatic wireless weather station with remote graphical

- application and alerts. In *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 605–609. IEEE, 2016. [15](#), [16](#)
- [100] KNV Satyanarayana, SRN Reddy, PVYN SaiTeja, and B Habibuddin. IoT based smart weather station using Raspberry-PI3. *J Chem Pharm Sci*, 10:1–6, 2016. [15](#)
- [101] Trifun Savić and Milutin Radonjić. One approach to weather station design based on Raspberry Pi platform. In *2015 23rd Telecommunications Forum Telfor (TELFOR)*, pages 623–626. IEEE, 2015. [15](#)
- [102] Marco Schwartz. *ESP8266 Internet of Things Cookbook*. Packt Publishing Ltd, 2017. [9](#), [12](#)
- [103] C Shearer. The CRISP-DM: The new blueprint for Data Mining. *Journal of Data Warehousing*, 5(4), 2000. [45](#)
- [104] Jana Sillmann, Thordis Thorarinsdottir, Noel Keenlyside, Nathalie Schaller, Lisa V Alexander, Gabriele Hegerl, Sonia I Seneviratne, Robert Vautard, Xuebin Zhang, and Francis W Zwiers. Understanding, modeling and predicting weather and climate extremes: Challenges and opportunities. *Weather and climate extremes*, 18:65–74, 2017. [1](#)
- [105] Ina SMITH, Tshiamo MOTSHEGWA, and Susan VELDSMAN. Policy, Infrastructure, Skills and Incentives Driving African Data Sharing: The African Open Science Platform Project. In *PV2018: Proceedings of the 2018 conference on adding value and preserving data*, page 28, 2018. [2](#)
- [106] United States. National Weather Service Report. 2017. [17](#)
- [107] Souhaib Ben Taieb, Gianluca Bontempi, Amir F Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert systems with applications*, 39(8):7067–7083, 2012. [40](#)
- [108] Souhaib Ben Taieb, Rob J Hyndman, et al. *Recursive and direct multi-step forecasting: the best of both worlds*, volume 19. Citeseer, 2012. [5](#), [40](#)

- [109] Souhaib Ben Taieb, Antti Sorjamaa, and Gianluca Bontempi. Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomputing*, 73(10-12):1950–1957, 2010. [40](#)
- [110] Mmaphaka Tau. How weather services inform DRR/DRM in South Africa: success, challenges and opportunities. 2018. [47](#)
- [111] Nicholas H Tollervey. *Programming with MicroPython: Embedded Programming with Microcontrollers and Python*. " O'Reilly Media, Inc.", 2017. [9](#), [12](#), [25](#)
- [112] Hristos Tyralis and Georgia Papacharalampous. Variable selection in time series forecasting using random forests. *Algorithms*, 10(4):114, 2017. [51](#)
- [113] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5), 1999. [27](#)
- [114] Costas A Varotsos, Chris G Tzanis, and Nicholas V Sarlis. On the progress of the 2015–2016 El Niño event. *Atmospheric Chemistry and Physics*, 16(4):2007, 2016. [56](#)
- [115] Robert L Vislocky and George S Young. The use of perfect prog forecasts to improve model output statistics forecasts of precipitation probability. *Weather and Forecasting*, 4(2):202–209, 1989. [8](#)
- [116] Modise Wiston and Mphale KM. Weather Forecasting: From the Early Weather Wizards to Modern-day Weather Predictions. *Journal of Climatology & Weather Forecasting*, 2018. [8](#)
- [117] Y Yang, LT Wilson, and J Wang. Development of an automated climatic data scraping, filtering and display system. *Computers and Electronics in Agriculture*, 71(1):77–87, 2010. [5](#), [11](#), [25](#)
- [118] Günther Zängl, Daniel Reinert, Pilar Rípodas, and Michael Baldauf. The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. *Quarterly Journal of the Royal Meteorological Society*, 141(687):563–579, 2015. [8](#)

Situational Analysis Questionnaire

This set of questionnaire serves to acquire opinions concerning the application of climate informatics for weather and climate services in Botswana. Climate informatics is cross-disciplinary field that combines climate science, predictive analytics (data mining, machine learning, statistics, and mathematics). The aim of this project is to foster new collaborations to overcome cross-disciplinary language barriers and to emphasize communication between relevant stakeholders. The ultimate goal is to grow the climate informatics community and accelerate discovery in Botswana and the SADC region.

This situational analysis aims to inform the development of the project that is ongoing at current, with the general objective of facilitating daily operations in weather and climate services. The whole project entails harvesting of climatological data, as well as using it for operational services such as in weather analytics and forecasting of particular variables at local level. Please answer the following questions, and add some additional notes that can be used to enhance better understanding of the problem under consideration.

1. Are you in access of an (some) automated weather stations (AWS)? If yes, what kind of products do you get from such systems?
 - Yes, we do have an AWS. It provides us with all sorts of weather related data depending on the sensors mounted. In BIUST AWS we have Air temperature/relative humidity AT/RH sensor, wind speed, wind direction sensor, Pyranometer, Barometric pressure sensor, tipping bucket rain gauge and dust monitor.
2. The AWS is expected to harvest various weather variables, what kind of clients do you normally service with these data?
 - Mostly students and researchers, but we do get other clients like contractors working in around BIUST requesting the data and PPCF department (projects).
3. Can you please mention the reason why the clients request the data?
 - Students use it for laboratory experiments, mostly those doing final year require the data for use in their projects. Researchers likewise, as well as contractors, use it for justifications and estimations on project delays and completion and other related project calculations.

4. How often do you expect the clients requesting the data? (Daily, weekly, or monthly)?
- **Weekly**
5. Do you have an application interface that links the data generated from the installed AWS to some interface like website for dissemination?
- **Not at the moment. Data is polled from sensors and displayed on the dedicated computer in the room next to the station. To have it ready for use, I have to visit the station weekly to copy the data and paste it in Microsoft Excel, save it a memory stick, then I can then email it to clients.**
6. Do you conduct research on weather analytics and forecasting using the collected data? If yes mention them, and how you use the data.
- **No. I collect data for the purpose of dissemination to clients, depending on their research, some might be using it for weather analytics and forecasting.**
7. Is there any operational High Performance Computing (HPC) system you have access to? If yes, please mention the specifications.
- **None**
8. If you answered yes to the above question, please mention the various ways you use the HPC system regarding your day to day activities.
- **Not applicable**
9. Do you have some storage facilities to save the harvested data?
- **No, only on the on site dedicated desktop computer. I store some of the Excel spreadsheet generated on my personal memory stick which I update at the beginning of each year.**
10. What kind of back-up measures do you have in place in case of data loss?
- **None. We currently also lose a lot of valuable data during power outages as the computer runs on main supply. A UPS has been suggested but it is not yet in place.**
11. Do you have access to other climatological data generated nationally or locally? If any, please mention the sources.
- **Not any I know**
12. Are you aware of any data driven technologies that are currently deployed in the world of work? If yes, mention some of the ones you know, and the ones you use. If no, state the reason why they are not in place.
- **None, I usually use the data for my laboratory experiments so that students could use local data in their calculations instead of using artificial data.**

13. Are there any data policies that prohibit you to share your observation and simulations for research purposes?

- The station was procured as a way to assist researchers with local data and there are no policies that prohibit sharing of data that I'm aware of.

14. What other problems do you experience with the systems and the generated data?

- Data loss due to power outages.
- I have to periodically visit the AWS to check if the system is running.
- I have to manually organise data in excel using cut and paste.

15. What solutions do you have in place to tackle such problems?

- No solution yet, but a remote monitoring and access would be ideal.
- If this could somehow be automated and emailed periodically.
- I have suggested a UPS, still waiting for it.

16. What do you think can be done to facilitate smooth running of the operational activities within the context of AWS?

- Remote monitoring access and automatic data repackaging would add value.

17. Please write down any additional information you think can add value to this studies.

- I feel currently the AWS is under-utilised as the data is not displayed anywhere like school website to disseminate local data. Also, data access is limiting as I have to first go to the station to access it, repackage it manually using copy and paste, only then I'm able to send it to whomever needs it. The process is tedious and time consuming. Monitoring is also tedious as I have to personally visit the station just to ensure its running. If there was a way that could facilitate me to remotely access and monitor the system from a centralised point and possibly have a database/site where clients could access the data that could be vital to us and the clients. Currently clients have to contact me to book for data request. I believe the AWS could serve the BIUST community and other clients better if most of the repetitive processes and data access were automated.

Signature:



Date:

21/08/19

Thank you for your time partitioned to this questionnaire.

