MODELLING STREAMFLOW RESPONSE TO HYDRO-CLIMATIC VARIABLES IN THE UPPER MKOMAZI RIVER, SOUTH AFRICA

Submitted in fulfillment of the requirements of the degree of

Master of Technology: Civil Engineering

in the Faculty of Engineering and the Built Environment at the

Durban University of Technology

Oluwaseun Kunle OYEBODE

Approved for final submission:

Supervisor: _____ **Date:** _____

Dr Josiah Adeyemo

Co-supervisor: _____ **Date:** _____

Prof. Fred Otieno

April 2014

ABSTRACT

Streamflow modelling remains crucial to decision-making especially when it concerns planning and management of water resources systems in water-stressed regions. This study proposes a suitable method for streamflow modelling irrespective of the limited availability of historical datasets. Two data-driven modelling techniques were applied comparatively so as to achieve this aim. Genetic programming (GP), an evolutionary algorithm approach and a differential evolution (DE)-trained artificial neural network (ANN) were used for streamflow prediction in the upper Mkomazi River, South Africa. Historical records of streamflow and meteorological variables for a 19-year period (1994-2012) were used for model development and also in the selection of predictor variables into the input vector space of the models. In both approaches, individual monthly predictive models were developed for each month of the year using a 1-year lead time. Two case studies were considered in development of the ANN models. Case study 1 involved the use of correlation analysis in selecting input variables as employed during GP model development, while the DE algorithm was used for training and optimizing the model parameters. However in case study 2, genetic programming was incorporated as a screening tool for determining the dimensionality of the ANN models, while the learning process was further fine-tuned by subjecting the DE algorithm to sensitivity analysis. Altogether, the performance of the three sets of predictive models were evaluated comparatively using three statistical measures namely, Mean Absolute Percent Error (MAPE), Root Mean-Squared Error (RMSE) and coefficient of determination (R^2).

Results showed better predictive performance by the GP models both during the training and validation phases when compared with the ANNs. Although the ANN models developed in case study 1 gave satisfactory results during the

training phase, they were unable to extensively replicate those results during the validation phase. It was found that results from case study 1 were considerably influenced by the problems of overfitting and memorization, which are typical of ANNs when subjected to small amount of datasets. However, results from case study 2 showed great improvement across the three evaluation criteria, as the overfitting and memorization problems were significantly minimized, thus leading to improved accuracy in the predictions of the ANN models. It was concluded that the conjunctive use of the two evolutionary computation methods (GP and DE) can be used to improve the performance of artificial neural networks models, especially when availability of datasets is limited. In addition, the GP models can be deployed as predictive tools for the purpose of planning and management of water resources within the Mkomazi region and KwaZulu-Natal province as a whole.

DECLARATION

I hereby declare that the work described in this thesis is my original work and has not previously been submitted in its entirety or in part for a degree in any other university. I further declare that this work does not infringe or violate the right of others, as all the sources cited or quoted are indicated and acknowledged by means of a comprehensive list of references.

Oluwaseun Kunle Oyebode

DEDICATION

To God Almighty

The Everlasting Father

The one who knows the end even before the beginning

The one in the know of all

The Ultimate Teacher

ACKNOWLEDGEMENTS

All glory and adoration be unto God Almighty for making the completion of this study a reality. All through the course of this study, I have experienced the joy of God with me, the assurance of God for me, and I testify to the mystery and the wonder of God in me. HE is the Pillar that holds my life. HE is worthy to be praised. I will forever love you LORD.

I would like to express my sincere appreciation to the following individuals and institutions that formed an integral part of the successful completion of this research:

Dr Josiah Adeyemo, for supervising this research and providing invaluable guidance, assistance and motivation throughout the duration of this study. His critical appraisals and suggestions have been priceless at every phase of the study. He read all my draft papers and his meticulous comments are immensely appreciated. Without his key eye for details, enthusiasm and intelligence, this thesis would not have been possible.

Professor Fred Otieno, for co-supervising this work and for providing guidance in application and approval of my bursary immediately I moved to South Africa despite arriving when the university was already winding-down for the academic year.

My beloved mother, Mrs V.O. Oyebode, my sister, Bukky and my brother, Kola, for their indefatigable, enthusiastic, moral, inspirational and spiritual support. Thank you for believing in me. You are simply the best. I love you all.

Mr Oluwatosin Olofintoye, a senior friend, brother and brilliant research fellow, for his pedagogical assistance throughout the course of this research, most especially during the development of the ANN models. Many of the useful discussions we had helped to hone my work and provide new ideas and

direction. Your critical suggestions and constructive criticism further invigorated my drive towards reaching the global optimum of this study. Brother Tosin, I have no doubts in my mind that you have been divinely positioned in this regard, to help surmount the challenges I faced in the course of this study. I will always be grateful.

Professor Olufunso Olorunsogo, one who has been divinely designated as a father and mentor. His inspirational and fatherly pieces of advice, both academically and emotionally can never be underestimated.

Olubayo Moses Babatunde (BayBabs), a friend, colleague and brother from another mother. BayBabs read all my draft papers and reviewed my thesis, chapter by chapter before final submission to my supervisor. Thank you for the countless hours (even during the wee hours of the morning) which you have dedicated for the perfection of this thesis. Your inspiration at all times remains indelible. I will forever appreciate you.

Department of Water Affairs and the South African Weather Service, for providing me with the requisite data for carrying out this research.

Durban University of Technology and the Postgraduate Research Directorate, for providing the scholarship to pursue this degree and thereby making my dream come true. The series of trainings and capacity building workshops have been instrumental to the success of this research.

And to all who have contributed to the success of this research, but space would not allow me to mention, I say a BIG thank you. I love you all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

ACF	Auto Correlation Function
AIMGP	Automatic Induction of Machine Code by Genetic Programming
AMI	Average Mutual Information
ANFIS	Adaptive Neural Fuzzy Inference System
ANN	Artificial Neural Network
ANN-1	Artificial Neural Network Model developed in Case Study 1
ANN-2	Artificial Neural Network Model developed in Case Study 2
ARMA	Auto Regressive Moving Average
ASCE	American Society of Civil Engineers
BNN	Bayesian Neural Network
BP	Back Propagation
BPSO	Binary Particle Swarm Optimization
CART	Classification and Regression Trees
CC	Correlation Coefficient
CFNN	Counter-propagation Neural Network
CG	Conjugate Gradient
CI	Computational Intelligence
cms	Cumecs
CR	Crossover Constant
D	Number of Weights and Biases
DDMs	Data-Driven Models
DE	Differential Evolution
DE-ANN	Differential Evolution-trained Artificial Neural Networks
DWA	Department of Water Affairs (South Africa)

E	Nash Sutcliffe Efficiency
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EP	Evolutionary Programming
ERM	Empirical Risk Minimization
ES	Evolution Strategies
Evo-NN	Evolutionary Neural Network
F	Mutation Scale Factor
FFNN	Feed-Forward Neural Network
FRBS	Fuzzy Rule-Based Systems
GA	Genetic Algorithm
GCM	Global Climate Model
GP	Genetic Programming
HBV	*Hydrologiska Bryång Vattenbalansavdelning*
HSPF	Hydrologic Simulation Program - FORTRAN
i	Counter
IBL	Instance-Based Learning
IUH	Instantaneous Unit Hydrology
K	Number of Nearest Neighbours
K-NN	K-Nearest Neighbours
LGP	Linear Genetic Programming
LM	Levenberg-Marquardt
MAP	Mean Annual Precipitation
MAPE	Mean Absolute Percentage Error
MLP	Multilayer Perceptron

MLR	Multivariate Linear Regression
MSE	Mean Squared Error
MT	Model Tree
n	Number of Data Points
NP	Population Size
P_c	Crossover Probability
P_m	Mutation Probability
PMB	Pietermaritzburg
PMI	Partial Mutual Information
PSO	Particle Swarm Optimization
PSR	Phase Space Reconstruction
Q_o	Observed Streamflow
Q_p	Predicted Streamflow
$\overline{Q_0}$	Mean value of Observed Streamflow
$\overline{Q_p}$	Mean value of Predicted Streamflow
Q_t	Streamflow for a given month in the t^{th} year
Q_{t-1}	Streamflow for the same month in the $(t-1)^{th}$ year
Q_{t-2}	Streamflow for the same month in the $(t-2)^{th}$ year
Q_{t+1}	Streamflow modelled for the given month during the next year
R^2	Coefficient of Determination
$R1_t$	Rainfall for a given month in the t^{th} year for Pietermaritzburg
$R2_t$	Rainfall for a given month in the t^{th} year for Shaleburn
$R3_t$	Rainfall for a given month in the t^{th} year for Giant Castle
RBF	Radial Basis Function
RMSE	Root Mean Squared Error

RNN	Recurrent Neural Networks
SA	South Africa
SAWS	South Africa Weather Service
SDR	Standard Deviation Reduction
SOFM	Self-Organizing Feature Maps
SRM	Structured Risk Minimization
SSA	Singular Spectrum Analysis
SSE	Sum of Squared Errors
SVM	Support Vector Machines
SVR	Support Vector Regression
t	Time counter in years
$T1_t$	Temperature for a given month in the t^{th} year for Pietermaritzburg
$T2_t$	Temperature for a given month in the t^{th} year for Shaleburn
$T3_t$	Temperature for a given month in the t^{th} year for Giant Castle
TSAD	Total Sum of Absolute Deviations
TSK	Takagi-Sugeno-Kang
VBA	Visual Basic for Applications
VC	Vapnik-Chervonenkis
WRC	Water Research Commission (WRC)
x	Weighted Sum of Inputs to a Neuron
X_a, X_b	Pair of Real Valued Vectors
X_c	Target Vector

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The need to manage water resources in arid and semi-arid regions has always been of high importance to water managers and decision-makers, most especially in this era of climatic variations. South Africa, being a water-stressed country and the 30th driest in the world (Crowley and van Vuuren 2013), has not been left out in formulating various policies and strategies towards ensuring continuous availability of water. The rising profile of water demand in majority of South Africa's catchment areas is evident as the demand for water is far above the quantity available for supply, thereby leading to water stress (Figure 1).

Water researchers, the government and other related stakeholders have been making concerted efforts towards developing various approaches to managing the relatively little amount of water in different regions across the country. However, increase in population, economic growth, improved standard of living, higher agricultural water demand among other factors continue to place freshwater resources under agglomerative pressure. Consequently, constant availability of water for domestic, industrial, agricultural, energy, mining, ecological, transportation and recreational purposes can be considered to be under threat.

The pressure on freshwater resources is now being further aggravated by impacts of climatic variations on the water cycle (Graham *et al.* 2011). Streamflow, a fundamental component of the water cycle and a major source of freshwater availability for human, animals, plants and natural ecosystems is severely being impacted upon (Makkeasorn *et al.* 2008). In river hydrology,

variation in streamflow remains a major consequence of the complex, nonlinear and dynamic nature of hydro-meteorological variables within and around a river basin (Maity and Kashid 2011). Thus, prediction of streamflow both on short term and long term basis is critically important as it forms the basis upon which water managers, consumers, policy makers and other stakeholders put in place planning, allocation, control and adaptive strategies in order to ensure water availability and security.

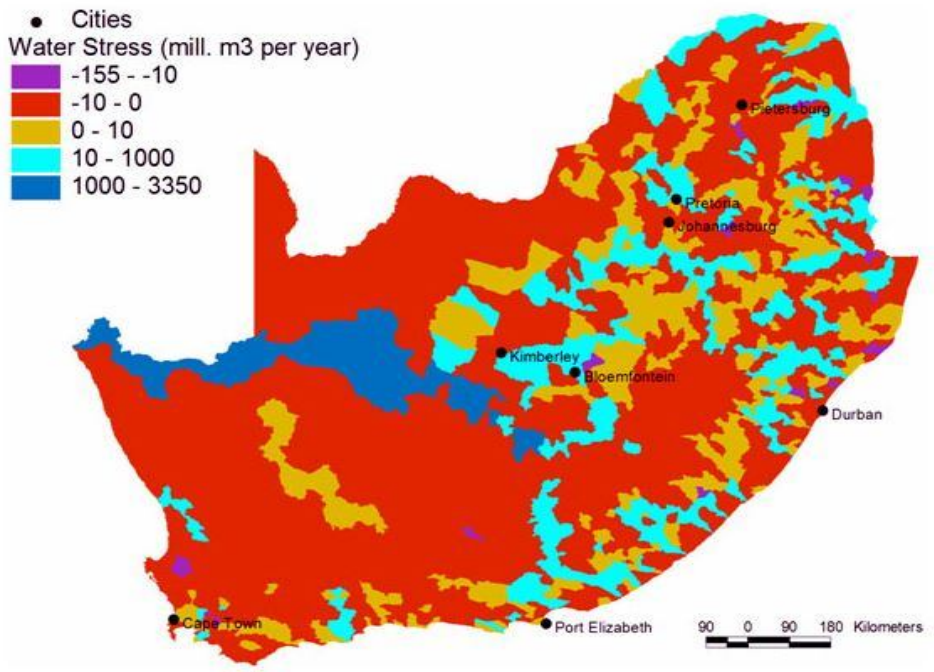

Figure 1: 2005 Annual water balance in SA catchments (Colvin *et al*. 2009)

While short term predictions are made to provide signals about dangers of flood and drought, long term predictions help in providing information for long term water supply strategies (Kisi and Cigizoglu 2007). Furthermore, such information is needed when making decisions on the construction and localization of reservoirs on a river, as well as in optimization of reservoir operations such as in hydropower generation and irrigation water scheduling (Wang *et al.* 2009).

Streamflow predictions also influence the making of financially-related decisions, especially when there is a need to maximize returns on investments made on available water resources (Robertson and Wang 2009). Presently, the South African government requires a sum of 700 billion rand ($71 billion) to meet the country's growing demand for water, with 45% of the sum already budgeted for (Crowley and van Vuuren 2013). Thus, results from accurate and reliable streamflow predictive/modelling studies are of crucial importance to all stakeholders as it yields significant economic and social benefits.

1.2 MODELLING APPROACHES IN HYDROLOGICAL STUDIES

Considering the importance of hydrological modelling, a significant number of modelling techniques have been developed and adopted for the purpose of streamflow prediction. Based on the internal and spatial representations of hydrological processes within a catchment, modelling techniques can be categorized into two. The two categories are namely, process-driven models and data-driven models (DDMs) (Solomatine and Ostfeld 2008).

The process-driven models are generally referred to as "*knowledge-driven*" models based on their ability to provide detailed representation and interpretation of hydrological processes. This is achieved by incorporating laws based on physics of water movement in a catchment. The process-based models include the lumped conceptual and distributed physically-based models (Babovic and Keijzer 2002). Examples of popular process-based models include, the European Hydrological System (SHE) (Abbot *et al.* 1986), the ACRU model – developed in South Africa (Schulze 1995), the HBV (*Hydrologiska Bryång Vattenbalansavdelning*) model (Bergstrom 1995; Lindström *et al.* 1997), the Hydrologic Simulation Program – Fortran (HSPF) model (Johanson *et al.* 1984), and the HYDROLOG model (Porter and McMahon 1971).

However, the complexities and extensive detailing involved in the representation of hydrological processes within a river catchment make the development of models from first principle extremely challenging, thereby inflicting certain drawbacks on the use of process-based models. The major drawbacks to the use of process-based models arise from mixed problems such as mis-calibration, over-parameterization, parameter insensitivity or redundancy, high computational requirements and huge data demand. These limitations tend to generate some uncertainties in the predictive capability of the process-based models, and thus affect the reliability of their results. Attempts have been made to reduce the influences of these uncertainties through uncertainty evaluation techniques (Bastola *et al.* 2011; Brigode *et al.* 2013; Montanari and Di Baldassarre 2013), but the processes involved are rather computationally expensive.

DDMs, on the other hand, define the relationships between system state variables (input, internal and output) variables while characterizing the behaviour of hydrological processes within a river catchment. DDMs achieve this by taking into account only few assumptions on the physics of the system being modelled. These models rely majorly on the methods of computational intelligence and machine learning, and thus assume the presence of a considerable amount of data describing the physics of the modelled system (Solomatine and Ostfeld 2008). Popular DDMs include (i) artificial neural networks (ANNs) (Moradkhani *et al.* 2004; Coulibaly and Evora 2007; Besaw *et al.* 2010; Heng and Suetsugi 2013); (ii) fuzzy rule-based systems (Maskey *et al.* 2004; Shiri and Kisi 2010; Kisi and Shiri 2011) (iii) tree-based methods (Solomatine and Dulal 2003; Galelli and Castelletti 2013) (iv) evolutionary computation methods such as genetic programming (GP) (Guven 2009; Kashid *et al.* 2010; Ni *et al.* 2012) and (v) support vector machines (SVM) (Auria and Moro 2008; Shabri and Suhartono 2012; Zakaria and Shabri 2012).

DDMs are relatively quicker to develop and easier to use when compared to the process-based models. In addition, due to the ability of DDMs to directly define input-output relationships, the large computational and data requirements often associated with process-based models are, to some extent, reduced in DDMs. The use of DDMs is also seen as a promising technique to solving the sensitivity and uncertainty challenges inherent in the use of process-based models (Tripathi *et al.* 2006; Shrestha and Solomatine 2008; Selle and Muttil 2011). In the light of this, DDMs are now being considered as an alternative and promising approach that will complement or replace the knowledge-driven models (Londhe and Charhate 2010; Mohanty *et al.* 2013). A more in-depth appraisal of the application of DDMs in hydrological studies and in particular streamflow modelling is reviewed in Chapter 2 (Literature Review) of the thesis.

1.3 PROBLEM STATEMENT

The availability of data has always been a major factor that influences the accuracy and reliability of predictions in hydrological studies. Furthermore, it has been established that in order to achieve accurate and reliable predictions, large datasets are often required for the purpose of calibration/training and validation in both categories of models (process-based and data-driven) earlier discussed (Babovic and Keijzer 2000). However, these huge amounts of datasets are limited to certain regions and often unavailable in some areas especially in developing countries such as South Africa (Ni *et al.* 2010). Thus, limitation of datasets is considered as a major obstacle to the implementation of hydrological modelling studies in data sparse regions.

By and large, due to the complexity of hydrological processes, no single modelling technique can be considered to be a “miraculous cure-all” or general enough to outclass others in all modelling circumstances (Maier and Dandy 2000; Shamseldin 2004). This implies that the choice of a modelling technique to be employed in any given situation is “problem specific and data dependent”

(Bhattacharya *et al.* 2001). Consequently, there is a need for the discovery and adoption of suitable methods that may serve as alternative approaches for carrying out hydrological modelling studies while using the limited amount of datasets available. Therefore, this study aims to offer solutions in this regard by investigating the potential of two evolutionary-driven techniques (GP and DE-ANN) for streamflow prediction under limited availability of datasets.

1.4 STUDY OBJECTIVES

The aim of this study was to model the impacts of hydro-climatic variables on streamflow dynamics in the upper Mkomazi River. The specific objectives of this study include:

(a) To quantify the impacts of local climatic variables on streamflow in the upper Mkomazi River.
(b) To develop models that could be used for effective streamflow prediction in the upper Mkomazi River using two data-driven modelling techniques namely, genetic programming (GP) and differential evolution (DE)-trained artificial neural networks (ANNs).
(c) To evaluate the potentials of the developed models individually using standard model evaluation criteria.
(d) To carry out performance comparison of the models so as to determine a suitable approach for streamflow prediction under limited availability of datasets in the upper Mkomazi River.
(e) To provide means of improving the performance of the developed models (ANNs) in order to achieve better predictive accuracy and reliability.

1.5 RESEARCH METHODOLOGY

Two data-driven modelling techniques were used in this study. The two modelling techniques are genetic programming and artificial neural networks.

Genetic programming (GP), developed by (Koza 1992) is a computational intelligence approach and population based search which is founded on the principle of natural selection (survival of the fittest). GP is a member of the evolutionary algorithm (EA) family which genetically breeds a population of computer programs to evolve models that best give the representation of a system. GP has been applied in a wide-range of disciplines such as the financial (Garcia-Almanza and Tsang 2007; Esfahanipour and Mousavi 2011), medical (Brameier and Banzhaf 2001; Tsakonas *et al.* 2004) and other science-related and engineering fields (Sætrom *et al.* 2005; Shokir 2008; Ganesan *et al.* 2011). However, GP is a relatively new and expanding method which has found applications in various water-related studies such as in sediment transport modelling (Guven and Kişi 2011; Sirdari *et al.* 2011), streamflow prediction (Whigham and Crapper 2001; Maity and Kashid 2011), climate prediction (Parasuraman *et al.* 2007; Kasiviswanathan *et al.* 2011), ecological modelling (Muttil and Lee 2005), uncertainty assessment studies (Selle and Muttil 2011) and so on.

The artificial neural networks (ANNs) are one of the most extensively used data-driven modelling techniques. ANN is also a computational intelligence approach which is inspired by neuroscience. ANNs can detect and learn correlated patterns between input datasets and corresponding output values (Jha 2007). The ability of ANNs to undergo training, results in the acquisition of an adaptive learning feature which it uses in solving problems. Numerous learning algorithms have been used for the purpose of training and subjecting ANNs to adaptive learning. The popular ones include methods based on gradient descent such as back propagation (BP) algorithm, quick propagation (QP) algorithm and Levenberg Marquardt (LM) algorithm, and evolutionary-heuristic methods such as genetic algorithm (GA) and differential evolution (DE) algorithm.

Thus, ANNs are capable of solving complex and nonlinear problems with little or no *a priori* knowledge of the system being modelled. By extension, the ability of ANNs to assimilate complex and nonlinear input-output interactions makes it suitable for use in the field of water resources. Over the last two decades, ANN has found application in various fields of water resources which include function approximation, classification and forecasting studies (Coulibaly *et al.* 2001; Cigizoglu 2005; Toth 2009; Yonaba *et al.* 2010).

This study employed both the GP and ANN approaches for monthly streamflow prediction in the upper Mkomazi River under limited availability of datasets. A linear genetic programming (LGP) technique was used for the implementation of GP algorithm, while the ANN models were trained using differential evolution (DE) algorithm. Most streamflow prediction studies have only employed the use of one evolutionary-driven modelling technique without comparing its performance with different evolutionary techniques. In this study, two evolutionary-driven modelling techniques are investigated comparatively, so as to achieve more reliable results, especially under limited availability of datasets.

Monthly streamflow prediction models were developed using local hydro-climatic variables namely, streamflow, rainfall and temperature from different weather stations within the area of study. The impacts of the variables on the modelling processes were examined. Both GP and ANN modelling approaches were subjected to sensitivity analysis to produce optimal models. The potential and suitability of the developed models were determined and their performance analyzed comparatively using three model evaluation criteria namely, mean absolute percent error (MAPE), root mean-square error (RMSE) and co-efficient of determination (R^2).

1.6 DESCRIPTION OF STUDY AREA

The Mkomazi catchment is located in the KwaZulu-Natal province of South Africa (Figure 2), and it is the third largest river in the province. The catchment situated around 29°17'24"E and 29°35'24"S, derives its source from the upper Drakensberg mountains and discharges into the Indian Ocean, draining an area of about 4 400km^2. The river is of approximately 160km in length and elevated at about 3 300m above sea level, stretching from the Northwest to the Southeast region. The major water use activities within the catchment include irrigation, commercial afforestation, tourism and recreational activities, as well as paper production which exist near the Mkomazi River mouth. Thus, the river plays a vital role in catering for water needs within the catchment.

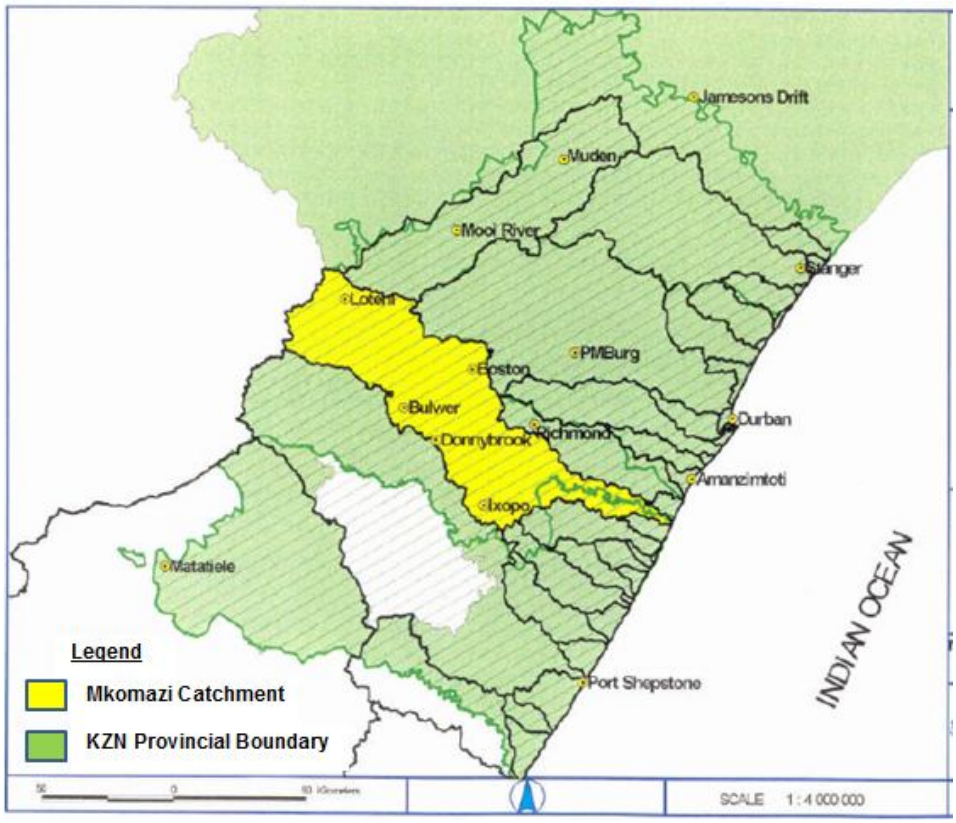

Figure 2: Location of the Mkomazi catchment within the KZN province (DWA 1999a)

The climate is characterized by high seasonality with dry winters and summer rainfall season as Mean Annual Precipitation (MAP) varies between 700 and 1200mm year^{-1} with highly intra- and inter-seasonal streamflows (Flugel and Marker 2003). The MAP is higher in the upper, higher elevations of the river catchment (950mm to about 1 200mm) (Taylor *et al.* 2003), and as a result, a significant amount of catchment runoff is generated in the upper part of the catchment. Hence, this serves as one of the justifications of the choice of the upper Mkomazi River catchment as illustrative study area of this research.

The maximum rainfall occurs in the summer months, particularly between January and February, and minimum rainfall between the months of June and July. The mean annual temperature varies between 15°C in the “High Berg” areas of the Drakensberg and 24°C in the central area of the catchment.

1.6.1 Rationale for the study

The upper Mkomazi River was selected for the purpose of streamflow modelling due to the following reasons:

1. Most of the catchment runoff originates at the upper Mkomazi River Catchment, with flows experienced in the river all-year round (DWA 2004).
2. The overall population distribution in the Mkomazi region is relatively low, hence the use of available water resources can be said to be conservative (Taylor *et al.* 2003).
3. Currently (2013), the river flows directly into the Indian Ocean, as there are no major dams on the river. This translates to the fact that the high abundance of water resources is not being adequately harnessed (DWA 2009).
4. Due to the high pressure and demand for water in the Pietermaritzburg (PMB) and Durban regions, it was concluded from a pre-feasibility study

conducted by the Department of Water Affairs (DWA) that water be transferred out of the Mkomazi River to augment the Umgeni River (DWA 1999b). This will help cater for the huge water demands of the highly populated and industrialized PMB and Durban areas.

5. Recent developments indicate that two potential impoundment sites (Impendle and Smithfield) to be located around the upper-middle Mkomazi catchment have been identified and proposed by DWA for the actualization of the Mkomazi-Umgeni water transfer scheme (DWA 2004).
6. The availability of historical datasets of the catchment area is however limited (DWA 1999c; Rivers-Moore *et al.* 2007).

Following these recent developments, accurate estimation of the streamflow fluctuations in the upper Mkomazi could help understand its dynamics, especially in this era of climatic variations. Furthermore, the prediction of streamflow is of crucial importance irrespective of the limited availability of datasets, as it will help in planning for future water allocation and reservoir operations.

1.7 SIGNIFICANCE OF THE STUDY

Going by the current developments in the Mkomazi-Umgeni region, results from this study would help national and regional water management institutions such as the Department of Water Affairs (DWA), South African Water Research Commission (WRC) and other stakeholders, in planning for adaptive water resource allocation within KwaZulu-Natal province. Monthly models developed in this study could also be deployed as predictive tools for water resource management and for use in future reservoir operations. Furthermore, this study could be adopted as a model for conducting similar studies, especially in regions with limited availability of datasets.

1.8 THESIS OVERVIEW

This thesis is organized into five chapters. The general structure of the thesis and the focal point of each chapter can be described as follows:

Chapter 1 Introduction

This chapter gives a general background to this study, introduces current issues with respect to water resources management in South Africa, identifies research problems and provides a brief review of modelling techniques employed in water resources management. The chapter also outlines the scope, objectives, significance of the study and research methodology. Finally, a description of the study area, the rationale for its study, structure of the thesis and publications from the study are presented.

Chapter 2 Literature Review

This chapter provides a comprehensive review of popular data-driven modelling techniques and modern trends with respect to their applications in hydrological modelling. Modelling approaches used in this study namely, genetic programming (GP) and artificial neural networks (ANNs) are also given elaborate description.

Chapter 3 Genetic Programming for Streamflow Prediction

This chapter begins with the analysis of hydro-climatic datasets obtained from the study area, and thereafter delves into the mainstream of the application of genetic programming for monthly streamflow prediction. The techniques used in data analysis, input variable selection, data pre-processing and model development are discussed in details. Also, the toolkit used for the implementation of GP, the objective function and the model structure as well as genetic operators are fully explained. Finally, the potentials of the developed

models are examined using standard performance evaluation criteria and their corresponding results discussed extensively.

Chapter 4 Artificial Neural Networks for Streamflow Prediction and Performance Comparison between GP and ANNs

This chapter describes the methods used in the development and application of artificial neural networks models for monthly streamflow prediction. Furthermore, the application of differential evolution (DE) algorithm and parameters used for ANN training is discussed. Two case studies are considered in this chapter. Case study 1 involves the use of the same sets of data employed in developing the GP models. This is done to avoid any form of bias when comparing the performance of the derived models. However, in case study 2, GP is incorporated as part of the optimization process of the ANNs, while also fine-tuning its learning process. The performance of the ANN models derived in both case studies were then compared with that of the GP-derived models so as to determine the suitable approach to be adopted for use. The chapter ends with the discussion of the results from the comparative study.

Chapter 5 Conclusions and Recommendations

This chapter presents a general summary based on the results of the previous chapters. It also provides suggestions and recommendations for future research.

Each chapter is concluded by the details of research output(s) from the chapter. The datasets used in this study and the program codes representing the GP models are presented in the Appendix at end of the thesis.

This thesis represents a compilation of manuscripts where each chapter is an individual entity and therefore, some repetitions between chapters are unavoidable.

1.9 PUBLICATIONS

This study has resulted in the following publications:

(a) Journal articles

1. Oluwaseun Oyebode, Josiah Adeyemo and Fred Otieno (2014). Monthly streamflow prediction with limited hydro-climatic variables in the upper Mkomazi River, South Africa using genetic programming. *Fresenius Environmental Bulletin*. Vol. 23: No 3.
2. Josiah Adeyemo, Oluwaseun Oyebode and Fred Otieno (2014). Performance comparison of two data-driven modelling techniques for long term streamflow prediction using limited datasets. *Journal of the South African Institution of Civil Engineering (SAICE)*. Under review.
3. Oluwaseun Oyebode, Fred Otieno and Josiah Adeyemo (2014). Review of three data-driven modelling techniques for hydrological modelling and forecasting. *Fresenius Environmental Bulletin.* Vol. 23: No. 6.

(b) Conference papers

4. Oluwaseun Oyebode and Josiah Adeyemo (2014). An evolutionary approach for improving artificial neural networks performance in river flow forecasting. Paper accepted for presentation at the *International Journal of Arts and Sciences' (IJAS) American Canadian Conference for Academic Disciplines*, 19 – 22 May, 2014, Toronto, Canada.
5. Oluwaseun Oyebode, Josiah Adeyemo and Fred Otieno (2014). Uncertainty sources in climate change impact modelling of water resource systems. Paper accepted for presentation at the *International Journal of Arts and Sciences' (IJAS) American Canadian Conference for Academic Disciplines*, 19 – 22 May, 2014, Toronto, Canada.
6. Oluwaseun Oyebode and Josiah Adeyemo (2014). Genetic Programming: Principles, Applications and Opportunities for Hydrological Modelling. Paper accepted for presentation at the *7th Int'l Conference on Environmental Sci. & Tech.,* 9 – 13 June, 2014, Houston, Texas USA.

CHAPTER 2

LITERATURE REVIEW

2.1 DDM TECHNIQUES IN STREAMFLOW MODELLING

Streamflow modelling plays a crucial role in the planning and management of water resources, most especially in water stressed regions, where the need to effectively manage the available water resources is of critical importance. Therefore, effective streamflow prediction, both on short-term and long-term basis remains essential for efficient operation of water resources systems (Londhe and Charhate 2010). As a result, numerous techniques have been applied by water experts in the quest for understanding the dynamics of streamflow – majorly a product of complex and nonlinear nature of hydro-climatic interactions.

As discussed in Chapter 1, models used for hydrological modelling purposes can be categorized into two, namely, process-based models and data-driven models. However, due to the aforementioned limitations and uncertainties associated with process-based models, the use of data-driven modelling techniques will be the focus of this review and study. Data-driven models have become quite popular in hydrological studies.

This chapter provides a comprehensive review of seven popular data-driven modelling techniques which include support vector machines (SVM), K-nearest Neighbours (K-NN), chaos theory-based methods, model trees (MTs), fuzzy rule-based systems (FRBS), artificial neural networks (ANNs) and genetic programming (GP). The review of each of these modelling approaches encapsulates an introduction to the modelling approach, applications in hydrological modelling, strengths and limitations, areas of concern and performance improvement methods. Other issues discussed in this review

include the processes involved in data-driven modelling such as input determination, data-preprocessing and model performance evaluation.

2.2 SUPPORT VECTOR MACHINES (SVMs)

2.2.1 Introduction

Support vector machines (SVMs) are non-linear parametric models which originate from the statistical learning theory (Vapnik 1998). The working principle of SVMs is however different from that of other DDMs. In most DDMs, an Empirical Risk Minimization (ERM) principle is adopted with the aim of minimizing the difference between observed and predicted values. In contrast, SVMs initialize by defining a reasonable limit or boundary on the generalization error using a Structural Risk Minimization (SRM) principle (Elshorbagy *et al.* 2010). Thereafter, SVMs proceed to search for the optimal structure of the model via the selection of model training parameters which guarantees an exclusive global minimum of the error surface. This approach enables SVMs to have a good performance in terms of generalization.

Although, SVMs were initially developed for classification purposes, they have now been broadened for use in regression and prediction purposes, and thus being referred to as support vector regression (SVR) (Deo 2009). By this extension, SVMs now possess the ability to capture both linear and nonlinear patterns in a given set of data. They achieve this by employing a mapping function which transforms the data space into a higher domain which exhibits multiple dimensionalities (Jordaan 2002). The mapping function which is implemented by using a specified kernel may either be a linear, polynomial, sigmoidal, radial basis or hybrid function. However, the preferred kernel in hydrological modelling studies is the Gaussian radial basis function (Liong and Sivapragasam 2002; Lin *et al.* 2006; Yu *et al.* 2006). Thus, SVMs are able to give representation of both linear and nonlinear processes. Going by the

principle of operation of SVMs, their structural make-up is considered to be similar to that of ANNs. The structural make-up of both SVMs and ANNs can be represented as two-layered networks, where the weights are nonlinear in the first layer and linear in the output layer. However, the major difference between the two approaches is in their learning approach. While ANN adopts an adaptive learning approach in optimizing all its parameters, SVMs select the parameters for the first layer to be the training input vectors. Thus, SVMs create a model with good generalization using a minimized Vapnik-Chervonenkis (VC) dimension – model capacity or complexity (Cherkassky and Mulier 2007). Owing to the ability of SVMs to characterize both linear and nonlinear systems, they are found suitable for use in hydrological modelling and predictive studies.

2.2.2 Application of SVMs in hydrological modelling

Although, SVMs have not received wide application in hydrological modelling studies compared with other DDMs, their application is increasingly getting attention. SVMs have been applied in water-related studies such as rainfall-runoff modelling, river flow prediction, flood forecasting, water level prediction in lakes and reservoirs, sediment modelling, etcetera (Dibike et al. 2001; Yu et al. 2006; Sudheer et al. 2013). Likewise, their performance has been compared with other techniques (Cannas et al. 2004; Asefa et al. 2006; Lin et al. 2006).

Sivapragasam *et al.* (2001) in a novel study applied a SVM in predicting rainfall and streamflow in two different catchments in Denmark and Singapore. The SVM was coupled to a data preprocessor by using a Single Spectrum Analysis (SSA) technique. A combination of the linear and radial basis function (RBF) kernels was proposed. Results showed that the SVM method yielded significant accuracy in both rainfall and streamflow predictions across the two catchments.

Dibike *et al.* (2001) applied SVM for land use classification and rainfall-runoff prediction using a polynomial and RBF kernel. Results showed better

performance than ANN-derived models, with the RBF kernel also performing better than the polynomial one. Liong and Sivapragasam (2002) also applied a SVR to river-stage forecasting at multiple lead-times in Bangladesh. It was reported that SVR performed better than models developed using ANN. Vojinovic and Kecman (2004) successfully applied SVM for contaminant transport modelling in Stream Wairu in New Zealand. They however cautioned that SVMs are susceptible to overfitting problems if the number of support vectors employed is higher than the number of training samples.

Cannas *et al.* (2004) compared the performance of SVM and multi-layer perceptron (MLP) in rainfall-runoff modelling. They observed that under non-stationary conditions, a well-trained MLP performed better than the SVM. Sivapragasam and Muttil (2005) employed SVM to implement and extrapolate discharge rating curves in three gauging stations in Washington. Results obtained showed that SVMs performed significant better than logarithmic and higher-order polynomial methods. They also found that SVM was better suited for extrapolation purposes than ANN.

Tripathi *et al.* (2006) used a SVM-based downscaling model (DM) to statistically downscale outputs from a global climate model (GCM) in India. The SVM-based DM was coupled to a GCM to obtain future projection of precipitation at a monthly time scale. Results were analyzed to ascertain plausible impacts of climate change over India. Results showed SVMs as a promising tool for statistical downscaling purposes in climate change studies. Rasouli *et al.* (2012) applied SVM for daily streamflow forecasting at various meteorological variables and large-scale climate indices in British Columbia, Canada. Results indicated that SVM performed satisfactorily in capturing the input-output relationships and streamflow regime of river basin.

2.2.3 Areas of concern

Going by application of SVMs in modelling hydrological systems, some areas of concern have been identified.

Asefa *et al.* (2006) applied SVM for forecasting streamflow at multiple time scales in an ungauged river basin in Utah, USA. Local climatological variables were used conjunctively with atmospheric circulation indices. A RBF kernel was used during implementation of the SVM, while a cross-validation approach was used for estimating the SVM parameters. The performance of the SVM method was evaluated comparatively with that of a transfer function noise (TFN). It was reported that the SVMs produced an overall better performance in terms of RMSE values than the TFN models, although the TFN model gave a smoother prediction especially for the high annual flows. Estimation of prediction uncertainty in SVMs was suggested, as they noted that weaknesses arise in the selection of kernels and hyper-parameters.

Lin *et al.* (2006) through a comparative study demonstrated the capability of SVMs to predict long-term flow discharges while using a hydrological time series with nonlinear features. Results indicated that the SVM models were able to give more accurate predictions compared with that of ANN and ARMA-derived models. However, they pointed out that high computational time is required while applying SVMs to solve problems with large data-size. Their final submission was in line with that of Asefa *et al.* (2006), as they also emphasized that challenges of kernel and model parameter selection are inherent in the use of SVMs. Behzad *et al.* (2009) in a more recent study however shared a contrasting view to that of Lin *et al.* (2006) in terms of computational time of SVMs. They opined based on results of their comparative study with a genetic algorithm (GA)-inspired ANN that SVMs are considerably faster with the same or higher accuracy.

In a study conducted by Bray and Han (2004) in the Bird Creek region, USA, SVM was applied to forecast runoff with focus of the identification of a suitable model structure and significant model parameters. The relationships between several model structures, kernel functions, model parameters, scaling factors and input vectors were investigated. This was done with the aim of providing valuable information for efficient model structure identification for the benefit for future research. Results showed that the TFN model used as a basis for comparison performed better than SVM in short-term forecasts, as the SVM encountered difficulties in searching for global optima. They remarked that irrespective of the nonlinear features and flexibility of the SVM input vectors, the process of optimizing SVMs is very complex and that parameter sensitivity has great influence on computational time. The need for future exploration was however suggested as it was noted that some uncertainties are being generated in the use of SVMs in hydrological modelling studies.

Following the nature of the contrasting views that surround the use of SVMs, it can be concluded that the use of SVMs in water-related studies exhibits certain degree of uncertainty. Hence, this might be considered a reason for its limited application in water-related studies.

2.2.4 Performance improvement methods

Having established some areas of concern in the use of SVM, efforts have been made by some researchers to address the aforementioned grey areas so as to improve the performance of SVM. One major approach that has been employed by researchers is the hybridization of SVM with other DDMs.

Chau and Wu (2010) developed a hybrid model by the integration of an ANN, SVR and SSA for daily rainfall prediction in China. The joint effect of the ANN-SVR model and SSA was then subjected to investigation. Results showed that

the hybrid model performed better than other individual models, as the incorporation of SSA ensured significant improvement in the rainfall forecast.

Kisi *et al.* (2012a) also applied a discrete wavelet transform and SVM conjunctively in forecasting monthly streamflow in Turkey. The performance of the hybrid model was thereafter compared with that of a single SVM. Results found that hybrid models could improve the forecast accuracy of the SVM model in forecasting monthly streamflows. Shabri and Suhartono (2012) developed a least-square support vector machine (LSSVM) for streamflow forecasting in Malaysia and compared its performance to ARIMA, ANN and SVM models. Results obtained showed that the hybridization of the SVM and least square methods could influence the performance of SVM models positively. Likewise, Bhagwat and Maity (2013) explored the potential of a LS-SVR for daily streamflow forecasting in India. Results obtained were in firm agreement with that of Shabri and Suhartono (2012), as the hybrid model was found to have better correspondence between observed and predicted values when compared with individual models.

It has also been noted that the performance of SVM models depends on the selection of kernel function and proper setting of its hyper-parameters (Rasouli *et al.* 2012; Sudheer *et al.* 2013). However, the effective calibration and optimization of SVM parameters as well as choice of kernel function has always been a major challenge and source of uncertainty in results obtained from SVMs (Lin *et al.* 2006). Consequently, researchers have developed various methods towards addressing this shortcoming. These include the development of algorithms for selection of kernels and optimization of SVM model parameters.

Wu *et al.* (2008) proposed a distributed SVR (D-SVR) in which a linear regression (LR) and two-step GA algorithm was used alongside Gaussian RBF kernel to find the optimal model parameters. The D-SVR model was applied for river flow prediction in the Yangtze River, China. Results indicated that the

proposed model gave better performance in comparison to the conventional SVR models. It was further reported that the computational time was dramatically reduced when compared with the conventional SVR. Rasouli *et al.* (2012) coupled a SVR to a GA optimization model, to determine the optimal hyper-parameters and kernel functions of the SVR model. The SVR model performed satisfactorily when employed to forecast short-term streamflow variability in Canada. A Bayesian neural network (BNN) was however found to have outperformed the SVR model.

More recently, in order to achieve appropriate setting of SVM model parameters, Sudheer *et al.* (2013) adopted an alternative method. A quantum-behaved particle swarm optimization (QPSO) method was used to optimize and fine-tune the SVM model parameters. The SVM-QPSO model was used to forecast monthly streamflows in two river gauging stations in India. Results showed that the SVM-QPSO model outperformed simple SVM and PSO models. Furthermore, the results showed clearly poor performance of the simple SVM models when the model parameters and their settings are not properly selected.

2.2.5 Advantages and disadvantages

The advantages and disadvantages associated with its use of SVMs in hydrological modelling studies are presented.

First of all, SVM is not a black-box model, but can be referred to as a semi-black-box model. The reason being that the model parameters are explicit in the manner with which the training patterns arrive at the final solution. Also, SVM showcases a definite model structure due to the fact that optimization in SVM is convex in nature and therefore has only one optimal point (Auria and Moro 2008). This serves as an advantage over the ANNs, as ANN models are usually characterized by several local optima, thus resulting into an arduous search for the most suitable optimal point (Jin *et al.* 2003).

Another advantage of SVM is its simplicity and potency in handling multi-dimensional inputs. Unlike in ANNs, where the process of optimizing model parameters (while using multi-dimensional inputs) culminates into complicated network architectures, the number of model parameters in SVMs remains constant with increase in input dimension (Jordaan 2002). Finally, the execution of the SRM principle in SVM rather than that of ERM as used in other DDMs facilitate better generalization on unseen datasets (Sudheer *et al.* 2013). Thus, the SRM principle makes SVM more robust against data characterized by outliers and noise (Sivapragasam *et al.* 2001; Bray and Han 2004).

One of the major limitations of SVM is its slow response and high computational demands, which is typical of SVM when presented with large amount of datasets (Lin *et al.* 2006). Vojinovic and Kecman (2004) pointed out that SVM may suffer from severe over-fitting if the ratio of support vectors to training samples is very high. Following the need for appropriate selection of kernel functions and hyper-parameters, it has been found that the search for an optimum model in SVM is often exhaustive and prohibitive in nature, resulting in very long computational times (Bray and Han 2004). Therefore, the process of finding an appropriate kernel for each specific task can be said to be highly laborious. The use of historical datasets as support vectors in SVM also limits its ability to extrapolate outside the range of the training datasets. Finally, Bray and Han (2004) remarked that SVM produces only point predictions, thereby making it unsuitable for use in probabilistic forecasts.

2.3 K-NEAREST NEIGHBOUR (K-NN) METHOD

2.3.1 Introduction

The K-nearest neighbour method belongs to the class of methods based on the working principle of instance-based learning (IBL) algorithms. IBL algorithms are algorithms that initialize by storing information from the training samples using

specific instances, and delays generalization effort until need arises for the prediction of a new query instance (Akbari *et al.* 2011). They apply experiential knowledge gained from initial events to generate details about relatively new instances. They achieve this by retrieving salient information from a set of nearest neighbours and building localized models based on them. The K-NN method is a typical representative of IBL, and thus operates by describing complex functions as a collection of less complex local approximations, with the letter, K, symbolizing the number of nearest neighbours. K-NN combines the target variables of K selected neighbours to determine the target outputs of a given test pattern. The pattern is represented by a limited number of illustrative observations referred to as “features”, and consequently characterized by a vector known as “feature vector” (Shamseldin and O’Connor 1996). This enables K-NN to recognize the feature vectors as a subgroup of the original pattern. Thus, K-NN is considered to be intuitive in nature, though it also exhibits highly influential statistical features (Karlsson and Yakowitz 1987).

In K-NN, the nearest neighbours are interpreted as a function of a Euclidean distance which is a measure of the proximity or similarity of a feature vector of query distance and any feature vector of the training sample (Galeati 1990). The Euclidean distance is often estimated as a weighted Euclidean norm. Furthermore, based on the Euclidean distance, each of the K neighbours is assigned a weight factor, so as to reveal the relative impact on the prediction value. The weights are computed such that they generate the lowest mean square error of forecasting over the training samples. A number of kernels have been used for the implementation of K-NN. They include linear, inverse, square inverse, exponential and Gaussian kernels (Akbari *et al.* 2011).

2.3.2 Application of K-NN in hydrological modelling

Karlsson and Yakowitz (1987) was the first to subject the K-NN method to use in hydrologic studies. K-NN method was applied to solve a univariate rainfall-runoff

forecasting problem. Results obtained showed a competitive performance between the K-NN, autoregressive moving average with auxiliary input (ARMAX) method and instantaneous unit hydrology (IUH) forecast methods. The satisfactory results obtained thereafter defined a robust theoretical foundation for subsequent use of the K-NN method. Galeati (1990) investigated the potential of K-NN in predicting daily average discharge in a rocky basin in Italy. Results yielded comparable results between K-NN and an autoregressive model with exogenous input (ARX). Kember *et al.* (1993) used the K-NN method to forecast daily river flow at a single site, and found it to provide improved forecasts. Shamseldin and O'Connor (1996) applied K-NN to fine-tune parameters of a linear perturbation model in river flow forecasting study, obtaining an improved and more reliable forecast. Solomatine *et al.* (2008) compared the performance of K-NN, ANN and M5 model trees for hourly and daily rainfall prediction. Results demonstrated that K-NN is comparable to other DDMs, especially when a Gaussian kernel is employed.

K-NN has also found application in climate change impact studies. Yates *et al.* (2003) found out that K-NN is capable of generating alternative climate information when conditioned upon hypothetical climate scenarios. Leander and Buishand (2007) applied the nearest neighbour method to resample outputs from a regional climate model (RCM), and its performance was remarkably satisfactory. Bannayan and Hoogenboom (2008) further testified to the reliable performance of the K-NN method when used for resampling daily temperature and precipitation events. Sharif and Burn (2006) used an improved K-NN for perturbation of historical datasets in a climate change assessment study of the upper Thames River Basin, Canada. Results showed that the K-NN was effective in producing the desired precipitation amounts.

2.3.3 Areas of concern

Some issues relating to the efficacy of the K-NN method have been raised via its application in some assessment studies**.** Toth *et al.* (2000) investigated the potential of the K-NN method comparatively with the ARMA and ANN methods for short-range rainfall prediction. Results demonstrated that the K-NN model failed to provide noteworthy predictive performance when compared with the ARMA and ANN methods. It was discovered that the improvement of the performance with an increasing number of nearest neighbours was less noticeable, with no marginal improvement in overall performance when K is increased beyond a certain limit. An investigative study conducted by Scheuber (2010) further corroborates Toth *et al.* (2000)'s findings, as he remarked that the selection of suitable parameters in the development of K-NN is quite challenging and could have negative impacts on algorithm performance. He further stressed that increasing K translates to the introduction of spatially more distant reference data, which consequently leads to higher degree of bias.

Kim and Tomppo (2006) carried out a prediction error uncertainty assessment of K-NN method, and found out that K-NN lacks the logical approach to compute error estimates for domains of arbitrary size. This is in agreement with results from Maltamo *et al.* (2003) and Gjertsen (2007)'s study in which over- and under-estimations of the lowest and highest historic observations were evident. This further shows that limitations exist to the methodological and analytical characteristics of the K-NN method.

2.3.4 Performance improvement methods

Several methods have been devised by experts towards improving the performance of the K-NN algorithm**.** Akbari *et al.* (2011) established that inconsistencies do occur in query instances in K-NN, which thereafter show up in the data points of output values, thus leading to the deterioration forecasting results. A clustered K-NN (CKNN) was introduced to capture inconsistencies in

data points, and was found to be robust against a set of noisy data. In addition, the CKNN demonstrated high level efficacy for daily inflow forecasting of a reservoir in Iran.

Magnussen *et al.* (2010) tested a model-based calibration method to reduce the out-of-sample extrapolation bias often associated with K-NN. Results showed that the calibrated K-NN predictions were considerably closer to observed values than regular non-calibrated predictions. Prairie *et al.* (2006) also developed a modified K-NN which involved the use of a probability metric to resample residuals from a traditional K-NN. The approach entails giving more weight to the nearest Neighbours and less to the farthest. The resultant model was applied to monthly streamflow in the Colorado River, United States, and was found to exhibit better performance in terms of capturing the patterns inherent in the datasets. Finally, in order to simplify the computational task of the K-NN algorithm, there is a need for reduction in the dimensionality of the K-NN feature space through its synergetic use with other transformation methods such as principal component analysis (PCA) (Kim and Tomppo 2006).

2.3.5 Advantages and disadvantages

The K-NN is a learning algorithm which exhibits simplicity and robustness, and thus can tolerate noise and irrelevant attributes. K-NN also possesses the ability to give a good representation of probabilistic and overlapping concepts simultaneously, and as well capable of naturally exploiting inter-attribute relationships (Aha *et al.* 1991). K-NN also gives room for the identification of past events (instances), which makes it more explicit in nature when compared with the ANN. This feature makes it suitable for use in hydrological studies, and hence its acceptability by forecasters.

On the other hand, K-NN does not have the ability to discover any input-output mapping functions, not even *a posteriori* like the ANN does (Benyahya *et al.*

2007). Thus, it has no extrapolation ability when presented with unfamiliar input vectors, meaning, it has no ability whatsoever to predict values higher than those in the range of the historic observations (Galeati 1990; Hessenmoller and Elsenhans 2002). This serves as a major disadvantage to the use of K-NN, as its credibility is severely limited when used for real forecasting.

2.4 CHAOS THEORY-BASED METHODS

2.4.1 Introduction

It has been established that hydrological processes is often characterized by high degree of complexity, nonlinearity and dynamism (Sivakumar 2003). However, certain aspects of the system may be less nonlinear than others and the nature of nonlinearity may not always be clear (Tsonis 2001). The hydrological domain can be considered as a chaotic dynamic system. In the nonlinear science literature, the term “chaos” refers to circumstances where complex and random-looking behaviours originate from simple nonlinear deterministic systems, showing sensitive dependency on initial conditions (Sivakumar and Berndtsson 2010). Thus, a deterministic system is considered to be chaotic whenever its sensitivity depends on the initial conditions. Generally, due to the fact that the experimental initial conditions of these systems are imperfectly understood, coupled with their dependency on initial conditions, the nonlinear systems is said to be essentially unpredictable (Youssef *et al.* 2013).

According to Sivakumar and Berndtsson (2010), there are three fundamental properties inherent in chaos theory when related to hydrologic systems and processes. The first property is referred to as nonlinear interdependence which represents the ‘general’ nature of hydrologic phenomena, whereas the second and third represents their ‘deterministic’ and ‘stochastic’ natures respectively. Thus, some methods have being applied for the purpose of chaos identification

and prediction. These methods include auto-correlation function (ACF), average mutual information (AMI), correlation dimension, power spectrum, Lyaponov exponent, Kolmogorov entropy, phase space reconstruction (PSR), etcetera (Li *et al.* 2013). Such chaos identification methods help in (i) determining the degree of dependence present in successive values of a time series (ii) estimating the dimensionality (iii) defining the delay time and reconstructing phase space (iv) the differentiation of chaotic signals from noise and (v) investigating convergence/divergence and predictability.

The phase space reconstruction (PSR) however is one of the most widely used method for chaos identification. It operates by representing the dynamics of a system using trajectories in 'phase space' graph or diagram (Elshorbagy *et al.* 2002a). This property implies that two trajectories emanating from two independent but neigbhouring conditions separate exponentially with time (Boccaletti *et al.* 2000). The trajectories of the phase space diagram describe the evolution of the system from the preliminary state to target points called 'attractors', and thus give a historical representation of the system. The region of attraction of these trajectories which in depicted in the phase space diagrams ensures the provision of qualitative information on the degree of complexity of the system. Thus, the ability of chaos theory-based method to predict the future from complex, nonlinear and dynamic historic instances makes it suitable for use in hydrological modelling studies.

2.4.2 Application of chaos theory in hydrological modelling

The application of chaos theory in hydrologic studies dates back to the late 1980s in Boston when Rodriguez-Iturbe *et al.* (1989) pioneered it use in modelling storm rainfall recorded at 15-seconds interval. A satisfactory realization was obtained from the results as trajectories of the system in the phase space gave a preliminary support to the presence of chaotic dynamics in the historical observations. Since then, investigations of nonlinearity and

application of chaos theory to hydrological processes have received increased attention. Chaos theory-based methods have found application of across various aspects of hydrology such as prediction of rainfall-runoff, river flow, lake water-levels, missing data estimation, noise reduction and determination of chaos in hydrological time series, etcetera (Jayawardena and Lai 1994; Solomatine *et al.* 2000; Sivakumar *et al.* 2001; Elshorbagy *et al.* 2002a, 2002b; Sivakumar *et al.* 2002; Wang *et al.* 2006b; Li *et al.* 2013).

Jayawardena and Lai (1994) used chaos identification methods to determine the presence of chaos in daily rainfall and streamflow time series measured from two stations in Hong Kong. Results confirmed the presence of chaos in the time series datasets. The chaos reconstruction method was found to be superior to ARMA. Sivakumar *et al.* (2001) investigated the possibility of the existence of chaos in monthly rainfall and runoff series at the Gota River Basin in Sweden. Results provided substantial evidence that combined rainfall-runoff processes may exhibit chaotic characteristics. In the same vein, Elshorbagy *et al.* (2002a) successfully applied the correlation dimension method to identify chaos in daily flows in the English River, Canada. The analysis of chaos approach was also productive in configuring two models for the purpose of estimating missing data.

Sivakumar *et al.* (2002) applied PSR to forecast river flow dynamics of the Chao Phraya River in Thailand. The performance of the PSR method was estimated comparatively with that of ANN. Though, it was reported that the two methods performed satisfactory, the PSR method was found to have considerable edge over the ANN in all facets of prediction. Li *et al.* (2013) used six chaos identification methods to determine the existence and intensity of chaos at three individual time scales. Results revealed low-dimensional chaos and minimal fraction of nonlinear intensity in the runoff dynamics at different time scales. The results also produced reasonable predictions to indicate the efficiency of the nonlinear prediction method.

2.4.3 Areas of concern

Going by the series of applications mentioned in the preceding section, it can be noted that some genuine efforts has been made towards the exploration of the potential of chaos theory for modelling and prediction of hydrological processes. However, the reliability of the chaos identification and prediction methods in identifying chaotic patterns has been under considerable debate. While some studies have denied the presence of chaos in hydrological systems (Wilcox *et al.* 1991; Pasternack 1999; Khan *et al.* 2005), others shared contrasting views over the presence of low-dimensional chaos in hydrological processes (Schertzer *et al.* 2002; Wang *et al.* 2006b). Thus, doubts and criticisms are being raised with regards to some key decision factors that influence its application. Among the factors are: data size, data noise, presence of zeros, delay time selection and selection of suitable parameter values (Sivakumar and Berndtsson 2010). Solomatine and Ostfeld (2008), in a similar manner, stated that chaos theory can only be applied successfully when the times series under study exhibits certain attributes such as periodicity and considerable nonlinearity, and when the time series is of adequate (considerable) length. Thus, chaos theory method has no universal applicability.

In light of the foregoing, further exploration is still desirable, as there is an absence of consensus on the type of nonlinearity that exists in hydrological processes and in the selection of key decision factors. More understanding into the degree of nonlinearity will give direction with respect to the type of model (linear or nonlinear) to be applied to a particular system.

Finally, the true potential of chaos theory in hydrology can only be realized when efforts are directed towards solving more challenging problems such as hydrologic scaling and model parameterization problems, rather than chaos identification and prediction from historical data (Sivakumar 2009).

2.4.4 Performance improvement methods

More recently, some studies have focused on improving the performance and acceptability of the chaos theory method. Cui and Jiang (2012) used a combination of the PSR method and a Binary Particle Swarm Optimization (BPSO) method for chaotic time series prediction. The BPSO was incorporated to search for the optimal parameters (time delay and embedding dimension) in a new local linear model by minimizing standard prediction errors. Results showed the viability and efficacy of proposed approach, as improvement in the predictive accuracy of the local linear model was evident. Jia *et al.* (2011) also developed a hybrid algorithm termed CGPSO, which is a combination of canonical PSO with chaotic and Gaussian local search methods. The CGPSO was used for parameter optimization, and was found to be successful in solving high-dimensional problems.

Zhu and Hao (2012) developed a hybrid model by combining chaos theory with a back-propagation neural network (BPNN). The hybrid model was applied in predicting chaotic flood sequences in the lower Yellow River in China. A considerable improvement in forecast accuracy was seen in the results produced by the hybrid model.

2.4.5 Advantages and disadvantages

The major advantage of the chaos theory-based method is in its ability to determine the extent of nonlinearity in hydrological processes. Chaos theory-based methods thus help in providing accurate predictions from historical observations, given that the historical time series exhibits chaotic nature. This characteristic ability of chaos theory-based methods however appears to be its limitation, as it makes it suitable for only chaotic hydrological series, thereby lacking universal applicability. In reality, a given set of historical time-series must indeed exhibit certain properties in order for chaos theory-based method to achieve meaningful success (Solomatine and Ostfeld 2008).

In addition, the selection of parameters and chaos identification methods can be considered to be challenging. Chaotic systems are highly sensitive to parameter variations and to initial conditions, because the chaos is a source of oscillation and instability (Youssef *et al.* 2013). Therefore, modellers face great difficulty in determining the appropriate parameters, and also in exploring different types of chaos identification methods one after the other (Li *et al.* 2013). Although, the development of performance improvement methods such as hybridization and optimization algorithms may result in minimized predictive error (Cui and Jiang 2012), it may also lead to higher computational demands (Ma 2013).

2.5 MODEL TREES (MTs)

2.5.1 Introduction

The model tree (MT) is a piecewise linear model unlike other nonlinear models such as the ANN and GP. MTs are extensions of classification and regression trees, in which computational process is represented by a hierarchical tree-like structure (Deo 2009). It comprises of a root node or decision point that subdivides into several other nodes and leaves. The process of developing the nodes and branches into a tree is based on the idea of splitting the input space into mutually exclusive domains, according to a predefined splitting criterion, progressively narrowing down the size of the domains (Galelli and Castelletti 2013). When the number of instances in a domain becomes smaller than the predefined value, the splitting of that domain is finally brought into a halt with the creation of a leaf. Each time a new instance is fed into the tree, it follows after a specified path in accordance with the splitting rules defined in the tree-building procedure. A linear regression (LR) model is thereafter developed for each domain, and thus formulates a piecewise linear function for the estimation of nonlinear relationship between input-output variables as shown in Figure 3.

The growing of model trees is carried out using algorithmic rules, which majorly is a function of the splitting criterion utilized. The M5 algorithm (Quinlan 1992) is commonly used for inducing a MT and its working principle is as follows. Assuming there exist a set of training samples (initial instance), *P*, characterized by values of a fixed set of (input) attributes and a corresponding target (output) value. The objective is to develop a model that relates a target value of the training samples to the values of their input attributes based on a divide-and-conquer method (Londhe and Charhate 2010).

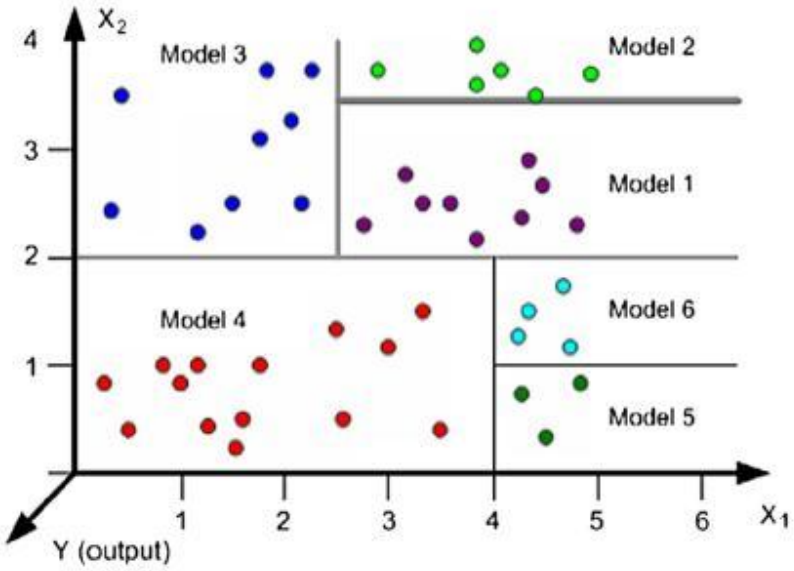

Figure 3: Splitting of the input space such as $X_1 \times X_2$ by M5 model tree algorithm; each model is a linear regression model $y = a_0 + a_1x_1 + a_2x_2$. Source: Jung *et al*. (2010).

The overall quality of the model will be determined by the accuracy with which it predicts the target values of set of new unseen data (new instance). The set *P* is either associated with a leaf, or some test is chosen that splits *P* into subsets corresponding to the test outcomes and the same process is applied recursively to the subsets. The splitting criterion for the M5 algorithm is based on treating

the standard deviation of the class values that reach the node, as a measure of the error at that node. Thus, the variable that maximizes this error reduction is chosen for splitting at that node. The mathematical expression for representing the standard deviation reduction (SDR) is given in equation (1):

$$SDR = sd(P) - \sum_i \frac{|P_i|}{|P|} sd(P_i) \quad (1)$$

where *SDR* = standard deviation reduction; *sd(P)* = standard deviation of all the training samples having a total number, *P*; P_i = i-th subset of *P*; and $sd(P_i)$ = standard deviation of the i-th subset.

Upon the exploration of all possible splits of input space, the M5 algorithm finally selects the one with the maximum value of SDR, which it then uses to develop linear regression models in the individual domains. Splitting terminates when the outputs of all the data that reach the node vary slightly or when only a few remain (Shrestha and Solomatine 2008). Pruning of the trees is carried out right away to prevent the problem of over-fitting which may occur as a result of monotonic increase in the training samples during tree growth (Jung *et al.* 2010). Finally, a 'smoothing' operation is performed to compensate for sharp discontinuities that inevitably occur between adjacent linear models at the leaves of the pruned tree. The smoothing operation thus updates the predicted values from neigbhouring equations in order to achieve better agreement as a whole.

2.5.2 Application of MTs in hydrological modelling

The M5 MT technique can be considered to be relatively new in solving hydrological problems, with its first application in rainfall-runoff prediction reported by Kompare *et al.* (1997). Water-related areas in which MT have been successfully applied include rainfall-runoff and streamflow modelling

(Solomatine and Dulal 2003; Khan and See 2006; Jagtap 2013); sedimentation modelling (Bhattacharya and Solomatine 2006); modelling of water pollutants (Ostfeld and Salomons 2005; Preis *et al.* 2006); and climate change impact modelling (Goyal and Ojha 2013; Nasseri *et al.* 2013).

Comparative studies have also been conducted with the aim of estimating the potential of M5 MTs against other DDMs. Solomatine and Dulal (2003) applied M5 MTs for rainfall-runoff modelling at different hourly time-slices, and compared its performance to that of ANNs. The performance of both models were said to be comparable, although the ANNs performed slightly better at higher lead times. However, M5 MTs produced more interpretable models and also allowed for development of modular models of varying complexity and accuracy. Khan and See (2006) employed one statistical model and three DDMs namely MLR, ANN, M5 MTs and evolutionary neural network (Evo-NN) in river level forecasting. The study was carried out in the Ouse River catchment located in Northern England. Results showed that the M5 and Evo-NN models provided the best performance based on global performance measures. Furthermore, it was observed that the M5 model demonstrated its ability to make explicit its internal structures, unlike other black-box models such as the ANNs.

In a quest for more accurate predictive models in hydrology, M5 MTs have gained some recognition in the development of modular and hybrid models, due to its transparent and intelligible nature. Solomatine and Xue (2004) built a modular model comprising of M5 MT and ANN which was applied to flood forecasting in the upper Huai River, China. Flood samples with different hydrological features also split into groups using separate M5 and ANN models. Improved accuracy in predicting high floods was generated by the modular model when compared with both individual models. The authors also attested to the fact that the incorporation of M5 MTs ensured transparency of the internal features inherent in the hydrological processes. Likewise, Bhattacharya and

Solomatine (2005) constructed a model with an ANN model and M5 MT for river stage-discharge modelling. The model was found to be superior in accuracy to a conventional stage-discharge rating curve, most especially during periods of high flow. They finally submitted that the M5 algorithm did not only allow for higher accuracy, but was also being transparent, simple, verifiable and easily demonstrable. Results from Ajmera and Goyal (2012)'s stage-discharge modelling study also supported this claim, as the M5 MTs outperformed three different ANN algorithms as well as the conventional stage-discharge method.

2.5.3 Areas of concern

Some issues relating to the application of M5 model trees in hydrological studies have been identified. One of such issues which have been reported in literature is the partitioning or splitting problem. Solomatine and Dulal (2003) observed from their study that results from their M5 model at higher lead times and peak flows produced sub-standard predictions compared with the ANN model. This was ascribed to the splitting criteria used to build linear models at the leaves. They noticed that model trees do not use all available attributes to make linear models at any leaf. Only attributes which fulfill the condition of certain criteria (such as SDR) go under one sub-tree, terminating to a leaf. This therefore may have resulted into the non-inclusion of influencing attributes. In addition, the resultant model tree was deemed to be so large, and an attempt made to prune it to a smaller size led to deterioration of accuracy. Thus, it can be inferred that unsupervised pruning operation of model trees could result into poor predictive performance.

Londhe and Charhate (2010) also reported cases of over-estimations of peak values in their river flow forecasts, which were attributed to the absence of influencing attributes in the development of the linear models. Bhattacharya and Solomatine (2006) additionally suggested that further improvement in terms of performance could be achieved by including additional information about

physical processes, using larger datasets or by exploring other machine learning methods.

In furtherance to the abovementioned, the ability of MTs to rapidly increase computational requirements when confronted with high dimensionality have also attracted a bit of concern. Although, such ability enables MTs to learn efficiently and undertake tasks with high dimensionality (Solomatine and Dulal 2003); it could however lead to extremely high computational demands (Ganesan 2011).

2.5.4 Performance improvement methods

Some efforts have been directed towards addressing the aforementioned areas of concern. An M5' algorithm, a modification to the M5 algorithm was presented by Wang and Witten (1996). M5' allows for pruning of the tree size with minimal penalty in prediction performance via the incorporation of a pruning factor, which can be specified by the user. Thus, the M5' algorithm seemingly outperformed the original M5 algorithm. Samadi *et al.* (2012) tested M5' MT for prediction of scour depth below free overall spillways, and evaluated its performance against classification and regression trees (CART). The results indicated that M5' MT produced better predictions than the CART method. Mafi *et al.* (2013) also applied M5' for modelling long-shore sediment transport rate (LSTR), and its performance compared to that of existing empirical equations initially proposed for such purpose. Results found that equations derived from the M5' MT method predicted LSTR more accurately than the existing formulas, producing lesser error estimates. This result is also agreement with that obtained from Nasseri *et al.* (2013)'s climate change modelling study. They employed a combination of M5' algorithm and three other nonlinear data mining methods in developing a nonlinear data mining downscaling model (NDMDM). The performance of the NDMDM model was evaluated comparatively with a popular statistical downscaling model (SDSM), using daily precipitation events. Results indicated

better performance of the NDMDM model when a combination of M5' and multivariate adaptive regression splines (MARS) methods was employed.

Asides the applications of the M5' algorithm, some experts have also proposed other methods aimed at solving the partitioning problem in M5 model trees. Rather than use piecewise linear regression models at the leaf nodes of the model tree, Jung *et al.* (2010) employed partial least square regression (PLSR), which is an extension of multivariate linear regression. The M5-PLSR MTs was thereafter compared with the M5' MTs, MLF- and RBF-ANN and K-NN for algal growth prediction in a Korean reservoir. Results show improved prediction by both M5' and M5-PLSR MTs using partitioned datasets, with the M5-PLSR MTs performing better than other algorithms via the use of more closely correlated multivariate datasets.

Hong and Chen (2012) proposed an extension of sample efficient regression tree (SERT) approach, which entails the fusion of the forward selection of regression analysis and the regression tree methodologies. This was done with the aim of maximizing the degree of freedom of the datasets and also to obtain unbiased model estimates. The outcome of the study was the realization of an unbiased MT. Recently, an innovative method termed turning point regression tree induction (TPRTI) was developed by Amalaman *et al.* (2013) to determine optimal split points in MTs. The workings of the TPRTI method include: division a set of data into subsets using a sliding window, computation of a centroid for each subset and use of the centroid for identifying turning points which indicates where general trend changes in the input space. The novel approach was compared with the M5 algorithm using synthetic and real-life datasets for experimentation. Results from the TPRTI method showed higher predictive accuracy and improvement in scalability and low model complexity when compared with the M5 algorithm.

However, irrespective of the recent developmental concepts which have resulted into improved accuracy of MTs, the need to reduce its high computational demands is still of concern to modellers (Ganesan 2011). Galelli and Castelletti (2013) recently advised on the need for development of techniques that will strike a balance between accuracy and computational requirements.

2.5.5 Advantages and disadvantages

The applications of MTs in hydrological studies have been found to have certain advantages. Such advantages include its ability to produce simple, accurate, transparent, provable and understandable models (Bhattacharya and Solomatine 2005; Jung *et al.* 2010). M5-derived models have also been found to consistently showcase high rate of convergence, especially under fewer events (Singh *et al.* 2010; Ajmera and Goyal 2012). The pruning operation performed on newly built model trees also help to counter overfitting problems (Jung *et al.* 2010), provided the operation is well supervised. Furthermore, the partitioning of the input space of model trees allows for the combination of several local linear models, which consequently results in improved model accuracy (Sattari *et al.* 2013).

However, certain drawbacks to the use of model trees have also been identified. They include: (i) partitioning problems which normally occur when the ratio of instances (observations) is smaller than the number of attributes (variables) (Jung *et al.* 2010); (ii) need for high level of expertise for model implementation, especially as it relates to the achievement of optimal partitioning and pruning (Solomatine and Siek 2006); (iii) high computational demands and fallible performance when used to model data with high dimensionality and nonlinear features (Pal *et al.* 2012; Garg and Jothiprakash 2013); and (iv) generation of equations that are easily verifiable but not realistic in terms of physical interpretation (Londhe and Charhate 2010) .

2.6 FUZZY RULE-BASED SYSTEMS (FRBS)

2.6.1 Introduction

Fuzzy rule-based systems are based on the method of fuzzy logic, established by (Zadeh 1965). FRBS are built by simulating the reasoning process of humans in order to achieve transparency in modelling processes (Bourdin *et al.* 2012). In FRBS, knowledge is represented using IF-THEN rules (Adriaenssens *et al.* 2004). A typical rule is represented as IF-(antecedent part)-THEN (consequence part) (Katambara and Ndiritu 2009). The initialization procedure of an FRBS model entails choosing input and output variables which it uses in defining fuzzy sets. The FRBS model thereafter uses the fuzzy sets to construct membership functions on the input domain of the model. This is achieved by partitioning the domain into a number of overlapping regions. The membership functions are represented using quantitative and linguistic terms. Several types of membership functions that can be used for the fuzzy set in the antecedent of the rules include triangular and Gaussian functions.

Relationship between membership functions of the inputs and that of the outputs are expressed by using linguistic logical statements based on the subjective knowledge of the modeller. For example, river flow can be categorized as “low”, “intermediate” and “high”. Thus, matching of the inputs and outputs with fuzzy rules can be expressed as:

“IF *Input* 1 is LOW and *Input* 2 is HIGH THEN *Output* is INTERMEDIATE”

The structure of the rule could either be a fuzzy set such as the Mamdani model (Mamdani 1974), or a function, often linear referred to as a Takagi-Sugeno-Kang (TSK) model (Takagi and Sugeno 1985; Sugeno and Kang 1988). Upon the formulation of a rule, an iterative and tuning process is introduced to relate observations to the rules. Finally, FRBS combines the fuzzy rules through an inference engine, and “defuzzification” is used to collapse the fuzzy or estimated

model output into a single crisp value (Bourdin *et al.* 2012). Conventional defuzzification methods include center of area (or gravity) method, bisector methods, and some other methods which focuses on the maximum membership value attained by the set. Following the working principle of FRBS, hydrological systems can be modelled through the processing of historical observations and mapping of input-output variables, thus forming a DDM.

2.6.2 Applications of FRBS in hydrological modelling

FRBS have found application in several water-related studies, such as rainfall-runoff modelling, management of reservoir operations, streamflow and water-level forecasting, ecological and sediment yield modelling (See and Openshaw 2000; Xiong *et al.* 2001; Adriaenssens *et al.* 2004; Chang and Chang 2006; Katambara and Ndiritu 2009; Kisi *et al.* 2012b; Goyal *et al.* 2013).

FRBS have also been found to be effective for river flow prediction purposes. Valença and Ludermir (2000) used a neuro-fuzzy network model for monthly streamflow forecasting for a hydropower plant in Brazil. Results showed that the neuro-fuzzy model provided better predictions than models based on the traditional Box-Jenkins method. Aqil *et al.* (2007b) evaluated the potential of a neuro-fuzzy model for the purpose of predicting flow from local source in the Citarum River in Indonesia. The performance of the neuro-fuzzy model was compared with a MLR. It was reported that the neuro-fuzzy model gave better performance for low and medium flows, but underestimated the magnitude of high flows. Katambara and Ndiritu (2009) applied FRBS for simulating daily streamflows at three reaches of the Letaba River in South Africa. Satisfactory results were obtained from the FRBS model despite the irregular and intermittent water abstractions that characterize the river.

Recently, the use of an adaptive neural fuzzy inference systems (ANFIS), a hybrid of FRBS and ANN approaches have been widely reported in river flow

prediction studies (Chau *et al.* 2005; Keskin *et al.* 2006; Wang *et al.* 2009), with results showing that the ANFIS performed slightly better when compared with ARMA and/or ANN.

FRBS have also found application in the development of rainfall-runoff models. See and Openshaw (2000) integrated a hybrid neural network, an ARMA model and a fuzzy rule-based model for rainfall-runoff forecasting. Hundecha *et al.* (2001) formulated a set of fuzzy rule-based routines to independently simulate components of snowmelt, evaporation, runoff and basin response for a physically-based (HBV) model. With the aim of automatically generating IF-THEN rules from historical observations, the first-order Takagi-Sugeno models were used to simulate rainfall-runoff transformation (Xiong *et al.* 2001; Vernieuwe *et al.* 2005; Jacquin and Shamseldin 2006). Luchetta and Manetti (2003) employed fuzzy logic to predict rainfall-runoff dynamics, and comparison with ANN approach indicated a better performance by the fuzzy method. Nayak *et al.* (2007) developed a rainfall-runoff model by combining fuzzy logic and ANN. Results from the hybrid models were found to be comparable with an ANFIS model, with a suggestion that the hybrid model could be used as a promising alternative to ANFIS for rainfall-runoff modelling purposes.

FRBS have also been used conjunctively with other models or optimization algorithms. Chang and Chen (2001) used a counter-propagation fuzzy-neural network (CFNN) rainfall-runoff model for hourly streamflow forecasting, and was found to be better in terms of prediction accuracy when compared with the ARMAX model. Maskey *et al.* (2004) used a combination of FBRS and GA for treating precipitation uncertainty in a rainfall-runoff model. Shiri and Kisi (2010) applied a hybrid wavelet neuro-fuzzy model to investigate daily, monthly and annual streamflows in the Filyos River in Turkey. Bagis and Karaboga (2004) compared FRBS and neural-fuzzy network system, and results indicated that the former was effective for regular reservoir operations, while the latter was

effective primarily for flood control. Katambara and Ndiritu (2010) developed a calibrated hybrid conceptual-fuzzy-logic model and investigated its potential in simulating daily streamflow in the Letaba River, South Africa. The model performance was satisfactory regardless of the intricacy of the system and insufficiency of relevant data. The study suggested that hybrid conceptual-fuzzy logic-modelling could be adopted for more detailed and reliable planning analysis than single FRBS modelling, especially when used to model complex data-scarce river systems.

2.6.3 Areas of concern

Despite the successful application of FRBS in modelling studies, some areas of concern have been identified. An area that seems to be of major challenge in the use of FRBS is the identification of the optimal number of rules to achieve the best performance. Abebe *et al.* (2000) noted that improper selection of fuzzy-rules could lead to extreme generalization and overfitting, especially when faced with data insufficiency. It was however suggested to carry out test runs in order to determine the optimal number of rules. Aqil *et al.* (2007a) also attributed the failure of their neuro-fuzzy model in underestimations of high flows to the FRBS, which was unable to make proper fuzzy rules corresponding to a high-range from the training datasets. They however encouraged the inclusion of another input variable for the refinement of the fuzzy rule. Katambara and Ndiritu (2010) corroborated this claim by stating that their stand-alone fuzzy-systems performed poorly in simulating baseline recession due to the lack of subjective transformation rainfall and evaporation, and therefore failed to obtain realistic modelling. Thus, the representation of input-output relationships by FRBS may be hampered if optimal fuzzy rules are not employed.

In addition, since the determination of optimal rules in FRBS entails choosing appropriate input variables and membership functions, the choice of input variables and membership functions is therefore important. It has been

established that FRBS rules increase exponentially with increase in input variables or membership functions, imposing a curse of dimensionality of the system, which translates into poor model performance (Li *et al.* 2010). Thus improper selection of input variables and membership functions may result into addition of unnecessary variables that will create a more complex model than required, and further complicate the rule selection process.

Another issue that is of concern in the application of FRBS to hydrological modellers is the partitioning of the input domain. In FRBS, the number of partition represents the number of fuzzy sets, and the corresponding membership function defined in that order (Nayak *et al.* 2005). As a result, the partitioning of the input domain is vital to achieving optimal configuration of the model. The partitioning of input domain is done using different methods – grid partitioning and fuzzy clustering. However, the challenge remains as to which method to employ, as both methods come with their strength and limitations. The major limitation of grid partitioning are that the number of rules increase exponentially (Nayak *et al.* 2005), and that the membership functions of the variables are independent of each other, leading to the disregard of the relationship between the variables (Vernieuwe *et al.* 2005). Moreover, attempt to optimize the antecedent parameters leads to the more complexity. In fuzzy clustering, the antecedent parameters are obtained from fuzzy clusters. The drawback to the use of fuzzy clustering is that, the falling off of any data point from the cluster center or outside the cluster affects the model performance.

2.6.4 Performance improvement methods

Following the above-mentioned areas of concern, some techniques have been employed with respect to improving the performance of FRBS. Chang and Chang (2001) introduced GA for the purpose of synergizing it with the ANFIS model. GA was used to achieve several objectives such as searching for the optimal reservoir operation histogram, construction of suitable model structure

and parameters for the ANFIS model and estimation of optimal water release from the reservoir. Abolpour *et al.* (2007) developed a new approach called "adaptive neural fuzzy reinforcement learning" (ANFRL) which was derived by the combination of ANFIS and fuzzy reinforcement learning. The ANFRL method was used for optimizing allocation of water resources and to obtain optimal values of decision parameters. Results also showed that ANFRL brought about an increase of 16% in water allocation; a value considered not attainable by the regular ANFIS model. However, two limitations to the use of ANFRL were identified. The first is its requirement for long series of hydrological data in deriving a robust set of rules, and the second – the performance of the ANFRL-derived models is a function of the ability of the ANFIS model to handle data variability. Thus, if the ANFIS model does not yield suitable estimation of water resources variability, then the ANFRL results will not be accurate.

Recently, Akrami *et al.* (2013) applied a new method earlier proposed by Jovanovic *et al.* (2004) to model the dynamic nonlinear behaviour of rainfall. The new method, referred to as modified ANFIS (MANFIS) entails modification to the structure of the conventional ANFIS model, with the aim of improving its performance. MANFIS was used to find the optimal number of rules, discover the appropriate membership functions and learning algorithm. A hybrid learning algorithm which combines the least square and back-propagation gradient descent methods was used to train membership function parameters. Results showed that the MANFIS method when compared to the conventional ANFIS method produced faster convergence and low computational while maintaining outstanding performance. However, improved performance was not noticeable when using additional input member functions.

Generally, majority of the studies have demonstrated that FRBS has good predictive abilities. However, some studies have shown that the predictions produced by FRBS are quite inferior to that of other simple conventional models,

such as ARMA (See and Openshaw 2000) or MLR (Jacquin *et al.* 2004). These results therefore substantiate the need to subject the quality of a model to test for each given situation, as there is no such flawless model that will perform well, at all times, in solving all modelling problems (Maier and Dandy 2000).

2.6.5 Advantages and disadvantages

Going by the applications of FRBS discussed in the previous sections, the advantages that can be derived from its use as stated by Jacquin and Shamseldin (2006) include: (i) ability to infer the nature of complex systems purely from data, while also providing insight about their internal operations; (ii) ability to present knowledge that can easily be interpretable by humans; (iii) subjective knowledge by expert can be incorporated into the model in a natural and transparent way; (iv) flexibility of use, as their architecture and inference mechanisms can be adapted to a given modelling problem. Additional advantages of the FRBS method according to Aqil *et al.* (2007a) include: (v) ability to handle large amount of noisy data from dynamic and nonlinear systems, especially where the underlying physics of the system is not fully understood; (vi) ability to improve the performance of other models, and (vii) fast model development with less computation time, provided the input vector space is well-dimensioned.

The disadvantages of FRBS however include: (i) inability to provide adequate representation of input-output relationships in cases where too many variables are required or when used for modelling highly complex systems (Li *et al.* 2010); (ii) suffers from "the curse of dimensionality", where the number of fuzzy rules increases exponentially with little increment of inputs, leading additional complexity and higher computational time (Akrami *et al.* 2013); (iii) attempts to reduce the number of rules generally decreases model generalization ability (Li *et al.* 2010) (iv) lacks an appropriate set of guidelines for calibrating model

parameters in a way that will maximize model interpretability (Jacquin and Shamseldin 2009).

2.7 ARTIFICIAL NEURAL NETWORKS (ANNs)

2.7.1 Introduction

The inspiration for development of ANN originated from the desire to develop a system that mirrors the working principles of the human brain, in making intelligent decisions. ANNs consist of a large pool of processing units (generally referred to as neurons or nodes) which receive, process, and send information to each over a large number of weighted connections. Thus, ANNs gain experiential knowledge through a process of finding an optimal set of weights for the connections and threshold values (biases) for the neurons. In the same manner as a human brain works, the knowledge gained by ANNs is stored and made available for future use (Elshorbagy *et al.* 2010).

ANNs can be classified as single, bilayer and multilayers according to the number of layers, and as feed-forward, recurrent and self-organizing according to the direction of information flow and processing (ASCE 2000a). The most popular ANNs are multilayer feed-forward neural networks, which are mainly multilayer perceptron (MLP) and radial basis function networks (RBF). The MLP is the most widely-used in hydrological studies, thus this section concentrates on the MLP network. Other network topologies that have found application in hydrology include RBF (Lin and Chen 2004; Kagoda *et al.* 2010), Bayesian neural network (BNN) (Khan and Coulibaly 2006; Zhang *et al.* 2009), Kohenen self-organizing feature maps (SOFM) (Hsu *et al.* 2002; Kalteh and Berndtsson 2007; Kalteh *et al.* 2008), and Stacked neural network (SNN) (Boucher *et al.* 2009; Yonaba *et al.* 2010).

A MLP network comprise of an input layer, one or more hidden layers and an output layer. Figure 4 presents a three-layer feed-forward neural network with

each layer one or more neurons. Each neuron is represented by a circle and each connection weight by a line. The structure of an individual neuron is also presented. Each individual neuron computes an output, based on the weighted sum of all its inputs, according to a nonlinear function called the activation function (Kalteh *et al.* 2008). Thus, in reality, the function of the neuron is to receive input from neigbhouring or external sources, and compute an output signal which is propagated to other units.

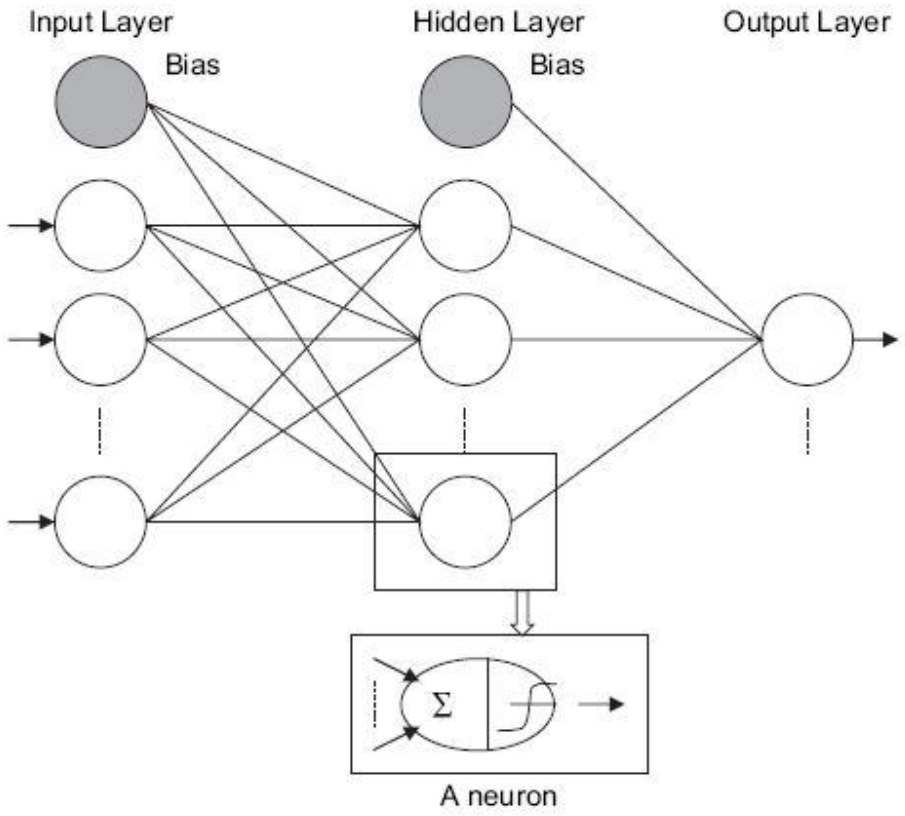

Figure 4: Structure of a feed-forward neural network (FFNN) (Kalteh et al. 2008)

The activation function used in an ANN is dependent on the type, training algorithm and the data scaling technique. The activation function may be logistic sigmoid, linear, threshold, Gaussian or hyperbolic tangent sigmoid functions. The activation functions that are commonly used in hydrological studies are the

sigmoidal-type functions such as the hyperbolic tangent and logistic sigmoidal-type functions (Maier and Dandy 2000), which can be computed using equations (2) and (3) respectively.

$$f(x) = \frac{2}{1+e^{-2(x)}} - 1 \tag{2}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

In equations (2) and (3), x is the weighted sum of inputs to the neuron, and $f(x)$ is the neuron's output.

The connection weights reflect the contribution of each input to the neuron, and the threshold value (bias) must be exceeded before the node can be activated. The connection weights and biases are parameters that can be determined by training the network using a number of training samples.

In order to develop an ANN, there are series of processes to follow. According to Maier and Dandy (2000); Dawson and Wilby (2001) and Abrahart *et al.* (2012), model development majorly entails data gathering, data splitting, data handling/preprocessing (i.e. data cleaning and standardization), determination of model inputs, selection of ANN (in terms of model architecture, activation function and training algorithm), network training and testing. These processes are however not strictly sequential or exclusive, but oftentimes mixed. The following sections present a discussion on each of the processes.

2.7.2 Data splitting

In data-driven modelling, the norm is to split a given set of historical observations into training and validation sets. In data splitting, there is a need to ensure that the training and validation datasets are representatives of the same population, because ANNs are unable to extrapolate beyond the range of the

training datasets. It has been established that when validation datasets fall outside the range of the training datasets, there is high tendency to produce poor predictions (Maier and Dandy 2000). This usually occurs when a limited amount of dataset is employed.

According to Maier and Dandy (2000), measures that have been applied to maximize utilization of available datasets include the use of “hold-out method” (Masters 1993) which involves withholding a small portion of the data for validation and training the network on the remaining datasets. Once the generalization ability of the network is obtained using the validation set, a different subset of data is withheld and the above process repeated, until the generalization ability is ascertained for all of the available data. Another method that is being used to maximize available data is the “cross-validation” technique. This method involves using an independent test set to evaluate the performance of the model at different phases of learning. In cross-validation method, the norm is to split the available data into three parts namely: training, testing and validation sets. The training set is used to train a number of different ANN configurations. The test set is also used to decide when to stop training so as to avoid overfitting, and also to determine the model with optimal network architecture. The validation set is used to evaluate the selected model against unseen data. The method however requires large amount of data and is not normally applied when the amount of datasets is limited. A comprehensive review of other methods for maximizing available training data has been presented by Maier and Dandy (2000).

2.7.3 Data handling/preprocessing

Having established the need to split available datasets into subsets, the need for efficient handling/pre-processing of the datasets is of high importance, as variability exists across the different subsets. Thus, data pre-processing can have significant impacts on model performance, as it ensures the reduction of

noise in data. Maier and Dandy (2000) advocated that all variables should be standardized to ensure that they all receive equal attention during the training process. Generally, standardization refers to rescaling of datasets as to fall within the limits of the activation function employed in the output layer. Minns and Hall (1996) further advised that scaling of data must be done to a range that is proportionate with the bounds of activation function in the output layer, otherwise input variables measured at different scales will to a certain extent control the training process (Dawson and Wilby 2001).

Another method used in data-preprocessing is normalization, which involves rescaling the data to a Gaussian function (often achieved by subtracting the mean and dividing by the standard deviation). Maier and Dandy (2000) however cautioned against the scaling of data to the extreme ranges of the activation function as this reduces the size of the weight updates, thereby resulting to flat spots during training. Abrahart *et al.* (2012) advocated for a systematic inquiry into the use of deseasonalized hydrological data, as series of mixed results show that such inputs may either produce better or worse results than those obtained from the original records with no preprocessing. Other noise-reduction techniques that have been used in water resources applications include wavelet transforms (WT) (Partal and Cigizoglu 2008; Kişi 2009) and single spectrum analysis (SSA) (Sivapragasam *et al.* 2001; Baratta *et al.* 2003).

2.7.4 Determination of model inputs

One of the initial steps in carrying out any modelling study is to collect data of causal variables that may possibly influence the modelled hydrological system. These set of data are often large in size, and thus need to be subjected to a screening process. This will ensure the selection of only input variables that will give the best representation of the system being modelled. Although, DDMs possess the ability to determine the important model inputs, presenting a large amount of inputs to ANNs and relying on the network to determine the important

ones usually increase the amount of data required to effectively compute the connection weights (Lachtermacher and Fuller 1994). Furthermore, inclusion of irrelevant inputs increases the network size, heightens model complexity, reduces model interpretability, slows down the learning process and consequently leads to mis-convergence (Bowden *et al.* 2005a).

Several methods of input determination have been employed in hydrological studies. According to Bowden *et al.* (2005a), the main input determination methods can be broadly categorized into five namely: (i) methods based on *a priori* knowledge of the modelled system; (ii) correlation analysis; (iii) a heuristic approach; (iv) knowledge extraction; and (v) composite methods.

Methods based on *a priori* knowledge of the modelled system have been widely applied in various water-related modelling studies. The use of *a priori* methods in determining the appropriate input variables rely on the level of understanding of the modeller on the modelled system (ASCE 2000a). This implies that effectiveness of the approach is dependent on the expert's knowledge, and hence, very subjective and case-dependent.

Correlation-based methods are often used for analytical purposes, most especially when the system to be modelled is less understood. They help provide insight into the relationships that exists between the system processes. Correlation-based methods that have been used for input variable selection include linear correlation analysis and nonlinear correlation analysis. Methods based on linear correlation analysis include cross correlation (CC) (Sudheer *et al.* 2002; Aqil *et al.* 2007b), autocorrelation (AC) (De Vos and Rientjes 2005; Londhe and Charhate 2010), and partial autocorrelation (PAC) (Sudheer and Jain 2004; Jain and Kumar 2007). Nonlinear correlations analysis is sometimes employed due to the inability of linear correlation analysis to capture nonlinear relationships. Nonlinear correlation methods used in hydrological studies include average mutual information (AMI) (Abebe and Price 2003; Bhattacharya and

Solomatine 2005), and partial mutual information (PMI) (Bowden *et al.* 2005b; Fernando *et al.* 2009).

Heuristic methods involve training several ANN models using different combination of inputs until the best model is identified (Baratti *et al.* 2003; Toth 2009). Another heuristic approach used in hydrological studies is the stepwise selection approach. The stepwise selection approach serves as an alternative to the former approach, as it eliminates the need for enumerating all the input combinations. It involves training individual networks for each input variable. Two stepwise approaches usually utilized include forward selection and backward selection stepwise approach. The limitation to the use of the heuristic approach is that they are based on trial-and-error, thus there is no guarantee to finding the globally best subsets. In addition, the implementation of the stepwise approach is also highly computationally expensive (Bowden *et al.* 2005a).

The knowledge extraction method as the name implies, entails extracting knowledge from the trained ANNs. The most widely-used method of extracting knowledge from trained ANNs is the sensitivity method (Dibike and Coulibaly 2006; Khan and Coulibaly 2006). In the sensitivity method, graphical representations of the sensitivities of the inputs are critically examined and the significant inputs are selected based on human assessment. However, the difficulty with this method is the determination of the plausible value to perturb the input by choosing the suitable cut-off point for input significance (Bowden *et al.* 2005a).

Composite methods refer to the conjunctive use of any of the afore-mentioned approaches. Silverman and Dracup (2000) used a combination of *a priori* knowledge and trial-and-error method to determine ANN model inputs for long-term precipitation forecasts in California, USA. In addition, the conjunctive use of *a priori* knowledge and other methods have been endorsed as a viable approach for determining suitable model inputs and their corresponding lags (Solomatine

and Dulal 2003; Yu *et al.* 2006). Other methods used in the determination of model inputs but not captured in the above categories include principal component analysis (PCA) which is often used when the number of inputs is large (Hu *et al.* 2007; Al-Alawi *et al.* 2008), and evolutionary optimization methods such as GA and multi-objective GA to simultaneously optimize the model inputs and network architecture (Abrahart *et al.* 1999; Giustolisi and Simeone 2006).

2.7.5 Determination of network architecture

Determination of the network architecture is one of the most important, but also one of the most difficult tasks in model building process (Maier and Dandy 2000). The design of ANN architecture consists of the number of hidden layers, number of neurons in the input layer, hidden layer(s) and output layer. The most widely adopted ANN architecture for prediction and forecasting of hydrological variables is the one hidden-layered FFNN (Dibike *et al.* 2001), since its networks are considered to provide adequate complexity to precisely simulate the dynamic and nonlinear behaviour of hydrological processes (De Vos and Rientjes 2005). Therefore, the major task in building FFNN architecture is either specifying model inputs and the number of hidden layer nodes, or only the number of hidden layer nodes, given that the number of model inputs has been predetermined. The optimal network architecture generally strikes a balance between generalization ability and network complexity (Maier *et al.* 2010).

According to Maier *et al.* (2010), methods for determining the optimal network architecture for ANNs can be categorized into three namely, global, stepwise, or ad-hoc methods. The global methods comprise of methods which uses the principle of evolutionary search to find the optimal network architecture. Examples of such methods include GA, DE, PSO, simulated annealing etcetera (Yao and Liu 1997; Giustolisi and Simeone 2006; Subudhi and Jena 2011). These methods allows for simultaneous optimization of the network parameters

(network weights and biases) and the model structure (number of hidden layer nodes). Maier *et al.* (2010) further opined that if the global methods are used appropriately, they are likely to produce the best model structure and/or parameters; however, they are computationally expensive.

An alternative to the use of global methods is to employ a stepwise trial-and-error approach, which first gives an assumption of a basic ANN structure, and thereafter adjusts each trial with the aim of achieving a structure that is neither too complex nor too simple. The stepwise methods can be further subdivided into two categories, one based on pruning algorithms, and the other on constructive algorithms (Abrahart *et al.* 1999; Sharma and Chandra 2010). The pruning algorithm initializes with a sufficiently complex model structure and sequentially removes network parameters and the associated neurons one after the other, until model performance deteriorates significantly. Constructive algorithms however start-off with the simplest model structure and sequentially adds hidden neurons one at a time, until no significant improvement in model performance is noticed. The use of pruning and constructive algorithms also require high computational resources, as different model structures need to be trained and manually examined before obtaining the optimal structure (Maier *et al.* 2010).

The determination of appropriate model structure and optimal number of hidden layer nodes via pure trial-and-error approach, *a priori* knowledge or other approaches outside the two methods earlier discussed can be categorized as ad-hoc. Maier *et al.* (2010)'s review of the afore-mentioned approaches however found that the ad-hoc methods was by far the most popular, followed by the constructive and stepwise approaches which have also been reasonably adopted. Conversely, it was observed that the use of global optimization methods have only received little attention. They finally submitted that the use of

global optimization methods requires further attention, in order to ensure that the best possible ANNs are developed.

2.7.6 Network training

The objective of ANN training is to find a set of optimal model parameters (connection weights) that enables a model to give the best representation of the input-output relationship of a particular system. This usually involves minimizing the error measure between the network output and target outputs, so as to achieve optimal generalization ability of the model. However, finding the model parameters or combination of model parameters that will produce the smallest error estimates is not a simple task. The degree of difficulty in searching for the optimal set of model parameters is dependent on the "ruggedness" or "smoothness" of the error surface (Maier *et al.* 2010).

Due to the difficulty associated with ANN training, optimization algorithms are usually employed for training ANNs. These training algorithms either belong to local or global optimization methods. Local methods are usually gradient based, and can either be first-order or second-order methods. The most frequently used training algorithm in the local methods is the back-propagation (BP) algorithm (Rumelhart *et al.* 1986), which is based on the first-order gradient of the slope of the objective function. Other approaches which belong to the second-order class include the Levenberg-Marquardt (LM) algorithm and the conjugate gradient (CG) algorithm. The local methods are generally computationally efficient algorithms, and have been adopted in a wide range of studies (Kisi 2005; Corzo and Solomatine 2007; Coulibaly and Evora 2007; Kisi and Cigizoglu 2007; Toth 2009; Wang *et al.* 2009; Elshorbagy *et al.* 2010; Ghanbarpour *et al.* 2012; Kisi and Shiri 2012). Kisi and Cigizoglu (2007) and Maier *et al.* (2010) however pointed out that the local methods are susceptible to being trapped in local optima and may also generate negative values, if the error surface is fairly rugged.

On the other hand, global optimization methods such as Genetic Algorithm (GA), Differential Algorithm (DE), Evolutionary Programming (EP), Particle Swarm Optimization (PSO), and the Shuffled-Complex Evolution are becoming more popular in hydrology. These global optimization methods have been used to optimize a pre-defined set of model parameters (Hsu *et al.* 2002; Dorado *et al.* 2003; Parasuraman and Elshorbagy 2007; Kagoda *et al.* 2010; Piotrowski and Napiorkowski 2011; Dhamge *et al.* 2012; Mathur 2012), or used in conjunction with the local optimization methods which include the BP, LM, CG and DE algorithms (Muleta and Nicklow 2004; Chen and Chang 2009; Mihalache and Leon 2009; Subudhi and Jena 2009; Kişi 2010). In comparison to the local methods, the global optimization methods have increased ability to overcome local optima and obtain more steady solutions, but at the expense of computational efficiency.

In the main time, some stochastic-based approaches such as Bayesian methods have also been used to obtain distributions of model parameters, rather than searching for single parameter values (Mohanty *et al.* 2010; Rasouli *et al.* 2012). The major benefit of these methods is that prediction intervals can be automatically obtained for the model predictions.

2.7.7 Applications of ANNs in hydrological modelling

Over the past two decades, the use of ANNs for the prediction and forecasting of water resource variables has been firmly established. ANNs have been successfully applied in diverse fields of water resources such as rainfall estimation, rainfall-runoff modelling, streamflow modelling, groundwater modelling, sediment transport modelling, water quality, hydrological time series, precipitation forecasting, climate change impact modelling, reservoir operations, eco-modelling and uncertainty estimation (ASCE 2000b; Maier and Dandy 2000; Dawson and Wilby 2001; Maier *et al.* 2010; Abrahart *et al.* 2012).

With specific reference to streamflow prediction, numerous researchers have adopted the cause-effect form of modelling to predict streamflow. Historical observations of causative variables such as rainfall, temperature, soil moisture content, evaporation, and runoff have been used to predict streamflow either individually or in combinatory form (Thirumalaiah and Deo 2000; Dolling and Varas 2002; Baratti *et al.* 2003; Raghuwanshi *et al.* 2006; Anctil *et al.* 2008). However, in some studies, only previous values of streamflow were used for streamflow prediction, as in univariate time series modelling (Cigizoglu 2005; Kisi 2005; Jain and Kumar 2007; Kisi and Cigizoglu 2007; Londhe and Charhate 2010).

Meanwhile, some studies focused on performance comparison between ANN and traditional statistical methods such as AR and ARMA (Thirumalaiah and Deo 2000; Abrahart and See 2002; Kisi 2003; Castellano-Méndez *et al.* 2004; Kisi 2005; Jain and Kumar 2007). Generally, majority of studies have proven ANNs to have better predictive performance than the traditional statistical methods. In addition, the dominance of ANN over linear regression attributes to the nonlinear dynamics of hydrological processes, which are inadequately represented by the regression methods.

Furthermore, some researchers employed modular or hybrid procedures for streamflow prediction. Abrahart and See (2000) introduced a hybrid network solution based on the clustering of historical observations with a self-organizing map (SOM) neural network. Chang and Chen (2001) developed a hybrid neural network by the fusion of a neural network and fuzzy arithmetic, which was successful in streamflow prediction. Sivakumar *et al.* (2002) used ANN in combination with the PSR method for river flow prediction. Wang *et al.* (2006a) also developed a hybrid model to forecast daily streamflow in the Yellow River, China. Jain and Kumar (2007) employed a combination of ANNs and traditional time series techniques for monthly streamflow prediction.

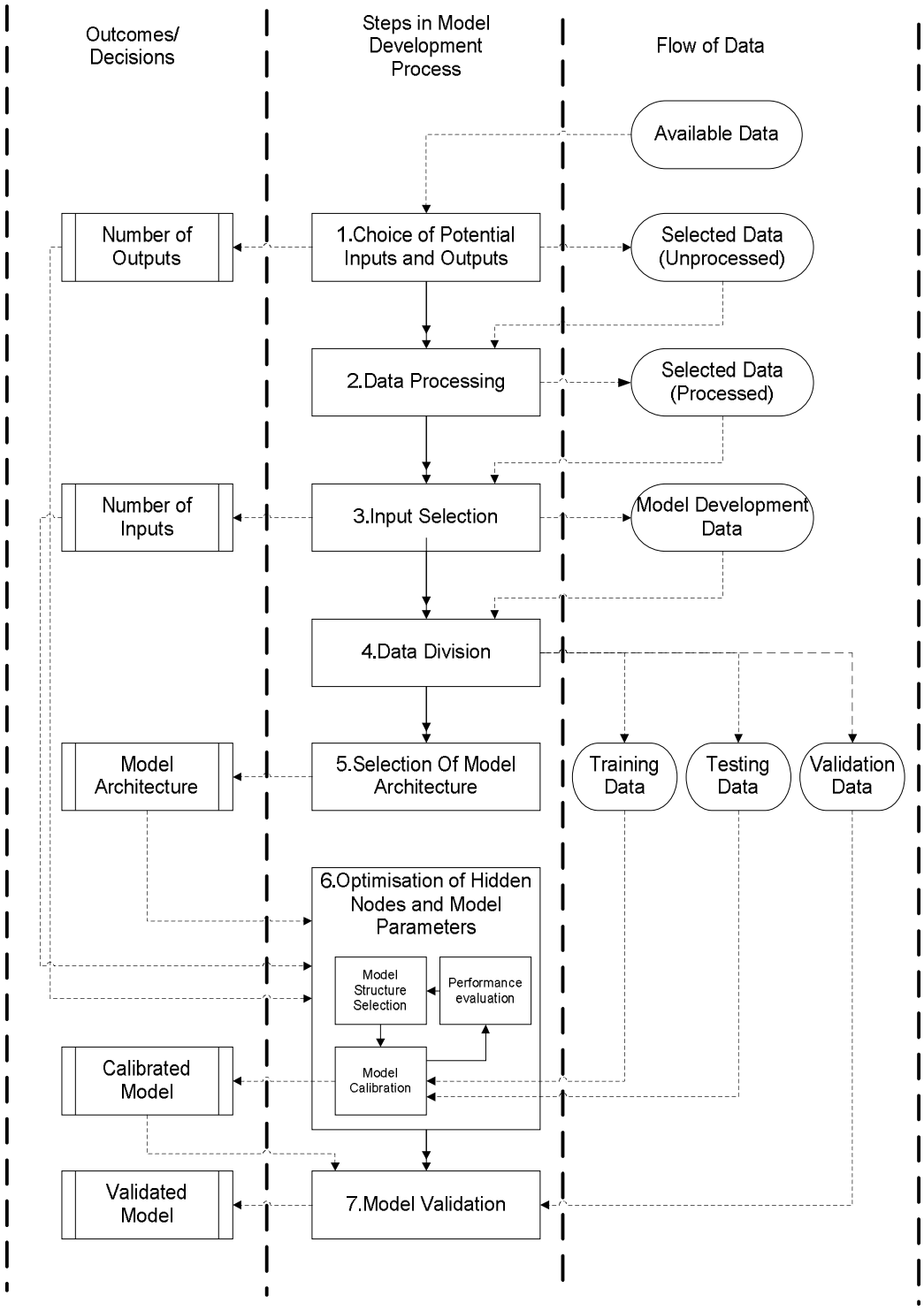

Figure 5: Steps in ANN development process. *Source*: Maier *et al.* (2010)

Besides, ANNs are also coupled with other DDMs such as FRBS, GP, Bayesian methods etcetera (Dorado *et al.* 2003; Nayak *et al.* 2004; Khan and Coulibaly 2006; Aqil *et al.* 2007a). While the use of modular ANNs ensures higher forecast accuracy by splitting the various hydrological processes among individual models (Solomatine and Ostfeld 2008), the synergic use of ANNs and other DDMs however improves forecast accuracy by making-up for the limitations that might be inherent in any of the techniques.

Finally, ANNs have also been deployed as complementary tools in water-related studies such as, for providing error updates in other process-based and other data-based models (Toth *et al.* 2000; Al-Alawi *et al.* 2008), for infilling of missing hydrological records (Coulibaly and Evora 2007), and for estimating prediction intervals of model outputs (Shrestha and Solomatine 2006; Shrestha *et al.* 2009).

2.7.8 Areas of concern

The major areas of concern in the use of ANNs relates to the methodologies employed in building the model network (Figure 5). Most of these issues have been discussed one way or the other while enumerating the model development processes in previous sections, and have also been raised in the reviews of Maier and Dandy (2000); ASCE (2000b); Dawson and Wilby (2001); and Maier *et al.* (2010). However, the synopsis of the major areas of concern in the application of ANNs as presented by Abrahart *et al.* (2010) and Abrahart *et al.* (2012) is as follows: (i) the fact that ANNs are “black-box” models raise questions on their ability to reveal the physics of underlying processes, as the final product and the development process are not fully transparent. Thus, the need for knowledge-based networks and rule-extraction techniques aimed at opening the box were suggested; (ii) identification of the optimal training set is challenging, as there are no fixed rules for the selection or division of datasets for model development purposes; (iii) in terms of ANN model structures, the use

of simple models may produce diminished performance, conversely, complex models may also produce insufficient generalization and its parameters could be more difficult to calibrate or interpret. Thus, striking a balance between costs and benefits still remains a source of concern; (iv) the extrapolation ability of ANNs is also a subject of concern. Although, ANNs are seen as good interpolators, however, predictions made beyond the range of the training set (i.e. extrapolation) in ANNs tend to generate poor and unreliable predictions. In order to give room for extrapolation, methods such as scaling of activation functions have been employed, but at the expense of introducing an extra, initially arbitrary parameter.

2.7.9 Performance improvement methods

Various methods have been suggested for the improvement of the performance of ANNs. These performance improvements range from experimentation with different ANN algorithms and architecture, to the development of hybrid models and dynamic solutions. Abrahart *et al.* (2004) suggested five general directions in which the improvement of ANNs could be pursued in other to make meaningful progress in the application of ANNs in hydrology. A brief summary of the five directions include: (i) improvement of existing ANN models by the investigation of current ANN hydrological problems; (ii) more emphasis on their comparison with operational and process-based models; (iii) the development and building of more robust and more efficient modelling evaluation standards; (iv) further research into the understanding of their internal mechanisms and the practical meaning of each component; and (iv) the development of integrated and hybrid ANN tools for hydrological studies.

2.7.10 Advantages and disadvantages

Various advantages can be derived from the use of ANNs in hydrology. These advantages include: (i) capability of modelling complex, nonlinear hydrological processes without any assumption of the relationship patterns between input

and output variables; (ii) ability to self-adjust to a given set information; (iii) requirement of little expertise of the problem under consideration to achieve successful applications; (iv) showcase compactness and flexibility in their model structure, thus can be easily integrated into other DDMs; and (v) relatively low demand for computation resources when compared to the some other modelling techniques.

However, a major limitation to the use of ANNs is in their inability to produce transparent models, in that their internal operations are somewhat obscure and not interpretable. Another major drawback is that the optimal network configuration for each modelling circumstance can differ, as the problem of "no theoretical support" persists throughout almost all areas of design; even the most basic matters must be determined through experimentation (Abrahart *et al.* 2004). Furthermore, there are no standard or fixed rules for governing appropriate model design and development, thus making it impossible to establish a suitable model *a priori* (Abrahart *et al.* 2012). The inability to incorporate knowledge acquired from existing physical laws into ANNs also serves as a drawback to its use. Finally and most importantly, ANNs are highly susceptible to over-parameterization and over-fitting problems, especially when not properly put to use.

2.8 GENETIC PROGRAMMING (GP)

2.8.1 Introduction

Genetic programming (GP), developed by Koza (1992) belongs to the class of evolutionary algorithms (EA), which is based on the "principle of survival of the fittest", adopted from the process of natural evolution and genetics. GP is however a relatively new addition to the group of other EA techniques such as evolutionary programming (EP), genetic algorithms (GA) and evolution strategies (ES) (Babovic and Keijzer 2000). GP however differs from GA in that

its solution is a computer program or an equation as against a set of binary strings in the GA (Londhe and Charhate 2010). GP is a data-driven modelling technique which performs its operation via a population-based search, in which computer programs or equations that are perceived to be candidate solutions to a given problem are randomly generated and bred using specific genetic operators. GP differs from other DDMs such as ANNs, FRBS and MTs in that it provides mathematically meaningful structures (with optimum parameters) while relating input-output variables of the system (Babovic and Rao 2010). GP has found application mostly in the area of symbolic regression, where the aim is to find a functional relationship between input and output variables. These functional relationships may either be linear, quadratic or higher order polynomial. GP however defines the relationships by optimizing the model structure and the numerical coefficients of the model simultaneously. The GP algorithm is characterized by two major components, which are the terminal and function sets. These two sets contain the major building blocks used to construct the population members of the GP search space.

2.8.2 Representation of GP

GP programs are usually expressed as syntax trees, consisting of terminal and function sets. The terminal set, generally referred to as “Terminals”, consists of the independent variables and constants (known as the “tree leaves”) which are inputs to the problem (Poli *et al.* 2008). For instance, the terminal set may simply consists of causative variables $(x_1, x_2, x_3, ..., x_n)$ of a particular hydrological process. Following that the GP algorithm is being used as a regression technique, the numerical constants (coefficients) that match the chosen model structure to the target output are thereafter determined. Thus, the input variables and numerical constants constitute the terminal set. This is achieved with the aim of searching for a formula that uses the input variables to produce the target output (Babovic and Keijzer 2000). The function set consists of a number of

domain-specific functions that are combined with the terminal set to enable the algorithm generate candidate solutions to the problem. The function set may consist of basic arithmetic operators $(+,-,\times,\div,sqrt)$, mathematical functions (sin, cos, tan, log, ln, e^x), Boolean operators (AND, OR, NOT), logical expressions (IF-THEN-ELSE), iterative functions (DO-UNTIL), and any other user-defined function. The program can however be represented linearly or in an explicit tree representation. Figure 6 demonstrates a syntax tree notation of a mathematical model, $y' = \sin(x_2) + x_3\sqrt{x_1}$ which is a combination of function and terminal points usually referred to as nodes (where y' is the predicted output).

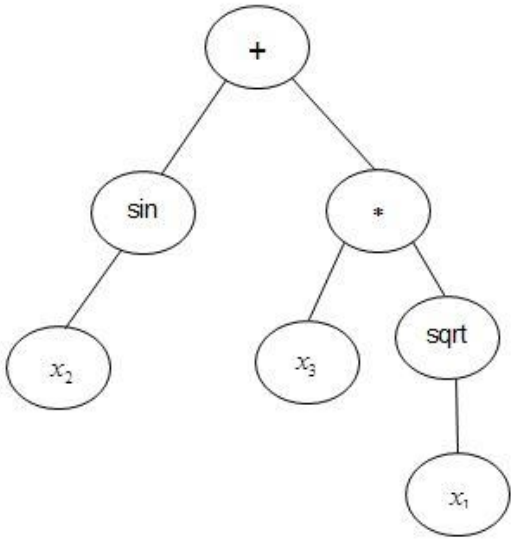

Figure 6: Tree-representation of the mathematical expression

The set of permissible functions and terminals together constitute the primitive set and a major part of the search space of the GP system. Babovic and Keijzer (2000) noted that the primitive set of a GP program must exhibit two properties namely, a "closure" property and a "sufficiency" property. A closure property is a property of a function set which enables it to swap any sub-tree to another location in the tree. The closure property also ensures that all functions can

accept all possible inputs from the terminals, so as to return a well-defined value. For example, the use of a division function was presented with “illegal inputs” such as a zero value will return an undefined value, which may distort the operation of the algorithm (Koza 1992). Thus, protection is usually given to such functions by allowing them to return a specific value when confronted with such illegal inputs. Following the above example, the division function may be set to return a value of 1 when confronted with a zero value. i.e. division (x) = 1. The sufficiency property however ensures that elements of the primitive set are able to express a solution to the given problem. Poli *et al.* (2008) stated that “when a primitive set is insufficient, GP can only develop programs that approximate the desired one”. It was however added that such an approximation, in many cases, can be very close and good for the user’s purpose.

Having established the process of constructing the primitive set of the GP program, it is important to determine how good the elements of the population space are. Thus, a “fitness function” (also referred to as “objective function”) is assigned to each population member in order to measure the performance of each individual program. The objective function measures the accuracy of each computer program by computing the difference between the actual and the predicted values. The error measures may be in form of root mean-square error (RMSE), mean square error (MSE), etcetera (Babovic and Keijzer 2000). This function can either be minimized or maximized depending on the objective of the modelling problem. Furthermore, a number of “hits” can also be employed to evaluate performance of each program. The number of hits is a function of the number of data points correctly predicted within some frequency interval of tolerance. A major limitation to this approach is that each data point is classified as a “hit” or a “miss”, thus, making it possible for one model to be inaccurate than another but still share the same objective function.

2.8.3 Initialization of GP

The first step in implementing GP is to randomly create an initial population for a given population size. The initial population is often randomly created to provide for a satisfactory initial coverage of the search space. In addition, it is a cheap computational process that requires no *a priori* knowledge from the modeller. Different methods are employed to randomly generate the initial population of the search space.

These methods include the full method, the grow method and the ramped half-and-half method. In the full method, the tree nodes are generated only from the function set until a pre-assigned maximum tree depth is reached, where only terminals can be chosen. The method is referred to as full method, because it generates full trees, with all the leaves at the same depth. Unlike the full method, the grow method gives room for creation of trees with variable sizes or shapes. The grow method also allows for selection of nodes from the whole primitive set, until the pre-defined depth limit is reached, where only terminals can be selected.

The ramped half-and-half method however is a combination of the full and grow method, which allows for the creation of trees with various sizes and shapes, but also with equal number of trees for the specified depth. The ramped half-and-half method was by preferred Koza (1992) due to its easy implementation and usage, as well as its good coverage of the search space. Poli *et al.* (2008) however noted that whenever the ramped half-and-half method is employed, high difficulty is often experienced in controlling the statistical distribution of the important properties such as size and shape of the generated syntax trees.

Once the initial population is randomly created, the next step is to create the next generation of population, which involves selecting better program solutions.

2.8.4 Selection

From the principles of genetics, better individuals are more likely to produce more offspring than inferior individuals. Thus, GP employs a selection process (an optimization force in EA), to focus on worthwhile regions of the search space by mapping the objective function values to the number of offspring produced (Babovic and Keijzer 2000). Selection involves the use of genetic operators to probabilistically pick individuals based on their performance, as evaluated using the objective function. The better the fitness of an individual, the greater the chance of that individual being carried over into the next generation. Although, different selection methods are being used within the EA domain such as truncation selection and fitness proportionate selection; the most popular selection method employed in GP is the "tournament" selection method.

Tournament selection involves choosing two individual programs randomly from the population, and comparing them to one another. The "fitter" program wins the tournament. A major characteristic of the tournament method is that it only looks at which program is better than another, and not how much better (Poli *et al.* 2008). This is done in order to ensure that extra-ordinary programs do not outshine others throughout the selection process, and thereafter populate the entire search space with its offspring. Hence, it gives programs with average quality the opportunity to produce offspring. The advantages of the tournament selection over other methods are: (i) it is easier to implement, and (ii) it provides automatic fitness rescaling, and hence, its wide acceptance in GP.

2.8.5 Population structures in GP

In GP, different structural configurations are used to populate the search space with candidate solutions. According to Babovic and Keijzer (2000), the population structures can be classified as: (i) Steady-state population; (ii) Generational population (iii) Panmictic and distributed populations.

In the steady-state population structure, the population size remains unchanged during the GP run. However, in a generational population structure, an intermediate mating pool is created for the selected programs to occupy so as to form the next generation of offspring. The generation population structure may or may not ensure the reproduction of some programs into the next generation. When a population structure ensures that some programs in the mating pool are reproduced to the next generation, such a structure is referred to as an “Elitist” population model (Babovic and Keijzer 2000). Elitism ensures that the best program individuals survive the selection process in order to preserve the best-so-far individual programs in the new generation.

A panmictic population structure entails an extremely mixed population which considers the fitness of the entire population. Therefore, selection process is performed on a global scale. A distributed population structure involves the spatial distribution of the population of candidate solutions into multiple semi-independent subpopulations called “demes” (Poli *et al.* 2008). This gives room for the execution of selection process on a local scale. The distributed structure also allows for occasional migration of individuals majorly between adjacent demes for exchange of genetic material as illustrated in Figure 7.

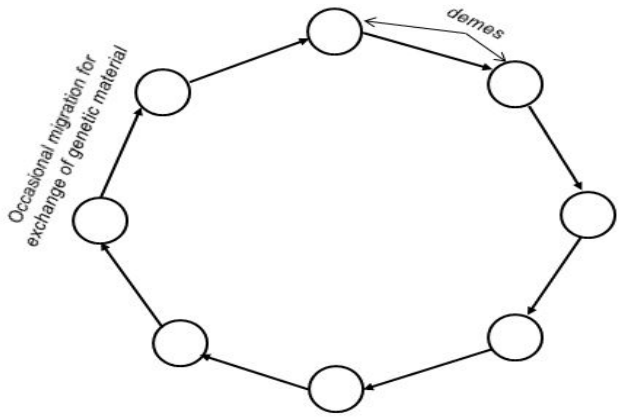

Figure 7: Distribution of the population space into demes, allowing for exchange of genetic material.

This genetically inspired mechanism is usually referred to as “Island model” (Babovic and Keijzer 2000; Brameier and Banzhaf 2001). A major advantage of the distributed population over other methods is that the distribution of the population space into demes facilitates the occurrence of parallel evolution. The parallel evolution consequently makes the GP algorithm less susceptible to convergence to local optima (Babovic and Keijzer 2000), as a local optima in one deme might be overcome by other demes with better search direction. Furthermore, the distributed population structure ensures that the evolution progresses faster than in a single population structure (Brameier and Banzhaf 2001).

2.8.6 Genetic operators

In GP, two genetic operators namely, crossover and mutation are often used to transform selected best programs into a new generation of programs. These genetic operators operates by applying slight modifications to the structure of selected programs in order to achieve better or fitter programs (Figure 8). The purpose of the crossover operator is to generate new programs which did not exist in the old population, to allow for thorough sampling of the search space. Crossover is performed by selecting two parent programs from the mating pool and swapping some corresponding sections across a randomly chosen point to produce two different offspring programs of different characteristics. The number of programs experiencing crossover is dependent on a predefined probability of crossover, P_c.

The mutation operation however involves random modification of a structural member of a selected parent program to create a new offspring program. The modification is also performed based on a probability of mutation, P_m. The evolution process is performed over successive generations until a preset termination criterion is met, and the program generated upon termination of the

run is finally selected as the best program that gives the most accurate description of the modelled system.

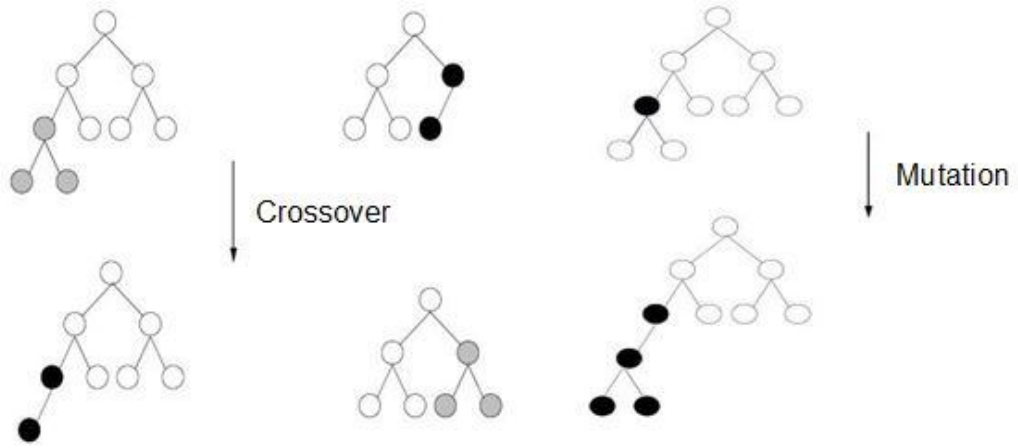

Figure 8: Crossover and mutation being performed on parent programs

Generally, GP evolve models in two different forms. These models can either be evolved in form of computer programs or in form of equations (Deo 2009). Program-based models consists of codes that may be written in different programming languages such as Assembly, Java, C or C++, which enables the source files to be called and new data sent to them. On the other hand, in equation-based approach, the input-output relationships are defined by evolving models in form of mathematical formulae. Some good examples of program-based and equation-based GP implementation kits include Discipulus™ (Francone 1998), and GPKernel respectively.

2.8.7 Application of GP in hydrological modelling

Regardless of its status of being a relatively young and growing branch of EA, GP has been successfully applied to solve a wide range of water-related problems. Thus, literature on its applications is vast and multifaceted. In the light of this, some representative examples of the major applications of GP in water-related fields is presented in Table 1.

Table 1: Application of GP in various fields of hydrological modelling

S/N	Authors & Year	Application Area	Input Variables	Location	Forecast Interval	Comparisons	Remarks on GP
1.	Khu *et al.* (2001)	Real-time runoff forecasting	Rainfall	Orgeval Catchment, France	+1	AR, Kalman Filter	GP was found to be a better updating tool when compared to AR and Kalman filter method
2.	Sheta and Mahmoud (2001)	River flow forecasting	Previous flow	Nile River, Sudan	+1	AR	GP produced simpler models with less error estimates
3.	Babovic and Keijzer (2002)	Hourly Rainfall-runoff modeling	Rainfall & Runoff	Orgeval Catchment, France	+1	NAM Conceptual model	Better forecasting accuracy produced by GP. Integration of both models suggested
4.	Whigham and Crapper (2001)	Rainfall-runoff modelling	Rainfall	Teifi & Namoi Basin (Wales & Australia)	Function Approx.	IHACRES (Deterministic model)	GP produced an impressive performance just as IHACRES, but without requiring any causal relationship
5.	Liong *et al.* (2002)	Rainfall-runoff modelling	Runoff & Rainfall intensity	UBT Catchment, Singapore	+15, +30, +45 (minutes)	Naïve Rainfall-runoff model	GP induced R-R relationship was seen as viable alternaive to the naïve model
6.	Dorado *et al.* (2003)	Rainfall-runoff prediction	Rainfall	Old Victoria Catchment, Spain	N/A	Traditional SCS Unit Hydrograph, NARMAX	Combination of GP & ANN performed better than the ARMAX & traditional models
7.	Jayawardena *et al.* (2005)	Rainfall-runoff modelling	Rainfall & Runoff	Hok Tau Basin, Hong Kong; Shanqio & Shunhan Basin, China	+1	N/A	GP performance was not very satisfactory, as it was unable to capture high peak discharge magnitudes.
8.	Muttil and Lee (2005)	Daily prediction of algal blooms	Chlorophyll, dissolved oxygen, water temp, sola radiation & wind speed	Kat O Station & Tolo Harbour, Hong Kong	+1	ANN, ARMA	GP results was in agreement with that of ANN, but was able to identify key input variables. The GP performance was also better than that of the ARMA models
9.	Bautu and Bautu (2006)	Weather prediction	Temperature, Precipitation & Pressure	Rennes, France & Froson Sweden	+2	AR, ANN	GP exhibited its self-adaptive nature when unsupervised; producing better input-output relationships than other models

10.	Makkeasorn *et al.* (2008)	Daily Streamflow forecasting	NEXRAD rainfall, SST & Met. data	Choke Canyon Watershed	+30, +7, +3 days	ANN	GP-derived models performed better than the ANN models
11.	Elshorbagy and El-Baroudy (2009)	Prediction of Soil moisture content	Net radiation, Precipitation, Air temp. & Soil temp.	Mildred Lake Mine Site, Canada	N/A	Evolutionary Polynomial Regression (EPR)	A program-based GP tool (Discipulus™) performed comparably with EPR, but better than an equation-based GP tool (GPLAB). Tool uncertainty was identified.
12.	Maity and Kashid (2009)	Streamflow prediction	Streamflow, ENSO & EQUINOO indices	Narmada River Basin	+1	N/A	Satisfactory performance was recorded by GP-derived models, as they were able to provide significant impacts of various input combinations
13.	Wang *et al.* (2009)	Forecasting of monthly discharge time series	Previous flows	Lancangjiang River, Asia	N/A	ARMA, ANN, ANFIS, SVM	Best performance was obtained by GP, ANFIS & SVM models with GP producing better results in the validation phase
14.	Londhe and Charhate (2010)	Daily streamflow forecasting	Previous streamflows	Narmada Basin, India	+1	ANN, MTs	GP performed marginally better than ANN & MTs, especially in normal & extreme events
15.	Ni *et al.* (2010)	Annual streamflow prediction	Precipitation & ET	West Malian River	Function Approx.	MLP, Grey system theory, MLR	GP performed better than other models under limited availability of datasets
16.	Selle and Muttil (2011)	Testing structure of hydro. Models	N/A	Southeastern Australia	N/A	Conceptual model	GP predictions supported the dominant processes contributing to deep percolation in conceptual models
17.	Guven (2009)	Modelling of daily flow rate	Previous flows	Schuylkill River, USA	+1	ANN (MLP, GRNN), AR	GP results were fairly better than the ANN & AR models
18.	Guven and Kişi (2011)	Daily susp. sediment modelling	Streamflow & Suspended Sediment	Two stations in the Tongue River, USA	+1	GEP, ANN	LGP performed better than ANN & GEP models, producing simple & explicit models
19.	Sivapragasam *et al.* (2011)	Long term inflow forecasting	Rainfall & Inflows	Tamarabarami Basin, India	Multiple	N/A	GP-derived models generated improved inflow forecasts, especially when rainfall values from neighbouring stations were included as inputs
20.	Zahiri and Azamathulla (2012)	Flow discharge Prediction	Depth ratio, coherence parameter & discharge rate	Mulitple number of Rivers	N/A	M5 model trees, Vertical divided channel method	Better accuracy produced by GP compared to other methods

2.8.8 Areas of concern

Researchers have pointed out some important issues pertaining to the application of GP in water-related studies. Some criticisms have trailed the generation of mathematical formulations by GP, as its combination of multiple elementary functions often result in extremely complex models which may be too difficult to interpret (Keijzer and Babovic 1999; Solomatine *et al.* 2008). Keijzer and Babovic (1999) argued that the complexity of these formulae may result in GP producing models with accurate syntax but meaningless semantics. A case was thereafter made for the introduction of dimension into the GP paradigm.

Another major challenge faced by modellers in the use of GP is the selection of appropriate parameter settings to control the algorithm run. The convergence of the GP algorithm to global optimum is dependent on the parameters that govern the evolution process (Babovic and Rao 2010). However, precaution needs to be taken while setting these parameters, as optimum settings vary from application to application. Thus, for a given problem, multiple runs using different parameter settings are often carried out and the solutions compared. This task is often considered as highly laborious, and also demands for higher computational resources.

The ability of GP to find the optimal solution at higher forecast horizons have also generated some concern to GP users, as forecast accuracy tends to deteriorate with increase in forecast horizon. Jayawardena *et al.* (2005) in their study, observed the inability of GP to capture complex rainfall-runoff transformation when applied to a steep-sloped catchment, characterized by high peak discharge magnitudes with steep rising and recession limbs. However, they acknowledged the satisfactory performance of GP when applied to two other catchments at smaller time intervals. Furthermore, Babovic and Keijzer (2000) earlier noted that the ability of GP to find the optimal solution depend on

the magnitude of the numerical values and its dimensions. Thus, difference in the magnitude and dimensions of input variables often makes it difficult for GP to scale the variables to workable values.

2.8.9 Performance improvement methods

A major improvement to the GP approach was suggested by Babovic and Keijzer (2000). They proposed a “dimensionally aware GP”, which requires the scaling of inputs and outputs to cast in dimensionless and proportionate terms. A dimension-based brood selection method was also introduced, which involves the use of a “culling function” to measure the goodness-of-dimension of candidate solutions. This ensures the selection of best solutions in terms of goodness-of-dimension, and thus results in the development of solutions with improved goodness-of-fit and less complexity.

Deschaine and Francone (2002) also introduced an improved version of a program-based GP toolkit which ranks and combines the best individual program solutions called “program models” into teams of solutions referred to as “team models”. This combination ensures better predictive accuracy in the team models than any of the individual program models. Thus, the ability of the GP model to generalize and converge towards global optimum is improved.

2.8.10 Advantages and disadvantages

GP has been found to exhibit several advantages over other DDMs. Its major advantage is in its ability to generate programs that can efficiently simulate complex processes using symbolic expressions (Elshorbagy and El-Baroudy 2009). Another advantage of GP over other robust methods such as ANN is that it generates a transparent and structured representation of the system being modelled, without requiring *a priori* identification of the model structure (Giustolisi and Savic 2006). This is unlike the ANN approach where the structure of the network and training algorithm have to be defined in advance, and only

the optimization of the network parameters (weights and biases) are performed. However, in GP both the model structure and its parameters are being optimized, as they are both part of the search process (Babovic and Keijzer 2002). This gives GP the ability to automatically identify the input variables that contribute beneficially to the model and disregard those that do not (Jayawardena *et al.* 2005), thus reducing the dimensionality of the model. Besides, GP evolves models capable of giving physical insight into the input-output interactions inherent in the modelled system, in contrast to the ANNs where difficulty still exists in extracting knowledge from the network parameters (Sudheer 2005).

On the other hand, GP has its own limitations. Principally, GP is not very powerful in finding constants, and more importantly, it tends to produce more complex functions as the forecast horizon increases (Giustolisi and Savic 2006).

2.9 PERFORMANCE EVALUATION MEASURES

In order to determine the potential of a developed model, its performance is usually appraised against one or more criteria. There are several measures that are being used to evaluate model performance. Each of the performance evaluation measures provides different information about the predictive ability of the model. The most popular measures are squared error measures which are based on the squares of the differences between observed and predicted output values. Examples of such measures are sum of squared errors (SSE), root mean-squared error (RMSE) and the Nash Sutcliffe efficiency (E). An attribute of squared error measures is that they tend to be dominated by high extreme values (Maier *et al.* 2010). As an alternative, absolute errors can be employed. These are measures based on absolute differences between observed and predicted output values. Such methods include mean absolute percentage error (MAPE) and total sum of absolute deviations (TSAD). Although, absolute errors

provide some insight into the degree of the errors, they do not provide insight on the performance of the model with respect to overall under- or over-prediction.

Another model evaluation measure often used is the Pearson coefficient of correlation (r) and the coefficient of determination (R^2). These two methods describe the degree of collinearity between observed values and predicted values. While correlation coefficient is an index of the level of linear relationship between observed and predicted values, R^2 however defines the proportion of variance in observed values as explained by the model (Moriasi *et al.* 2007). Though r and R^2 have been widely applied for model evaluation, these measures are over-sensitive to high extreme values, and insensitive to additive and proportional difference between observed and predicted output values (Legates and McCabe 1999).

Besides the aforementioned measures, some other performance evaluation measures can be obtained from literature. Such measures include normalized mean squared error (NMSE), threshold statistics (TS), measures based on relative error, non-parametric Wilcoxon rank-sum test, Akaike information criterion (AIC), Bayesian information criterion and a rescaled version of mean absolute percentage error (MAPE-R) (Thompson *et al.* 2001; Coulibaly *et al.* 2005; Jain and Kumar 2007; Maier *et al.* 2010; Swanson *et al.* 2011).

2.10 CONCLUSIONS

Following this extensive review of the application of popular DDMs in the hydrological domain, one cannot boldly say one modelling technique is superior to the other, as all DDMs have their strengths and drawbacks. With a wide-range of modelling options to choose from, the challenge remains as to which particular technique will generate the best results for a given task, as one cannot continue to subject each DDM to test one after the other. Thus, it is of crucial importance for modellers and other decision-makers to subject DDMs to test

comparatively, in order to determine the approach that best suites the given problem. Furthermore, another promising approach that has been producing improved results is the development of modular and hybrid models, which allows for complementary modelling. The evolutionary computation (EC) and global optimization techniques have showcased great potential in this respect, as they can be easily be integrated into other DDMs; same as the ANN and fuzzy rule-based techniques.

Thus, for the purpose of this research, the potential of two evolutionary-inspired techniques (GP and a DE-trained ANN) is investigated comparatively. Although the performance of evolutionary-inspired techniques have been well attested to when employed individually in water resources applications, however comparative studies of this nature are conspicuous by their near absence. This serves as an inciting cause of employing two evolutionary-inspired techniques; to determine the better model in terms of predictive performance under limited availability of datasets.

2.11 RESEARCH OUTPUT

1. Oluwaseun Oyebode, Fred Otieno and Josiah Adeyemo (2014). Review of three data-driven modelling techniques for hydrological modelling and forecasting. *Fresenius Environmental Bulletin.* Vol. 23: No. 6.
2. Oluwaseun Oyebode, Josiah Adeyemo and Fred Otieno (2014). Uncertainty sources in climate change impact modelling of water resource systems. Paper accepted for presentation at the *IJAS American Canadian Conference for Academic Disciplines,* 19 - 22 May, 2014, Toronto, Canada.
3. Oluwaseun Oyebode and Josiah Adeyemo (2014). Genetic Programming: Principles, Applications and Opportunities for Hydrological Modelling. Paper accepted for presentation at the *7th Int'l Conference on Environmental Sci. & Tech.*, 9 – 13 June, 2014, Houston, Texas, USA.

CHAPTER 3

GENETIC PROGRAMMING FOR STREAMFLOW PREDICTION

3.1 INTRODUCTION

Streamflow prediction remains crucial to decision-making especially when it concerns planning and management of water resources in water scarce regions such as South Africa. This is due to its role as a key component of the water cycle, serving as the primary source of water supply for numerous sectors which benefit from its availability. However, streamflow prediction comes with various complexities arising from nonlinear and dynamic nature of climatological and hydrological factors. Thus, effective streamflow prediction cannot be carried out without an appraisal of the fundamental factors that influence streamflow.

Owing to the importance of streamflow prediction, several hydrological models have been developed to provide a representation and interpretation of the complex and non-linear hydro-climatological processes. These models can be categorized into physically-based (process) models and data-driven models (DDMs). All these models require significant amount of data for their respective calibration and validation (Babovic and Keijzer 2002). The process-based models are deliberately configured to portray the behavioural characteristics of the catchment; thus, they are very effective in exploring the impacts of climatic variables on hydrological processes. However, difficulties in model parameterization, complexities in model formulation and huge training data requirements serve as major drawbacks to the use of these models, especially in areas where large hydrological datasets are often unavailable (Beven 2002; Ni *et al.* 2010). The application of process-based models is discussed in chapter 1.

Data-driven models (DDMs) on the other hand describe the behaviour of the hydro-climatological processes by taking into account only limited assumptions about the underlying physics of the system being modeled. They can be very effective for the specific circumstance for which they were developed. DDMs have been widely applied in numerous hydrological studies, and significant number of successes has been recorded**.** The popular methods include K-nearest neighbours (K-NN), artificial neural networks (ANNs), model trees (MTs), support vector machines (SVMs), chaos-based theory, fuzzy rule-based systems (FRBS) and genetic programming (GP). These data-driven modelling techniques are now being considered as an approach that will complement or replace the process-based models (Londhe and Charhate 2010). Chapter 2 provides a detailed discussion on the application of popular DDMs used in hydrological modelling studies.

Due to the advantages of the GP approach as highlighted in Section 2.8.9 of this thesis, GP was adopted for use in this study. The main objective of this chapter was to investigate the potential of the GP approach in predicting streamflow response to local hydro-climatic variables in the upper Mkomazi River, South Africa. Twelve GP models were developed using the few available datasets; one for each month of the year. The performance of the GP evolved models were evaluated using standard model evaluation measures.

3.2 METHODOLOGY

3.2.1 Linear genetic programming (LGP)

A detailed description of the working principles of GP is fully discussed in Section 2.8 of this thesis. It has been established in Section 2.8.2 that GP can be represented in different forms, which are majorly tree-based and linear-based programs. The representation used in this chapter is the linear genetic programming (LGP), and its basic concepts are briefly introduced.

LGP is an extension of the traditional tree-based GP. LGP evolves sequences of instructions from an imperative programming language such as C/C++, or from a machine language (Guven and Kişi 2011). The word “linear” refers to the structure of the (imperative) program representation, and does not represent functional genetic programs which are constrained to a linear list of nodes only (Brameier and Banzhaf 2007). The prominent attributes of LGP are the graph based flow that results from a multiple usage of indexed variable (register) contents, and evolving programs in a low-level language. This concept was subsequently expanded and resulted to the automatic induction of machine code by genetic programming (AIMGP) system (Brameier and Banzhaf 2001). In AIMGP, solutions are directly manipulated as binary machines codes, and executed without using an interpreter. This gives tremendous acceleration to the process of evolving program solutions (Bhattacharya *et al.* 2001).

Each individual program in LGP is represented by a variable-length sequence of instructions, which are generally clubbed together to form “instruction blocks”. The instruction blocks hold one or more native machine code instructions, depending on the size of the instructions. In LGP, the instruction set (i.e. the function set) can be composed of arithmetic operations, conditional branches and function calls. Each function implicitly includes an assignment to a variable, which facilitates the use of multiple program outputs in LGP, whereas in tree-based GP those side effects need to be incorporated explicitly (Brameier and Banzhaf 2001).

LGP utilizes a two-point string crossover, in which a section of the random position and random length is selected in both parent programs and swapped between them. If one of the resultant offspring programs exceeds a predefined maximum length, crossover is terminated and re-initialized by swapping uniform sections. Mutation operation is performed in LGP by modifying an operand or operator on the instruction set into another symbol over the same set. LGP also

employs a special type of mutation (termed *macro mutation*) which deletes or inserts an entire instruction (Guven and Kişi 2011). Further details on LGP can be found in the literature (Brameier and Banzhaf 2001, 2007).

In this study, LGP was implemented using an LGP toolkit called Discipulus™. A brief introduction of the Discipulus™ toolkit is presented in the next section.

3.2.2 Discipulus™

Discipulus™ is a program-based LGP toolkit developed by Francone (1998). Discipulus™ evolves models in the form of computer programs written in machine language. The machine language programs however can be decompiled into other programming languages such as Assembly, Delphi, C, C++ and Java. This enables the source files to be called and new data sent to them. Discipulus™ evolves two categories of computer programs namely, a “program model” and a “team model”.

A program model is a single program solution that defines the input-output relationship of a given system. A team model is a combination of single program models that are combined to produce better convergence than any of the single program models. A comprehensive description of the features of Discipulus™ can be found in Francone (2011). The latest version of the toolkit (Discipulus™ 5.2) was used for implementing LGP in this study.

3.3 STUDY AREA AND DATASETS

The upper Mkomazi River is located within the semi-arid province of KwaZulu-Natal in South Africa, and is the third largest in the province. The river has several large tributaries which include Loteni, Nzinga, Mkomanzana and Elands Rivers. The description of the river and its characteristics that make it suitable for this work is given in Section 1.6 of this thesis.

Datasets relating to the study area were provided by the Department of Water Affairs (DWA) and the South African Weather Service (SAWS). Historical monthly records of mean streamflow were obtained from DWA. Streamflow data for a 19-year period (1994 – 2012) from gauging station U1H005 (Mkomazi River @ Lot 93 1821) with geographical coordinates between 29°44'37.3" South longitudes and 29°54'17.8" East latitudes were applied in this study. However, datasets from two other gauging stations (U1H009 and U1T008) located within the upper catchment of the river were relatively short and not used. The SAWS provided the corresponding climatic data (rainfall and temperature) from three independent weather stations namely Pietermaritzburg, Shaleburn and Giant Castle stations, located within the study area.

Figures 9 and 10 show plots of the monthly averages of temperature, rainfall and streamflow for the period under study. It can be observed from the plots that rainfall and temperature datasets for the three weather stations exhibit the same distribution patterns, despite the fact that they are of different values. It can also be observed that the streamflow pattern follows after the rainfall and temperature distributions.

An increase in streamflow occurs between the months of October and February, after which it follows a downward trend until it reaches its minimum between the months of June and July. The same was observed for rainfall and temperature. It can be inferred that rainfall and temperature both contribute to the magnitude of streamflow in the river. Furthermore, Figures 11 and 12 show plots of annual streamflow against average annual temperature and annual rainfall respectively. However, a decline in annual streamflow from year 2000 and series of fluctuations thereafter are evident from the plots.

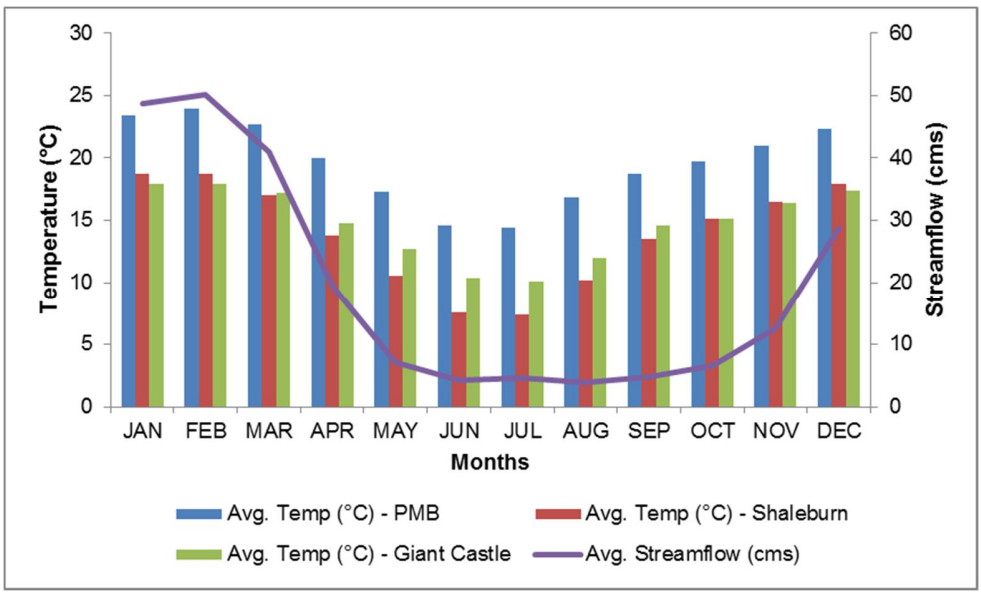

Figure 9: Plot of average monthly temperature at the three weather stations and average monthly streamflow measured at gauging station U1H005.

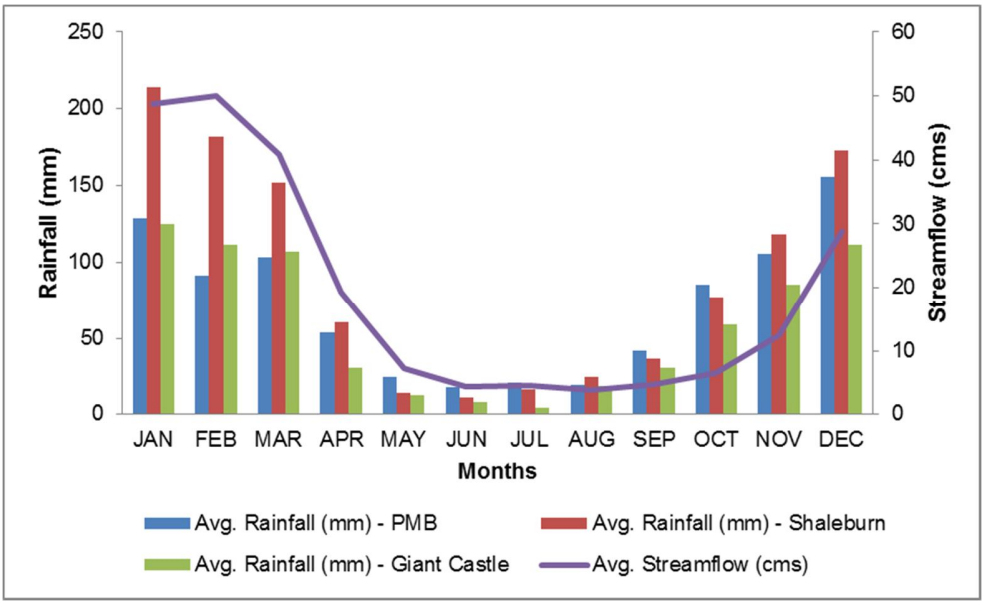

Figure 10: Plot of monthly rainfall at the three weather stations and average monthly streamflow measured at gauging station U1H005.

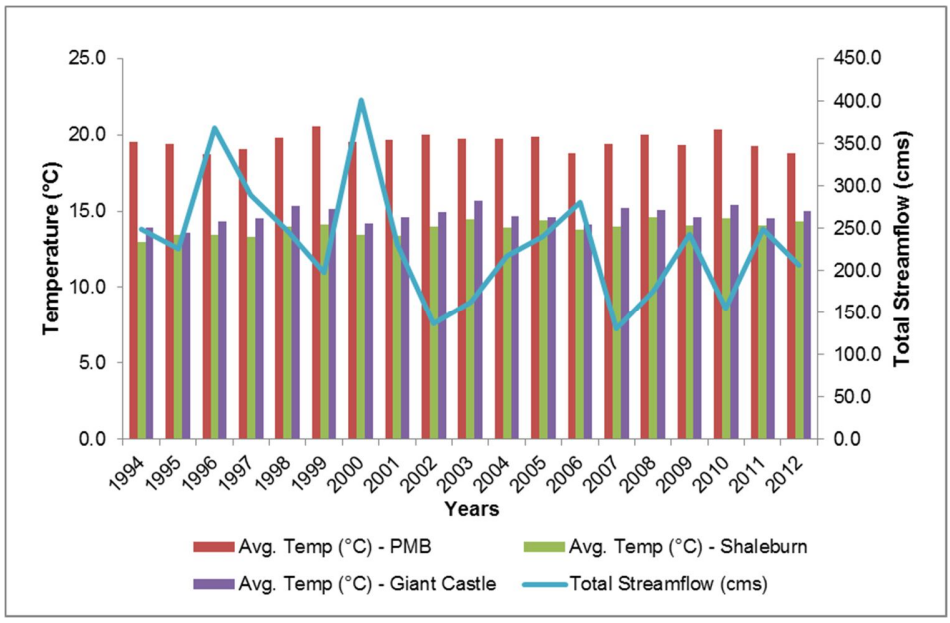

Figure 11: Plot of average annual temperature at the three weather stations vs. total streamflow at station U1H005 during the study period (1994 – 2012).

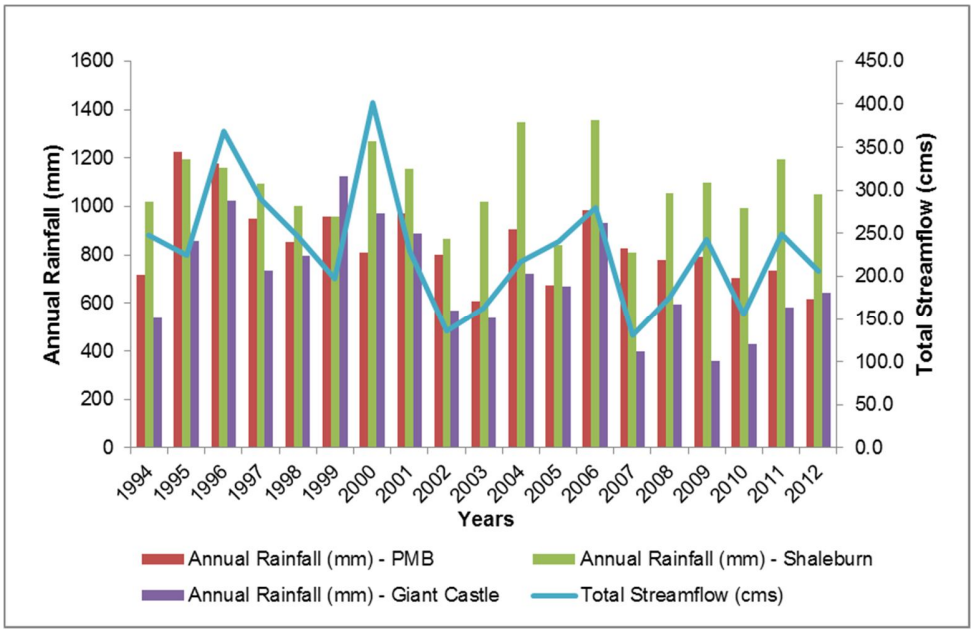

Figure 12: Plot of annual rainfall at the three weather stations vs. total streamflow at station U1H005 during the study period (1994 – 2012).

3.4 MODEL DEVELOPMENT

3.4.1 Selection of input variables

There are several processes that influence streamflow generation in river hydrology. These processes include precipitation, temperature, evaporation, soil moisture, vegetation cover, land use, etcetera (Loucks and van Beek 2005; Raghunath 2007). However, historical records of all these factors are often unavailable and limited to certain areas such as the site used in this study. Thus, the choice of input parameters was dependent on the few available datasets. The datasets made available by DWA and SAWS include streamflow, rainfall and temperature which cover a 19-year period (1994 - 2012).

As a result, the available datasets were used for the construction of the input vector space of the GP algorithm. The streamflow values during the same month of previous years were used alongside rainfall and temperature values from three independent and neigbhouring weather stations within the upper Mkomazi River Catchment. In a recent study, Sivapragasam *et al.* (2011) established that the use of climatic information from surrounding areas, when combining antecedent flows and climatological data in flow forecasting result in drastic improvement in forecast accuracy. This justifies the reason for the inclusion of rainfall and temperature data from three neigbhouring stations within the river catchment.

In order to effectively construct the input vector space of the GP algorithm, it is highly important to ensure the selection of only the input variables that will give the best representation of the system being modelled. Furthermore, as pointed out in Section 2.7.4 of this thesis, the inclusion of irrelevant inputs may lead to an increase in model complexity, reduction in model interpretability, higher computational demands, and consequently mis-convergence (Bowden *et al.* 2005a). Thus, the determination of the appropriate input variables entails finding

the lags of input variables that have a significant influence on the predicted output (streamflow). In this chapter, the dependency between input variables and the associated lag effect were determined using correlation analysis. Correlation analysis has been successfully applied for input variable determination in river hydrology by many researchers (Sudheer *et al.* 2002; Aqil *et al.* 2007b).

Table 2 presents the results of the correlation analysis used in selecting input variables into the GP input space. The results showed high correlation between the values of streamflow for the past three years and that of any given year. The results also indicated that streamflow for the given year highly corresponds to the rainfall and temperature values of the preceding year across the three independent weather stations. However, outcome of the analysis on higher number of lags other than the reported ones produced lower correlation values, and were therefore discarded.

Table 2: Results of correlation analysis showing the relationship between historical values of input parameters and target output

Input Parameters	**Target Output** (Q_{t+1})
Q_t	0.9983
Q_{t-1}	0.9957
Q_{t-2}	0.9966
$R1_t$	0.7256
$R2_t$	0.9203
$R3_t$	0.8856
$T1_t$	0.8605
$T2_t$	0.8046
$T3_t$	0.8072

The input vector spaces of the GP models were populated with a total of nine (9) input variables. These inputs comprise of streamflow values for a given month in the last three (3) years (Q_t, Q_{t-1}, Q_{t-2}), rainfall values from the three independent weather stations for the same month in the preceding year $(R1_t, R2_t, R3_t)$, and their corresponding temperature values $(T1_t, T2_t, T3_t)$. The numbers 1, 2 and 3 represent Pietermaritzburg (PMB), Shaleburn, and Giant Castle weather stations respectively.

The approach employed for monthly streamflow prediction in this study was to adopt a 1-year lead time. Therefore, the streamflow being modelled for a given month in the next year (Q_{t+1}) is designated as the target output. Furthermore, the prediction of streamflow on a monthly basis was done using individual monthly models. This is because the use of individual monthly models in high lead time predictions has been found to produce better predictions when compared with the adoption of a single model, which often produce poor predictions (Sivapragasam *et al.* 2011). Hence, a total of twelve individual monthly models (one for each month of the year) was developed using the GP technique. The mathematical representation of the 1-year lead time model adopted can be expressed as:

$$Q_{t+1} = f(Q_t, Q_{t-1}, Q_{t-2}, R1_t, R2_t, R3_t, T1_t, T2_t, T3_t) \quad (4)$$

Following the construction and formulation of the GP input vector space, the streamflow and the climatic datasets prepared for each month constituted sixteen (16) data points.

3.4.2 Data splitting

As part of the model development process in DDMs, the next step to take after selecting the appropriate variables into the model input vector space is the splitting of historical observations into training and validation sets. A data-

preprocessing feature in Discipulus™ called “Notitia” was used to split the datasets into training and validation sets. The data-preprocessing process ensured that the training and validation datasets are representatives of the same population, as DDMs tend to produce poor predictions when allowed to extrapolate beyond the range of the training datasets. Therefore, the datasets were split into two subsets using a random sampling method with about two-third of the datasets used for model training and the remaining one-third for validation. The training datasets were used for the purpose of model development while the validation datasets were used to check the predictive capability of the models.

3.4.3 Objective function and model fitness

As earlier discussed in Section 3.2.2, Discipulus™ toolkit was used for the implementation of GP in this study. The objective function employed by Discipulus™ is the minimization of the mean of the sum of squared errors that can be obtained between the predicted and the observed values of streamflow. The objective function, F, measures the fitness of evolved programs, and can be expressed mathematically as:

$$F = Min\left[\sum_{i=1}^{n}\frac{(Q_{o_i} - Q_{p_i})^2}{n}\right] \tag{5}$$

Q_{o_i} and Q_p are observed and predicted values of streamflow respectively, n is the number of data points, and i is a counter from 1 to the number of data points.

Furthermore, the ability of GP to screen and prioritize input variables during its run contributes to the fitness of the evolved programs. In this chapter, Discipulus™ was set up to express the contribution of each input variable as a function of its frequency of occurrence during the algorithm run so as to ensure

the accuracy of predictions. This also allows for computation of the influence of each input variable on the magnitude of streamflow in the area of study.

To initialize the algorithm run, the functional set was supplied with arithmetic, comparison and trigonometric functions. Thus, the instruction set of the GP algorithm comprised of the selected input variables, the functional set and randomly generated constants of between -1 and 1. Though, instructions majorly occur in different lengths, Discipulus™ imposes some uniformity on the arrangement by gluing instructions together into instruction blocks (Francone 2011). Each instruction block in Discipulus™ is 32 bits long, and may be composed of one or more native processor instructions.

3.4.4 GP algorithm setup

The setting of the LGP algorithm parameter is of crucial importance to the successful performance of the algorithm (Francone 2011). In this chapter, the default parameter settings were adopted as advised by the developer of Discipulus™. According to Francone (2011), the default settings for Discipulus™ project work quite well for most of the projects.

Table 3 presents the various parameters used to control the GP algorithm run in this study. The GP algorithm for each computation was run on an Intel Core i7 PC with 3.40GHz and 4GB RAM. The maximum size of each evolved program was restricted to 512, initializing with 80 instructions per program. This was done in order to prevent the phenomenon of bloating, which means over-growing of programs without limits and without any improvement in the fitness of the population (Bleuler *et al.* 2001).

Table 3: Summary of GP parameter settings used to control the algorithm run

Parameters	Value
Program size	Initial 80, maximum 512
Mutation frequency	95%
Crossover frequency	50%
Block mutation rate	30%
Instruction mutation rate	30%
Instruction data mutation rate	40%
Population size	500
Maximum number of run	300
Generations since start of run	10 000
Homologous crossover	95%
Number of demes	10
Migration rate	1%

The distributed population structure discussed in Section 2.8.5 was adopted in this study. This involves subdividing the population into multiple subpopulations or demes. This subdivision allows for occasional migration of individuals among demes for exchange of genetic material. Furthermore, it leads to the evolution of the entire population, quickens the evolution process, and consequently reduces the tendency of the algorithm converging towards local optima. However, the use of high migration rates between demes could cancel the effect of having separate demes. Thus, the migration rate was set at 1% as advised by Francone (2011).

3.4.5 Model performance evaluation

The need to evaluate the performance of evolved models is of high importance as it is a pointer to the potential of GP technique in providing accurate and reliable predictions. Theoretically, a good model should offer the least

differences between the predicted and observed values (Maity and Kashid 2009).

Three (3) standard model evaluation criteria namely, mean absolute percent error (MAPE), root mean square error (RMSE) and coefficient of determination (R^2) were used to investigate the performance of the monthly models developed in this study. The three model evaluation measures were computed using the mathematical expressions in equations (6), (7), and (8).

1. The Mean Absolute Percent Error (MAPE)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Q_p - Q_o}{Q_o}\right| \times 100 \tag{6}$$

2. The Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(Q_o - Q_p)^2}{n}} \tag{7}$$

3. Correlation of Determination (R^2)

$$R^2 = \left[\frac{\sum(Q_o - \overline{Q_o})(Q_p - \overline{Q_p})}{\sqrt{\sum(Q_o - \overline{Q_o})^2 \sum(Q_p - \overline{Q_p})^2}}\right]^2 \tag{8}$$

Q_o and Q_p represent observed and predicted streamflows respectively, while $\overline{Q_o}$ and $\overline{Q_p}$ represent their corresponding mean values.

Clearly, lower values of MAPE and RMSE would indicate better predictive accuracy of the model, while higher values of R^2 (close to 1.0) would indicate better predictive accuracy of the models.

3.5 RESULTS AND DISCUSSIONS

As earlier discussed, Discipulus™ evolves two categories of program solutions namely single program models and team models. It combines the best thirty single program models into five team models in order to achieve higher predictive accuracy. Results from this study are from the best team models that gave a better representation of the hydrological system of the upper Mkomazi River compared to that of single program models. The best team model that simultaneously provided the minimum error estimates and highest R^2 values was chosen in each of the monthly simulations carried out.

Table 4 presents the statistical performance of the individual monthly models in predicting streamflow at upper Mkomazi River. It can be observed that all the monthly models produced significantly low error estimates in terms of MAPE and RMSE while recording high R^2 values, both during the training and validation phases.

Table 4: Performance evaluation statistics for the individual monthly models

MONTH	TRAINING PHASE			VALIDATION PHASE		
	MAPE (%)	RMSE	R^2	MAPE (%)	RMSE	R^2
JAN	3.9401	1.4968	0.9964	3.4179	1.3535	0.9969
FEB	0.9966	0.4974	0.9994	3.0490	1.0126	0.9923
MAR	3.6570	1.2469	0.9982	4.3077	1.3797	0.9932
APR	5.6798	1.0710	0.9891	6.1725	1.1868	0.9741
MAY	2.6864	0.1982	0.9970	1.8900	0.1245	0.9954
JUN	1.1854	0.0607	0.9986	0.5932	0.0325	0.9990
JUL	1.2691	0.0558	0.9985	0.7575	0.0257	0.9999
AUG	4.2479	0.1047	0.9972	7.5730	0.3401	0.9881
SEP	11.1474	0.0607	0.9988	5.1472	0.1983	0.9740
OCT	3.1860	0.1607	0.9994	6.1092	0.2653	0.9905
NOV	4.2854	0.4855	0.9975	2.1735	0.2918	0.9978
DEC	6.0007	0.8642	0.9972	2.4802	0.5242	0.9978
AVG	**4.0235**	**0.5252**	**0.9973**	**3.6392**	**0.5612**	**0.9916**

During the training phase, all the models obtained a MAPE of between 1% and 11%, with models for the months of February and September producing the lowest and highest MAPE respectively. In like manner, RMSE values of between 0.06 and 1.50 were recorded by the developed models. The lowest RMSE values were produced by the June and July models, while the highest values were produced by the January and March models. All the models produced R^2 values approaching towards unity, similar to the experience reported by Wang *et al*. (2009). This shows that the models exhibited highly significant positive correlations between observed and predicted streamflows during training. Furthermore, the R^2 values produced in all were comparable, as the values ranged between 0.9891 – 0.9994.

During the validation phase, the MAPE estimates obtained by the models were between 0.59% and 7.57%. The June and July models produced the best performance, providing MAPE estimates of 0.59% and 0.76% respectively. The models for the months of April, August and September however generated the highest MAPE estimates, as the three models recorded estimates above 6%. Similarly, it can be observed that RMSE estimates in most of the models converged towards zero. This is in agreement with the results of Mehr *et al*. (2013) where LGP-derived models also produced very low error estimates during validation. The RMSE estimates during validation varied between 0.03 – 1.37, with the models of the months of June and July once again producing the lowest RMSE estimates, while the January and March models produced the highest estimates in the same fashion as they did during the training phase. Additionally, it can be observed that the models for the months with low flows (May, June and July) produced the least error estimates, while those with high flows (January – April) recorded higher error estimates. This may be attributed to the absence of significant number of high flow patterns in the training datasets as suggested by Mugumo (2012) in his streamflow forecasting study where low flows were also better simulated than high flows.

Generally, it can be observed that the error estimates (in terms of both MAPE and RMSE) in most of the models were better converged during the validation phase when compared to the training phase, with the exception of the models of the months of February, March, April, August and October, where slight increment can be noticed. These slight differences may be considered to be as a result of minimal contrast in patterns of the training and validation datasets of the months involved (Mugumo 2012). However, the excellent prediction is a pointer to the existence of a significant portion of similarities in their pattern. Nonetheless, it can be said that the GP models performed brilliantly considering that limited amount of datasets were employed in this study. Furthermore, it can be inferred that the subdivision of the population space into demes facilitated parallel evolution in the population space and consequently led to the high degree of convergence achieved by the monthly models during validation. This is evident in the results presented in Table 4, as all the models maintained R^2 values close to 1, as they did during the training phase; showing that the observed and predicted values were highly and positively correlated.

Figures 13, 14 and 15 further reveal the degree of correlation between the observed and predicted streamflow values. It can be observed that all the models showcased high accuracy and excellent performance in reproducing the streamflow dynamics in the upper Mkomazi River. Both the low, intermediate and high flows were substantially reproduced, including several sharp spikes that characterized the streamflows in some months. This further confirms the ability of GP to capture both normal and extreme events as pointed out by Londhe and Charhate (2010), and also show that the GP approach exhibits high degree of consistency in its predictions (Mehr *et al*. 2013). It can also be seen from the scatter plots that the observed and predicted streamflows are closely fit to the line of equality in all the models.

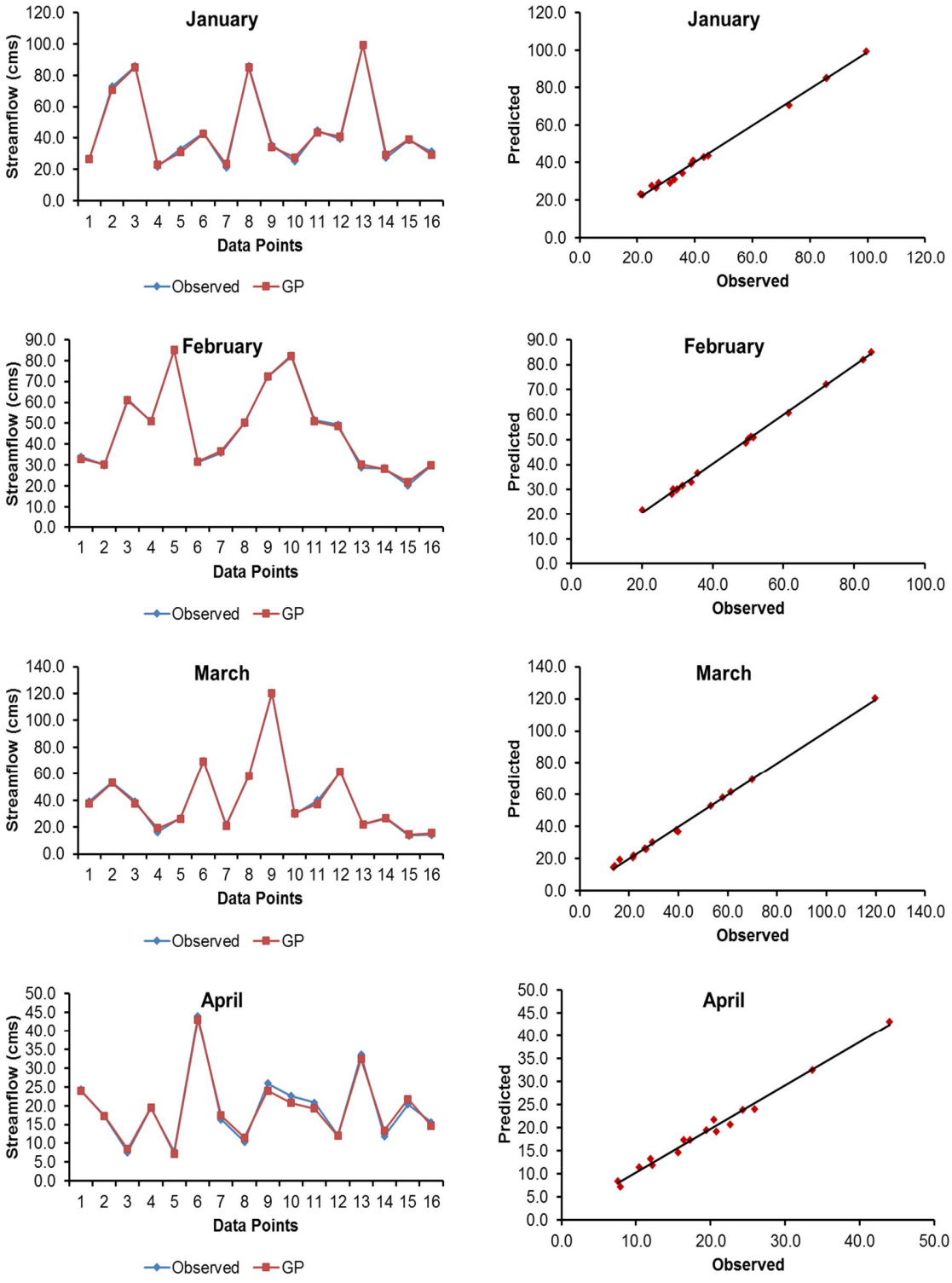

Figure 13: Plots of observed and GP-predicted streamflows (January - April)

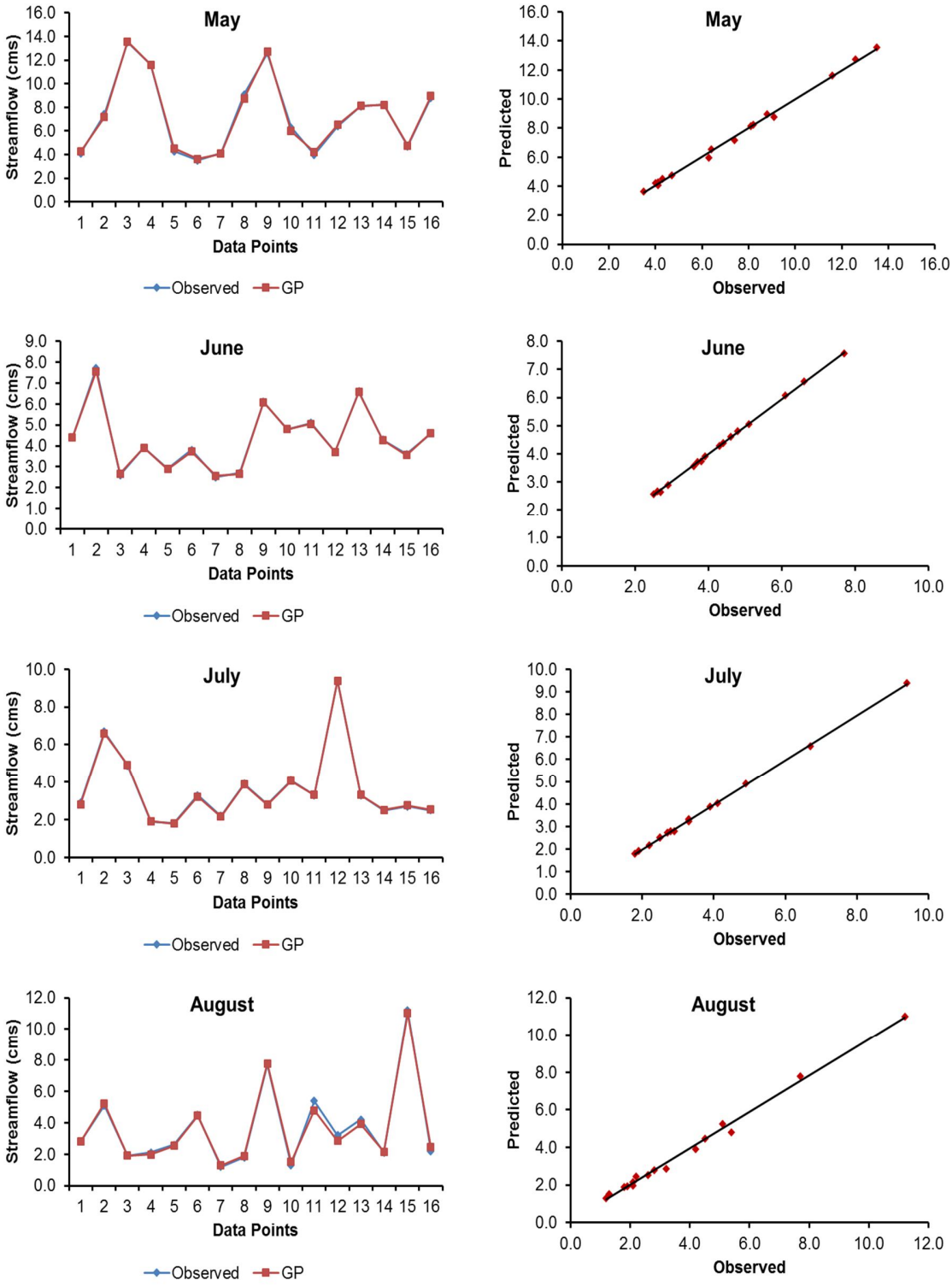

Figure 14: Plots of observed and GP-predicted streamflows (May - August)

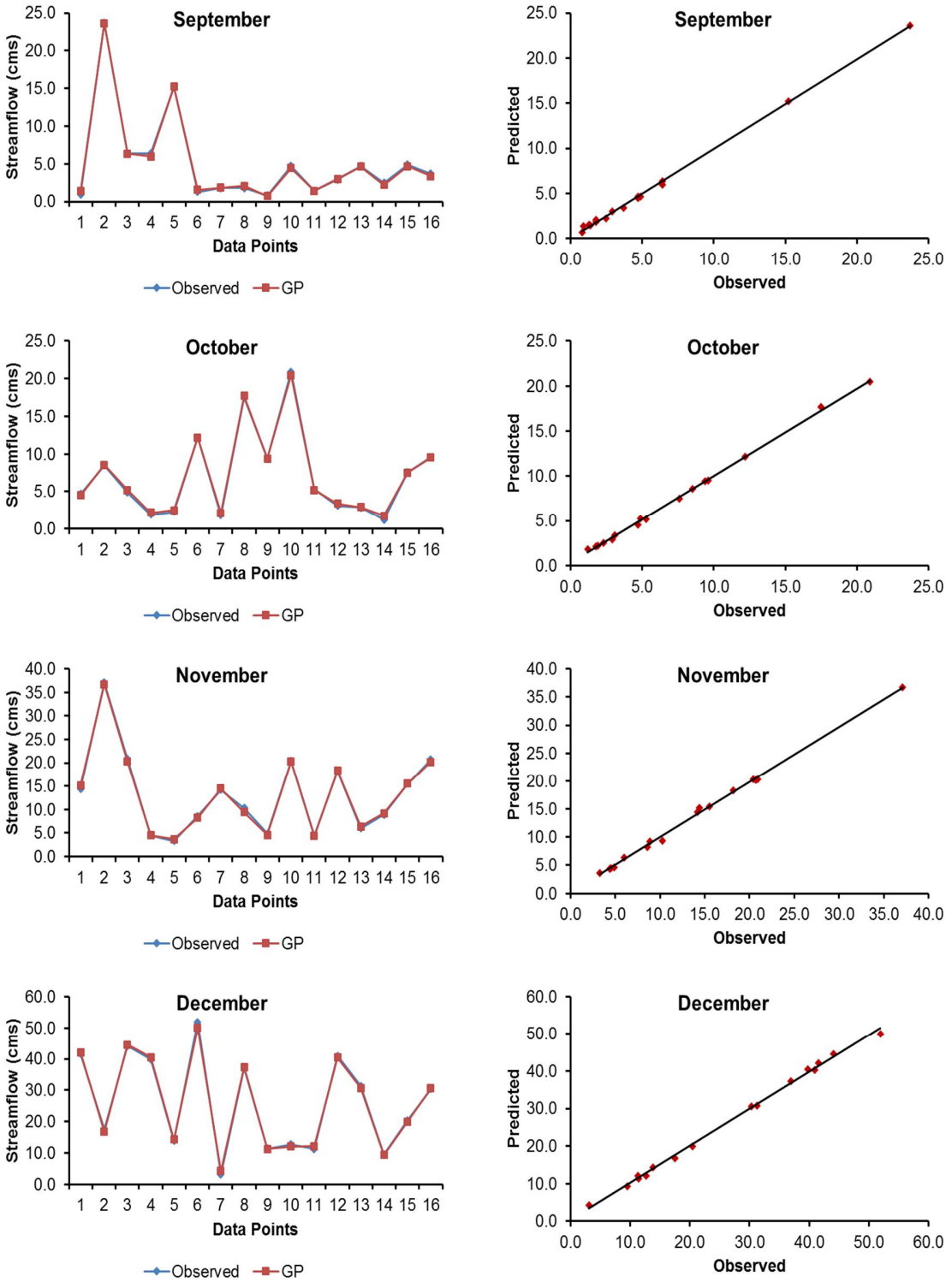

Figure 15: Plots of observed and GP-predicted streamflows (Sept. – Dec.)

The brilliant performance of GP is evident in the scatter plots of the months of March, August and September, despite the large differences that characterize their streamflow values. This further shows that GP is capable of capturing rare occurrence of large events (Londhe and Charhate 2010). The models of the months of June and July can be considered as the best models of this study, as they produced the least error estimates (Table 4), and also the closest agreement between observed and predicted streamflow values (Figure 14).

Generally, results obtained from the individual monthly models demonstrate that the GP approach is capable of providing an accurate and reliable estimate of streamflow in the upper Mkomazi River.

3.6 IMPACTS OF INPUT VARIABLES ON STREAMFLOW REGIME

One of the objectives of this study was to quantify the impacts of local climatic variables on streamflow in the upper Mkomazi River. To achieve this, the ability of GP to determine the impacts of each input variable on predictions made was subjected to use. Upon successful completion of each GP algorithm run, Discipulus™ computes the contribution of each input variable to the fitness of the best programs. The impact of each input variable is a function of its frequency of occurrence during the GP algorithm run.

Table 5 presents the impact of each of the input variables used for streamflow prediction in the study. The results are scaled between 0.0 and 1.0. A value of 1.0 in the frequency column indicates that the particular variable appeared in 100% of the thirty best programs.

Table 5: Impacts of input variables in terms of their frequency of occurrence in the best team model for each month

Inputs	Frequency of Input Variables												
	Jan	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**	**Avg.**
Q_t	0.83	0.97	0.67	0.67	0.43	0.53	0.50	0.77	0.60	0.30	0.63	0.43	**0.61**
Q_{t-1}	0.77	0.47	0.43	0.70	0.53	0.53	0.63	0.60	1.00	0.70	0.83	0.40	**0.63**
Q_{t-2}	0.50	0.40	0.63	0.83	0.53	0.50	0.90	1.00	0.80	0.93	0.77	0.60	**0.70**
$R1_t$	0.67	0.53	0.20	0.63	0.20	0.37	0.83	0.83	0.27	0.67	0.60	0.73	**0.54**
$R2_t$	0.63	0.53	0.43	0.40	0.53	0.73	0.97	0.20	0.50	0.70	0.53	0.53	**0.56**
$R3_t$	0.63	0.57	0.47	0.63	0.30	0.33	0.40	0.60	0.23	0.83	0.60	0.33	**0.49**
$T1_t$	0.53	0.70	0.97	0.97	0.23	0.30	0.50	0.30	0.60	0.23	0.50	0.63	**0.54**
$T2_t$	0.80	0.77	0.47	0.43	0.43	0.73	0.63	0.70	0.27	0.90	0.80	0.47	**0.62**
$T3_t$	0.60	0.47	0.57	0.37	0.53	0.73	0.23	0.57	0.27	0.27	0.67	0.57	**0.49**

Results from Table 5 show that the contributions of the input variables vary across the months of the year. However, on average, streamflow observations of the previous two years (Q_{t-2}) made the highest contribution towards the fitness of the program, with an average frequency of occurrence of 0.70. It can also be observed amongst the climatic variables that the rainfall and temperature inputs from the Shaleburn weather station had a relatively high average in terms of frequency of occurrence ($R2_t$ = 0.56, $T2_t$ = 0.62), when compared with those of PMB ($R1_t$ = 0.54, $T1_t$ = 0.54), and Giant Castle ($R3_t$ = 0.49, $T3_t$ = 0.49) stations.

These results show that GP was able to capture the climate variability within the upper Mkomazi River Catchment, in relation to the location of the gauging

station. Interestingly, the Shaleburn weather station is the closest to the gauging station used in this study, and hence the reason for its high frequency of occurrence. The excellent predictive performance of GP in this study can further be attributed to its ability to screen and prioritize input variables as remarked in Makkeasorn *et al.* (2008)'s streamflow forecasting study. In addition, results from Table 5 show that each of the input variables contributed to the fitness of the models. These results indicate that the inclusion of climatic information from independent weather stations within the river catchment also contributed to the predictive accuracy of the models. This supports the findings of Sivapragasam *et al*. (2011) where significant improvement in streamflow prediction accuracy was observed upon the inclusion of information from additional weather stations within the study area. This is a clear indication that the input variables employed in this study have been carefully and appropriately selected.

Program codes representing the GP models are presented in Appendix A. These programs have been decompiled into the C programming language. However, owing to the size of these codes, only the program codes for the month of January are presented. Codes for other monthly models would be deposited in the institutional repository, and made accessible to potential users upon request from the appropriate authority.

3.7 CONCLUSIONS

The application of genetic programming (GP) to predict streamflow response to hydro-climatic variables using limited amount of datasets in the upper Mkomazi River, South Africa has been demonstrated. Results show that GP exhibited good potentials in predicting monthly streamflow regardless of the use of limited datasets. A close agreement and significantly low error estimates were produced between observed and predicted streamflows in all the twelve monthly models developed. On average, the twelve models produced a MAPE, RMSE and R^2 estimates of 3.64%, 0.56 and 0.99 respectively during model validation.

GP also demonstrated its ability to screen and prioritize input variables in conformity to the hydrological regime of the upper Mkomazi River. The GP technique and the methodology employed in selecting the input variables both contributed to the accuracy and confidence of the streamflow predictions. This study further elucidates the potential of GP as a potent predictive tool for effective planning and management of water resources, even in the absence of significant amount of datasets.

3.8 RESEARCH OUTPUT

1. Oluwaseun Oyebode, Josiah Adeyemo and Fred Otieno (2014). Monthly streamflow prediction with limited hydro-climatic variables in the upper Mkomazi River, South Africa using genetic programming. *Fresenius Environmental Bulletin*. Vol. 23: No. 3.

CHAPTER 4

ARTIFICIAL NEURAL NETWORKS FOR STREAMFLOW PREDICTION AND PERFORMANCE COMPARISON BETWEEN GP AND ANNs

4.1 INTRODUCTION

The management of water resource systems has always been of crucial importance to water managers and decision makers, most especially in water-stressed countries such as South Africa. Water resources engineers and other related stakeholders have developed various approaches towards managing the relatively little amount of water in these regions in order to meet the increasing profile of water demand. However, explosive increase in population continues to place higher demands on the limited water resources. A major constituent of the hydrologic cycle – streamflow, remains a vital point of reference in water resources management due to its role as a major source of freshwater accessibility for the sustainability of man, animal and the natural environment. Thus, streamflow prediction is highly essential to the making of worthwhile decisions by all relevant stakeholders.

Various approaches that have been employed by researchers with the aim of predicting streamflow have been extensively discussed in chapter 2 of this thesis. However, in hydrological modelling studies, limitations in the availability of datasets often pose serious challenge to the achievement of accurate and reliable predictions (Babovic and Keijzer 2002). Therefore, there is a need to find out suitable approaches that could serve as alternatives in the bid to achieving accurate and reliable predictions in data sparse regions.

Evolutionary computation (EC) and global optimization techniques have gained much popularity in hydrological modelling studies due to their ability to produce

robust models, and also enhance faster convergence towards global optimum. The ease with which they can be integrated into other modelling techniques also serves as a major reason for their prominence. In this chapter, the performance of two evolutionary inspired data-driven modelling techniques, genetic programming (GP) and differential evolution (DE)-trained artificial neural networks (DE-ANN) were compared, to determine the better approach to be adopted for streamflow prediction in the upper Mkomazi River under limited availability of datasets.

Two case studies involving the use of ANN are presented in this chapter. Case study 1 involved the use of the same set of input variables employed in GP model development, while the second case study entailed the introduction of early stopping method, incorporation of GP as a screening tool, and the subjection of the ANN learning process to sensitivity analysis. For the purpose of identification, the ANNs in Case study 1 and 2 will hereafter be referred to as ANN-1 and ANN-2 respectively. The performance of the ANN-1 and ANN-2 derived models were investigated comparatively with that of the GP models (as evaluated earlier in chapter 3), so as to determine the most suitable approach for streamflow prediction in the upper Mkomazi River.

4.2 METHODOLOGY

4.2.1 Artificial neural networks (ANNs)

ANNs are computational intelligence (CI) techniques inspired by the neurological processing ability of the human brain. ANNs consist of a pool of simple processing units called neurons which communicate by sending signals to each other over a large number of weighted connections (Kröse and van der Smagt 1996). The operating principles of ANNs is based on parallel distributed information processing that is capable of storing experiential knowledge gained through the process of learning, and making it available for future use

(Elshorbagy *et al.* 2010). The processing units function by receiving inputs from external sources or other neurons in the network and computing output signals which is transmitted to other units. These processing units are found in layers commonly categorized as input, hidden and output layers. Nonlinearities inherent in the inputs of the system being modelled are transformed into a linear space by the use of an activation function in the hidden layer of the network. The commonly used activation functions are sigmoidal functions such as the logistic and hyperbolic tangent functions (Maier and Dandy 2000). The major network topologies that characterize the architecture of ANNs are the feed-forward neural networks (FFNN) and the re-current neural networks; with multilayer perceptron (MLP), radial basis function (RBF) networks, Kohonen's self-organizing feature maps (SOFM) and Elman-type RNN as the most popular ANNs (Coulibaly and Evora 2007; Jha 2007). Numerous specialized learning algorithms have been employed for the purpose of training and subjecting ANNs to adaptive learning. These methods have been fully discussed in Section 2.7.6 of this thesis. Hence, the ability of ANNs to assimilate complex and nonlinear input-output interactions makes it suitable for predictive studies in the field of water resources.

The multilayer feed-forward neural network (FFNN) architecture was employed in this study. The FFNN was trained using differential evolution (DE) algorithm. A brief introduction on working principles of DE is presented in the next section.

4.2.2 Differential evolution algorithm

Differential evolution (DE) is a relatively new technique in EA, which was introduced by Storn and Price (1997). DE is a population-based heuristic algorithm for global optimization. The working operation of the DE algorithm primarily involves four stages namely initialization, mutation, recombination and selection. DE differs from other population-based techniques in that it employs differential mutation (Chauhan *et al.* 2009). Differential mutation as a simple

adaptive scheme is used in DE to scale mutation increments to its correct magnitude. In like manner, DE uses a non-uniform crossover in which the parameter values of an offspring vector are inherited in uneven proportions from the parent vectors (Abdul-Kader 2009). Furthermore, the selection method utilized by DE is the tournament selection method, where the offspring vector competes against one or both of its parents (Abdul-Kader 2009).

Assuming a population of solutions of size, NP, within a *D*-dimensional search space and a defined number of real valued vectors are randomly initialized. Two arrays referred to as the primary and secondary arrays are used to hold the population of NP, *D*-dimensional, real valued arrays. The primary array holds the current vector population, while the secondary array accumulates vectors that are selected for the next generation. Every pair of vectors (X_a, X_b) defines a vector differential: $X_a - X_b$. Upon the random selection of vectors X_a and X_b, they undergo combination via mutation operation so as to produce new generation of vectors (a weighted differential of the vectors). Crossover operation is thereafter performed on the newly generated vectors. This involves the perturbation of the newly generated vectors with a randomly selected target vector, X_c to produce a trial vector. The scaling of the vectors is performed in DE by using a scaling factor, F, which is a user-defined constant in the range of 0 to 1.2. Similarly, the non-uniform crossover is implemented using a user-defined constant in the range $0 \leq CR \leq 1.0$ (Abdul-Kader 2009). The resulting trial vector is chosen for the next generation if and only if it generates a decrease in the value of the objective function. Consequently, evolution of vectors occurs over successive iterations and generations in order to explore the search space, and to locate the minima of the objective function. The iteration is finally brought to a halt when a predefined termination criterion is met.

DE has been successfully applied for solving complex engineering problems due to its ability to diffuse close to the global optimum solution (Qian *et al*. 2009;

Adeyemo and Otieno 2010; Pal *et al*. 2010). With particular reference to training of ANNs, DE have been found to produce better and quicker convergence using only small number of parameters for its algorithm setup, unlike other training algorithms such as BP, LM, CG and GA which are either susceptible to local optima, or require high computational times and huge number of iterations to obtain satisfactory results (Slowik and Bialko 2008; Abdul-Kader 2009; Mihalache and Leon 2009; Piotrowski and Napiorkowski 2011).

4.3 MODEL DEVELOPMENT

4.3.1 Case study 1

As earlier mentioned, the multilayer FFNN topology was investigated comparatively with the GP technique for streamflow prediction in this study. The architectural design of the FFNN models developed comprised of three layers: one input, one hidden and one output layer (Figure 16). In this first case study, the aim was to create a basis for comparison between the ANN-1-derived models and the GP models earlier derived in chapter 3. Thus, the modelling strategy employed was to subject the ANN-1 to the same set of data used in the development of the GP models. This is done to avoid any form of bias. Thus, the input layer of the ANN-1 models comprised of nine input nodes, representing the nine selected input variables, while the output node comprised of only one neuron (the target output). The optimal architecture of each monthly model was determined by incrementally varying the number of hidden layer nodes 2 to 10 using a single (one) stepping approach.

The DE algorithm which was utilized in training the network was run for 10 000 generations for each of the monthly models. A computer with the same specifications as the one used for the GP algorithm run (as stated in chapter 3) was also employed. The DE algorithm was however written using Visual Basic for Applications (VBA) programming language. The population size, NP, and

crossover constant, CR, were used to control the algorithm run, while the mutation scale factor, F, controlled the amplification of differential variation during the run. Following the suggestion of Price and Storn (2013), the DE control parameters, NP, CR and F were set at “D multiplied by 10”, 0.9 and 0.4 respectively, (where D is the number of weights and biases in the selected architecture). In terms of data-preprocessing, a logistic sigmoidal-type activation function of between 0 and 1 was used in the hidden layer of the FFNN, to rescale the inputs in the range 0.1 – 0.9. The rescaling of the inputs to the extreme ranges of the activation function was avoided in accordance with the advice of Maier and Dandy (2000). It has been earlier established in Section 2.7.3 of this thesis that the rescaling of input variables towards the extreme bounds of the activation function could reduce the size of the weight updates, thus resulting into flats spots during training (Maier and Dandy 2000). A linear activation function was however employed in the output layer, in order for the network to transform nonlinearities in the inputs into a linear space.

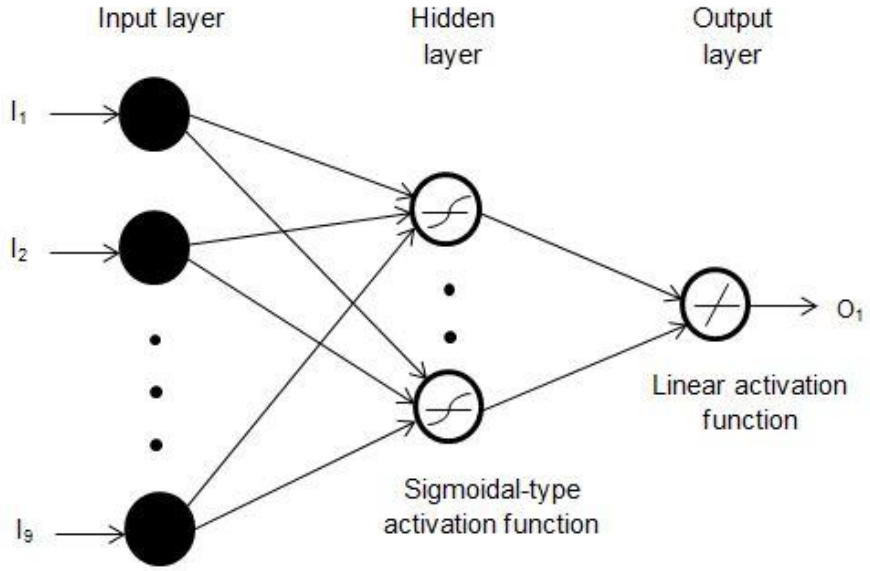

Figure 16: Architecture of the three-layered multilayer feed-forward neural network (FFNN)

4.3.2 Case study 2

As discussed in Section 2.7.10, one of the major problems that usually confront ANNs in hydrological modelling studies is over-parameterization of inputs. Thus, in this second case study, an attempt was made to curb the effects of over-parameterization which may likely occur in case study 1, considering the fact that limited amount of datasets was employed. The approach employed was to introduce GP as a screening tool for selecting the input variables into the input vector space of the ANN-2 models. Results from the GP technique already established the ability to screen and prioritize input variables according to their contribution to the fitness of the best program solutions. Following the input impact results obtained in Table 5, four out of the nine variables were selected for the development of ANN-2 models. The selection of the four input variables was based on their impacts on the predictive accuracy of the GP models. Thus, the dimensionality of the ANN-2 models was reduced compared with that of the ANN-1 models. In addition, different combinations of input variables were used across the twelve monthly models unlike in ANN-1 where the same combination was employed in all the models. Table 6 presents the selection and combinations of the four input variables used for the development of each monthly model.

Table 6: Combination of input variables used in developing the monthly models

Inputs	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Q_t	X	X	X	X				X	X			
Q_{t-1}	X			X	X	X	X		X		X	
Q_{t-2}			X	X	X		X	X	X	X	X	X
$R1_t$	X						X	X				X
$R2_t$		X			X	X	X			X		
$R3_t$										X		
$T1_t$		X	X	X					X			X
$T2_t$	X	X				X		X		X	X	
$T3_t$			X		X	X					X	X

Furthermore, the learning process of the ANN-2 models was fine-tuned by subjecting the DE algorithm to sensitivity analysis. This involved varying the parameter settings during the run to determine the optimal parameters that will generate the least errors. CR was varied between 0.5 – 0.9, while F was altered between 0.1 – 0.5. Both CR and F were adjusted incrementally using a stepping value of 0.1. The parameters could have been adjusted to span across their respective intervals; however, adopting this approach would definitely result in higher computational times. An early-stopping method was also introduced to avoid the problem of overfitting. Early stopping seeks to identify the point where minimum error on the validation datasets starts to increase, and immediately stops training to prevent overfitting. With the exception of the incorporation of GP and early-stopping methods, as well as the fine-tuning of the DE algorithm, all other parameter settings and procedural steps employed in developing the ANN-1 models were maintained.

In all, nine input variables were used in the ANN-1 models, while four were used in the ANN-2 models. Thus, the optimal network architecture in both case studies is a function of the optimal number of hidden nodes that returns the minimum error between observed and predicted values at the end of the run.

4.4 RESULTS AND DISCUSSIONS

The performance of all the models developed by the ANNs (ANN-1 and ANN-2) was subjected to test by adopting the three performance evaluation criteria used for the GP models (i.e. MAPE, RMSE and R^2). The performance of ANN models were evaluated against those obtained using the GP technique. The performance evaluation results are presented in Tables 7 and 8, while the optimal network architectures are presented in Table 9. The performance of the different models in predicting streamflow in the upper Mkomazi River is analyzed and presented with reference to each of the case studies.

Table 7: Comparison of MAPE, RMSE and R^2 values between GP and ANN during training

MONTH	TRAINING PHASE								
	MAPE (%)			RMSE			R^2		
	GP	ANN-1	ANN-2	GP	ANN-1	ANN-2	GP	ANN-1	ANN-2
JAN	3.9401	1.2865	3.2593	1.4968	0.6778	1.7259	0.9964	0.9992	0.9951
FEB	0.9966	0.4008	7.2021	0.4974	0.2689	5.7134	0.9994	0.9997	0.8653
MAR	3.6570	3.1207	91.2276	1.2469	1.1404	41.5205	0.9982	0.9989	0.1176
APR	5.6798	3.4220	50.1606	1.0710	0.7181	8.4563	0.9891	0.9949	0.2966
MAY	2.6864	0.6667	1.7045	0.1982	0.0694	0.1627	0.9970	0.9991	0.9961
JUN	1.1854	1E-08	0.1711	0.0607	7E-10	0.0081	0.9986	1.0000	1.0000
JUL	1.2691	0.2050	27.5932	0.0558	0.0150	1.3923	0.9985	1.0000	0.6224
AUG	4.2479	4.1465	15.3046	0.1047	0.1799	0.5216	0.9972	0.9918	0.9302
SEP	11.1474	5.3100	10.9293	0.0607	0.2615	0.4853	0.9988	0.9984	0.9946
OCT	3.1860	4.2952	58.9866	0.1607	0.4268	4.4800	0.9994	0.9953	0.4610
NOV	4.2854	2.2617	34.0076	0.4855	0.2183	3.5226	0.9975	0.9995	0.8667
DEC	6.0007	1E-10	23.6360	0.8642	3E-11	5.5912	0.9972	1.0000	0.8304
AVG	**4.0235**	**2.0929**	**27.0152**	**0.5252**	**0.3313**	**6.1317**	**0.9973**	**0.9981**	**0.7480**

Table 8: Comparison of MAPE, RMSE and R^2 values between GP and ANNs during validation

MONTH	VALIDATION PHASE								
	MAPE (%)			RMSE			R^2		
	GP	ANN-1	ANN-2	GP	ANN-1	ANN-2	GP	ANN-1	ANN-2
JAN	3.4179	65.407	45.769	1.3535	29.1835	13.0723	0.9969	0.0992	0.8444
FEB	3.0490	22.707	61.138	1.0126	18.3385	19.1202	0.9923	0.9651	0.9509
MAR	4.3077	35.350	17.067	1.3797	20.2504	10.6233	0.9932	0.3559	0.6991
APR	6.1725	21.983	18.862	1.1868	6.3037	5.3593	0.9741	0.5174	0.6993
MAY	1.8900	57.279	57.748	0.1245	3.8866	3.9262	0.9954	0.2272	0.2234
JUN	0.5932	40.584	9.4256	0.0325	1.6614	0.4419	0.9990	0.6659	0.7441
JUL	0.7575	87.680	21.531	0.0257	2.5899	0.6496	0.9999	0.5788	0.5137
AUG	7.5730	58.147	46.672	0.3401	3.9421	1.3623	0.9881	0.0080	0.9306
SEP	5.1472	100.67	107.54	0.1983	1.8091	1.8671	0.9740	0.2979	0.2917
OCT	6.1092	125.09	58.171	0.2653	3.3390	1.7698	0.9905	0.0959	0.6667
NOV	2.1735	255.03	11.610	0.2918	24.8077	1.3853	0.9978	0.1231	0.9707
DEC	2.4802	89.395	87.983	0.5242	16.0652	8.3457	0.9978	0.0917	0.9295
AVG	**3.6392**	**65.407**	**45.769**	**0.5612**	**11.0148**	**5.6602**	**0.9916**	**0.3355**	**0.7053**

4.4.1 Case study 1

It can be observed from Table 7 that both the GP and ANN-1 models provided very competitive performance during the training phase, with the ANN-1 models having a slight edge over the GP models. During training, the ANN-1 models converged better towards zero than the GP models, while producing the lowest errors in the months of June and December. An agreement in the maximum MAPE estimates of the GP and ANN-1 models can be noticed, as both techniques (GP and ANN-1) produced maximum MAPE estimates in the month of September – 5.31% and 11.1% respectively. It can also be noted that the RMSE estimates recorded by the ANN-1 models was in line with the orientation of its MAPE estimates, as the minimum RMSE values converged towards zero and were also generated in the months of June and December. The maximum RMSE of 1.14 was however produced in the March model. Although, the performance of the ANN-1 models was marginally better in most of the months during training, some dominance by the GP models were however evident in the August, September and October models. The R^2 values in both techniques (GP and ANN) were found to be comparable, as high correlation between observed and predicted streamflows were recorded. The R^2 values ranged between 0.9918 – 1.0000 in the ANN-1 models, and 0.9891 – 0.9994 in the GP models. Generally, the performance of the GP and ANN-1 models during training can be said to be comparable, but with the ANN-1 models producing better convergence in most cases. This is in correlation with the study of Ni *et al*. (2010) where ANN performed slightly better than GP during the training phase.

On the other hand, GP models performed considerably better than the ANN-1 models during validation, as the ANN-1 produced higher error estimates. The errors produced in the GP models were better converged towards zero, and were estimated to be 0.6% - 7.6% and 0.03 – 1.38 for MAPE and RMSE respectively. In furtherance to that, all the GP models maintained the highly

positive correlations recorded during training, with R^2 values of 0.9740 – 0.9999. This proves the consistency of GP and showcases its ability to produce good generalization across training and validation datasets as also recorded by Wang et al. (2009). However, the error estimates in the ANN-1 models increased substantially, yielding higher MAPE and RMSE estimates, while generating lower R^2 values. This may be considered as a result of poor learning which arises when small amount of datasets are used for model training in ANNs (Zhang *et al.* 2010). It can be inferred that ANN-1 simply memorized the learning process as evident in the training results, and thus led to its inability to produce adequate generalization on the validation datasets as also experienced by Nourani *et al*. (2011). Furthermore, it was noticed that the optimization of the network architecture (number of hidden layer nodes) resulted in higher computational demands in terms of training time and computer memory. More understanding in this regard can be found from Table 9 which presents the optimal network architecture of the individual ANN models as determined by the DE algorithm, and returned at each end of the run.

Table 9: Network architecture showing number of hidden layer nodes in the ANN models

Month	Optimal Network Architecture	
	ANN-1	ANN-2
Jan	9-10-1	4-4-1
Feb	9-7-1	4-3-1
Mar	9-10-1	4-3-1
Apr	9-7-1	4-3-1
May	9-10-1	4-4-1
Jun	9-5-1	4-5-1
July	9-7-1	4-3-5
Aug	9-8-1	4-4-1
Sep	9-7-1	4-5-1
Oct	9-9-1	4-4-1
Nov	9-10-1	4-3-1
Dec	9-4-1	4-5-1

It was found during the runs that the training speed became slower with increase in number of hidden layer nodes. The reduction in training speed can be considered as a function of the increment in number of synaptic connections between units and adjacent network layers. Thus, as the number of hidden layer nodes increased, greater amount of weight/load was imposed on the network. This observation agrees with Karthikeyan *et al.* (2013)'s recent submission in their groundwater level prediction study, that the number of hidden layer nodes influences computational time. The authors stated further that ANNs require an ample period of time for continuous training in order to achieve better convergence when used on small datasets. Some researchers however opined that the idea of increasing the computational time should not be seen as a guarantee to achieving better generalization, as this effort may yield no practical improvement in the results (Ilonen *et al.* 2003; Ghaffari *et al.* 2006; Corzo and Solomatine 2007). Thus, there was no basis for subjecting the models to further training in this study.

In contrast, the GP models produced better generalization at a faster learning rate; a product of its ability to distribute the semi-isolated population space into multiple subpopulations, called demes. As earlier observed in chapter 3, and as pointed out by Babovic and Keijzer (2000), the distribution of the demes gave room for occasional migration of individuals between adjacent demes, thereby leading to the occurrence of parallel evolution, while also ensuring faster convergence. Although, all the simulations in this study were carried out on the same computer, the average computational time of the GP and ANN-1 models were 4 and 8 hours respectively.

Figures 17 and 18 present plots of observed against predicted streamflow values. The plots clearly show some few under- and over-estimations of the observed values by the ANN-1 models.

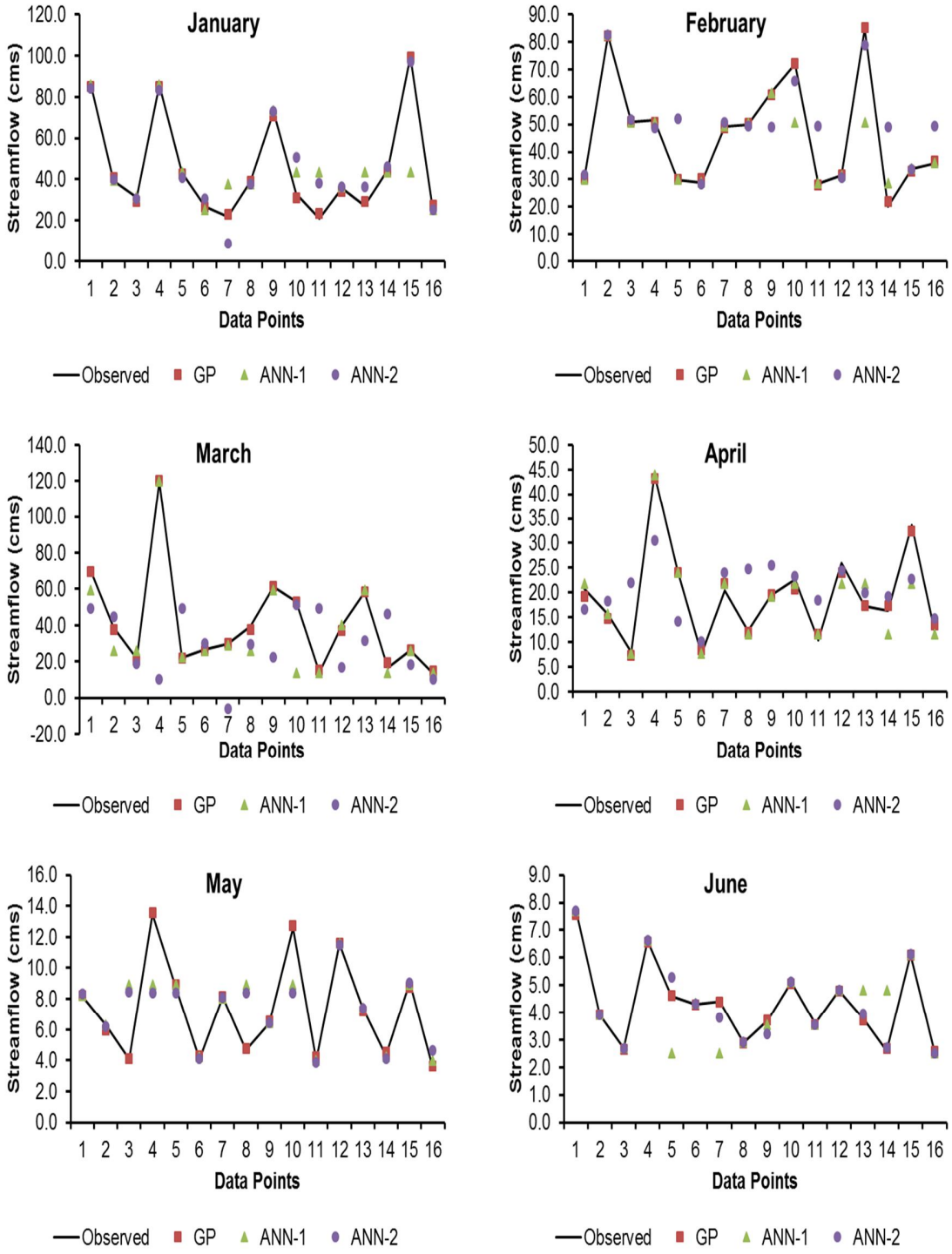

Figure 17: Observed and predicted streamflows by the GP and ANN models (January - June)

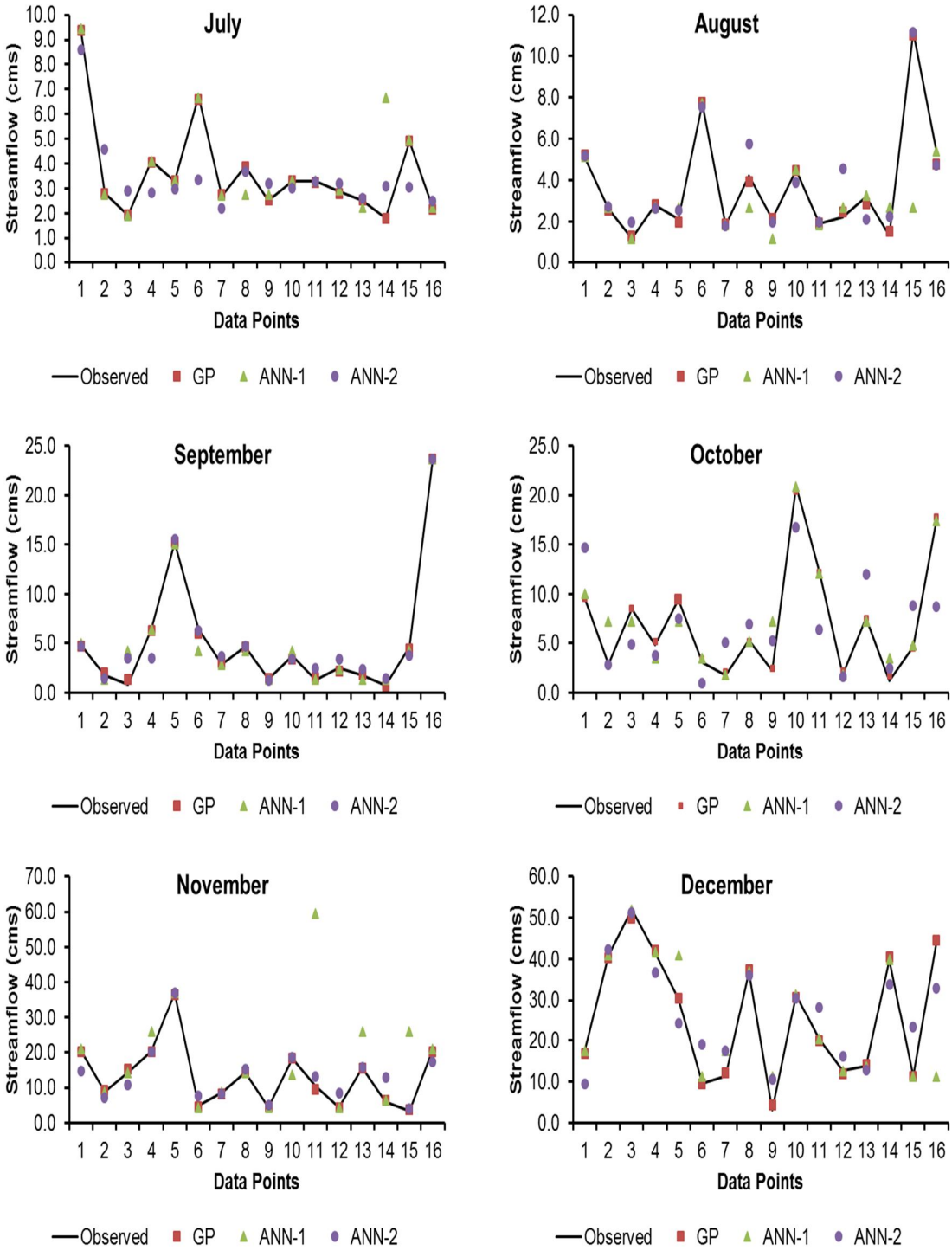

Figure 18: Observed and predicted streamflows by the GP and ANN models (July - December)

Despite ensuring that the validation datasets were within the range of the training datasets, the variations between observed and ANN-1 predicted values during the months of high flows (January, February and March) were more pronounced than that of the months characterized by low flows (June and July).

4.4.2 Case study 2

As presented in Tables 7 and 8, significant improvement can be noticed in the performance of the ANN-2 models compared to the ANN-1 models. Although, the error estimates produced during training were higher than those recorded by the ANN-1 models, it can be observed that the models achieved better convergence in the validation phase. This is evident as the errors produced during validation were far less than those produced during training. This implies that the overfitting and memorization problems that plagued the ANN-1 models were totally eliminated in the ANN-2 models. This is an indication that the introduction of the early stopping method in the second case study was effective in preventing the occurrence of overfitting in the ANN-2 models. Similar experience was reported in a rainfall-runoff modelling study conducted by Siou *et al*. (2012). In the study, early stopping was also found to have successfully prevented the occurrence of overfitting in ANN-derived models thereby resulting to improved forecast accuracy.

The superiority of the ANN-2 models over ANN-1 models is quite noticeable in the R^2 values, as the observed and ANN-2 predictions were reasonably correlated, with the exception of the May and September models. A computation of the relative MAPE, RMSE and R^2 differences between the ANN models during validation is presented in Table 10, with the ANN-2 models producing lower MAPE and RMSE estimates, and higher R^2 values in most of the months.

Table 10: Relative differences between the ANN models during validation

Month	Relative differences (%)		
	MAPE	RMSE	R^2
Jan	30.0	55.2	88.3
Feb	62.9	4.1	1.5
Mar	51.7	47.5	49.1
Apr	14.2	15.0	26.0
May	0.8	1.0	1.7
Jun	76.8	73.4	10.5
Jul	75.4	74.9	11.2
Aug	19.7	65.4	99.1
Sep	6.4	3.1	2.1
Oct	53.5	47.0	85.6
Nov	95.4	94.4	87.3
Dec	1.6	48.1	90.1
Average	**43.3**	**48.6**	**52.4**

It is clearly evident from the results that the incorporation of the GP technique and fine-tuning of the ANN learning process led to the improved performance of the ANN-2 models. The introduction of GP as a screening tool, to reduce the dimensionality of the models obviously facilitated quick convergence and reduced over-parameterization effects. In the same vein, the subjection of the DE algorithm to sensitivity analysis invigorated the search for better solutions, and consequently resulted into improved performance and reduced model complexity. Table 9 clearly shows that the ANN-2 models were able to produce better performance using a less complex architecture (number of hidden nodes). Thus, it can be said that the methodologies adopted in case study 2 prevented the imposition of higher amount of load on the network (as experienced in case study 1), and consequently translated into lesser computational time. This is in accordance with findings of Siou *et al*. (2012) where the introduction of early stopping and other optimization techniques also facilitated selection of optimal model complexity thereby translating into decreased computational time.

Although, results of the ANN-2 models were improved through the synergistic integration of the EC techniques (GP and DE), yet the ANN-2 models were unable to match the brilliant performance of the GP models in all aspects. The GP models exhibited dominance in terms of predictive accuracy, convergence rate and adaptation to rare occurrences of extreme events (Figure 17 and 18). The middling performance of the ANN models in this study points to the fact that the application of ANNs in modelling and prediction studies is problem specific and data dependent (Bhattacharya *et al.* 2001), and that high difficulty exists in modelling hydrological processes with limited datasets (Ni *et al.* 2010).

As a final point, the consistency and superiority of the GP technique in simulating the hydrological processes in the upper Mkomazi is evident, as the GP models were able to significantly reproduce the streamflow dynamics in the river for each month of the year. These results are very much in firm agreement with similar comparative studies such as Makkeasorn *et al.* (2008); Guven (2009) and Londhe and Charhate (2010), which have also found GP-derived models with superior performance when compared with the ANNs.

4.5 CONCLUSIONS

In this chapter, the performance of two data-driven modelling techniques namely genetic programming (GP) and differential evolution (DE)-trained artificial neural networks (ANNs) were investigated comparatively for monthly streamflow prediction using limited amount of datasets. Two case studies were considered in the application of ANNs. Correlation analysis, used for determining the predictor variables in the GP technique was adopted in the first case study. However, in the second case study, GP, DE and early stopping methods were integrated for optimization purposes. The ANN models in the first case study were found to be plagued by the problems of overfitting and over-parameterization, which is typical of ANNs when confronted with small-sized datasets. The ANN models developed in the second case study however

showcased improved performance through the conjunctive effort of early stopping method, GP and DE, but their performance was not enough to match that of the GP-derived models. In general, the performance of the GP models was found to be superior to the ANNs both in terms of predictive accuracy and computational speed despite the use of limited datasets.

4.6 RESEARCH OUTPUT

1. Josiah Adeyemo, Oluwaseun Oyebode and Fred Otieno (2014). Performance comparison of two data-driven modelling techniques for long term streamflow prediction using limited datasets. *Journal of the South African Institution of Civil Engineering (SAICE)*. Under review.
2. Oluwaseun Oyebode, Josiah Adeyemo and Fred Otieno (2014). An evolutionary approach for improving artificial neural networks performance in river flow forecasting. Paper accepted for presentation at the *International Journal of Arts and Sciences' (IJAS) American Canadian Conference for Academic Disciplines*, 19 – 22 May, 2014, Toronto, Canada.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 GENERAL CONCLUSIONS

In this study, two data-driven modelling techniques were applied to model the impacts of hydro-climatic variables on streamflow dynamics in the upper Mkomazi River, South Africa. The aim was to determine a suitable technique capable of representing the streamflow processes on monthly basis under limited availability of datasets. Genetic programming (GP) was first used in chapter 3, while its performance was investigated comparatively with that of differential evolution-trained artificial neural network (DE-ANN) models in chapter 4. Results obtained from both chapters have shown that the main objective of this study which was to model the impacts of hydro-climatic variables on streamflow dynamics in the upper Mkomazi River using limited datasets was achieved. Models developed using the GP technique was highly successful in reproducing the streamflow dynamics of the river.

As mentioned in chapter 1, in achieving the aim, this study has five specific objectives which were:

(a) To quantify the impacts of local climatic variables on streamflow in the upper Mkomazi River.
(b) To develop models that could be used for effective streamflow prediction in the upper Mkomazi River using two data-driven modelling techniques namely, genetic programming (GP) and differential evolution (DE)-trained artificial neural networks (ANN).
(c) To evaluate the potentials of the developed models individually using standard model evaluation criteria.

(d) To carry out performance comparison of the models so as to determine a suitable approach for streamflow prediction under limited availability of datasets in the upper Mkomazi River.
(e) To provide means of improving the performance of the developed models in order to achieve better predictive accuracy and reliability.

The specific objective (a) above was studied in chapter 3, as GP was used to determine the impacts of input variables used in this study. The determination of these impacts was a function of the contribution of each input variable to the fitness of the program models developed. GP's ability to screen and prioritize input variables during its algorithm run was instrumental to the achievement of this aim. Rainfall and temperature observations from the Shaleburn weather station for the same month in a given year, and streamflow observations for the same month in the previous two years were found to produce the highest impact on streamflow generation in the upper Mkomazi River. The input impact results further indicated that the methods applied in selecting the input variables used in this study were appropriately evaluated, as all inputs were shown to have contributed to the fitness of the developed models.

Also in chapter 3, GP was successfully applied to develop individual monthly models which were found capable of simulating the streamflow processes in the upper Mkomazi River despite using the few available datasets. The GP models were able to capture the streamflow patterns. The low, intermediate and high flows were substantially reproduced by GP. The occasional spikes which depict rare occurrences of extreme events were adequately captured by the GP models. The GP models also exhibited great potentials in terms of quick convergence towards global optimum, while also producing good generalization ability.

In chapter 4, the DE-ANNs were employed in order to have a basis for comparison with the GP models, thereby addressing the specific objectives (b), (c) and (e). Results from the two case studies considered have shown the superiority of the GP models over the ANNs. The ANNs were unable to adequately reproduce the streamflow dynamics of the upper Mkomazi River as effectively achieved by the GP models. The ANNs developed in case study 1 in particular suffered from overfitting and over-parameterization problems which were evident in their results across the three model performance criteria used in this study (MAPE, RMSE and R^2). A significant improvement in the performance of the ANNs developed in the second case study was however achieved. It was demonstrated in chapter 4 that the incorporation of GP, introduction of early stopping method and fine-tuning of the DE control parameters could lead to improvement in the performance of ANN-derived models.

One of the advantages derived from the use of GP in this study was its ability to screen and prioritize inputs variables which was found to be effective in reducing the dimensionality of the ANN models in the second case study. This led to the minimization of over-parameterization effects. Similarly, the introduction of the early stopping method was able to totally eliminate overfitting problems which were experienced in case study 1. Furthermore, the DE algorithm also demonstrated robustness in improving the performance of the ANN models via invigorated search for better solutions. Thus, this study has showcased various techniques that can be used to improve the performance of the ANN models in streamflow modelling.

Although difficulty exists in modelling hydrological processes using small amount of datasets, this study has demonstrated the consistency, efficacy and reliability of GP in modelling complex hydrological processes in the absence of large-sized historical records. Results from this study further indicate that evolutionary computation techniques can be easily integrated in any physically-based model

or other DDMs for the purpose of complementary modelling as remarked in chapter 2.

There is no doubt that the performance of the GP models developed in this study would improve when presented with larger amounts of datasets, thereby resulting into better representation of the hydro-climatic processes in the upper Mkomazi River catchment.

The GP models developed in this study can be deployed as predictive tools for planning and management of water resources within the Mkomazi region, most especially in optimizing the operations of the proposed dams when finally built. Furthermore, this study demonstrates GP as an alternative technique to be considered in hydrological modelling studies, especially as it relates to modelling of hydrological processes in data sparse regions.

5.2 RECOMMENDATIONS FOR FUTURE RESEARCH

The application of GP and other model development techniques used in this study can be extended for daily or hourly streamflow prediction (real-time forecasting) so as to provide information on short-term basis. Projections in this respect could help in giving signals on the impending dangers of flood and drought.

Following the success recorded in the application of GP in modelling the streamflow processes in the upper Mkomazi River, the methodology and techniques could be adopted for modelling of other regions of the catchment such as the lower Mkomazi River. Such studies could help in ensuring sustainability and maximization of available water resources to meet the water demands at the lower Mkomazi region, and also protect the wetlands, lakes and estuaries around the river mouth. The techniques used in this study could also be adopted for modelling purposes in other catchments in South Africa or Africa as a whole, especially in areas where limited datasets exists.

The GP models developed in this study which currently exist in C programming language can be developed into a decision support system (DSS) with user-friendly interface. This will enable data to be easily sent to the codes, thereby facilitating the application of the models by users with little knowledge of computing.

The incorporation of remote sensing data via the geographic information systems (GIS) into DDMs such as GP and ANN could also be subjected to test. GIS may help to generate more data which could facilitate the spatial representation of hydrological processes within a river catchment, provided the input variables have good relationships to the outputs being modelled.

On a final note, further studies can be directed towards the incorporation of promising evolutionary computation techniques such as GP and DE in climate change studies, especially as it concerns downscaling of climate projections from global climate models (GCMs), and also in uncertainty assessment in process-based hydrological models.

REFERENCES

Abbot, M., Bathurst, J., Cunge, J., O'Connell, P. and Rasmussen, J. 1986. An introduction to the European Hydrological System-Systeme Hydrologique Europeen," SHE", 1: History and philosophy of a physically-based, distributed modelling system. *Journal of Hydrology*, 87: 45-59.

Abdul-Kader, H. 2009. Neural Networks Training Based on Differential Evolution Algorithm Compared with Other Architectures for Weather Forecasting34. *IJCSNS*, 9 (3): 92.

Abebe, A. and Price, R. 2003. Managing uncertainty in hydrological models using complementary models. *Hydrological Sciences Journal*, 48 (5): 679-692.

Abebe, A., Solomatine, D. and Venneker, R. 2000. Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events. *Hydrological Sciences Journal*, 45 (3): 425-436.

Abolpour, B., Javan, M. and Karamouz, M. 2007. Water allocation improvement in river basin using adaptive neural fuzzy reinforcement learning approach. *Applied Soft Computing*, 7 (1): 265-285.

Abrahart, R., Kneale, P. E. and See, L. M. 2004. Towards a Hydrological Research Agenda. In: Abrahart, R., Kneale, P. E. and See, L. M. eds. *Neural networks for hydrological modeling.* Taylor & Francis, 289-304.

Abrahart, R., See, L. and Kneale, P. 1999. Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model. *Journal of hydroinformatics*, 1: 103-114.

Abrahart, R. J., Anctil, F., Coulibaly, P., Dawson, C. W., Mount, N. J., See, L. M., Shamseldin, A. Y., Solomatine, D. P., Toth, E. and Wilby, R. L. 2012. Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. *Progress in Physical Geography*, 36 (4): 480-513.

Abrahart, R. J. and See, L. 2000. Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes*, 14 (11-12): 2157-2172.

Abrahart, R. J. and See, L. 2002. Multi-model data fusion for river flow forecasting: an evaluation of six alternative methods based on two contrasting catchments. *Hydrology and Earth System Sciences Discussions*, 6 (4): 655-670.

Abrahart, R. J., See, L. M., Dawson, C. W., Shamseldin, A. Y. and Wilby, R. L. 2010. Nearly Two Decades of Neural Network Hydrologic Modelling. In: Sivakumar, B. and Berndtsson, R. eds. *Advances in data-based approaches for hydrologic modeling and forecasting.* Singapore: World Scientific, 267-346.

Adeyemo, J. and Otieno, F. 2010. Differential evolution algorithm for solving multi-objective crop planning model. *Agricultural Water Management*, 97 (6): 848-856.

Adriaenssens, V., Baets, B. D., Goethals, P. L. and Pauw, N. D. 2004. Fuzzy rule-based models for decision support in ecosystem management. *Science of the total environment*, 319 (1): 1-12.

Aha, D. W., Kibler, D. and Albert, M. K. 1991. Instance-based learning algorithms. *Machine learning*, 6 (1): 37-66.

Ajmera, T. K. and Goyal, M. K. 2012. Development of stage–discharge rating curve using model tree and neural networks: An application to Peachtree Creek in Atlanta. *Expert Systems with Applications*, 39 (5): 5702-5710.

Akbari, M., van Overloop, P. J. and Afshar, A. 2011. Clustered K nearest neighbor algorithm for daily inflow forecasting. *Water Resources Management*, 25 (5): 1341-1357.

Akrami, S. A., El-Shafie, A. and Jaafar, O. 2013. Improving Rainfall Forecasting Efficiency Using Modified Adaptive Neuro-Fuzzy Inference System (MANFIS). *Water Resources Management*: 1-17.

Al-Alawi, S. M., Abdul-Wahab, S. A. and Bakheit, C. S. 2008. Combining principal component regression and artificial neural networks for more accurate predictions of ground-level ozone. *Environmental Modelling & Software*, 23 (4): 396-403.

Amalaman, P. K., Eick, C. F. and Rizk, N. 2013. Using Turning Point Detection to Obtain Better Regression Trees. In: *Machine Learning and Data Mining in Pattern Recognition.* Springer, 325-339.

Anctil, F., Lauzon, N. and Filion, M. 2008. Added gains of soil moisture content observations for streamflow predictions using neural networks. *Journal of Hydrology*, 359 (3): 225-234.

Aqil, M., Kita, I., Yano, A. and Nishiyama, S. 2007a. Analysis and prediction of flow from local source in a river basin using a Neuro-fuzzy modeling tool. *Journal of environmental management*, 85 (1): 215-223.

Aqil, M., Kita, I., Yano, A. and Nishiyama, S. 2007b. A comparative study of artificial neural networks and neuro-fuzzy in continuous modeling of the daily and hourly behaviour of runoff. *Journal of Hydrology*, 337 (1): 22-34.

ASCE. 2000a. Artificial neural networks in hydrology. I: Preliminary concepts. *Journal of Hydrologic Engineering*, 5 (2): 115-123.

ASCE. 2000b. Artificial neural networks in hydrology. II: Hydrology Applications. *Journal of Hydrologic Engineering*, 5 (2): 124-137.

Asefa, T., Kemblowski, M., McKee, M. and Khalil, A. 2006. Multi-time scale stream flow predictions: The support vector machines approach. *Journal of Hydrology*, 318 (1): 7-16.

Auria, L. and Moro, R. A. 2008. Support Vector Machines (SVM) as a technique for solvency analysis. *DIW Berlin*: Available: http://www.diw.de/english/products/publications/discussion_papers/27539.html

Babovic, V. and Keijzer, M. 2000. Genetic programming as a model induction engine. *Journal of hydroinformatics*, 2: 35-60.

Babovic, V. and Keijzer, M. 2002. Rainfall runoff modelling based on genetic programming. *Nordic Hydrology*, 33 (5): 331-346.

Babovic, V. and Rao, R. 2010. Evolutionary Computing in Hydrology. In: Sivakumar, B. and Berndtsson, R. eds. *Advances in data-based approaches for hydrologic modeling and forecasting.* Singapore: World Scientific, 347-369.

Bagis, A. and Karaboga, D. 2004. Artificial neural networks and fuzzy logic based control of spillway gates of dams. *Hydrological Processes*, 18 (13): 2485-2501.

Bannayan, M. and Hoogenboom, G. 2008. Predicting realizations of daily weather data for climate forecasts using the non-parametric nearest-neighbour re-sampling technique. *International Journal of Climatology*, 28 (10): 1357-1368.

Baratta, D., Cicioni, G., Masulli, F. and Studer, L. 2003. Application of an ensemble technique based on singular spectrum analysis to daily rainfall forecasting. *Neural Networks*, 16 (3): 375-387.

Baratti, R., Cannas, B., Fanni, A., Pintus, M., Sechi, G. M. and Toreno, N. 2003. River flow forecast for reservoir management through neural networks. *Neurocomputing*, 55 (3): 421-437.

Bastola, S., Murphy, C. and Sweeney, J. 2011. The role of hydrological modelling uncertainties in climate change impact assessments of Irish river catchments. *Advances in Water Resources*, 34 (5): 562-576.

Bautu, A. and Bautu, E. 2006. Meteorological data analysis and prediction by means of genetic programming. In: Proceedings of *Proceedings of the 5th Workshop on Mathematical Modeling of Environmental and Life Sciences Problems Constanta, Romania.* 35-42

Behzad, M., Asghari, K., Eazi, M. and Palhang, M. 2009. Generalization performance of support vector machines and neural networks in runoff modeling. *Expert Systems with Applications*, 36 (4): 7624-7629.

Benyahya, L., Caissie, D., St-Hilaire, A., Ouarda, T. B. and Bobée, B. 2007. A review of statistical water temperature models. *Canadian Water Resources Journal*, 32 (3): 179-192.

Bergstrom, S. 1995. The HBV Model. In: Singh, V. P. ed. *Computer models of watershed hydrology.* Littleton, Colo: Water Resources Publications, 443-476.

Besaw, L. E., Rizzo, D. M., Bierman, P. R. and Hackett, W. R. 2010. Advances in ungauged streamflow prediction using artificial neural networks. *Journal of Hydrology*, 386 (1): 27-37.

Beven, K. 2002. Towards an alternative blueprint for a physically based digitally simulated hydrologic response modelling system. *Hydrological Processes*, 16 (2): 189-206.

Bhagwat, P. P. and Maity, R. 2013. Hydroclimatic streamflow prediction using Least Square-Support Vector Regression. *ISH Journal of Hydraulic Engineering*, (ahead-of-print): 1-9.

Bhattacharya, B. and Solomatine, D. P. 2005. Neural networks and M5 model trees in modelling water level–discharge relationship. *Neurocomputing*, 63: 381-396.

Bhattacharya, B. and Solomatine, D. P. 2006. Machine learning in sedimentation modelling. *Neural Networks*, 19 (2): 208-214.

Bhattacharya, M., Abraham, A. and Nath, B. 2001. A Linear Genetic Programming Approach for Modeling Electricity Demand Prediction in Victoria. In: Abraham, A. and Koppen, M. eds. Proceedings of *International Workshop on Hybrid Intelligent Systems.* Adelaide, Heidelberg: Physica-Verlag, 379-393

Bleuler, S., Brack, M., Thiele, L. and Zitzler, E. 2001. Multiobjective genetic programming: Reducing bloat using SPEA2. In: Proceedings of *2001 Congress on Evolutionary Computation.* IEEE, 536-543.
Boccaletti, S., Grebogi, C., Lai, Y.-C., Mancini, H. and Maza, D. 2000. The control of chaos: theory and applications. *Physics reports*, 329 (3): 103-197.

Boucher, M., Perreault, L. and Anctil, F. 2009. Tools for the assessment of hydrological ensemble forecasts obtained by neural networks. *Journal of hydroinformatics*, 11 (3-4): 297-307.

Bourdin, D. R., Fleming, S. W. and Stull, R. B. 2012. Streamflow Modelling: A Primer on Applications, Approaches and Challenges. *Atmosphere-Ocean*, 50 (4): 507-536.

Bowden, G. J., Dandy, G. C. and Maier, H. R. 2005a. Input determination for neural network models in water resources applications. Part 1—background and methodology. *Journal of Hydrology*, 301 (1): 75-92.

Bowden, G. J., Maier, H. R. and Dandy, G. C. 2005b. Input determination for neural network models in water resources applications. Part 2. Case study: forecasting salinity in a river. *Journal of Hydrology*, 301 (1): 93-107.

Brameier, M. F. and Banzhaf, W. 2001. A comparison of linear genetic programming and neural networks in medical data mining. *Evolutionary Computation, IEEE Transactions on*, 5 (1): 17-26.

Brameier, M. F. and Banzhaf, W. 2007. *Linear genetic programming*. USA: Springer.

Bray, M. and Han, D. 2004. Identification of support vector machines for runoff modelling. *Journal of hydroinformatics*, 6: 265-280.

Brigode, P., Oudin, L. and Perrin, C. 2013. Hydrological model parameter instability: A source of additional uncertainty in estimating the hydrological impacts of climate change? *Journal of Hydrology*, 476: 410-425.

Cannas, B., Montisci, A., Fanni, A., See, L. and Sechi, G. M. 2004. Comparing artificial neural networks and support vector machines for modelling rainfall-runoff. In: Proceedings of *6th International Conference on Hydroinformatics-Liong, Phoon & Babovic (eds).* World Scientific.

Castellano-Méndez, M. a., González-Manteiga, W., Febrero-Bande, M., Manuel Prada-Sánchez, J. and Lozano-Calderón, R. 2004. Modelling of the monthly and

daily behaviour of the runoff of the Xallas river using Box–Jenkins and neural networks methods. *Journal of Hydrology*, 296 (1): 38-58.

Chang, F.-J. and Chang, Y.-T. 2006. Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. *Advances in Water Resources*, 29 (1): 1-10.

Chang, F.-J. and Chen, Y.-C. 2001. A counterpropagation fuzzy-neural network modeling approach to real time streamflow prediction. *Journal of Hydrology*, 245 (1): 153-164.

Chang, L. C. and Chang, F. J. 2001. Intelligent control for modelling of real-time reservoir operation. *Hydrological Processes*, 15 (9): 1621-1634.

Chau, K. and Wu, C. 2010. A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. *Journal of hydroinformatics*, 12 (4): 458-473.

Chau, K., Wu, C. and Li, Y. 2005. Comparison of several flood forecasting models in Yangtze River. *Journal of Hydrologic Engineering*, 10 (6): 485-491.

Chauhan, N., Ravi, V. and Karthik Chandra, D. 2009. Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks. *Expert Systems with Applications*, 36 (4): 7659-7665.

Chen, Y.-h. and Chang, F.-J. 2009. Evolutionary artificial neural networks for hydrological systems forecasting. *Journal of Hydrology*, 367 (1): 125-137.

Cherkassky, V. and Mulier, F. M. 2007. *Learning from data: concepts, theory, and methods*. Wiley. com.

Cigizoglu, H. K. 2005. Application of generalized regression neural networks to intermittent flow forecasting and estimation. *Journal of Hydrologic Engineering*, 10 (4): 336-341.

Colvin, C., Le Maitre, D. and Lötter, D. 2009. *Water Resources and Climate Change Case Study, South African Risk and Vulnerability Atlas*. Available: http://www.rvatlas.org/casestudies/study05.php (Accessed 27 August 2013).

Corzo, G. and Solomatine, D. 2007. Baseflow separation techniques for modular artificial neural network modelling in flow forecasting. *Hydrological Sciences Journal*, 52 (3): 491-507.

Coulibaly, P., Anctil, F. and Bobee, B. 2001. Multivariate reservoir inflow forecasting using temporal neural networks. *Journal of Hydrologic Engineering*, 6 (5): 367-376.

Coulibaly, P. and Evora, N. 2007. Comparison of neural network methods for infilling missing daily weather records. *Journal of Hydrology*, 341 (1): 27-41.

Coulibaly, P., Haché, M., Fortin, V. and Bobée, B. 2005. Improving daily reservoir inflow forecasts with model combination. *Journal of Hydrologic Engineering*, 10 (2): 91-99.

Crowley, K. and van Vuuren, A. J. 2013. *South Africa Water Expenditure Needs Rise to $71 Billion*. 30 July 2013. Available: http://www.bloomberg.com/news/2013-07-30/south-africa-water-infrastructure-requires-700-billion-rand.html (Accessed 27 August 2013).

Cui, X. and Jiang, M. 2012. Chaotic Time Series Prediction Based On Binary Particle Swarm Optimization. *AASRI Procedia*, 1: 377-383.

Dawson, C. and Wilby, R. 2001. Hydrological modelling using artificial neural networks. *Progress in Physical Geography*, 25 (1): 80-108.

De Vos, N. and Rientjes, T. 2005. Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation. *Hydrology and Earth System Sciences Discussions*, 9 (1/2): 111-126.

Deo, M. C. 2009. Recent Data Driven Methods and Applications in Coastal and Hydrologic Data Analysis. *ISH Journal of Hydraulic Engineering*, 15 (sup1): 310-327.

Deschaine, L. M. and Francone, F. D. 2002. Comparison of Discipulus™ Linear Genetic Programming Soft-ware with Support Vector Machines, Classification Trees, Neural Networks and Human Experts. *Register Machine Learning Technologies Inc*.

Dhamge, N. R., Atmapoojya, S. and Kadu, M. S. 2012. Genetic Algorithm Driven ANN Model for Runoff Estimation. *Procedia Technology*, 6: 501-508.

Dibike, Y. B. and Coulibaly, P. 2006. Temporal neural networks for downscaling climate variability and extremes. *Neural Networks*, 19 (2): 135-144.

Dibike, Y. B., Velickov, S., Solomatine, D. and Abbott, M. B. 2001. Model induction with support vector machines: introduction and applications. *Journal of Computing in Civil Engineering*, 15 (3): 208-216.

Dolling, O. R. and Varas, E. A. 2002. Artificial neural networks for streamflow prediction. *Journal of Hydraulic Research*, 40 (5): 547-554.

Dorado, J., RabuñAL, J. R., Pazos, A., Rivero, D., Santos, A. and Puertas, J. 2003. Prediction and modeling of the rainfall-runoff transformation of a typical urban basin using ANN and GP. *Applied Artificial Intelligence*, 17 (4): 329-343.

DWA. 1999a. *Mkomazi/Mooi-Mgeni Transfer Scheme Pre-Feasibility Study: Mkomazi-Mgeni Transfer Scheme. Main Report.* South Africa: Department of Water Affairs in association with Umgeni Water.

DWA. 1999b. *Mkomazi/Mooi-Mgeni Transfer Scheme Pre-Feasibility Study: Mkomazi-Mgeni Transfer Scheme. Supporting Report No 4 - Hydrology and Water Resources.* South Africa: Department of Water Affairs and Forestry in association with Umgeni Water.

DWA. 1999c. *Mkomazi/Mooi-Mgeni Transfer Scheme Pre-Feasibility Study: Mkomazi-Mgeni Transfer Scheme. Supporting Report No 5 - Environmental.* South Africa: Department of Water Affairs and Forestry in association with Umgeni Water.

DWA. 2004. *Internal Strategic Perspective: Mvoti to Mzimkulu Water Management Area.* Department of Water Affairs and Forestry, South Africa. Available: http://www.dwaf.gov.za/Documents/Other/WMA/11/optimised/MVOTI%20TO%20MZIMKULU%20REPORT.pdf (Accessed 02 September 2013).

DWA. 2009. *Water for Growth and Development Framework: Version 7.* South Africa: Department of Water Affairs and Forestry. Available: http://www.dwaf.gov.za/WFGD/documents/WFGD_Frameworkv7.pdf (Accessed 02 September 2013).

Elshorbagy, A., Corzo, G., Srinivasulu, S. and Solomatine, D. 2010. Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology—part 1: concepts and methodology. *Hydrology and Earth System Sciences*, 14 (10): 1931-1941.

Elshorbagy, A. and El-Baroudy, I. 2009. Investigating the capabilities of evolutionary data-driven techniques using the challenging estimation of soil moisture content. *Journal of hydroinformatics*, 11 (3-4): 237-251.

Elshorbagy, A., Simonovic, S. and Panu, U. 2002a. Estimation of missing streamflow data using principles of chaos theory. *Journal of Hydrology*, 255 (1): 123-133.

Elshorbagy, A., Simonovic, S. and Panu, U. 2002b. Noise reduction in chaotic hydrologic time series: facts and doubts. *Journal of Hydrology*, 256 (3): 147-165.

Esfahanipour, A. and Mousavi, S. 2011. A genetic programming model to generate risk-adjusted technical trading rules in stock markets. *Expert Systems with Applications*, 38 (7): 8438-8445.

Fernando, T. M. K. G., Maier, H. R. and Dandy, G. C. 2009. Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. *Journal of Hydrology*, 367 (3): 165-176.

Flugel, W. and Marker, M. 2003. The response units concept and its application for the assessment of hydrologically related erosion processes in semiarid catchments of Southern Africa. *ASTM SPECIAL TECHNICAL PUBLICATION*, 1420: 163-177.

Francone, F. D. 1998. Discipulus™ Software Owner's Manual, version 3.0 DRAFT. *Register Machine Learning Technologies, Inc*.

Francone, F. D. 2011. Register Machine Learning Technologies, Inc. Discipulus Users Manual, Version 5. Available online at www.rmltech.com

Galeati, G. 1990. A comparison of parametric and non-parametric methods for runoff forecasting. *Hydrological Sciences Journal*, 35 (1): 79-94.

Galelli, S. and Castelletti, A. 2013. Assessing the predictive capability of randomized tree-based ensembles in streamflow modelling. *Hydrol. Earth Syst. Sci. Discuss*, 10: 1617-1655.

Ganesan, S. 2011. Regression Leaf Forest: A Fast and Accurate Learning Method for Large & High Dimensional Data Sets. MSc, University of Georgia.

Ganesan, T., Vasant, P. and Elamvazuthi, I. 2011. Optimization of nonlinear geological structure mapping using hybrid neuro-genetic techniques. *Mathematical and Computer Modelling*, 54 (11): 2913-2922.

Garcia-Almanza, A. L. and Tsang, E. P. 2007. Detection of stock price movements using chance discovery and genetic programming. *International*

Journal of Knowledge-based and Intelligent Engineering Systems, 11 (5): 329-344.

Garg, V. and Jothiprakash, V. 2013. Evaluation of reservoir sedimentation using data driven techniques. *Applied Soft Computing,* 13 (8): 3567-3581.

Ghaffari, A., Abdollahi, H., Khoshayand, M., Bozchalooi, I. S., Dadgar, A. and Rafiee-Tehrani, M. 2006. Performance comparison of neural network training algorithms in modeling of bimodal drug delivery. *International journal of pharmaceutics*, 327 (1): 126-138.

Ghanbarpour, M. R., Amiri, M., Zarei, M. and Darvari, Z. 2012. Comparison of stream flow predicted in a forest watershed using different modelling procedures: ARMA, ANN, SWRRB, and IHACRES models. *International Journal of River Basin Management*, 10 (3): 281-292.

Giustolisi, O. and Savic, D. 2006. A symbolic data-driven technique based on evolutionary polynomial regression. *Journal of hydroinformatics*, 8 (3): 207-222.

Giustolisi, O. and Simeone, V. 2006. Optimal design of artificial neural networks by a multi-objective strategy: groundwater level predictions. *Hydrological Sciences Journal*, 51 (3): 502-523.

Gjertsen, A. K. 2007. Accuracy of forest mapping based on Landsat TM data and a kNN-based method. *Remote Sensing of Environment*, 110 (4): 420-430.

Goyal, M. K. and Ojha, C. 2013. Evaluation of Rule and Decision Tree Induction Algorithms for Generating Climate Change Scenarios for Temperature and Pan Evaporation on a Lake Basin. *Journal of Hydrologic Engineering*.

Goyal, M. K., Ojha, C., Singh, R., Swamee, P. and Nema, R. 2013. Application of ANN, fuzzy logic and decision tree algorithms for the development of reservoir operating rules. *Water Resources Management*, 27 (3): 911-925.

Graham, L., Andersson, L., Horan, M., Kunz, R., Lumsden, T., Schulze, R., Warburton, M., Wilk, J. and Yang, W. 2011. Using multiple climate projections for assessing hydrological response to climate change in the Thukela River Basin, South Africa. *Physics and Chemistry of the Earth, Parts A/B/C*, 36 (14): 727-735.

Guven, A. 2009. Linear genetic programming for time-series modelling of daily flow rate. *Journal of earth system science*, 118 (2): 137-146.

Guven, A. and Kişi, Ö. 2011. Estimation of suspended sediment yield in natural rivers using machine-coded linear genetic programming. *Water Resources Management*, 25 (2): 691-704.

Heng, S. and Suetsugi, T. 2013. Using Artificial Neural Network to Estimate Sediment Load in Ungauged Catchments of the Tonle Sap River Basin, Cambodia. *Journal of Water Resource and Protection*, 5 (2): 111-123.

Hessenmoller, D. and Elsenhans, A. 2002. Comparison between parametric and non-parametric methods to predict the annual increment of beech. *Allgemeine Forst Und Jagdzeitung*, 173 (11-12): 216-223.

Hong, A. and Chen, A. 2012. Piecewise regression model construction with sample efficient regression tree (SERT) and applications to semiconductor yield analysis. *Journal of Process Control*, 22 (7): 1307-1317.

Hsu, K. l., Gupta, H. V., Gao, X., Sorooshian, S. and Imam, B. 2002. Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis. *Water Resources Research*, 38 (12): 38-31-38-17.

Hu, T., Wu, F. and Zhang, X. 2007. Rainfall-runoff modeling using principal component analysis and neural network. *Nordic Hydrology*, 38 (3): 235-248.

Hundecha, Y., Bardossy, A. and WERNER, H.-W. 2001. Development of a fuzzy logic-based rainfall-runoff model. *Hydrological Sciences Journal*, 46 (3): 363-376.

Ilonen, J., Kamarainen, J.-K. and Lampinen, J. 2003. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 17 (1): 93-105.

Jacquin, A. and Shamseldin, A. 2009. Review of the application of fuzzy inference systems in river flow forecasting. *Journal of hydroinformatics*, 11 (3-4): 202-210.

Jacquin, A., Shamseldin, A. Y., Webb, B., Arnell, N., Onof, C., MacIntyre, N., Gurney, R. and Kirby, C. 2004. Development of a river flow forecasting model for the Blue Nile River using Takagi-Sugeno-Kang fuzzy systems. In: Proceedings of *Hydrology: science and practice for the 21st century. Proceedings of the British Hydrological Society International Conference, Imperial College, London, July 2004.* British Hydrological Society, 291-296

Jacquin, A. P. and Shamseldin, A. Y. 2006. Development of rainfall–runoff models using Takagi–Sugeno fuzzy inference systems. *Journal of Hydrology*, 329 (1): 154-173.

Jagtap, S. A. 2013. Rainfall Runoff Modeling Using M5 Model Tree Techniques. *International Journal of Global Technology Initiatives*, 2 (1): A79-A83.

Jain, A. and Kumar, A. M. 2007. Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, 7 (2): 585-592.

Jayawardena, A. and Lai, F. 1994. Analysis and prediction of chaos in rainfall and stream flow time series. *Journal of Hydrology*, 153 (1): 23-52.

Jayawardena, A., Muttil, N. and Fernando, T. 2005. Rainfall-runoff modelling using genetic programming. In: Proceedings of *Proceedings of the MODSIM 2005 international congress on modelling and simulation: advances and applications for management and decision making. Melbourne, Australia.* 1841-1847.

Jha, G. K. 2007. *Artificial neural networks and its applications*. Available: http://www.iasri.res.in/ebook/ebadat/5-Modeling%20and%20Forecasting%20Techniques%20in%20Agriculture/5-ANN_GKJHA_2007.pdf (Accessed 10 June 2013).

Jia, D., Zheng, G., Qu, B. and Khan, M. K. 2011. A hybrid particle swarm optimization algorithm for high-dimensional problems. *Computers & Industrial Engineering*, 61 (4): 1117-1122.

Jin, H., Ooi, B. C., Shen, H. T., Yu, C. and Zhou, A. Y. 2003. An adaptive and efficient dimensionality reduction algorithm for high-dimensional indexing. In: Proceedings of *Data Engineering, 2003. Proceedings. 19th International Conference on.* IEEE, 87-98

Johanson, R. C., Imhoff, J. C., Kittle, J. L. and Donigan, A. 1984. Hydrological Simulation Program--FORTRAN(HSPF): Users Manual for Release 8. 0. *EPA-600/3-84-006 June 1984. Environmental Protection Agency, Athens, GA. 767 p, 138 p, 49 Ref, 5 Append. Contract/Grant*, (68-01).

Jordaan, E. M. 2002. Development of Robust Inferential Sensors: Industrial application of support vector machines for regression. PhD Thesis. Eindhoven University of Technology, Eindhoven, the Netherlands,

Jovanovic, B. B., Reljin, I. S. and Reljin, B. D. 2004. Modified ANFIS architecture-improving efficiency of ANFIS technique. In: Proceedings of *Neural*

Network Applications in Electrical Engineering, 2004. NEUREL 2004. 2004 7th Seminar on. IEEE, 215-220

Jung, N., Popescu, I., Kelderman, P., Solomatine, D. and Price, R. 2010. Application of model trees and other machine learning techniques for algal growth prediction in Yongdam reservoir, Republic of Korea. *Journal of hydroinformatics*, 12 (3): 262-274.

Kagoda, P. A., Ndiritu, J., Ntuli, C. and Mwaka, B. 2010. Application of radial basis function neural networks to short-term streamflow forecasting. *Physics and Chemistry of the Earth, Parts A/B/C*, 35 (13): 571-581.

Kalteh, A., Hjorth, P. and Berndtsson, R. 2008. Review of the self-organizing map (SOM) approach in water resources: Analysis, modelling and application. *Environmental Modelling & Software*, 23 (7): 835-845.

Kalteh, A. M. and Berndtsson, R. 2007. Interpolating monthly precipitation by self-organizing map (SOM) and multilayer perceptron (MLP). *Hydrological Sciences Journal*, 52 (2): 305-317.

Karlsson, M. and Yakowitz, S. 1987. Nearest-neighbor methods for nonparametric rainfall-runoff forecasting. *Water Resources Research*, 23 (7): 1300-1308.

Karthikeyan, L., Kumar, D. N., Graillot, D. and Gaur, S. 2013. Prediction of Ground Water Levels in the Uplands of a Tropical Coastal Riparian Wetland using Artificial Neural Networks. *Water Resources Management*, 27 (3): 871-883.

Kashid, S. S., Ghosh, S. and Maity, R. 2010. Streamflow prediction using multi-site rainfall obtained from hydroclimatic teleconnection. *Journal of Hydrology*, 395 (1-2): 23-38.

Kasiviswanathan, K., Pandian, R. S. R., Saravanan, S. and Agarwal, A. 2011. Genetic programming approach on evaporation losses and its effect on climate change for Vaipar Basin. *International Journal of Computer Science*, 8 (2): 269-274.

Katambara, Z. and Ndiritu, J. 2009. A fuzzy inference system for modelling streamflow: Case of Letaba River, South Africa. *Physics and Chemistry of the Earth, Parts A/B/C*, 34 (10): 688-700.

Katambara, Z. and Ndiritu, J. G. 2010. A hybrid conceptual–fuzzy inference streamflow modelling for the Letaba River system in South Africa. *Physics and Chemistry of the Earth, Parts A/B/C*, 35 (13): 582-595.

Keijzer, M. and Babovic, V. 1999. Dimensionally aware genetic programming. In: *Proceedings of the Genetic and Evolutionary computation Conference.* 1069-1076.

Kember, G., Flower, A. and Holubeshen, J. 1993. Forecasting river flow using nonlinear dynamics. *Stochastic Hydrology and Hydraulics*, 7 (3): 205-212.

Keskin, M. E., Taylan, D. and Terzi, O. 2006. Adaptive neural-based fuzzy inference system (ANFIS) approach for modelling hydrological time series. *Hydrological Sciences Journal*, 51 (4): 588-598.

Khan, M. S. and Coulibaly, P. 2006. Bayesian neural network for rainfall-runoff modeling. *Water Resources Research*, 42 (7): W07409.

Khan, S., Ganguly, A. and Saigal, S. 2005. Detection and predictive modeling of chaos in finite hydrological time series. *Nonlinear Processes in Geophysics*, 12 (1): 41-53.

Khan, S. A. and See, L. 2006. Rainfall-runoff modelling using data driven and statistical methods. In: Proceedings of *Advances in Space Technologies, 2006 International Conference on.* IEEE, 16-20

Khu, S. T., Liong, S. Y., Babovic, V., Madsen, H. and Muttil, N. 2001. Genetic Programming And Its Application In Real-Time Runoff Forecasting. *JAWRA Journal of the American Water Resources Association*, 37 (2): 439-451.

Kim, H.-J. and Tomppo, E. 2006. Model-based prediction error uncertainty estimation for< i> k</i>-nn method. *Remote Sensing of Environment*, 104 (3): 257-263.

Kisi, Ö. 2003. River flow modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 9 (1): 60-63.

Kisi, Ö. 2005. Daily river flow forecasting using artificial neural networks and auto-regressive models. *Turkish J. Eng. Env. Sci*, 29: 9-20.

Kişi, Ö. 2009. Neural networks and wavelet conjunction model for intermittent streamflow forecasting. *Journal of Hydrologic Engineering*, 14 (8): 773-782.

Kişi, Ö. 2010. River suspended sediment concentration modeling using a neural differential evolution approach. *Journal of Hydrology*, 389 (1): 227-235.

Kisi, O. and Cigizoglu, H. K. 2007. Comparison of different ANN techniques in river flow prediction. *Civil Engineering and Environmental Systems*, 24 (3): 211-231.

Kisi, O., Cimen, M. and Shiri, J. 2012a. Suspended sediment modeling using genetic programming and soft computing techniques. *Journal of Hydrology*. 450-451:48-58.

Kisi, O. and Shiri, J. 2011. Precipitation Forecasting Using Wavelet-Genetic Programming and Wavelet-Neuro-Fuzzy Conjunction Models. *Water Resources Management*, 25 (13): 3135-3152.

Kisi, O. and Shiri, J. 2012. River suspended sediment estimation by climatic variables implication: Comparative study among soft computing techniques. *Computers & Geosciences*, 43: 73-82.

Kisi, O., Shiri, J. and Nikoofar, B. 2012b. Forecasting daily lake levels using artificial intelligence approaches. *Computers & Geosciences*, 41: 169-180.

Kompare, B., Steinman, F., Cerar, U. and Dzeroski, S. 1997. Prediction of rainfall runoff from catchment by intelligent data analysis with machine learning tools within the artificial intelligence tools. *Acta hydrotechnica*, 16 (17): 79-94.

Koza, J. R. 1992. *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, Massachusetts, London, England: The MIT Press.

Kröse, B. and van der Smagt, P. 1996. *An introduction to neural networks*. Eight ed. Amsterdam: The University of Amsterdam.

Lachtermacher, G. and Fuller, J. D. 1994. Backpropagation in hydrological time series forecasting. In: *Stochastic and statistical methods in hydrology and environmental engineering.* Springer, 229-242.

Leander, R. and Buishand, T. A. 2007. Resampling of regional climate model output for the simulation of extreme river flows. *Journal of Hydrology*, 332 (3): 487-496.

Legates, D. R. and McCabe, G. J. 1999. Evaluating the use of “goodness-of-fit” measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35 (1): 233-241.

Li, T.-S., Tong, S.-C. and Feng, G. 2010. A novel robust adaptive-fuzzy-tracking control for a class of nonlinearmulti-input/multi-output systems. *Fuzzy Systems, IEEE Transactions on*, 18 (1): 150-160.

Li, X., Gao, G., Hu, T., Ma, H. and Li, T. 2013. Multiple time scales analysis of runoff series based on the Chaos Theory. *Desalination and Water Treatment*, (ahead-of-print): 1-9.

Lin, G.-F. and Chen, L.-H. 2004. A non-linear rainfall-runoff model using radial basis function network. *Journal of Hydrology*, 289 (1): 1-8.

Lin, J.-Y., Cheng, C.-T. and Chau, K.-W. 2006. Using support vector machines for long-term discharge prediction. *Hydrological Sciences Journal*, 51 (4): 599-612.

Lindström, G., Johansson, B., Persson, M., Gardelin, M. and Bergström, S. 1997. Development and test of the distributed HBV-96 hydrological model. *Journal of Hydrology*, 201 (1): 272-288.

Liong, S. Y., Gautam, T. R., Khu, S. T., Babovic, V., Keijzer, M. and Muttil, N. 2002. Genetic Programming: A New Paradigm in Rainfall Runoff Modeling. *JAWRA Journal of the American Water Resources Association*, 38 (3): 705-718.

Liong, S. Y. and Sivapragasam, C. 2002. Flood Stage Forecasting With Support Vector Machines. *JAWRA Journal of the American Water Resources Association*, 38 (1): 173-186.

Londhe, S. and Charhate, S. 2010. Comparison of data-driven modelling techniques for river flow forecasting. *Hydrological Sciences Journal*, 55 (7): 1163-1174.

Loucks, D. and van Beek, E. 2005. *Water Resources Systems Planning and Management: An Introduction to Methods, Models and Applications*. Delft Hydraulics, The Netherlands: UNESCO.

Luchetta, A. and Manetti, S. 2003. A real time hydrological forecasting system using a fuzzy clustering approach. *Computers & Geosciences*, 29 (9): 1111-1117.

Ma, M. 2013. Correlation Dimension analysis of complex hydrological systems: what information can the method provide? Berlin, Freie Universität Berlin, Diss., 2013.

Mafi, S., Yeganeh-Bakhtiary, A. and Kazeminezhad, M. H. 2013. Prediction formula for longshore sediment transport rate with M5'algorithm.

Magnussen, S., Tomppo, E. and McRoberts, R. E. 2010. A model-assisted k-nearest neighbour approach to remove extrapolation bias. *Scandinavian Journal of Forest Research*, 25 (2): 174-184.

Maier, H. R. and Dandy, G. C. 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling & Software*, 15 (1): 101-124.

Maier, H. R., Jain, A., Dandy, G. C. and Sudheer, K. P. 2010. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environmental Modelling & Software*, 25 (8): 891-909.

Maity, R. and Kashid, S. 2011. Importance analysis of local and global climate inputs for basin-scale streamflow prediction. *Water Resources Research*, 47 (11): W11504.

Maity, R. and Kashid, S. S. 2009. Hydroclimatological Approach for Monthly Streamflow Prediction Using Genetic Programming. *ISH Journal of Hydraulic Engineering*, 15 (2): 89-107.

Makkeasorn, A., Chang, N. B. and Zhou, X. 2008. Short-term streamflow forecasting with global climate change implications – A comparative study between genetic programming and neural network models. *Journal of Hydrology*, 352 (3-4): 336-354.

Maltamo, M., Malinen, J., Kangas, A., Härkönen, S. and Pasanen, A. M. 2003. Most similar neighbour-based stand variable estimation for use in inventory by compartments in Finland. *Forestry*, 76 (4): 449-464.

Mamdani, E. H. 1974. Application of fuzzy algorithms for control of simple dynamic plant. *Electrical Engineers, Proceedings of the Institution of*, 121 (12): 1585-1588.

Maskey, S., Guinot, V. and Price, R. K. 2004. Treatment of precipitation uncertainty in rainfall-runoff modelling: a fuzzy set approach. *Advances in Water Resources*, 27 (9): 889-898.

Masters, T. 1993. *Practical neural network recipes in C++*. Morgan Kaufmann.

Mathur, S. 2012. Particle swarm optimization trained neural network for aquifer parameter estimation. *KSCE Journal of Civil Engineering*, 16 (3): 298-307.

Mehr, A., Kahya, E. and Olyaie, E. 2013. Streamflow prediction using linear genetic programming in comparison with a neuro-wavelet technique. *Journal of Hydrology*, 505: 240-249.

Mihalache, C. R. and Leon, F. 2009. Functional approximation using neuro-genetic hybrid systems. *Buletinul Institutului Politehnic Din Iasi*, Tomul LV (LIX): 87-102.

Minns, A. and Hall, M. 1996. Artificial neural networks as rainfall-runoff models. *Hydrological Sciences Journal*, 41 (3): 399-417.

Mohanty, S., Jha, M. K., Kumar, A. and Panda, D. 2013. Comparative Evaluation of Numerical Model and Artificial Neural Network for Simulating Groundwater Flow in Kathajodi-Surua Inter-Basin of Odisha, India. *Journal of Hydrology*, 495: 38-51.

Mohanty, S., Jha, M. K., Kumar, A. and Sudheer, K. 2010. Artificial neural network modeling for groundwater level forecasting in a River Island of Eastern India. *Water Resources Management*, 24 (9): 1845-1865.

Montanari, A. and Di Baldassarre, G. 2013. Data errors and hydrological modelling: The role of model structure to propagate observation uncertainty. *Advances in Water Resources*, (51): 498-504.

Moradkhani, H., Hsu, K.-l., Gupta, H. V. and Sorooshian, S. 2004. Improved streamflow forecasting using self-organizing radial basis function artificial neural networks. *Journal of Hydrology*, 295 (1): 246-262.

Moriasi, D., Arnold, J., Van Liew, M., Bingner, R., Harmel, R. and Veith, T. 2007. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50 (3): 885-900.

Mugumo, M. 2012. A simple operating model of the Van der Kloof Reservoir using ANN streamflow forecasts. MSc Dissertation, University of the Witwatersrand, South Africa.

Muleta, M. K. and Nicklow, J. W. 2004. Joint application of artificial neural networks and evolutionary algorithms to watershed management. *Water Resources Management*, 18 (5): 459-482.

Muttil, N. and Lee, J. H. W. 2005. Genetic programming for analysis and real-time prediction of coastal algal blooms. *Ecological Modelling*, 189 (3-4): 363-376.

Nasseri, M., Tavakol-Davani, H. and Zahraie, B. 2013. Performance assessment of different data mining methods in statistical downscaling of daily precipitation. *Journal of Hydrology*, 492: 1-14.

Nayak, P., Sudheer, K. and Jain, S. 2007. Rainfall-runoff modeling through hybrid intelligent system. *Water Resources Research*, 43 (7).

Nayak, P., Sudheer, K., Rangan, D. and Ramasastri, K. 2004. A neuro-fuzzy computing technique for modeling hydrological time series. *Journal of Hydrology*, 291 (1): 52-66.

Nayak, P., Sudheer, K., Rangan, D. and Ramasastri, K. 2005. Short-term flood forecasting with a neurofuzzy model. *Water Resources Research*, 41 (4): 1-16.

Ni, Q., Wang, L., Ye, R., Yang, F. and Sivakumar, M. 2010. Evolutionary modeling for streamflow forecasting with minimal datasets: a case study in the West Malian River, China. *Environmental Engineering Science*, 27 (5): 377-385.

Ni, Q., Wang, L., Zheng, B. and Sivakumar, M. 2012. Evolutionary Algorithm for Water Storage Forecasting Response to Climate Change with Small Data Sets: The Wolonghu Wetland, China. *Environmental Engineering Science*, 29 (8): 814-820.

Nourani, V., Kisi, Ö. and Komasi, M. 2011. Two hybrid artificial intelligence approaches for modeling rainfall–runoff process. *Journal of Hydrology*, 402 (1): 41-59.

Ostfeld, A. and Salomons, S. 2005. A hybrid genetic—instance based learning algorithm for CE-QUAL-W2 calibration. *Journal of Hydrology*, 310 (1): 122-142.

Pal, S., Qu, B., Das, S. and Suganthan, P. 2010. Optimal synthesis of linear antenna arrays with multi-objective differential evolution. *Progress in Electromagnetics Research, PIER B*, 21: 87-111.

Pal, M., Singh, N. and Tiwari, N. 2012. M5 model tree for pier scour prediction using field dataset. *KSCE Journal of Civil Engineering*, 16 (6): 1079-1084.

Parasuraman, K. and Elshorbagy, A. 2007. Cluster-based hydrologic prediction using genetic algorithm-trained neural networks. *Journal of Hydrologic Engineering*, 12 (1): 52-62.

Parasuraman, K., Elshorbagy, A. and Carey, S. K. 2007. Modelling the dynamics of the evapotranspiration process using genetic programming. *Hydrological Sciences Journal*, 52 (3): 563-578.

Partal, T. and Cigizoglu, H. K. 2008. Estimation and forecasting of daily suspended sediment data using wavelet–neural networks. *Journal of Hydrology*, 358 (3): 317-331.

Pasternack, G. B. 1999. Does the river run wild? Assessing chaos in hydrological systems. *Advances in Water Resources*, 23 (3): 253-260.

Piotrowski, A. P. and Napiorkowski, J. J. 2011. Optimizing neural networks for river flow forecasting–Evolutionary Computation methods versus the Levenberg–Marquardt approach. *Journal of Hydrology*, 407 (1): 12-27.

Poli, R., Langdon, W. W. B. and McPhee, N. F. 2008. *A Field Guide to Genetic Programming*. Lulu Enterprises UK Limited.

Porter, J. and McMahon, T. 1971. A model for the simulation of streamflow data from climatic records. *Journal of Hydrology*, 13: 297-324.

Prairie, J. R., Rajagopalan, B., Fulp, T. J. and Zagona, E. A. 2006. Modified K-NN model for stochastic streamflow simulation. *Journal of Hydrologic Engineering*, 11 (4): 371-378.

Preis, A., Tubaltzev, A. and Ostfeld, A. 2006. Kinneret watershed analysis tool: a cell-based decision tree model for watershed flow and pollutants predictions. *Water science and technology*, 53 (10): 29-36.

Price, K. and Storn, R. 2013. *Differential Evolution (DE) for Continuous Function Approximation*. Available: http://www1.icsi.berkeley.edu/~storn/code.html (Accessed 19 July 2013).

Qian, B., Wang, L., Hu, R., Huang, D. and Wang, X. 2009. A DE-based approach to no-wait flow-shop scheduling. *Computers & Industrial Engineering*, 57 (3): 787-805.

Quinlan, J. R. 1992. Learning with continuous classes. In: Proceedings of *Proceedings of the 5th Australian joint Conference on Artificial Intelligence.* Singapore, 343-348.

Raghunath, H. M. 2007. *Hydrology: principles, analysis and design*. Revised Second ed. New Delhi: New Age International (P) Ltd.

Raghuwanshi, N., Singh, R. and Reddy, L. 2006. Runoff and sediment yield modeling using artificial neural networks: Upper Siwane River, India. *Journal of Hydrologic Engineering*, 11 (1): 71-79.

Rasouli, K., Hsieh, W. W. and Cannon, A. J. 2012. Daily streamflow forecasting by machine learning methods with weather and climate inputs. *Journal of Hydrology*, 414-415: 284-293.

Rivers-Moore, N., Goodman, P. and Nkosi, M. 2007. An assessment of the freshwater natural capital in KwaZulu-Natal for conservation planning. *Water SA*, 33 (5): 665-674.

Robertson, D. and Wang, Q. 2009. Selecting predictors for seasonal streamflow predictions using a Bayesian joint probability (BJP) modelling approach. In: Proceedings of *18th World IMACS/MODSIM Congress.* Cairns, 13-17 July 2009, 376-382.

Rodriguez-Iturbe, I., Febres De Power, B., Sharifi, M. B. and Georgakakos, K. P. 1989. Chaos in rainfall. *Water Resources Research*, 25 (7): 1667-1675.

Rumelhart, D., Hinton, G. E. and Williams, R. J. 1986. Learning internal representations by error propagation. In: Rumelhart, D. E. and McClelland, J. L. eds. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition.* Cambridge, MA The MIT Press, 318–362.

Sætrom, P., Sneve, R., Kristiansen, K. I., Snøve, O., Grünfeld, T., Rognes, T. and Seeberg, E. 2005. Predicting non-coding RNA genes in Escherichia coli with boosted genetic programming. *Nucleic acids research*, 33 (10): 3263-3270.

Samadi, M., Jabbari, E. and Azamathulla, H. M. 2012. Assessment of M5′ model tree and classification and regression trees for prediction of scour depth below free overfall spillways. *Neural Computing and Applications*: 1-10.

Sattari, M. T., Pal, M., Apaydin, H. and Ozturk, F. 2013. M5 model tree application in daily river flow forecasting in Sohu Stream, Turkey. *Water Resources*, 40 (3): 233-242.

Schertzer, D., Tchiguirinskaia, I., Lovejoy, S., Hubert, P., Bendjoudi, H. and Larchevêque, M. 2002. DISCUSSION of “Evidence of chaos in the rainfall-runoff process” Which chaos in the rainfall-runoff process? *Hydrological Sciences Journal*, 47 (1): 139-148.

Scheuber, M. 2010. Potentials and limits of the k-nearest-neighbour method for regionalising sample-based data in forestry. *European Journal of Forest Research*, 129 (5): 825-832.

Schulze, R. E. 1995. *Hydrology and agrohydrology: A text to accompany the ACRU 3.00 agrohydrological modelling system*. Water Research Commission.

See, L. and Openshaw, S. 2000. A hybrid multi-model approach to river level forecasting. *Hydrological Sciences Journal*, 45 (4): 523-536.

Selle, B. and Muttil, N. 2011. Testing the structure of a hydrological model using Genetic Programming. *Journal of Hydrology*, 397 (1): 1-9.

Shabri, A. and Suhartono. 2012. Streamflow forecasting using least-squares support vector machines. *Hydrological Sciences Journal*, 57 (7): 1275-1293.

Shamseldin, A. Y. 2004. Hybrid neural network modelling solutions. In: Abrahart, R. J., Kneale, P. E. and See, L. M. eds. *Neural networks for hydrological modeling.* Leiden: A.A. Balkema Publishers, 61-79.

Shamseldin, A. Y. and O'Connor, K. M. 1996. A nearest neighbour linear perturbation model for river flow forecasting. *Journal of Hydrology*, 179 (1): 353-375.

Sharif, M. and Burn, D. H. 2006. Simulating climate change scenarios using an improved K-nearest neighbor model. *Journal of Hydrology*, 325 (1): 179-196.

Sharma, S. K. and Chandra, P. 2010. Constructive Neural Networks: A Review. *International Journal of Engineering Science and Technology*, 2 (12): 7847-7855.

Sheta, A. F. and Mahmoud, A. 2001. Forecasting using genetic programming. In: Proceedings of *System Theory, 2001. Proceedings of the 33rd Southeastern Symposium on.* IEEE, 343-347

Shiri, J. and Kisi, O. 2010. Short-term and long-term streamflow forecasting using a wavelet and neuro-fuzzy conjunction model. *Journal of Hydrology*, 394 (3-4): 486-493.

Shokir, E. M. E.-M. 2008. Dewpoint pressure model for gas condensate reservoirs based on genetic programming. *Energy & Fuels*, 22 (5): 3194-3200.

Shrestha, D., Kayastha, N. and Solomatine, D. 2009. A novel approach to parameter uncertainty analysis of hydrological models using neural networks. *Hydrology and Earth System Sciences*, 13 (7): 1235-1248.

Shrestha, D. L. and Solomatine, D. P. 2006. Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks*, 19 (2): 225-235.

Shrestha, D. L. and Solomatine, D. P. 2008. Data-driven approaches for estimating uncertainty in rainfall-runoff modelling. *International Journal of River Basin Management*, 6 (2): 109-122.

Silverman, D. and Dracup, J. A. 2000. Artificial neural networks and long-range precipitation prediction in California. *Journal of applied meteorology*, 39 (1): 57-66.

Singh, K. K., Pal, M. and Singh, V. 2010. Estimation of mean annual flood in indian catchments using backpropagation neural network and M5 model tree. *Water Resources Management*, 24 (10): 2007-2019.

Siou, L. K. A., Johannet, A., Valérie, B. E. and Pistre, S. 2012. Optimization of the generalization capability for rainfall–runoff modeling by neural networks: the case of the Lez aquifer (southern France). *Environmental Earth Sciences*, 65 (8): 2365-2375.

Sirdari, Z. Z., Ghani, A. A. and Hasan, Z. A. 2011. Prediction of bed load transport in Kurau River based on genetic programming. Paper presented at the *3rd International Conference on Managing Rivers in the 21st century: Sustainable Solutions for Global Crisis of Flooding, Pollution and Water Scarcity.* Penang, Malaysia, 385-390.

Sivakumar, B. 2003. Forecasting monthly streamflow dynamics in the western United States: a nonlinear dynamical approach. *Environmental Modelling & Software*, 18 (8): 721-728.

Sivakumar, B. 2009. Nonlinear dynamics and chaos in hydrologic systems: latest developments and a look forward. *Stochastic Environmental Research and Risk Assessment*, 23 (7): 1027-1036.

Sivakumar, B. and Berndtsson, R. 2010. Nonlinear Dynamics and Chaos in Hydrology. In: Sivakumar, B. and Berndtsson, R. eds. *Advances in data-based approaches for hydrologic modeling and forecasting.* Singapore: World Scientific, 411-461.

Sivakumar, B., Berndtsson, R., Olsson, J. and Jinno, K. 2001. Evidence of chaos in the rainfall-runoff process. *Hydrological Sciences Journal*, 46 (1): 131-145.

Sivakumar, B., Jayawardena, A. and Fernando, T. 2002. River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches. *Journal of Hydrology*, 265 (1): 225-245.

Sivapragasam, C., Liong, S. and Pasha, M. 2001. Rainfall and runoff forecasting with SSA-SVM approach. *Journal of hydroinformatics*, 3: 141-152.

Sivapragasam, C. and Muttil, N. 2005. Discharge rating curve extension–A new approach. *Water Resources Management*, 19 (5): 505-520.

Sivapragasam, C., Muttil, N. and Arun, V. 2011. Long term flow forecasting for water resources planning in a river basin. Paper presented at the *International Congress on Modelling and Simulation.* Perth, Australia, 12-16 December 2011. Available: http://mssanz.org.au/modsim2011 (Accessed 22 February 2013).

Slowik, A. and Bialko, M. 2008. Training of artificial neural networks using differential evolution algorithm. In: Proceedings of *Human System Interactions, 2008 Conference on.* IEEE, 60-65

Solomatine, D. and Dulal, K. N. 2003. Model trees as an alternative to neural networks in rainfall—Runoff modelling. *Hydrological Sciences Journal*, 48 (3): 399-411.

Solomatine, D., Maskey, M. and Shrestha, D. L. 2008. Instance-based learning compared to other data-driven methods in hydrological forecasting. *Hydrological Processes*, 22 (2): 275-287.

Solomatine, D. and Ostfeld, A. 2008. Data-driven modelling: some past experiences and new approaches. *Journal of hydroinformatics*, 10 (1): 3-22.

Solomatine, D., Rojas, C., Velickov, S. and Wust, H. 2000. Chaos theory in predicting surge water levels in the North Sea. In: Proceedings of *Proc. 4th Int. Conference on Hydroinformatics, Cedar-Rapids.*

Solomatine, D. P. and Siek, M. B. 2006. Modular learning models in forecasting natural phenomena. *Neural Networks*, 19 (2): 215-224.

Solomatine, D. P. and Xue, Y. 2004. M5 model trees and neural networks: application to flood forecasting in the upper reach of the Huai River in China. *Journal of Hydrologic Engineering*, 9 (6): 491-501.

Storn, R. and Price, K. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11 (4): 341-359.

Subudhi, B. and Jena, D. 2009. An improved differential evolution trained neural network scheme for nonlinear system identification. *International Journal of Automation and Computing*, 6 (2): 137-144.

Subudhi, B. and Jena, D. 2011. Nonlinear system identification using memetic differential evolution trained neural networks. *Neurocomputing*, 74 (10): 1696-1709.

Sudheer, C., Anand, N., Panigrahi, B. and Mathur, S. 2013. Streamflow forecasting by SVM with quantum behaved particle swarm optimization. *Neurocomputing*.

Sudheer, K. 2005. Knowledge extraction from trained neural network river flow models. *Journal of Hydrologic Engineering*, 10 (4): 264-269.

Sudheer, K., Gosain, A. and Ramasastri, K. 2002. A data-driven algorithm for constructing artificial neural network rainfall-runoff models. *Hydrological Processes*, 16 (6): 1325-1330.

Sudheer, K. and Jain, A. 2004. Explaining the internal behaviour of artificial neural network river flow models. *Hydrological Processes*, 18 (4): 833-844.

Sugeno, M. and Kang, G. 1988. Structure identification of fuzzy model. *Fuzzy sets and systems*, 28 (1): 15-33.

Swanson, D. A., Tayman, J. and Bryan, T. 2011. MAPE-R: a rescaled measure of accuracy for cross-sectional subnational population forecasts. *Journal of Population Research*, 28 (2-3): 225-243.

Takagi, T. and Sugeno, M. 1985. Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on*, (1): 116-132.

Taylor, V., Schulze, R. and Jewitt, G. 2003. Application of the indicators of hydrological alteration method to the Mkomazi River, KwaZulu-Natal, South Africa. *African Journal of Aquatic Science*, 28 (1): 1-11.

Thirumalaiah, K. and Deo, M. C. 2000. Hydrological forecasting using neural networks. *Journal of Hydrologic Engineering*, 5 (2): 180-189.

Thompson, J. A., Bell, J. C. and Butler, C. A. 2001. Digital elevation model resolution: effects on terrain attribute calculation and quantitative soil-landscape modeling. *Geoderma*, 100 (1): 67-89.

Toth, E. 2009. Classification of hydro-meteorological conditions and multiple artificial neural networks for streamflow forecasting. *Hydrology and Earth System Sciences*, 13 (9): 1555-1566.

Toth, E., Brath, A. and Montanari, A. 2000. Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of Hydrology*, 239 (1): 132-147.

Tripathi, S., Srinivas, V. and Nanjundiah, R. S. 2006. Downscaling of precipitation for climate change scenarios: a support vector machine approach. *Journal of Hydrology*, 330 (3): 621-640.

Tsakonas, A., Dounias, G., Jantzen, J., Axer, H., Bjerregaard, B. and von Keyserlingk, D. G. 2004. Evolving rule-based systems in two medical domains using genetic programming. *Artificial Intelligence in Medicine*, 32 (3): 195-216.

Tsonis, A. 2001. Probing the linearity and nonlinearity in the transitions of the atmospheric circulation. *Nonlinear Processes in Geophysics*, 8 (6): 341-345.

Valença, M. and Ludermir, T. 2000. Monthly stream flow forecasting using an neural fuzzy network model. In: Proceedings of *Neural Networks, 2000. Proceedings. Sixth Brazilian Symposium on.* Rio de Janeiro, 22-25 November 2000. IEEE, 117-119.

Vapnik, V. N. 1998. *Statistical learning theory*. New York: John Wiley & Sons.

Vernieuwe, H., Georgieva, O., De Baets, B., Pauwels, V., Verhoest, N. E. and De Troch, F. P. 2005. Comparison of data-driven Takagi–Sugeno models of rainfall–discharge dynamics. *Journal of Hydrology*, 302 (1): 173-186.

Vojinovic, Z. and Kecman, V. 2004. Contaminant Transport Modelling with Support Vector Machine Model–An Alternative to Classical Advection-Dispersion Equation. In: Proceedings of *Proceedings of the 6th International Conference.* World Scientific, 24.

Wang, W.-C., Chau, K.-W., Cheng, C.-T. and Qiu, L. 2009. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *Journal of Hydrology*, 374 (3): 294-306.

Wang, W., Gelder, P. H. V., Vrijling, J. and Ma, J. 2006a. Forecasting daily streamflow using hybrid ANN models. *Journal of Hydrology*, 324 (1): 383-399.

Wang, W., Vrijling, J., Van Gelder, P. H. and Ma, J. 2006b. Testing for nonlinearity of streamflow processes at different timescales. *Journal of Hydrology*, 322 (1): 247-268.

Wang, Y. and Witten, I. H. 1996. Induction of model trees for predicting continuous classes. *Working Paper 96/23: Department of Computer Science, University of Waikato, New Zealand*.

Whigham, P. and Crapper, P. 2001. Modelling rainfall-runoff using genetic programming. *Mathematical and Computer Modelling*, 33 (6): 707-721.

Wilcox, B. P., Seyfried, M. S. and Matison, T. H. 1991. Searching for chaotic dynamics in snowmelt runoff. *Water Resources Research*, 27 (6): 1005-1010.

Wu, C., Chau, K. and Li, Y. 2008. River stage prediction based on a distributed support vector regression. *Journal of Hydrology*, 358 (1): 96-111.

Xiong, L., Shamseldin, A. Y. and O'connor, K. M. 2001. A non-linear combination of the forecasts of rainfall-runoff models by the first-order Takagi–Sugeno fuzzy system. *Journal of Hydrology*, 245 (1): 196-217.

Yao, X. and Liu, Y. 1997. A new evolutionary system for evolving artificial neural networks. *Neural Networks, IEEE Transactions on*, 8 (3): 694-713.

Yates, D., Gangopadhyay, S., Rajagopalan, B. and Strzepek, K. 2003. A technique for generating regional climate scenarios using a nearest-neighbor algorithm. *Water Resources Research*, 39 (7).

Yonaba, H., Anctil, F. and Fortin, V. 2010. Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting. *Journal of Hydrologic Engineering*, 15 (4): 275-283.

Youssef, T., Chadli, M., Karimi, H. and Zelmat, M. 2013. Chaos Synchronization Based on Unknown Input Proportional Multiple-Integral Fuzzy Observer. *Abstract and Applied Analysis*, Volume 2013: 1-11.

Yu, P.-S., Chen, S.-T. and Chang, I.-F. 2006. Support vector regression for real-time flood stage forecasting. *Journal of Hydrology*, 328 (3): 704-716.

Zadeh, L. A. 1965. Fuzzy sets. *Information and control*, 8 (3): 338-353.

Zahiri, A. and Azamathulla, H. M. 2012. Comparison between linear genetic programming and M5 tree models to predict flow discharge in compound channels. *Neural Computing and Applications*: 1-8.

Zakaria, Z. A. and Shabri, A. 2012. Streamflow Forecasting at Ungauged Sites Using Support Vector Machines. *Applied Mathematical Sciences*, 6 (60): 3003-3014.

Zhang, L., Merényi, E., Grundy, W. M. and Young, E. F. 2010. Inference of Surface Parameters from Near-Infrared Spectra of Crystalline H2OH2O Ice with Neural Learning. *Publications of the Astronomical Society of the Pacific*, 122 (893): 839-852.

Zhang, X., Liang, F., Srinivasan, R. and Van Liew, M. 2009. Estimating uncertainty of streamflow simulation using Bayesian neural networks. *Water Resources Research*, 45 (2).

Zhu, C. and Hao, Z. 2012. Experiment and Simulation of Flood Increase along Lower Yellow River. *Journal of Computers*, 7 (8): 1817-1824.

APPENDIX A

SOURCE CODES FOR JANUARY GP MODEL IN C/C++

```
#include <math.h>
#include <float.h>
#define TRUNC(x)(((x)>=0) ? floor(x) : ceil(x))
#define C_FPREM (_finite(f[0]/f[1]) ? f[0]-(TRUNC(f[0]/f[1])*f[1]) : f[0]/f[1])
#define C_F2XM1 (((fabs(f[0])<=1) && (!_isnan(f[0]))) ? (pow(2,f[0])-1) :
((!_finite(f[0]) && !_isnan(f[0]) && (f[0]<0)) ? -1 : f[0]))


float DiscipulusCFunctionSubC0(float v[])
{
  long double f[8];
  long double tmp = 0;
  int cflag = 0;

  f[0]=f[1]=f[2]=f[3]=f[4]=f[5]=f[6]=f[7]=0;

  double Qt=v[0] ;
  double Qt-1=v[1] ;
  double Qt-2=v[2] ;
  double R1t=v[3] ;
  double R2t=v[4] ;
  double R3t=v[5] ;
  double T1t=v[6] ;
  double T2t=v[7] ;
```

```
double T3t=v[8] ;

L0:   cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L1:   tmp=f[1]; f[1]=f[0]; f[0]=tmp;
L2:   f[0]-=0.2174909114837647f;
L3:   f[0]-=T3t;
L4:   f[0]*=1.77857518196106f;
L5:   f[0]=cos(f[0]);
L6:   f[1]+=f[0];
L7:   f[0]=sin(f[0]);
L8:   cflag=((_isnan(f[0]) || _isnan(f[0])) ? true : (f[0] < f[0]));
L9:   tmp=f[1]; f[1]=f[0]; f[0]=tmp;
L10:  f[0]+=f[1];
L11:  f[0]=sqrt(f[0]);
L12:  cflag=((_isnan(f[0]) || _isnan(f[0])) ? true : (f[0] < f[0]));
L13:  f[0]-=f[1];
L14:  f[0]=fabs(f[0]);
L15:  f[0]/=f[0];
L16:  f[0]+=f[0];
L17:  tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L18:  f[0]=sqrt(f[0]);
L19:  if (cflag) f[0] = f[0];
L20:  if (cflag) f[0] = f[0];
L21:  f[0]*=pow(2,TRUNC(f[1]));
L22:  f[0]-=f[0];
L23:  f[0]=-f[0];
L24:  f[0]=cos(f[0]);
L25:  f[0]/=T2t;
L26:  f[0]=cos(f[0]);
```

```
L27:  f[0]/=R1t;
L28:  f[0]/=-0.9636838436126709f;
L29:  if (cflag) f[0] = f[1];
L30:  f[0]=fabs(f[0]);
L31:  f[0]*=Qt-1;
L32:  f[0]*=0.2174909114837647f;
L33:  f[0]=sqrt(f[0]);
L34:  f[1]*=f[0];
L35:  f[0]*=f[1];
L36:  f[1]*=f[0];
L37:  f[0]*=pow(2,TRUNC(f[1]));
L38:  f[1]*=f[0];
L39:  f[0]-=0.2174909114837647f;
L40:  f[0]-=T3t;
L41:  f[0]+=Qt-2;
L42:  f[0]=cos(f[0]);
L43:  f[1]+=f[0];
L44:  f[0]-=-1.642645597457886f;
L45:  f[0]+=f[1];
L46:  f[0]+=f[0];
L47:  f[0]/=f[1];
L48:  f[0]=sqrt(f[0]);
L49:  f[0]+=R3t;
L50:  f[0]-=R2t;
L51:  f[0]/=-1.196667432785034f;
L52:  f[0]*=pow(2,TRUNC(f[1]));
L53:  f[0]=fabs(f[0]);
L54:  f[0]+=R3t;
L55:  f[0]-=R2t;
```

```
  L56:  f[0]+=Qt-2;
  L57:  f[0]+=T1t;
  L58:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
  L59:  f[0]/=f[1];
  L60:  f[0]*=0.2174909114837647f;
  L61:  f[0]+=f[0];
  L62:  f[0]+=f[1];
  L63:  f[0]=fabs(f[0]);
  L64:  if (cflag) f[0] = f[1];
  L65:  f[0]+=f[1];
  L66:  f[0]-=1.987620830535889f;
  L67:  f[0]+=R3t;
  L68:  f[0]-=R2t;
  L69:  f[0]+=Qt;
  L70:  f[0]*=0.2174909114837647f;
  L71:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
  L72:  f[1]*=f[0];
  L73:  f[0]=-f[0];
  L74:  f[0]=fabs(f[0]);
  L75:  f[1]+=f[0];
  L76:  if (cflag) f[0] = f[1];
  L77:  f[0]+=T2t;
  L78:

  return f[0];
}


float DiscipulusCFunctionSubC1(float v[])
```

```
{
  long double f[8];
  long double tmp = 0;
  int cflag = 0;

  f[0]=f[1]=f[2]=f[3]=f[4]=f[5]=f[6]=f[7]=0;

  double Qt=v[0] ;
  double Qt-1=v[1] ;
  double Qt-2=v[2] ;
  double R1t=v[3] ;
  double R2t=v[4] ;
  double R3t=v[5] ;
  double T1t=v[6] ;
  double T2t=v[7] ;
  double T3t=v[8] ;

  L0:   f[0]-=f[0];
  L1:   f[0]=sqrt(f[0]);
  L2:   f[0]*=1.530829906463623f;
  L3:   f[0]+=Qt;
  L4:   f[0]*=Qt-2;
  L5:   f[0]+=f[0];
  L6:   cflag=((_isnan(f[0]) || _isnan(f[0])) ? true : (f[0] < f[0]));
  L7:   f[0]=cos(f[0]);
  L8:   cflag=((_isnan(f[0]) || _isnan(f[0])) ? true : (f[0] < f[0]));
  L9:   tmp=f[1]; f[1]=f[0]; f[0]=tmp;
  L10:  f[0]=-f[0];
  L11:  if (cflag) f[0] = f[1];
```

```
L12:  f[0]*=pow(2,TRUNC(f[1]));
L13:  f[0]+=-1.427085638046265f;
L14:  f[0]/=f[1];
L15:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L16:  tmp=f[1]; f[1]=f[0]; f[0]=tmp;
L17:  f[1]/=f[0];
L18:  f[0]+=f[0];
L19:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L20:  f[0]+=f[1];
L21:  tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L22:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L23:  if (!cflag) f[0] = f[0];
L24:  f[0]=-f[0];
L25:  if (cflag) f[0] = f[1];
L26:  f[0]*=f[1];
L27:  f[0]/=f[0];
L28:  cflag=((_isnan(f[0]) || _isnan(f[0])) ? true : (f[0] < f[0]));
L29:  f[0]=cos(f[0]);
L30:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L31:  f[0]-=T3t;
L32:  f[0]=sin(f[0]);
L33:  f[1]+=f[0];
L34:  f[0]/=Qt-1;
L35:  f[0]*=1.252994060516357f;
L36:  f[0]+=f[0];
L37:  if (!cflag) f[0] = f[1];
L38:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L39:  f[1]*=f[0];
L40:  if (!cflag) f[0] = f[0];
```

```
L41:  if (!cflag) f[0] = f[0];
L42:  f[0]-=1.501374244689941f;
L43:  f[1]/=f[0];
L44:  if (cflag) f[0] = f[0];
L45:  tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L46:  f[0]+=T3t;
L47:  f[0]+=f[1];
L48:  f[0]=-f[0];
L49:  f[0]=-f[0];
L50:  if (cflag) f[0] = f[0];
L51:  f[0]+=f[0];
L52:  f[0]-=f[1];
L53:  f[0]+=f[1];
L54:  if (!cflag) f[0] = f[0];
L55:  if (!cflag) f[0] = f[0];
L56:  if (!cflag) f[0] = f[0];
L57:  f[0]-=-1.924433708190918f;
L58:  f[0]*=Qt-2;
L59:  f[1]*=f[0];
L60:  tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L61:  f[0]-=f[0];
L62:  tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L63:  f[1]+=f[0];
L64:  f[0]*=f[1];
L65:  tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L66:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L67:  f[0]=cos(f[0]);
L68:  f[0]+=f[0];
L69:  f[0]-=f[1];
```

```
L70:  if (!cflag) f[0] = f[1];
L71:  f[0]*=Qt-2;
L72:  f[1]/=f[0];
L73:  f[0]-=T2t;
L74:  f[1]/=f[0];
L75:  f[0]-=f[0];
L76:  f[0]*=f[0];
L77:  f[0]+=T3t;
L78:  f[0]+=1.987620830535889f;
L79:  f[0]*=Qt-2;
L80:  f[1]*=f[0];
L81:  f[0]=cos(f[0]);
L82:  f[0]*=f[0];
L83:  f[0]+=f[1];
L84:  f[0]*=Qt-2;
L85:  f[0]*=f[0];
L86:  f[0]=fabs(f[0]);
L87:  if (!cflag) f[0] = f[0];
L88:  f[1]/=f[0];
L89:  f[0]=-f[0];
L90:  if (!cflag) f[0] = f[0];
L91:  f[0]+=f[1];
L92:  f[0]*=f[1];
L93:  f[0]+=R1t;
L94:  if (!cflag) f[0] = f[1];
L95:  cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L96:  f[0]+=f[0];
L97:  f[0]+=T3t;
L98:  f[0]+=1.987620830535889f;
```

```
L99:  f[0]+=-1.196667432785034f;
L100:         f[0]*=Qt-2;
L101:         f[0]*=Qt-2;
L102:         f[1]*=f[0];
L103:         f[0]=cos(f[0]);
L104:         f[0]*=f[0];
L105:         f[0]-=f[1];
L106:         f[0]*=Qt-2;
L107:         tmp=f[1]; f[1]=f[0]; f[0]=tmp;
L108:         f[0]=sin(f[0]);
L109:         f[0]-=-1.549970149993897f;
L110:         f[0]=-f[0];
L111:         cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L112:         f[0]*=f[0];
L113:         f[1]-=f[0];
L114:         f[0]+=f[0];
L115:         f[0]+=f[1];
L116:         if (cflag) f[0] = f[0];
L117:         tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L118:         if (cflag) f[0] = f[0];
L119:         f[1]-=f[0];
L120:         f[0]=fabs(f[0]);
L121:         cflag=((_isnan(f[0]) || _isnan(f[0])) ? true : (f[0] < f[0]));
L122:         f[1]-=f[0];
L123:         cflag=((_isnan(f[0]) || _isnan(f[1])) ? true : (f[0] < f[1]));
L124:         f[0]+=-0.494312047958374f;
L125:         f[0]/=1.048232078552246f;
L126:         f[0]+=-0.9486191272735596f;
L127:         f[1]/=f[0];
```

```
  L128:         f[0]/=1.084159851074219f;
  L129:         f[0]=fabs(f[0]);
  L130:         f[0]-=f[1];
  L131:         f[0]-=-1.364777803421021f;
  L132:         f[0]=-f[0];
  L133:         f[0]=-f[0];
  L134:         f[0]+=1.530829906463623f;
  L135:         f[0]/=1.048232078552246f;
  L136:         f[0]=fabs(f[0]);
  L137:         f[0]-=f[1];
  L138:         f[0]/=1.048232078552246f;
  L139:         f[0]-=-1.549970149993897f;
  L140:         f[0]-=-1.364777803421021f;
  L141:         f[0]-=f[1];
  L142:         f[0]-=f[1];
  L143:         f[0]=-f[0];
  L144:         f[0]=-f[0];
  L145:         f[0]-=-1.364777803421021f;
  L146:

  return f[0];
}


float DiscipulusCFunction(float v[])
{
        float f[1];
        float fltTmp;
        long lngValidPredictions;
```

```
	f[0]=0;
	lngValidPredictions=0;

	fltTmp=DiscipulusCFunctionSubC0(v);
	if (_finite(fltTmp) && (fltTmp>=-10000.000000 &&
fltTmp<=10000.000000))
	{
		f[0]+=fltTmp;
		++lngValidPredictions;
	};
	fltTmp=DiscipulusCFunctionSubC1(v);
	if (_finite(fltTmp) && (fltTmp>=-10000.000000 &&
fltTmp<=10000.000000))
	{
		f[0]+=fltTmp;
		++lngValidPredictions;
	};
	fltTmp=DiscipulusCFunctionSubC2(v);
	if (_finite(fltTmp) && (fltTmp>=-10000.000000 &&
fltTmp<=10000.000000))
	{
		f[0]+=fltTmp;
		++lngValidPredictions;
	};
	fltTmp=DiscipulusCFunctionSubC3(v);
	if (_finite(fltTmp) && (fltTmp>=-10000.000000 &&
fltTmp<=10000.000000))
	{
```

```
                f[0]+=fltTmp;
                ++lngValidPredictions;
        };
        fltTmp=DiscipulusCFunctionSubC4(v);
        if (_finite(fltTmp) && (fltTmp>=-10000.000000 &&
fltTmp<=10000.000000))
        {
                f[0]+=fltTmp;
                ++lngValidPredictions;
        };
        fltTmp=DiscipulusCFunctionSubC5(v);
        if (_finite(fltTmp) && (fltTmp>=-10000.000000 &&
fltTmp<=10000.000000))
        {
                f[0]+=fltTmp;
                ++lngValidPredictions;
        };
        fltTmp=DiscipulusCFunctionSubC6(v);
        if (_finite(fltTmp) && (fltTmp>=-10000.000000 &&
fltTmp<=10000.000000))
        {
                f[0]+=fltTmp;
                ++lngValidPredictions;
        };
        if (lngValidPredictions)
        {
                f[0]/=lngValidPredictions;
        }
        else
```

```
        {
                f[0]=0.000000;
        }

  return f[0];
}
float DiscipulusCRegressionFunction(float  v [])
{
   float ret = DiscipulusCFunction(v) ;
   return ret;
}

// Copyright, 2013, RML Technologies.
// This program was evolved with Discipulus(tm).
// This program and any information derived from this program
// may be used solely for pure research purposes and publication
// of results therefrom in accordance with the Discipulus
// License agreement. This notice may not be removed from this
// program or any copy thereof.
```